

**O`ZBEKISTON RESPUBLIKASI OLIY VA O`RTA MAXSUS TA`LIM  
VAZIRLIGI**

**BUXORO MUHANDISLIK TEXNOLOGIYA INSTITUTI**

**“AXBOROT KOMMUNIKATSIYA TEXNOLOGIYALARI”  
KAFEDRASI**

**“KIMYOVIY INJINIRING UCHUN  
AXBOROT TEXNOLOGIYALARI”  
fanidan**

---

---

**MA`RUZA MATNLARI**

---

---

Bilim sohasi:	300000 - Ishlab chiqarish va texnik soha.
Ta'lim sohasi:	320000 - Ishlab chiqarish texnologiyalari.
Ta'lim yo'nalishi:	5321400 - Kimyoviy texnologiya (noorganik moddalar)
	5321400 – Kimyoviy texnologiya (organik moddalar)
	5321400 – Kimyoviy texnologiya (yuqori molekulali birikmalar)

**1 - qism**

**BUXORO 2018**

Ma'ruza matnida "Kimyoviy injiniring uchun axborot texnologiyalari" fanini o`qitishning modul tizimidan foydalangan holda rivojlangan xorijiy mamlakatlardagi ilg'or ta'lim texnologiyalari, ularni qo'llash bo'yicha uslubiy tavsiyalar bayon etilgan. Ushbu tavsiyalar ma'ruza mashg'ulotlari texnologiyalarini ishlab chiqish usul va vositalari, o`quv maqsadlari asosidagi muhim belgilaridan iborat ta'lim texnologiyalari qoidalarini hisobga olgan holda ishlab chiqilgan.

**Tuzuvchilar:**

Sohibov T.F. - BuxMTI, "Axborot kommunikasiya texnologiyalari"  
kafedrasi katta o`qituvchisi.

**Taqrizchilar:**

Jalolov O.I. - Buxoro davlat universiteti "Axborot texnologiyalari" kafedrasi dotsenti, f.–  
m. f. n.

Razzakov Sh.I BuxMTI dotsenti, f.– m. f. n.

Ma'ruza matni yangi pedagogik va axborot texnologiyalarining modul tizimidan foydalangan holda yaratilgan fan bo'yicha o`quv materiallar majmuasi seriyasidan. - Buxoro.: BuxMTI, 2018 yil.

## MUNDARIJA

Kirish .....	4
1 modul. Kimyo injiniring uchun axborot texnologiyalari tushunchasi va mohiyati .....	5
1 mavzu. “Kimyoviy injiniring uchun axborot texnologiyalari” faniga kirish .....	5
2 modul. MS Excel dasturini kimyo injiniringda qo'llash .....	23
2 mavzu. MS Excel electron jadvali haqida umumiy tushunchalar .....	23
3 mavzu. MS Excel asoslari .....	52
4 mavzu. MS Excel electron jadvali yordamida misollar echish .....	64
3 modul. C++ algoritmik tili .....	72
5 mavzu. Algoritmiklash asoslari. Algoritm turlari va xossalari .....	72
6 mavzu. C++ algoritmik tili. Kirish .....	87
7 mavzu. Asosiy ma'lumot turlari .....	99
8 mavzu. Operatorlar va standart funksiyalar .....	113
9 mavzu. C++ dasturlash tizimida takrorlanuvchi jarayonlarni dasturlash .....	120
Foydalanilgan adabiyotlar .....	138

## KIRISH.

Hozirgi vaqtda ishlab chiqarishning barcha sohalarida axborot texnologiyalaridan keng foydalanilmoqda. Zamonaviy texnik tizimlaridagi barcha muhandislik muammolari yangi yondoshishni talab etadi. Avtomatlashtirilgan loyihalash tizimlari. Geometric modellash dasturlari, tarmoqdagi ma'lumotlar bazalarini ishlab chiqarish jarayonidagi va texnik tizimlardagi muammolarni yechishda effektiv qo'llanilmoqda. Shuningdek hozirgi kun talabalarning yangi bilimlarni egallashda ta'lim jarayonida va mustaqil ta'limda axborot texnologiyalaridan unumli foydalana olishi va zamonaviy texnologiyalarni tadbiq etishi talab qilinadi.

Axborot kommunikatsiya texnologiyalari tez rivojlanayotgan yo`nalish bo`lib, hozirda u qamrab olmagan biror sohani topish qiyin. Texnik boshqaruv tizimlarida va muhandislik masalalarini yechishda axborot texnologiyalarining o`rni katta. "Kimyoviy injiniringi uchun axborot texnologiyalari" fanini o`qitishdan maqsad zamonaviy axborotli fikrlash va ilmiy dunyoqarashni shakllantirilgan holda talabalarni kompyuter imkoniyatlaridan foydalanish, kompyuter bilan muloqot o`rnatish usullarini o`rgatish va unda turli masalalarni yecha olishga yo`naltirishdan iborat. Jumladan, yo`nalishlar sohalaridagi masalalarni echishda, soha masalalarini loyihalashtirishda, mahsulot dizaynini ishlab chiqishda, sanoat mahsulotlarini tizimli tahlilida, tizim va kommunikatsiyani loyihalashda, muhandislik tahlilida soha muhandislik ishlarini loyihalash jarayonlarini avtomatlashtirishda, texnik tizimlarda ma'lumotlar almashish jarayonida zamonaviy dasturiy vositalardan va texnologiyalardan hamda Internet texnologiyalaridan mukammal foydalanish katta ahamiyatga ega.

Ushbu ma'ruza matnida 5321400 - Kimyoviy texnologiya (noorganik moddalar), 5321400 – Kimyoviy texnologiya (organik moddalar) va 5321400 – Kimyoviy texnologiya (yuqori molekulali birikmalar) yo`nalishi bo'yicha ta'lim oladigan bakalavrlar uchun "Kimyoviy injiniringi uchun axborot texnologiyalari" fanidan o`quv-uslubiy ma'lumotlar berilgan bo`lib, uni o`zlashtirish davomida talabalar o`zlari uchun zaruriy nazariy ma'lumotlarni oladilar va amaliy ko`nikmalar hosil qiladilar. Bu esa ularga kelgusida yetuk kadrlar bo`lib yetishishiga ko`maklashadi.

## **1 modul. KIMYO INJINIRING UCHUN AXBOROT TEXNOLOGIYALARI TUSHUNCHASI VA MOHIYATI**

### **1 mavzu. “Kimyoviy injiniring uchun axborot texnologiyalari” faniga kirish**

#### **Reja:**

1. “Kimyoviy injiniring uchun axborot texnologiyalari” tushunchasi va mohiyati.
2. Axborot texnologiyalarining vujudga kelishi va rivojlanishini asosiy bosqichlari. “Kimyoviy injiniring uchun axborot texnologiyalari” fani va uning bugungi kundagi ahamiyati.
3. Mamlakatda kompyuterlashtirish, ishlab chiqarishda axborot texnologiyalarini keng qo'llash va axborot kommunikatsiya texnologiyalarini rivojlantirish ushbu fanni o'qitishdagi roli va ahamiyati.
4. AKT sohasini texnik yo'nalishlarida tadbiiq etish tamoyillari
5. Tizimli va amaliy dasturiy ta'minotning rivojlanish tendensiyalari.

**Tayanch iboralar:** Injiniring, computer science, informatika, jarayon, axborot texnologiyalari, axborot-kommunikatsiya texnologiyalari, elektron hukumat, kompyuter tizimi, kompyuter platformasi, apparat platformasi, dastur platformasi, dastur, dasturiy ta'minot, tizimli va amaliy dasturiy ta'minot, operatsion tizim, utilita, arxivator, antivirus, drayver, tizim, avtomatlashtirilgan boshqarish tizimi

#### **1. “Kimyoviy injiniring uchun axborot texnologiyalari” tushunchasi va mohiyati**

Kimyo jamiyatda fanlar orasida asosiy o'rinlardan birini egallaydi. Kimyo hayotimizni taminlab turuvchi vosita bo'libgina qolmasdan, kundalik yuzaga kelgan muammolarni hal qilishda asosiy qurol bo'lib ham xizmat qiladi. Shuni hisobga olgan akademik V.A.Koptyug “Nazariy va Amaliy Kimyo Xalqaro Komiteti” (International Union of Pure and Applied Chemistry) ning 35 Bosh Assambleyasi yig'ilishida "Kimyoning boshqa fanlarga va jamiyatdagi progressga ta'siri kattaligini keng jamoatchilikka tushuntirish kerakligani" uqtiradi. Kimyo dunyoni tashkil etgan elementlarni ham, shu elementlardan hosil bo'lgan turli - tuman oddiy va murakkab moddalarni ham, ularning bir-biriga aylanishiga doir murakkab qonunlarni ham o'rganadi.

Kimyo kundalik turmushda kuzatilayotgan hodisalar va o'tkazilayotgan tajribalarning mohiyatini tushuntiradi.

Kimyo avvalo insoniyat uchun eng kerakli materiallarni tabiiy moddalardan olish uchun zarur metallar, keramika, bog'lovchi moddalar, farfor, shisha, o'g'itlar, farmastevtik materiallar, kauchuk, plastmassa, sun'iy tola, yoqilg'i oziq-ovqat mahsulotlari, kompozition materiallar va ko'pgina boshqa materiallar olishga qaratilgandir. Buning uchun kimyo o'zining asosiy muammosini echishi kerak: ya'ni ayni modda qanday kimyoviy elementlardan tashkil topgan va kerakli materiallarni olish uchun qanday o'zgartishlar qilish kerak. Bundan kimyoning asosiy vazifasi - xossasi oldindan ma'lum moddani olish va modda xossasini boshqarish yo'llarini bilishi yoki aniqlashdan iborat.

Faqatgina tabiiy fanlar emas, balki kimyo fani ham - organik kimyo, anorganik kimyo, fizik kimyo va analitik kimyoga ajraldi, lekin ularning asosida kimyoviy tavsif - yani moddaning tarkibi, tuzilishi va kimyoviy bog'lanishi yotadi. Kimyoviy bilimlar ko'lami juda keng. Kimyo fani - atrofimizni o'rab olgan butun olamni, uning xilma - xil shakllarini va olamda sodir bo'ladigan turli tuman hodisalarni tekshiruvchi tabiat fanlaridan biridir. Tabiat, butun olam inson ongidan tashqarida va inson ongiga bog'liq bo'lmagan holda mavjuddir. Olam materiyadan iborat, mavjud narsalarning hammasi doimo harakat qilib turadigan materiyaning har xil turlaridir. Moddani chuqur bilish va undan inson farovonligi yo'lida foydalanish kimyoni hozirgi asosiy masalasidir.

Kimyoda kimyoviy jarayonlar amalga oshiriladi va bu jarayonlarni amalga oshirishda murakkab hisoblashlar bajariladi. Bu hisoblashlarni tez va aniq bajarish uchun zamonaviy kompyuter texnikalaridan unumli foydalaniladi. Kimyoviy jarayonlarni borishi va undagi holatlarni tavsiflash uchun birqancha maxsus dasturiy ta'minotlardan foydalaniladi. Shuning uchun hozirgi zamonaviy kimyo texnologiyalarini kompyuter ishtirokisiz tasavvur qilib bo'lmaydi. Shu sababli kimyo texnologiyalarini yanada rivojlantirish uchun kimyo sohasining

bo'lajak mutaxassislariga maxsus "Kimyoviy injiniring uchun axborot texnologiyalari" fani ham o'qitiladi.

Injinirihg – ishlab chiqarish jarayonlarini loyihalash, tadbiiq va tashkil qilish hamda ish jarayonini mu'tadil davom etishini ta'minlashni amalga oshirish xizmati.

Injinirihg keng qamrovli tushuncha bo'lib, uni turli mutaxassislar turlicha talqin qilishadi. Ko'pchilikning fikricha injiniring asosan ingenerlik ishi, jarayonlarni loyihalash va tadbiiq qilish, marketing, moliya-iqtisod, menejment singari sohalarni bilishni taqozo qiladi. Bu bilimlarni ishlab chiqarish jarayoniga tadbiiq qilishda axborot texnologiyalari sohasida ham etarlicha bilimga ega bo'lishlik talab qilinadi.

"Kimyoviy injiniring uchun axborot texnologiyalari" fani kimyo sohasida axborot texnologiyalarining o'ziga xos xususiyatlari va imkoniyatlarini hamda axborot texnologiyalarini inson faoliyatining turli sohalarida qo'llanilish asoslarini o'rgatishdan iboratdir. Bunda asosiy e'tibor axborot texnologiyalarini kimyo sohasida qo'llanilishiga qaratilgandir. Shuningdek talabalarni axborot texnologiyalarini texnik va dasturiy vositalari bilan tanishtirish va bu vositalarni ishlab chiqarish, ilmiy tadqiqot ishlari hamda o'quv jarayonlariga tadbiiq etish usullari va ularning o'ziga xos xususiyatlarini o'rgatishdan iboratdir.

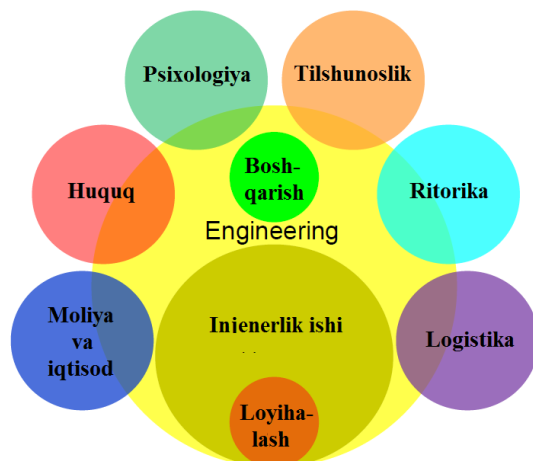
## **2. Axborot texnologiyalarining vujudga kelishi va rivojlanishini asosiy bosqichlari.** **"Kimyoviy injiniring uchun axborot texnologiyalari" fani va uning bugungi kundagi ahamiyati.**

Dunyo miqyosida 1950 yil elektron hisoblash mashinalari (EHM) taraqqiyotining boshlanishi bo'ldi. (Birinch hisoblash mashina AQShning Pensilvaniya Universitetida Djon Mochli va Prespera Ekerta rahbarligida 1946 yilda yaratilgan. U ENIAC deb nomlangan. Uning og'irligi 30 tonna, 18000 elektron lampadan iborat va 500 amalni 1 sekundda bajarar edi.) EHM larning takomillashtirilib borilishi va ularning ishlab chiqarishning barcha jabhalarida qo'llash kengayib borganligi sababli, unda ishlovchi mutaxassislarni tayyorlashga jahon miqyosida 1950 yillardan kirishilgan. Keyinchalik, nafaqat unda ishlovchi mutaxassislar, balki boshqa soha mutaxassislari ham bu sohada bilimga ega bo'lishi ehtiyoji vujudga keldi. Natijada EHM haqida boshlangich bilimlarni beruvchi maxsus fan yuzaga keldi. Bu fan ba'zi bir davlatlarda "Informatika", ba'zilarida esa "Computer science" atamasi bilan atala boshlandi.

1962 yilda Frantsiyada Filipp Dreyfus tomonidan kiritilgan "Information" (axborot) hamda "Automatique" (avtomatika) so'zlarining qo'shilmasidan hosil bo'lgan "Informatique" (informatika) atamasi qo'llanila boshlandi. "Informatika" atamasi "Informatique" (Frantsiya), "Informatik" (Germaniya), "Informatica" (Italiya, Niderlandiya), "Informatica" (Ispaniya, Portugaliya), "informatika" (slavyan tillarida) nomi bilan qo'llanila boshlandi. Bu axborotga avtomatik ravishda ishlov berish fani degan ma'noni anglatadi. 1978 yil xalqaro ilmiy kongress "Informatika" atamasini qo'llashni tasdiqladi. G'arbiy Evropaning ko'pchilik mamlakatlarida hamda AQSh va Kanadada "Informatika" atamasi o'rniga uning sinonimi "Computer science" (kompyuter haqidagi fan) atamasi qo'llaniladi. "Informatika" (bizda ushbu atama ishlatiladi) EHMning rivojlanishi tufayli yuzaga keldi, unga asoslanadi, usiz mavjud bo'la olmaydi va o'z navbatida uning rivojiga, yangilanishiga o'z hissasini qo'shadi. Uning asosiy texnik bazasi EHM, resursi – axborotdir. Informatika fani ko'p qirrali funksiyalarga ega bo'lgani uchun unga har xil ta'rif berilgan.

*Informatika* – ma'lumotlarni hosil qilish, saqlash, uzatish va qayta ishlash usullarini hisoblash texnikasi vositalari yordamida tizimlashtiruvchi, aloqadorligini o'rnatuvchi va ularni boshqaruvchi inson faoliyatining aniq sohasidir.

*Informatika* - ilmiy axborotlarning struktura va umumiy xossalarini, shuningdek ularni



Injiniring - bilimlarning integral sohasidir

yaratish, o'zgartirish, uzatish va ulardan inson faoliyatining turli sohalarida foydalanish qonuniyatlarini o'rganadigan fan.

*Informatika* — bu axborotning nafaqat umumiy xususiyatlari, balki unga avtomatlashtirilgan ishlov berishning uslublari, jarayonlari va texnik vositalarini o'rganuvchi fandır.

*Informatika* - axborotlar haqidagi fan bo'lib, uning qayta ishlashdagi takomillashgan usul va vositalarini o'rgatadi.

*Informatika* - axborotlarni EHM yordamida tasvirlash, saqlab turish, uzatish va ishlov berish usullarini o'rganadigan fandır.

1973 yildan boshlab EHM tarixining yangi sahifasi, shaxsiy kompyuterlar sahifasi boshlandi.

*Kompyuter* (ingl: *computer*, rus: *kompyuter*). Hisoblarni bajarish, shu jumladan elektron shakldagi axborotni oldindan belgilangan algoritm bo'yicha qabul qilish, qayta ishlash, saqlash uchun mo'ljallangan mashina. Kompyuter so'zi ingliz tilidagi to compute, computer, so'zlarining hosilasi bo'lib, ular "hisoblash", "hisoblagich" deb tarjima qilinadi. Dastlab ingliz tilida bu so'z, mexanik qurilmani jalb qilib yoki uning ko'magisiz arifmetik hisoblarni bajaradigan insonni anglatgan. Keyinchalik uning ma'nosi mashinalarning o'ziga ko'chirildi, biroq, zamonaviy kompyuterlar matematika bilan bevosita bog'liq bo'lmagan ko'plab masalalarni ham bajaradilar. XX asrning 90-yillaridan boshlab, kompyuter atamasi elektron hisoblash mashinalari (EHM) atamasini amalda siqib chiqardi.<sup>1</sup>

Kompyuter hozirda faqat hisoblovchi bo'lmasdan matnlar ustida, tovush, video va boshqa ma'lumotlar ustida ham amallar bajaradi.

Kompyuter tizimi deganda muvofiq dastur orqali ma'lumotlarga ishlov berish imkoniga ega bo'lgan ixtiyoriy texnikalarni tushunamiz. Ularga super kompyuterlardan to oddiy kal'kulyatorgacha bo'lgan texnikalar kiradi. Ular turli xil kompyuter platformasiga ega.

Kompyuter platformasi - bu taklif qilinayotgan cheklovlar va imkoniyatlar doirasida dasturlar yoki ob'ektga oid modullar bajariladigan ixtiyoriy mavjud muhitdir.

Kompyuter platformasi apparat va dastur platformasi qismlariga ajratiladi.

Kompyuterning apparat platformasi – bu operatsion tizim va amaliy dasturiy ta'minot suyanadigan yuksak darajada tashkillashtirilgan hisoblash tizimining quyi qatlami modelidir.

Kompyuterning apparat platformasi uning arxitekturasini ifodalaydi. Shuning uchun, ba'zan kompyuterning apparat platformasi atamasi o'rniga qisqacha kompyuter platformasi yoki kompyuter arxitekturasini so'zi ishlatiladi. U asosan tizimli platalar, shinalar, protsessorlar, xotira va chip singari qurilmalardan tashkil topadi. Ushbu komponentlarga muvofiq kompyuter platformalari bir-biridan farq qiladi. Bir kompyuter platformasi uchun ishlab chiqilgan qurilmani boshqa kompyuter platformasida ishlatib bo'lmaydi. O'z navbatida har bir kompyuter platformasi uchun aniq bir dasturiy ta'minot oilasi (operatsion tizimlar, amaliy dasturiy ta'minot) mavjud.

Kompyuterning dasturiy platformasi – bu operatsion tizim va amaliy dasturiy ta'minot oilalari sinfidir.

Turli xil kompyuter platformalari o'z imkoniyatlariga ko'ra umumiy tarzda quyidagicha tavsiflanadi:

1. Superkompyuter.
2. Meynfreym yoki "Korxonalar serveri".
3. Server.
4. Shaxsiy kompyuter, SHK.



<sup>1</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

5. Notebook.
6. Netbook.
7. Planshet.
8. Smartfon.

Hozirgi kunda kompyuter texnikalari va ulardan foydalanishni o'rgatuvchi fan "Informatika" atamasi o'rniga ko'proq "Axborot texnologiyalari" atamasi ishlatilmoqda.

*Axborot (ingl: information, rus: informatsiya).*

1 Taqdim etilish shaklidan qat'iy nazar shaxs, predmet, dalil, voqea, hodisa va jarayonlar haqidagi ma'lumotlar.

2 Dalil, voqea, hodisa, predmet, jarayon kabi ob'ektlar haqidagi bilim (ma'lumotlar) hamda tushunchalar yoki buyruqlar.

3 Ma'lum xos matnda aniq ma'noga ega bo'lgan tushunchalarni ichiga oluvchi dalil, voqea, hodisa, predmet, jarayon, taqdimot kabi ob'ektlar haqidagi bilim (ma'lumotlar)<sup>1</sup>.

*Texnologiya (ingl: technology, rus: texnologiya).* Muayyan ishlab chiqarish sohasidagi usullar va jarayonlar majmuasi.<sup>1</sup>

*Jarayon (ingl: process, rus: protsess).*

1 Berilgan sharoitlarda erishiladigan maqsad yoki natija bilan belgilanuvchi voqealarning berilgan ketma-ketligi.

2 Dastur yoki uning qismi bajarilayotganda ro'y beradigan voqealarning berilgan ketma-ketligi<sup>1</sup>.

Axborot texnologiyalarida asosiy resurs axborotdir. Uning muhim xususiyati, beshta muhim xossaga ega bo'lishligidir. Bular axborotni yaratish, qabul qilish, saqlash, ishlov berish va uzatish xossaligidir.

Axborot texnologiyalari ko'pqirrali funksiyalarga ega bo'lib, u axborotlarga ishlov berishda kompyuterlardan foydalanish, hisoblash texnikalari va aloqa texnikalarini qamrab oladi. Unga ham har xil ta'rif berilgan, ya'ni:

*Axborot texnologiyalari* - axborotni to'plash, saqlash, izlash, unga ishlov berish va uni tarqatish uchun foydalaniladigan jami uslublar, qurilmalar, usullar va jarayonlardir.<sup>2</sup>

*Axborot texnologiyalari* - axborotlarni yig'ish, saqlash, uzatish, qayta ishlash usul va vositalari majmuyidir.

*Axborot texnologiyalari* - ma'lumotlarni to'plash, ularga ishlov berish, saqlash, uzatish, va ulardan foydalanish jarayonida hisoblash texnikasidan foydalanish yo'llari, usullari va uslublaridir.

*Axborot texnologiyalari* - hujjatlashtirilgan axborot, jumladan dasturli vositalarga ishlov berishning jami uslublari, yo'llari, usullari va vositalari hamda ulardan foydalanishning belgilangan tartibidir.

*Axborot texnologiyalari* - insonlar tomonidan axborotni yig'ish, saqlash, ishlov berish va tarqatish uchun foydalaniladigan jami uslublari, qurilmalar va ishlab chiqarish jarayonlaridir.

*Axborot texnologiyalari* - ob'ekt, jarayon yoki hodisa (axborot mahsulot) holati haqida yangi sifatdagi ma'lumotlarni olish uchun foydalanadigan ma'lumotlarni (birlamchi) yig'ish, ishlov berish va o'zlash vositalari, hamda usullari majmuasidir.

*Axborot texnologiyalari* - axborot oqimini yig'ish, saqlab turish, ishlov berish va uzatishni kompyuterlar yordamida amalga oshiradigan usullar majmuasidir.

Axborot texnologiyalari turli shakldagi axborotlar bilan ish ko'radi. Bular:

- belgili – turli ishoraviy belgilardan iborat axborotlar. Bular biror voqea hodisalar haqidagi axborotlarni uzatishda foydalaniladi;
- matnli – harf, raqam va belgilar to'plamidan tarkib topgan ma'lum ma'noni anglatuvchi so'zlardan iborat axborot;
- grafik – tasvirlardan iborat bo'lgan tasavvur ko'rinishidagi axborotlar;

<sup>2</sup> O'zbekiston Respublikasining "Axborotlashtirish to'g'risida" gi Qonyni. Toshkent shahri, 1993 yil, may.



- tovush – eshitish qobiliyati orqali qabul qilinadigan gap, musiqa, turli shovqin effektlar kabi axborotlar;
- video (lotincha video) –qarayman, ko'rayapman ma'nosini bildirib harakat va tovush effektlariga ega bo'lgan axborotlardir.

Axborot texnologiyalari o'z o'zicha mavjud bo'lmay, balki boshqa sohalaridagi muammolarini hal etish uchun yangi axboriy texnika va texnologiyalarini yaratishga qaratilgan kompleks ilmiy – texnik sohadir. U boshqa sohalar, hatto jarayonlar va hodisalar noformallashuvi tufayli miqdoriy uslublarni qo'llash mumkin emas deb hisoblanadigan sohalariga ham tadqiqot uslub va vositalarini taqdim etadi. Axborot texnologiyalarida kompyuter texnikasi sharofati tufayli amaliy ro'yobga chiqish mumkin bo'lgan matematik modellash uslublarining hal qilinishini alohida ajratib ko'rsatish lozim.<sup>3</sup>

Axborot texnologiyalari *amaliy fan sohasi* sifatida quyidagilar bilan shug'ullanadi:

- axborot jarayonlaridagi qonuniyatlarni o'rganish (axborotlarni yig'ish, qayta ishlash, tarqatish);
- inson faoliyatining turli sohalarida kommunikatsion – axborot modellarini yaratish;
- aniq bir sohalarida axborot tizimi va texnologiyalarini ishlab chiqish va ularning hayotiy bosqichini, ularni ishlab chiqarish, ishlashni va h-zolarni loyihalash, ishlab chiqish bosqichlari uchun tavsiyalar tayyorlash.

Zamonaviy axborot kommunikatsiya texnologiyalarining imkoniyatlari cheksiz bo'lib, u shiddat bilan rivojlanmoqda hamda XXI asr axborot-kommunikatsiya texnologiyalari va internet asriga aylanib bormoqda.

*Axborot-kommunikatsiya texnologiyalari (AKT) (ingl: information and communication technologies (ICT), rus: axborotno-kommunikatsionnie texnologii (IKT)).* Xususiy, umumiy va ishlab chiqarish kommunikatsiyasida axborotlar tayyorlash, qayta ishlash va etkazish bilan bog'liq bo'lgan ob'ektlar, harakatlar va qoidalar, shuningdek barcha texnologiyalar hamda sanab o'tilgan jarayonlarni birlashgan ravishda ta'minlovchi sohalar majmuasi. AKT tushunchasiga mikroelektronika, kompyuter va dasturiy ta'minot, telekommunikatsiyalar ishlab chiqish hamda ishlab chiqarish, Internetdan foydalanishni ta'minlash, Internetning axborot resurslarini ta'minlash, shuningdek sanab o'tilgan sohalar bilan bog'liq bo'lgan turli xil hodisalarva bu faoliyat sohalarini tartibga soluvchi qoidalar (rasmiylari kabi norasmiylari ham) kiradi. Bu erda radio, televidenie, mobil telefonlar, kompyuterlar va tarmoq uskunalari va dasturiy ta'minot, yo'ldosh tizimlari va h.k., shuningdek turli xizmatlar va ularga tegishli dasturlar, masalan, videoanjuman va masofaviy ta'lim singarilar qamrab olinadi.<sup>4</sup>

Bugungi kunda AKTrining jadal rivojlanishi quyidagilarda namoyon bo'lmoqda:

- yangi axborot va kommunikatsiya texnologiyalari shaxsiy kompyuter va internet ixtiro qilinishi sababli, jamiyat hayotiga juda tez, ya'ni bir inson umri davomida kirib keldi;
- axborotlar bilan ishlash kontseptsiyasining oddiyligi ayniqsa tasvir, ovoz va multimedia bilan ishlashni osonlashtirdi;
- kommunikatsiya usuli o'zgarib, uning imkoniyatlari kengayishi natijasida, yangi texnologiyalar fan, ta'lim, madaniyat va siësatga sezilarli darajada ta'sir ko'rsatishi kuzatildi.

### **3. Mamlakatda kompyuterlashtirish, ishlab chiqarishda axborot texnologiyalarini keng qo'llash va axborot kommunikatsiya texnologiyalarini rivojlantirish ushbu fanni o'qitishdagi roli va ahamiyati**

Axborotlashtirish jarayonini davlat tomonidan tartibga solishning muhimligi va uni tezlashtirish zaruriyatini hisobga olib, O'zbekiston Respublikasi Vazirlar Mahkamasining 1992 yil 8 dekabr qarori bilan "Fan va texnika bo'yicha Davlat Qo'mitasi" (FTDQ) qoshida "Axborotlashtirish bo'yicha bosh boshqarma" (Boshaxbor) tuzildi. Mazkur qarorda belgilab

<sup>3</sup> Michael Henderson, Geoff Romeo. Teaching and Digital Technologies Cambridge University Press 2015, p. 15-16.

<sup>4</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

berilgan asosiy vazifa va faoliyat yo'nalishlari doirasida O'z FTDQ tashabbusi bilan axborotlashtirish jarayonini rivojlantirish uchun bir qator qonunlar qabul qilindi.

1994 yil dekabrda Vazirlar Mahkamasi "O'zbekiston Respublikasini axborotlashtirish konseptsiyasi"ni ma'qulladi. Ushbu Kontseptsiyaning asosiy maqsad va unda qo'yilgan masalalar quyidagilardan iboratdir:

- o milliy axborot-hisoblash to'rini yaratish;
- o axborotlarga tovar sifatida yondashishning iqtisodiy, huquqiy va me'yoriy hujjatlarini yuritish;
- o axborotlarni qayta ishlashning jahon standartlariga rioya qilish;
- o informatika industriyasini mujassamlashtirish va rivojlantirish;
- o axborotlar texnologiyasi sohasidagi fundamental tadqiqotlarni rag'batlantirish va qo'llab-quvvatlash;
- o informatika vositalari foydalanuvchilarini tayyorlash tizimini muvofiqlashtirish.

Kontseptsiyaning asosiy qoidalari hisobga olingan "O'zbekiston Respublikasining axborotlashtirish dasturi" ishlab chiqildi. U uch maqsadli dasturni o'z ichiga oladi:

- a) milliy axborot-hisoblash tarmog'i;
- b) EHMni matematik va dasturiy ta'minlash;
- v) shaxsiy kompyuter.

Mazkur dasturda vazirlik va mahkama axborot tarmoqlari, Milliy axborot - hisoblash tarmog'ini yaratish, kompyuter va hisoblash texnikasi vositalarini ishlab chiqarishni tashkil etish, yangi axborot texnologiyalari sohasida kadrlar tayyorlashni takomillashtirish, hujjatlashtirishning me'yoriy - uslubiy va huquqiy tizimini yaratish va boshqalar joy olgan.

Respublikada kompyuterlashtirish va axborot kommunikatsiya texnologiyalarini rivojlantirishga doir muayyan vazifalar O'zbekiston Respublikasining birinchi Prezidenti I.A.Karimov tomonidan 2002 yil 30 mayda imzolangan "Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida" gi Farmonida belgilab berilgan bo'lib, O'zbekiston Respublikasi Vazirlar Mahkamasining 2002 yil 6 iyun 200-sonli "Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish chora-tadbirlari to'g'risida"gi Qarorida o'z aksini topgan. Ushbu vazifalar keyingi vaqtda Respublikamizda qabul qilingan bir qator qonunlar, Vazirlar mahkamasining qarorlari va boshqa normativ hujjatlarda o'z rivojini topdi

Ushbu farmonning qabul qilinishi O'zbekiston Respublikasi Vazirlar Mahkamasi huzurida "Kompyuterlashtirish va axborot-kommunikatsiya texnologiyalarini rivojlantirish bo'yicha Muvofiqlashtiruvchi Kengash" tuzilishiga, "O'zbekiston pochta va telekommunikatsiyalar agentligi" "O'zbekiston aloqa va axborotlashtirish agentligi" etib qayta tashkil etilishiga, "Kompyuter va axborot texnologiyalarini rivojlantirish hamda joriy etish markazi" tashkil qilinishiga, Toshkent elektrotexnika aloqa instituti Toshkent axborot texnologiyalari universitetiga aylantirilishiga olib keldi.

**a. O'zbekistonda AKT sohasidagi asosiy Qonunlar:**

- Elektron hisoblash mashinalari uchun yaratilgan dasturlar va ma'lumotlar bazalarining huquqiy himoyasi to'g'risida (1994 y.)
- Telekommunikatsiyalar to'g'risida (1999 y.)
- Axborot erkinligi printsiplari va kafolatlari to'g'risida (2002 y.)
- Elektron raqamli imzo to'g'risidagi (2003 y.)
- Axborotlashtirish to'g'risida (2003 y.)
- Elektron hujjat aylanishi to'g'risidagi (2004 y.)
- Elektron tijorat to'g'risidagi (2004 y.)
- Elektron to'lovlar to'g'risidagi (2005 y.)
- Axborot-kutubxona faoliyati to'g'risida (2011 y.)

**b. O'zbekiston Respublikasi Prezidentining AKT sohasidagi Farmon va Qarorlari.**

- Telekommunikatsiyalar sohasida boshqaruvni takomillashtirishga doir chora-tadbirlar to'g'risida (2000 y.).

- Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida (2002 y.).
- Axborot texnologiyalari sohasida kadrlar tayyorlash tizimini takomillashtirish to'g'risida (2005 y.).
- Axborot-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida (2005 y.).
- O'zbekiston Respublikasining jamoat ta'lim axborot tarmog'ini tashkil etish to'g'risida (2005 y.).
- Respublika aholisini axborot-kutubxona bilan ta'minlashni tashkil etish to'g'risida (2006 y.).
- O'zbekiston Respublikasida axborotni kriptografik muhofaza qilishni tashkil etish chora-tadbirlari to'g'risida (2007 y.).
- Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to'g'risida. PQ-1730, 21.03.2012 y.
- O'zbekiston Respublikasi aloqa, axborotlashtirish va telekommunikatsiya texnologiyalari davlat qo'mitasini tashkil etish to'g'risida. PF-4475, 16.10.2012.
- "Axborot-kommunikatsiya texnologiyalari sohasida kadrlar tayyorlash tizimini yanada takomillashtirish chora tadbirlarito'g'risida"gi PQ-1942, 26.03.2013.
- O'zbekiston Respublikasi Milliy axborot-kommunikatsiya tizimini yanada trivojlantirish chora-tadbirlari to'g'risidagi PQ-1989, 27.06.2013 y.

***v. O'zbekiston Respublikasi Vazirlar mahkamasining AKT sohasidagi qarorlari va me'yoriy hujjatlari.***

- 2001-2005 yillarda kompyuter va axborot texnologiyalarini rivojlantirish, Internetning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqishni tashkil etish chora-tadbirlari to'g'risida (2001 yil).
- Toshkent axborot texnologiyalari universiteti faoliyatini tashkil etish to'g'risida (2002 y.).
- Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish chora-tadbirlari to'g'risida (2002 y.).
- "ZiyoNet" axborot tarmog'ini yanada rivojlantirish to'g'risida (2005 y.).
- Axborotlashtirish sohasida normativ-huquqiy bazani takomillashtirish to'g'risida (2005 y.).
- Davlat va xo'jalik boshqaruvi, Mahalliy davlat hokimiyati organlarining axborot-kommunikatsiya texnologiyalaridan foydalangan holda yuridik va jismoniy shaxslar bilan o'zaro aloqadorligini yanada takomillashtirish chora-tadbirlari to'g'risida (2007 y.).
- Internet tarmog'ida O'zbekiston Respublikasining Hukumat portalini yanada rivojlantirish chora-tadbirlari to'g'risida (2007 y.).
- Internet tarmog'ida O'zbekiston Respublikasining Hukumat portaliga axborotlarni taqdim etish va joylashtirish tartibi to'g'risida (2009 y.).
- Davlat va xo'jalik boshqaruvi, mahalliy davlat hokimiyati organlari xodimlarining malakasi va ko'nikmalarini oshirishga doir qo'shimcha chora-tadbirlar hamda ularni ishda kompyuter texnikasi va axborot-kommunikatsiya texnologiyalaridan foydalanish yuzasidan attestatsiyadan o'tkazish tartibi to'g'risida (2011y.).
- Interaktiv davlat xizmatlari ko'rsatishni hisobga olgan holda Internet tarmog'ida O'zbekiston Respublikasining Hukumat portali faoliyatini yanada takomillashtirish chora-tadbirlari to'g'risida (2012 y.).
- O'zbekiston Respublikasi Aloqa, axborotlashtirish va telekommunikatsiya texnologiyalari davlat qo'mitasi huzuridagi "Elektron hukumat" tizimini rivojlantirish markazi hamda Axborot xavfsizligini ta'minlash markazi faoliyatini tashkil etish chora-tadbirlari to'g'risida (2013y.).

Umuman, mamlakatimizda axborot-kommunikatsiya texnologiyalarini rivojlantirishga qaratilgan me'yoriy - huquqiy baza shakllantirildi, ya'ni tarmoqqa doir 12 ta qonun, 6 ta turdosh

qonun, O'zbekiston Respublikasi Prezidentining 3 ta farmoni, O'zbekiston Respublikasi Prezidenti va Vazirlar Mahkamasining 40 dan ortiq qarori hamda 600 dan ortiq qonunosti hujjatlari qabul qilindi.

**Respublikada kompyuterlashtirish va AKTni rivojlantirish dasturi va undagi vazifalar va boshqaruvdagi roli.** 2012 yil, 21-mart kuni imzolangan, PQ-1730 Prezident qarori orqali, o'z ichiga 21 ta aniq chora-tadbirlarni qamrab olgan, 2012-2014 yillarga mo'ljallangan maxsus dastur tasdiqlandi va uning ijrosi maxsus Muvofiqlashtiruvchi Kengashga yuklandi.

Vazirlar Mahkamasining 2012 yil 30 dekabrda, №378 "Interaktiv davlat xizmatlari ko'rsatishni hisobga olgan holda internet tarmog'ida O'zbekiston Respublikasining hukumat portalini faoliyatini yanada takomillashtirish chora-tadbirlari to'g'risida" qarori bilan, elektron hukumatni yanada rivojlantirish va davlat boshqaruviga zamonaviy axborot texnologiyalarini joriy etishda ko'maklashish maqsadida, yagona davlat interaktiv xizmatlari portalini (yagona portal) tashkil etish va uning texnik va texnologik asoslari bayon qilingan nizom qabul qilindi.

Vazirlar Mahkamasining 25-yanvar, 2013 yil, 18-sonli qarori bilan, davlat boshqaruvining samaradorligi va yuridik hamda jismoniy shaxslarga interaktiv davlat xizmatlari ko'rsatish sifatining oshirilishini ta'minlovchi "Elektron hukumat" tizimini shakllantirish alohida ustuvor vazifa qilib belgilandi. Qolaversa, mazkur qarorda "Elektron hukumat" tushunchasiga o'zbek tilida rasmiy tushuntirish – ta'rif berildi.

*Elektron hukumat tizimi* – axborot-kommunikatsiya texnologiyalarini qo'llash yo'li bilan yuridik va jismoniy shaxslar tomonidan davlat xizmatlari hamda davlat organlarning faoliyati natijalari to'g'risida axborot olish (almashish) tezkorligi va qulayligi sifat jihatdan yangi darajasini ta'minlovchi davlat organlarining faoliyatini tashkil qilish shakli.

O'zbekiston Respublikasi Prezidentining 2013 yil 27 iyundagi PQ-1989-son qarorida "Respublika milliy axborot-kommunikatsiya tizimlarini 2013-2020 yillar mobaynida rivojlantirish kompleks dasturi" tasdiqlandi. Ushbu kompleks dasturga asosan "Elektron hukumat" tizimi axborot tizimi komplekslari va ma'lumotlar bazasini yaratish bo'yicha tadbirlar va loyihalar ro'yxati tuzildi. Shuningdek, O'zbekiston Respublikasi Milliy axborot-kommunikatsiya tizimlarini 2013-2020 yillar mobaynida rivojlantirish kompleks dasturini amalga oshirishni muvofiqlashtiruvchi Respublika komissiyasi tashkil etildi.

O'zbekiston Respublikasida 2013-2020 yillar mobaynida telekommunikatsiya texnologiyalari, tarmoqlari va infrastrukturasi rivojlantirish dasturi quyidagilardan iborat:

- Keng polosali optik tarmoqlarni kengaytirish.
- Optik tolali aloqa liniyalarini qurish.
- BRAS qurilmalarini o'rnatib, keng polosali tarmoq imkoniyatlarini kengaytirish.
- "O'zMobayl" mobil tarmog'ini hududlarda rivojlantirish - 1-bosqich yuqori tezlikdagi internet xizmatlarini ko'rsatish (EVDO texnologiyasi).
- "O'zMobayl" mobil tarmog'ini hududlarda rivojlantirish - 2-bosqich yuqori tezlikdagi internet xizmatlarini ko'rsatish (LTE texnologiyasi).
- 3G, 4G LTE va boshqa mobil aloqa operatorlarini hisobga olgan holda mobil aloqa tarmoqlarini rivojlantirish.
- NGN texnologiyasi bo'yicha kommunikatsiya qurilmalarini kengaytirish.
- Kommutatsiya markazlarini modernizatsiyalash.
- Paketli kommutatsiya xalqaro markazlarini kengaytirish.
- O'zbekiston Respublikasi magistral tarmoqlarining o'tkazuvchanlik qobiliyatini kengaytirish.
- Ma'lumotlarni uzatish multiservis tarmoqlarini qurish.
- Magistral optik tolali aloqa liniyalarini qurish.
- Korporativ sektorga mul'timediyali xizmatlarni ko'rsatuvchi studiyalarni yaratish.
- Axborot-ma'lumot xizmatlari markazlarini yaratish (call-center).
- Ma'lumotlarni saqlash va qayta ishlash markazlarini yaratish "Data-tsentri", Quyidagi shaharlarda: Toshkent, Qo'qon, Buxoro.

- Tez-tez foydalaniladigan ma'lumotlarni saqlash markazlarini yaratish (keshlash markazlari).
- Davlat va xo'jalik boshqaruvi, mahalliy hokimiyat organlari korporativ va lokal-hisoblash tarmoqlarini yaratish va modernizatsiya qilish.
- “Elektron hukumat” tizimi ma'lumotlar bazasi va axborot tizimi komplekslarini yaratish bo'yicha tadbirlar va loyihalar ro'yxati:
  - “Elektron hukumat to'g'risida” Qonun loyahasini yaratish.
  - “Elektron hukumat” tizimi doirasida ma'lumotlar almashish va idoralararo o'zaro munosabat bo'yicha davlat standartlarini yaratish va tasdiqlash.
  - “Elektron hukumat” tizimida tadbirkorlik sub'ektlari va aholi bilan elektron shaklda o'zaro munosabatga o'tishni ta'minlashda qonunchilikni takomillashtirish.
  - Ma'lumotlarni yig'ish va qayta ishlash mexanizmlarini standartlashtirishni hisobga olgan holda ko'chmas mulkni ro'yxatdan o'tkazish va kadastr axborot tizimi kompleksini yaratish.
  - O'zbekiston Respublikasi Milliy geografik axborot tizimini yaratish.
  - Jismoniy shaxslar ma'lumotlar bazasini shakllantirish (unikal identifikator bilan).
  - Yuridik shaxslar ma'lumotlar bazasini shakllantirish (unikal identifikator bilan).
  - Spravochniklar va klassifikatorlar ma'lumotlar bazasini shakllantirish.
  - Avtoulovlarni ro'yxatdan o'tkazish bo'yicha yagona ma'lumotlar bazasini shakllantirish.
  - Interaktiv davlat xizmatlari yagona portalini yaratish va rivojlantirish.
  - Xizmatlarni ko'rsatuvchilarning billing tizimlari bilan integratsiyasini hisobga olib, real vaqt rejimida chakana to'lovlarni amalga oshiruvchi kliring tizimini yaratish.
  - “Elektron hukumat” idoralararo integratsion tizimini yaratish.
  - Kompyuter va axborot texnologiyalarini joriy etish va rivojlantirish markazi “O'zinfokom” strukturasi “Elektron hukumat” tizimi ma'lumotlarini qayta ishlovchi markaz yaratish.
  - “Elektron hukumat” tizimi foydalanuvchilarining yagona identifikatsiya tizimini yaratish va joriy etish.
  - Davlat xaridlari sohasida interaktiv xizmatlarni ko'rsatuvchi “Xarid” axborot tizimi kompleksini integratsiya qilish.
  - Soliq sohasida interaktiv xizmatlarni ko'rsatuvchi “Soliq” axborot tizimi kompleksini integratsiya qilish.
  - Bojxona yuklarini rasmiylashtirish bo'yicha interaktiv xizmatlarni ko'rsatuvchi “Bojxona” axborot tizimi kompleksini integratsiya qilish.
  - Litsenziyalash sohasida interaktiv xizmatlarni ko'rsatuvchi “Litsenziya” axborot tizimi kompleksini yaratish.
  - Davlat byudjeti ijrosida ma'lumotlarni saqlash, tizimlashtirish, qayta ishlash, yig'ish uchun “Byudjet” axborot tizimi kompleksini integratsiya qilish.
  - Sog'liqni saqlash Milliy integrallashgan axborot tizimini yanada rivojlantirish uchun hamda sog'liqni saqlash sohasida interaktiv xizmatlarni ko'rsatuvchi “Sog'liqni saqlash” axborot tizimi kompleksini yaratish.
  - Ta'lim sohasida interaktiv xizmatlarni ko'rsatuvchi “Ta'lim” axborot tizimi kompleksini yaratish.
  - Kommunal xo'jalik sohasida interaktiv xizmatlarni ko'rsatuvchi “Kommunal” axborot tizimi kompleksini yaratish.
  - Sudlar va notariuslar faoliyati haqida ma'lumotlarni saqlash, qayta ishlash, to'plash bo'yicha “Adliya-2” axborot tizimi kompleksini yaratish.
  - Davlat organlari haqida statistik ma'lumotlarni ta'minlash bo'yicha “Davlat boshqaruvi” axborot tizimi kompleksini yaratish.
  - Nafaqa ta'minoti sohasida interaktiv xizmatlarni ko'rsatuvchi “Nafaqa” axborot tizimi kompleksini integratsiya qilish.

○ “Elektron hukumat” tizimini joriy qilish va uning samarali ishlatilishida ishtirok etuvchi Davlat va xo`jalik boshqaruvi, mahalliy hokimiyat organlari xodimlarini AKT bo`yicha o`qitish va malakasini oshirish maqsadida “Elektron hukumat” o`quv markazini yaratish.

#### 4. AKT sohasini texnik yo`nalishlarida tadbiiq etish tamoyillari

**Texnik tizimlarda boshqarish uchun zamonaviy texnologiyalarni qo`llash.** Boshqarish tushunchasi qadimdan ma`lum bo`lib, jamiyatni boshqarish, savdo-sotiqni tashkil etish, kemani boshqarish va boshqa sohalarda boshqarish masalasi echilib kelinadi. Tikuvchi, haydovchi, uchuvchi kasblarining barchasi mashina va mexanizmlarni boshqarish bilan bog`liq ishlarni bajaradi. Dastgoh, robot, samolyot, magnitofon yoki yadro reaktori kabi qurilmalarni boshqarishni turlicha amalga oshirish mumkin. Masalan, biror amalni bajarib, natijaga qaraladi, so`ngra boshqa amal bajariladi va shu tartibda to`xtovsiz so`nggi natijaga erishilguncha amallar ketma-ket bajariladi. Shu tartibda operatorlar yadro reaktorini, kapitanlar kemani, uchuvchilar samolyotni, kosmonavtlar kosmik kemalarni boshqaradi.

*Boshqaruv yoki menejment (ingl: management, rus: menedjment).*

1. Kerakli natijalarga erishish maqsadida insonlar va ob`ektlarga ta`sir o`tkazish. Avtomatika, informatika, texnologik jarayonlarda boshqaruv ob`ektlarga, ularning faoliyatini ta`minlash uchun maqsadli ta`sir ko`rsatishni ta`minlaydi. U asosan qurilmalar va amaliy dasturlar majmui tomonidan amalga oshiriladi. Shu bilan birga, boshqaruvda xodimlar ham ishtirok etishi mumkin. Jamiyatlarning axborot faoliyatini boshqarishning usullari va vositalari axborot menejmenti deyiladi.

2. Maqsadga qaratilgan jami harakatlar. Bunga ahvol va boshqaruv ob`ektining holatini baholash, boshqaruv ta`sirlarini tanlash va amalga oshirish ham kiradi.<sup>5</sup>

Boshqarishning vazifasi ob`ektning bir holatdan boshqa yangi holatga o`tkazishdan iborat. Jamiyat taraqqiy etgan sari boshqariladigan ob`ektlar soni ko`payib, boshqarish muammolari murakkablashib boradi. Boshqarish masalasini echishda eng avvalo, boshqarish ob`ekti va boshqarish vositasi ta`minlangan bo`lishi shart.

Boshqarish – bu tashqi muhit va ob`ekt holati to`g`risidagi axborotni qayta ishlash asosida unga uzluksiz ta`sir ko`rsatish jarayonidir.

Boshqarish ob`ekti deb, boshqariluvchi va murakkab harakatdagi tizimlarga aytiladi.

Murakkab mashina va dastgohlar, korxonalar va muassasalar, hatto odamning o`zi ham, jamiyat ham, jonli mavjudotlar, o`simliklar dunyosi ham boshqarish ob`ekti hisoblanadi. Demak, boshqarish ob`ektlari jonli va jonsiz bo`lishi mumkin. Jonli boshqarish ob`ektiga quyidagilar misol bo`la oladi: inson, jamiyat, o`simliklar, hayvonot dunyosi va boshqalar. Jonsiz boshqarish ob`ektiga esa stonok, zavod, samolyot, avtomobil, robot va boshqalar misol bo`la oladi.

Tizim – o`zaro uzviy bog`langan hamda bir maqsadga yo`naltirilgan elementlar majmuasidir.

Tizim tarkibidagi ixtiyoriy elementga ko`rsatilgan ta`sir unga bog`liq bo`lgan boshqa elementlarga ham ta`sir etadi.

Boshqarish tizimi – boshqarish ob`ekti hamda boshqarish vositalarini uzviy bog`lab, ma`lum bir maqsadni amalga oshirishga qaratilgan boshqaruvni amalga oshiruvchi tizimdir.

Demak, boshqarish sub`ektlari, ya`ni boshqaruvchi vositalar hamda boshqarish ob`ektlari birgalikda boshqarish tizimini tashkil etadi.

Har qanday ishlab chiqarish tizimi asosida boshqarishsiz amalga oshmaydigan maqsadga yo`naltirilgan harakatlar yotadi.<sup>6</sup>

Aniq predmet sohasi yoki texnologik ishlab chiqarish jaryoni masalalarini yechish, jarayonni boshqarish va rejalashtirish bilan bog`liq bo`lgan sanoat masalalari ixtisoslashgan axborot tizimlari yordamida yechiladi. Bunday tizimlarning ko`p qismini maxsus algoritmlar va qaror qabul qilish dasturlari tashkil etadi. Bir tomondan, katta tashkiliy tizimlarda, samarali boshqarishni

<sup>5</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

<sup>6</sup> Michael Henderson, Geoff Romeo. Teaching and Digital Technologies Cambridge University Press 2015, p. 83-84

tashkil qilish uchun shunday masalalar uchraydiki, unda vaziyatning o'zgarishini tezda anglash va unga muvofiq qaror qabul qilish talab qilinadi. Boshqa tomondan, shunday masalalar bo'ladiki, ularda vaziyatning o'zgarishi bilan tezda qaror qabul qilish talab qilinmaydi. Shuning uchun, axborot tizimi turli xil masalalarni, turli rejimlarda qayta ishlaydi. Tabiiyki, axborot tizimlarini ishlab chiqarishga joriy etilishi, ishlab chiqarish samaradorligini oshirishga va sarflangan xarajatlarni qoplay olishiga yo'naltirilgan bo'lishi kerak. Hamma hollarda ham qimmatli hisoblash texnikalariga sarflangan xarajatlar, iqtisodiy samaradorlik sifatini oshirish hisobiga erishilgan yutuqlar evaziga qoplanishi kerak. Shu sababli, texnologik jarayonlarning mavjud ma'lumotlari va holatlariga asoslangan holda boshqaruv uchun optimal qaror qabul qila olish algoritmlari chuqur o'ylab tuzilgan ixtisoslashtirilgan avtomatlashtirilgan boshqarish tizimlarini yaratilishi, ishlab chiqarish muassalari faoliyatida katta ahamiyat kasb etadi.

*Avtomatlashtirilgan boshqarish tizimi (ABT) (ingl: automated management system, rus: avtomatizirovannaya sistema upravleniya (ASU)).* Matematik uslublar, texnik vositalar (kompyuterlar, aloqa vositalari, ma'lumotlarni chiqarish qurilmalari va h.k.) va tashkiliy majmualar to'plami. U belgilab olingan maqsadga muvofiq murakkab ob'ekt (jarayon)ni boshqarishni ta'minlaydi.

ABT asos va funktsional qismlarga bo'linadi. Asosga axboriy, texnikaviy va matematik ta'minotlar kiradi. Funktsional qismga aniq boshqaruv vazifalarini (loyihalash, moliyaviy-hisobot faoliyati va h.k.) avtomatlashtiruvchi o'zaro bog'liq dasturlar to'plami kiradi. Ikki turdagi ABT mavjud. Bular, ob'ektlar miqyosida avtomatlashtirilgan tizimlar (texnologik jarayonlar uchun – TJABT, korxonalar uchun – KABT, soha uchun – SABT) va funktsional avtomatlashtirilgan tizimlardir. Funktsional ABTga loyihalash, hisob-kitob, moddiy-texnika va boshqa ta'minotlar uchun mo'ljallangan ABT misol bo'la oladi.<sup>7</sup>

Texnologik jarayonlarda avtomatlashtirilgan boshqarish tizimlari (TJABT) – ma'lumotlarni qayta ishlash va avtomatik tarzda yig'ishni ta'minlovchi, qabul qilingan kriteriyalar asosida (texnik, texnologik, iqtisodiy) texnologik ob'ektlarni boshqarishda kerak bo'luvchi, inson-mashina o'rtasida va texnologik boshqarish ob'ektida boshqaruvchi ta'sirlarni amalga oshirish va tanlash uchun mo'ljallangan tizim.

Texnologik boshqarish ob'ekti (TBO) – texnologik qurilmalar, texnologik jarayon ish tartibi va tegishli ko'rsatmalar bo'yicha boshqarishni amalga oshirish majmualari.

Zamonaviy ABTlarga qo'yiladigan talablar tizimlar ochiqligi (aniqligi) bo'lib, foydalanilayotgan ma'lumotlar formatlari va protsedurali interfeys ta'rif va tavsiflari, asbob-uskuna va qurilmalarni unga mustaqil ravishda birlashtirishga ruxsat etilishi lozim.

Asbob-uskunalar va avtomatlashtirish vositalari – har xil sanoat tarmoqlarida texnologik jarayonni boshqarish va rostdash, nazorat qilish, avtomatlashtirilgan o'lchash tizimlarini qurish uchun mo'ljallangan va agregatlashtirish, unifikatsiyalash, tiplashtirish tamoyillarini ta'minlash, qurilma va asbob-uskunalar majmuasi hisoblanadi.

**O'zbekistonda AKTni qo'llanilishini rivojlantirish.** 2013 yil, 16-sentyabrdagi 250-sonli Hukumat Qarori bilan, O'zbekistonda elektron hukumat tizimini yanada rivojlantirish maqsadida, maxsus markaz – “Elektron hukumat tizimini rivojlantirish markazi” hamda, “Axborot xavfsizligini ta'minlash markazi” tashkil qilindi. Mazkur qaror bilan, O'zbekistonda elektron hukumat tizimini rivojlantirish va qo'llab quvvatlash, uning me'yoriy-huquqiy bazasi, uslubiy va texnologik ta'minoti va tizim uchun mas'ul tashkilotlarning moddiy texnik bazasini mustahkamlashga oid o'ta muhim ustuvor vazifalar aniq belgilab berildi. Bundan tashqari, ushbu qaror bilan, “Elektron hukumat tizimini rivojlantirish” hamda, “Axborot xavfsizligini ta'minlash markazi” larning tuzilmasi va faoliyati tartibini belgilovchi nizom qabul qilindi.

Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada rivojlantirish va ishlab chiqarishning barcha sohalariga keng tatbiq etish, axborot resurslari, tizim va tarmoqlarini jadal rivojlantirish, tadbirkorlik sub'ektlari va aholiga ko'rsatilayotgan interfaol davlat xizmati turlarini

---

<sup>7</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

kengaytirish va yaxshilashni rag'batlantirish maqsadida O'zbekiston Respublikasi Vazirlar Mahkamasi 2013 yil 31 dekabrda "O'zbekiston Respublikasida axborot-kommunikatsiya texnologiyalarini rivojlantirish holatini baholash tizimini joriy etish chora-tadbirlari to'g'risida"gi 355-sonli qarorni qabul qildi. Ushbu qarorning 1-ilovasida baholash tuzilmasi keltirilgan, ya'ni:

#### 1-ILOVA

### **Davlat va xo'jalik boshqaruvi organlari, mahalliy davlat hokimiyati organlari faoliyatida axborot-kommunikatsiya texnologiyalarini joriy etish va rivojlantirish samaradorligini reytingli baholash tizimi TUZILMASI**

T/r	Ko'rsatkich	Ko'rsatkichning salmog'i
	<b>Yakuniy reyting,</b>	<b>100%</b>
	shu jumladan:	
	<b>I. Rasmiy veb-sayt va interaktiv davlat xizmatlari ko'rsatish</b>	<b>60%</b>
1.1.	Internet tarmog'ida rasmiy veb-saytning mavjudligi va undan foydalana olish	1%
1.2.	Rasmiy veb-saytning qonun hujjatlari talablariga muvofiqligi	5%
1.3.	"Ochiq ma'lumotlar" bo'limining mazmuni	8%
1.4.	"Elektron qatnashish" uchun imkoniyatlar	16%
1.5.	Elektron davlat xizmatlari ko'rsatish bo'yicha talablarni bajarish	30%
	<b>II. Axborot tizimlari, axborot resurslari va elektron hujjatlar aylanishi</b>	<b>16%</b>
2.1.	"Elektron hukumat" tizimi doirasida foydalaniladigan axborot tizimlari	8%
2.2.	Idoralararo elektron hujjatlar aylanishi tizimining faoliyat ko'rsatishi	4%
2.3.	Boshqa davlat organlarining axborot tizimlari bilan o'zaro hamkorlik qiluvchi ma'lumotlar bazalari	4%
	<b>III. Axborot xavfsizligi</b>	<b>20%</b>
3.1.	Axborot xavfsizligini ta'minlash tizimlari va vositalarini tashkilotning moliyalashtirish manbalari bilan tasdiqlangan yaqin yillardagi faoliyatiga joriy etish tadbirlari rejalarining mavjudligi	3%
3.2.	Axborot xavfsizligi bo'yicha tashkiliy chora-tadbirlarning mavjudligi va ularning amalga oshirilishi holati	7%
3.3.	Axborot xavfsizligini ta'minlash va xodimlarini himoya qilish bo'yicha jismoniy chora-tadbirlarning mavjudligi va amalga oshirilishi holati	4%
3.4.	Axborot xavfsizligini ta'minlash texnik vositalarining mavjudligi va holati	6%
	<b>IV. Tashkiliy jihatdan baholash</b>	<b>4%</b>
4.1.	Tashkilotning moliyalashtirish manbalari bilan tasdiqlangan, yaqin yillardagi faoliyatiga axborot-kommunikatsiya texnologiyalarini joriy etish va rivojlantirish tadbirlari rejalarining mavjudligi	1%
4.2.	Davlat xizmatlarini interaktiv shaklga bosqichma-bosqich o'tkazish bo'yicha o'tkazishning jadvallari va sanalari ko'rsatilgan holdagi chora-tadbirlar dasturining mavjudligi	1%
4.3.	Axborot-kommunikatsiya texnologiyalarini joriy etish va rivojlantirish yuzasidan rahbariyat orasidan mas'ul shaxsning mavjudligi	1%
4.4.	Axborot-kommunikatsiya texnologiyalarini joriy etish va rivojlantirish yuzasidan mas'ul bo'lgan ixtisoslashtirilgan tarkibiy bo'linmaning yoki tegishli xizmatlar outsorsingi bo'yicha shartnomaning mavjudligi	1%

#### **5. Tizimli va amaliy dasturiy ta'minotning rivojlanish tendensiyalari**

Kompyuterlar ikkita tashkiliy qismlardan iborat, yani apparat (hardware) taminot va dasturiy (software) taminot qismlari.



Apparat ta'minoti - bu, birinchi navbatda kompyuterning asosiy texnik qismlari va qo'shimcha (atrof) qurilmalaridir.

*Apparat ta'minoti (ingl: hardware, rus: apparatnoe obespechenie)* - hisoblash tizimi tarkibiga kiruvchi, uning faoliyat ko'rsatishi uchun zarur bo'lgan jami apparat vositalari. Apparat ta'minotiga kompyuter, tashqi qurilmalar, aloqa tarmoqlari va h.k. kiradi. Ular texnik nuqtai nazardan tizimning samarali ishini, undan foydalanuvchiga ma'lum xizmat turlarini ko'rsatishni ta'minlaydi.<sup>8</sup>

Kompyuter telefon, magnitafon yoki televizordan farqli o'laroq oldindan berilgan buyruqlarga asosan ixtiyoriy hajmdagi ma'lumotlar ustida axborot jarayonlarini amalga oshiruvchi qurilmadir. Axborot jarayonlarini amalga oshirishi uchun kompyuterda tushunarli va aniq bir tilda berilgan buyruqlar ketma-ketligi, ya'ni dasturlarni ishlatish kerak. Shunday dasturlarning majmui *kompyuterning dasturiy ta'minotini* tashkil qiladi va ularning imkoniyatlarini, quvvatlarini aks ettiruvchi asosiy vositalardan biri hisoblanadi.

*Dastur (ingl: software, rus: programma).*

1 Hisoblash mashinasiga algoritmi beruvchi ko'rsatmalar (buyruq yoki tasnif va operatorlar)ning ketma-ketligi. Dastur kompyuter tomonidan qaysi tartibda, qaysi ma'lumotlar ustidan va qaysi amallar bajarilishi va natija qaysi shaklda taqdim etilishini ko'rsatadi. Kompyuterning boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi. Dasturni mashina tilida yaratish – noqulay va katta mehnat talab qiluvchi jarayon. Shuning uchun kompyuter uchun dastur inson tomonidan dasturlash tillaridan birida yaratilib, keyin esa kompyuterning o'zi ushbu dasturni mashina tiliga o'giradi.

2 Ma'lum natijani olish uchun kompyuter va boshqa kompyuter qurilmalari faoliyati uchun mo'ljallangan jami ma'lumot va buyruqlarni taqdim etishning ob'ektiv shakli.

*Dasturiy ta'minot (ingl: software, rus: programmnoe obespechenie).* Axborotga ishlov berish tizimining barcha yoki ba'zi dasturlari, tartiblari, qoidalari va ularga tegishli hujjatlar.<sup>8</sup>

Dasturiy ta'minot kompyuter tizimining ajralmas tarkibiy qismi va texnik vositalarning mantiqiy davomidir. Muayyan kompyuterlarning qo'llanish sohasi uning dasturiy ta'minoti bilan aniqlanadi. Zamonaviy kompyuterning dasturiy ta'minoti kompyuterning ish jarayonini tashkil qilish, ular qurilmalarini o'zaro muvofiqlikda ishlashini ta'minlash, inson va kompyuter orasidagi muloqotni tashkil qilish, inson faoliyatining ko'ngil ochar, ishlab-chiqarish, boshqarish, ilmiy singari turli sohalariga taalluqli bo'lgan millionlab muammolarni hal qiluvchi dasturlarni o'z ichiga qamrab oladi.

Kompyuterning texnik va dasturiy ta'minoti orasida bog'lanish amalga oshirilishi tufayli ma'lum bir ish amalga oshiriladi. Kompyuterning turli texnik qismlari orasidagi o'zaro bog'lanish - bu, *texnik interfeysi*, dasturlar orasidagi o'zaro bog'lanish esa - *dasturiy interfeys*, texnik qismlari va dasturlar orasidagi o'zaro bog'lanish - *texnik - dasturiy interfeys* deyiladi.

Kompyuter tizimi bilan ishlashda uchinchi ishtirokchini, ya'ni insonni (foydalanuvchini) ham nazarda tutish lozim. Inson kompyuterning texnik, ham dasturiy ta'minotlari bilan muloqotda bo'ladi. Insonning dastur bilan va dasturni inson bilan o'zaro muloqoti - *foydalanuvchi interfeysi* deyiladi.

Barcha dasturiy ta'minotlarni uchta kategoriya bo'yicha tasniflash mumkin:

- A. Tizimli dasturiy ta'minot.
- B. Amaliy dasturiy ta'minot.
- C. Dasturlash texnologiyasining uskunaviy vositalari.

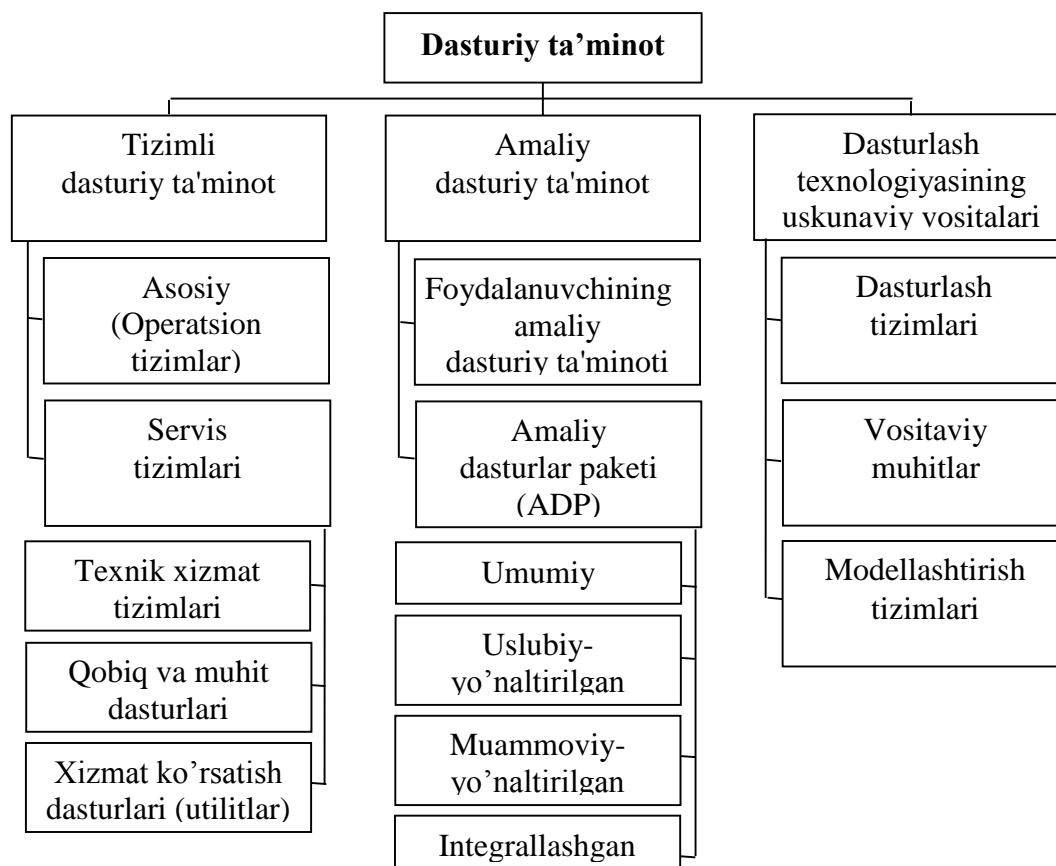
A. *Tizimli dasturiy ta'minot (ingl: system software, rus: sistemnoe programmnoe obespechenie).* Hisoblash tizimi tarkibiga kiruvchi jami tizimli dasturlar. Bu kompyuterdan foydalanish va xizmat ko'rsatish, hisoblash ishlarini tashkillashtirish va amaliy dasturlarni yaratishni avtomatlashtirish uchun zarur bo'lgan dasturiy ta'minot. Tizimli dasturiy ta'minotning eng muhim tarkibiy qismi – operatsion tizim – apparat vositalar uchun zarur qo'shimcha bo'lib,

---

<sup>8</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

odatda hisoblash tizimini ishlab chiqaruvchi tomonidan etkazib beriladi. Tizimli dasturiy ta'minotning boshqa tarkibiy qismlari kompyuterga foydalanuvchini qoniqtiradigan hisoblash tizimini yaratish uchun uning o'zi tomonidan o'rnatiladi.<sup>8</sup>

Tizimli dasturiy ta'minot (TDT) dasturlari kompyuterning protsessor, operativ xotira, kirish-chiqish kanallari, tarmoq qurilmalari singari resurslarini unumli boshqarish, foydalanilayotgan axborotning nusxalarini tuzish, kompyuter qurilmalarining ishlash imkoniyatini nazorat qilish, kompyuter haqida ma'lumotnoma taqdim qilish singari ishlarni bajaradi. Uning amaliy dasturiy ta'minotdan farqi shundaki, u aniq bir amaliy masalani echmaydi, balki boshqa dasturlarning ish jarayonini ta'minlaydi va kompyuter apparat ta'minoti resurslarini boshqaradi hamda ular bilan foydalanuvchi muloqotini tashkil qiladi. Tizimli dasturiy ta'minot ikki qismdan iborat. Ular: asosiy va servis dasturiy ta'minotdir.



*Asosiy dasturiy ta'minot (baze software)* - bu, kompyuter ishini ta'minlovchi dasturlarining eng kam to'plamidan iborat. Unga operatsion tizimlar kiradi.

*Operatsion tizim (OT)* – foydalanuvchi va kompyuter o'rtasidagi muloqotni tashkil etuvchi o'zaro bog'liq dasturlarning tizimli majmuasidir. Ular kompyuter resurslarini taqsimlash va ishlatishni hamda barcha apparat vositalari ishini boshqaradi.

*Servis (xizmat ko'rsatuvchi) dasturiy ta'minotga* asosiy dasturiy ta'minot imkoniyatlarini kengaytiruvchi va foydalanuvchining ish muhitini (interfeysni) qulayroq tashkil etuvchi dasturlar kiradi. Bular texnik xizmat tizimlari, qobiq va muhit dasturlari, Xizmat ko'rsatish dasturlari (utilitlar), tarmoq ishini ta'minlovchi va boshqa dasturlardir.

*Texnik xizmat tizimlari* - kompyuterning har bir qurilmalarini ishlash imkoniyatlarini tekshirish, testlash, diagnostika, nazorat qilish va yuzaga kelgan yoki potentsial nosozliklarni aniqlash.

Ularga tegishli bo'lgan ba'zi bir dasturlarni keltiramiz:

- Kompyuter ish qobiliyatini diagnostika qilish dasturlari.
- Diskli yig'uvchilar va ularda saqlanadigan axborotlarga xizmat qilishga mo'ljallangan dasturlar sinfi.

- Defragmentatsiyalash – ma'lumotlarni diskda joylashuvini tartiblash va ketma-ket joylashtirish maqsadida qo'llaniladi.
- Diskni tekshirish – noto'g'ri yozilgan yoki zararlangan fayllar va disk qismlarini qidirish, disk samarili ishlashi uchun ularni o'chirish yoki yopib qo'yish.
- Diskni tozalash – vaqtincha ishlatilgan, keraksiz fayllarni o'chirish, korzinani tozalash.
- Disk qaydini tuzish (razmetka) – fayl va papkalarni tartibga solish, xavfsizlikni ta'minlash maqsadlarida diskni bir nechta mantiqiy disklarga bo'lish.
- Zahiraviy nusxalash – disklarni yoki alohida fayllar zahirasini nusxalash hamda ushbu nusxalardan qayta tiklash imkoniyatini yaratadi.
- Diskni siqish – qattiq disk sig'imini oshirish maqsadida undagi ma'lumotlarni siqishni amalga oshiradi.
- Reestrlar bilan ishlash.
- Qurilmalarni monitoringini olib borish.
- Qurilmalarni testlash.

*Qobiq dasturlar* - dasturiy mahsulotlarning soddalashtirilgan sozlovchisi hisoblanadi. Qobiq dastur foydalanuvchiga sifat jihatidan yangi interfeys taqdim etadi. OTlarning sozlovchisi bo'lgan qobiq dasturlarlar operatsion qobiqlar deb ataladi. Ko'pgina qobiq dasturlar fayllar va kataloglar bilan ishlashga yo'naltirilgan. Masalan, Windows OT uchun Total Commander, FAR (File and Archive manager).

*Xizmat ko'rsatish dasturlari (utilitlar)* – yordamchi dasturlar bo'lib, ular ishlashda qulaylik va bekamu-ko'stlikni ta'minlashga xizmat qiladi.

*Utilitalar* - kompyuterning apparat vositalariga, ayniqsa disklarga, operatsion tizimlar va boshqa ilovalarga qo'shimcha servis xizmat qilishga mo'ljallangan dasturlar hisoblanadi<sup>9</sup>.

*Utilita (ingl: utility, rus: utilita)*. Kompyuter va kompyuter dasturlariga texnik xizmat ko'rsatish quroli bo'lmish xizmat dasturi. Utilitalar kompyuter tizimlarini sinovdan o'tkazish, operatsion tizim yoki uning qismlarini testlash va qayta tiklash, buzilgan yoki yo'qotilgan fayllarni qayta tiklash va h.k. uchun xizmat qiladi.<sup>10</sup>

Ularni vazifaviy alomatlarini bo'yicha qo'yidagicha tasniflash mumkin:

- *Arxivatorlar* – fayllarni siqish orqali hajmini qisqartirib, ma'lum bir arxivli fayl yaratuvchi dasturdir.
- *Antivirus dasturlari* - viruslarni topish va davolashni ta'minlaydi. Virus turli nomaqbul harakatlarni amalga oshirib, boshqa dasturlarga kirib olgan holda ko'payishga qodir bo'lgan dasturlardir.
- Magnitli disk sirtining sifatini tekshirish, mantiqiy va jismoniy darajadagi faylli tizimning saqlanishini nazorat qilish, disklarni siqish, disklarning sug'urta nusxalarini yaratish, tashqi manbalardagi ma'lumotlarni rezervlashtirish va boshqalarni ta'minlovchi disklarga xizmat ko'rsatish dasturlari.
- Disk muhiti sifatini nazorat qilish va optimallashtirish.
- Ma'lumotlarni himoya qilish, formatlash va tiklashni ta'minlovchi dasturlar.
- Kompakt disklarga yozish dasturlari.
- Tarmoqqa xizmat ko'rsatish dasturlari.
- *Drayverlar* - amaliy dasturlar va OTlar bilan qurilmalarni o'zaro hamkorligini ta'minlovchi maxsus dasturlar.

*Drayver (ingl: driver, rus: drayver)*. Boshqaruvchi dastur. Odatda, bu bajarilayotgan dasturning ma'lum moslama bilan o'zaro ishlashini ta'minlovchi va undan qulay foydalanishga yordam beruvchi amaliy tizimning dasturidir. Masalan, klaviatura, displey, sichqoncha, printer va shular kabi drayverlar mavjud. Drayver dasturlarning moslamaga qaratilgan buyruqlarini qabul qilib, ularni moslamani boshqarish buyruqlariga aylantiradi, shuningdek u xizmat ko'rsatilayotgan

<sup>9</sup> Justin Zobel, Writing for Computer Science, Third Edition, Department of Computing and Information Systems The University of Melbourne, Parkville Australia, 2014.

<sup>10</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

moslamadan uzilishlarni qayta ishlaydi. Bunda drayver moslamaning tuzilishidagi xususiyatlar va vaqtning voqeyi ko'lamidagi ishlash xususiyatlarini hisobga oladi. Moslama mumkin bo'lgan moslamalar ro'yxatiga kiritilgan bo'lsa, bunday moslama drayveri odatda amaliy tizim tarkibiga kiradi. Moslamalar drayverlari kompyuter yoqilganda avtomatik tarzda yuklanib, undan keyin foydalanuvchi uchun ko'rinmas tarzda bajariladi.<sup>11</sup>

Oddiy qilib aytganda drayver, qurilmalarni kompyuterga tanishtiradi va ular ishini boshqaradi.

Utilitalar quyidagi vazifalarni bajarishga yo'l qo'yadi:

- Qurilmaning ish unumdorligi monitoringi;
- Protsessor harorati, videoadapter, qattiq disk monitoringi;
- Qurilma parametrlarini boshqarish – CD disklarning maksimal aylanish tezligini cheklash, ventilyatorlarning aylani tezligini o'zgartirish;
- Ko'rsatkichlar nazorati – murojaatlarni va ma'lumotlarni yozish to'g'riligini tekshirish;
- Kengaytirilgan imkoniyatlar – ma'lumotlarni saqlab diskni formatlash va qayta tiklash, ma'lumotlarni tiklab bo'lmaydigan darajada o'chirish;

Ba'zi bir utilitlar operatsion tizim tarkibida taqdim qilinsa, ba'zilar esa mustaqil dasturiy ta'minot sifatida bozorda sotiladi.

B. Amaliy dasturiy ta'minot (ADT) - bu aniq bir predmet sohasi bo'yicha ma'lum bir masalalar sinfini yechishga muljallangan dasturlar majmuasidir.

Kompyuterning dasturiy ta'minoti orasida eng ko'p qo'llaniladigani ADT hisoblanadi. Bunga asosiy sabab, kompyuterlardan inson faoliyatining barcha sohalarida keng foydalanishi, turli predmet sohalarida tegishli muammolarni ular orqali yechilishidir. Ular tuzilishi bo'yicha oddiy yoki murakkab bo'lgan dasturlardan iborat bo'lishi mumkin. Masalan: matn ma'lumotlarni o'qish, yaratish va tahrirlash, rasm va tasvir ma'lumotlarni ko'rish, yaratish yoki o'zgartirish, musiqa va video ma'lumotlarni ko'rib chiqish va tahrirlash, ma'lumotlarni qogozga chop qilish, tasvirlar va hujjatlar nusxasini olish, matnlarni bir tildan boshqa tilga tarjima qilish, planirovchik yoki organayzerlar, ya'ni ish vaqtini rejalashtiruvchi, uchrashuvlar bayonnomalarini, jadvallarni tuzuvchi, telefon va yozuv kitoblarini olib boruvchi, skaner yordamida o'qilgan axborotni tanib oluvchi va matnli ifodaga o'zgartiruvchi, tarmoqdagi uzoq masofada joylashgan abonent bilan foydalanuvchi orasidagi o'zaro muloqotni tashkil etuvchi kommunikatsiya dasturlari va h-zolar.

*Foydalanuvchining amaliy dasturiy ta'minoti* – bu biror bir aniq muammoni echishga yo'naltirilgan dastur yoki dasturlar majmuasidir.

*Amaliy dasturlar paketlari (ADP)* - bu masalalarning belgilangan sinfini echish uchun hamkorlikdagi dasturlarning majmuasidir.

*Umumiy* maqsadli dasturlar ko'pchilik foydalanuvchilar ishlatadigan dasturlarni birlashtiradi va o'rganish uchun axborot texnologiyalariga oid maxsus bilimlarni talab etmaydi. Ularga quyidagilarni misol sifatida keltiramiz:

- Matn muharrirlari (masalan, MS Word, Word Perfect, Leksikon).
- Elektron jadvallar (masalan, MS Excel, Lotus 1-2-3, Quattro Pro).
- Taqdimot dasturlari (masalan, MS Power Point, Freelance Graphics, Harvard Graphics).
- Grafik muharrirlar (masalan, Sorel Draw, Adobe Photoshop).
- Nashriyot tizimlari (masalan, Page Maker, Venture Publisher).
- Avtomatik loyihalash tizimlari (masalan, AutoCad, BPWin, ERWin).
- Elektron lugatlar va tarjimon tizimlari (masalan, Prompt, Sokrat, Lingvo, Kontekst);
- Matnni tanish tizimlari (masalan, Fine Reader, Cunei Form).

Umumiy dasturlarning ko'pchiligi alohida paket sifatida birlashtirilib taqdim qilinadi. Masalan, ofis paketlari Microsoft Office, StarOffice va h-zo.

---

<sup>11</sup> Axborot kommunikatsiya texnologiyalari izohli lug'ati. BMT Taraqqiyot dasturining O'zbekistondagi vakolatxonasi Toshkent 2010.

*Uslubiy-yo`naltirilgan* dasturlar asosan matematik usullar orqali muammoni hal qilishga yo`naltirilgan bo`ladi. Masalan, ma`lumotlarga matematik ishlov berish (Mathematica, MathCad, MatLab, Maple), ma`lumotlarga statik ishlov berish (Statistica, Stat).

*Muammoviy-yo`naltirilgan* dasturlar aniq bir predmet sohasiga tegishli muammoni hal qilishga yo`naltirilgan bo`ladi. Masalan, huquqiy axborot tizimi (LexUz), buxgalteriya hisob-kitoblari (1S, maosh, komunal to`lovlar va h-zo).

*Integrallashgan* dasturlar bir vosita sifatida birlashtiriladi va ular aniq bir muammoni echishga yo`naltirilgan bo`ladi. Masalan, ma`lumotlar bazasini boshqarish tizimlari (MS Access, Oracle, MS SQL Server, My SQL, Informix), ta`lim dasturlari (CourseLab, Inspiring va h-zo).

A. Dasturlash texnologiyasining uskunaviy vositalari - yangi dasturlarni ishlab chiqish jarayonida qo`llaniladigan maxsus dasturlar majmuasidan iborat vositalardir.

Amaliyotda algoritm ijrochisi kompyuter hisoblanadi. Hozirgi zamon kompyuterlari inson qo`llaydigan rus, o`zbek yoki boshqa shu kabi biror-bir tilda tuzilgan algoritmni tushuna oladigan darajada takomillashgan emas. Shu bois, mashinaga mo`ljallangan buyruqlar kompyuter tushunadigan shaklda yozilishi zarur. Bu maqsadda algoritmik yoki dasturlash tillari deb ataluvchi sun`iy tillar qo`llaniladi. Algoritm ushbu sun`iy tilda yozilishi kerak. Ijrochi (kompyuter) berilgan buyruqlarni aniq va to`g`ri bajarishi uchun unga beriladigan buyruqlar aniq yozilishi va ijrochi tomonidan bir xil tushunilishi lozim. Shu sababli kompyuter tushunadigan til dasturlash tili, bu tilda yozilgan algoritm esa kompyuter dasturi deyiladi.

*Dasturlash tizimlariga* C++, Delphi, Visual Basic, Java, PHP singari dasturlash tillar kiradi.

*Vositaviy muhitlar* esa dasturlash tillari uchun matn muharriri, translyatorlar singarilarni taqdim qiladi. Ular: Borland Delphi, Microsoft Visual Basic, Borland C++, JavaScript, Java, PHP, Perl, ASP, LISP, Prolog, Action Script, Flash, C# muhitlari va h-zolar.

*Translyator* – bu dastur matnini mashina kodiga o`giruvchi dasturdir.

*Modellash tirish tizimlariga* imitatsiyali modellash tirish tizimi MatLab, biznes-jarayonlarni modellash tirish tizimi BpWin va ErWin singarilar kiradi.

Dasturiy ta`minot sanoati har kuni bozorga yuzlab yangi dasturiy mahsulotlarni taqdim qilmoqda. Ularning barchasini o`rganib chiqish, hatto shu soha mutaxassislari uchun ham etarlicha qiyin hisoblanadi. Shuningdek, foydalanuvchilarning ma`lum bir ehtiyojini qondirish uchun esa bir necha o`nlab dasturiy mahsulotlar taklif qilinadi. Ko`pchilik hollarda qaysi dasturiy mahsulotni tanlash masalasini oddiy foydalanuvchi hal qilishi lozim. Tanlash muammosi nuqtai nazarlaridan biri tanlangan dasturiy mahsulotni rivojlanish tendensiya (istiqboli) dasturiy ta`minotning umumiy rivojlanish istiqboliga muvofiqligi hisoblanadi. O`z navbatida, dasturiy ta`minotning umumiy rivojlanish istiqboli haqida bilimga ega bo`lishlik, u yoki bu dasturiy mahsulotni asoslangan holda tanlashga imkon beradi.

Dasturiy ta`minotning rivojlanishi asosiy istiqbollari yo`nalishlari quyidagilardir:

- o Dasturiy muhit komponentlari hamda ular orasidagi interfeyslarni standartlash - u yoki bu ilovani turli apparat platformalarda va turli operatsion tizim muhitlarida ishlatish hamda keng tarqalgan ilovalar bilan o`zaro hamkorligini ta`minlash.

- o Ob`ektga yo`naltirilgan loyihalash va dasturlashga yo`naltirilganlik – yangi texnologiyaga o`tishda aniq bir foydalanuvchining aniq bir talabiga mos u yoki bu ilovalar alohida modullarini va standrtlarini jamlash. Bu holda hajm va narx kamayadi va ishonchilik oshadi. Shuningdek, ilovani takomillashtirishda vujudga keladigan ko`pgina muammolar echiladi.

- o Foydalanuvchi talablarini shakllantirilganligi – tarmoqda ishlash hamda talab qilinadigan dastur resurslariga murojaatni hisobga olgan holda “yig`ish” texnologiyasiga yo`naltirilganlik, kelgusida u yoki bu ilovani “shaxsiy foydalanish” xususiyatiga ega bo`lishlikka yo`l qo`ymaydi hamda dasturiy ta`minot sarf-xarajatini kamaytiradi.

- o Foydalanuvchi interfeysini intellektuallashtirish, uni intuitiv tushuncha bilan ta`minlash, protsedurasizlik va kompyuter bilan muloqot tilini foydalanuvchi kasbiy tiliga yaqinlashtirish. Foydalanuvchi interfeysini aniq bir foydalanuvchining kompyuter bilan muloqoti xususiyati va talabiga muvofiq sozlash. Foydalanuvchi interfeysini tadbiiq qilganda mul`timedia vositalaridan foydalanish.

○ Dastur va dasturiy tizimning imkon darajasida intellektuallashtirish. Qiyin rasmiylashtiriladigan va ancha murakkab masalalarni echish uchun ilovalarni loyihalashtirganda sun'iy intellekt usullarini keng ishlatish orqali ilovalarni ancha "aqlli" qilish.

○ Dasturiy mahsulotning alohida komponent (modul)larini universallashtirish orqali dasturiy mahsulotni ham universallashtirish. Asta-sekin dasturiy mahsulotni ixtisoslashgan dasturiy ta'minot guruhidan chiqarib, uni universal amaliy dasturiy ta'minot guruhiga o'tkazishga erishish. Natijada, ushbu dasturiy mahsulot keng foydalanuvchilar uchun hammabop maqomiga erishadi.

○ Foydalanuvchilar biror muammoni dasturiy vosita yordamida hamkorlikda, guruh sifatida ishlab, hal qilishlariga yo'naltirish. SHu tufayli, dasturiy ta'minot ishlanmasini yaratayotganda e'tiborni ko'proq kommunikatsiya komponentlariga qaratish lozim.

○ Mobil qurilmalarda ham ishlay oladigan dasturiy ta'minotni ishlab chiqib, uni tadbir etilishiga erishish.

Shuni ta'kidlash lozimki, ko'rsatilgan istiqbolli yo'nalishlar yana bir qat'iy talabni, kompyuterningsonli va sifatli apparat tavsifiga ehtiyojni vujudga keltiradi. Ammo, hozirgi kunda apparatlarning potentsial imkoniyatlari ko'rsatilgan talablarni qondirishdan ancha yuqori bo'lgani uchun ko'rsatilgan istiqbolli yo'nalishlarni axborot texnologiyalarida to'liq tadbir qilishga imkon bera oladi.

### **Nazorat savollari:**

1. Informatika nima?
2. Kompyuter nima?
3. Axborot nima?
4. Texnologiya va jarayon nima?
5. Axborot texnologiyalari nima?
6. Axborot kommunikatsiya texnologiyalari nima?
7. Axborot texnologiyalari amaliy fan sohasi sifatida nima bilan shug'ullanadi?
8. O'zbekiston Respublikasida AKT sohasidagi Qonunlar.
9. O'zbekiston Respublikasi Prezidentining AKT sohasidagi farmonlari va qarorlari.
10. O'zbekiston Respublikasi Vazirlar mahkamasining AKT sohasidagi qarorlari va me'yoriy hujjatlari.
11. Respublikada kompyuterlashtirish va AKTni rivojlantirish dasturi va undagi vazifalar.
12. O'zbekiston Respublikasida 2013-2020 yillar mobaynida telekommunikatsiya texnologiyalari, tarmoqlari va infrastrukturasi rivojlantirish dasturi vazifasi nima?
13. Axborot texnologiyalarining ijtimoiy sohalar, iqtisodiy va boshqaruvdagi roli.
14. Tizim deganda nimani tushunasiz?
15. Qanday tizimlarni bilasiz?
16. Axborot tizimlari deganda nimani tushunasiz?
17. Axborot tizimlarining rivojlanish bosqichlarini tavsiflang.
18. Axborot tizimlarining tuzilmasi.
19. Avtomatlashtirilgan tizimi deganda nimani tushunasiz?
20. Avtomatik tizimi deganda nimani tushunasiz?
21. Avtomatlashtirilgan tizimda axborotga ishlov berish deganda nimani tushunasiz?
22. Boshqaruv deganda nimani tushunasiz?
23. Axborot tizimi ishini ta'minlovchi jarayonlar deganda nimani tushunasiz?
24. Avtomatlashtirilgan tizimda axborotga ishlov berish deganda nimani tushunasiz?
25. Axborot tizimlarini tashkil etish va boshqarish deganda nimani tushunasiz?
26. Avtomatlashtirilgan boshqarish tizimi deganda nimani tushunasiz?

## 2 modul. MS EXCEL DASTURINI KIMYO INJINIRINGDA QO'LLASH

### 2 mavzu. MS Excel electron jadvali haqida umumiy tushunchalar

#### Reja:

1. MS Excel electron jadvalining ishchi oynasi, menyu bo'limlari.
2. MS Excel electron jadvali va listlarni boshqarish.
3. Funktsiyalar masteri bilan ishlash.
4. Diogrammalar, grafiklar tuzish va boshqa shu kabi imkoniyatlardan foydalanish.

**Tayanch iboralar:** Elektron jadval, lenta, ustun, qator, katak, formula, parol, himoya.

#### 1. MS Excel electron jadvalining ishchi oynasi, menyu bo'limlari




Kundalik hayotimizda har-xil ma'lumotlarni va hisob-kitoblarni turli jadvallar ko'rinishida taqdim qilinishini ko'p ko'rganmiz. Bu holatda barcha ma'lumotlar jadval kataklarida tartib bilan berildi. Bu shakldagi jadvallarni MS Word matn muharririda chizib, unga turli ma'lumotlar joylashtirgach, jadval shaklidagi elektron hujjat vujudga keladi. Endi ushbu elektron hujjat faylini bir kompyuterdan boshqasiga olib o'tish, kerakli nusxada chop etish, kompyuter tarmoqlari orqali boshqa joylarga jo'natish mumkin bo'ladi. Agar ma'lumotni o'zgartirishga to'g'ri kelsa, elektron hujjat fayli MS Word matn muharririda ochiladi va kerakli o'zgartirishlar amalga oshiriladi. Bu jadvallarda hisob-kitob natijalari saqlanadi. Hozirgi kunda ma'lumotlarni jadval shaklida saqlovchi va ularda avtomatik ravishda hisob-kitob ishlarini bajaruvchi maxsus dasturlar yaratilgan bo'lib, ular elektron jadvallar deyiladi.

Dastur nuqtai nazaridan qaralsa elektron jadvallar, jadvallar ustida turli vazifalarni bajaruvchi dasturlar majmuasidir. Uning yordamida jadvallar tuzish, turli hisobotlarni amalga oshirish, berilgan matematik ifodalar, formulalar va funksiyalar yordamida masalalarni hal qilish, jadvaldagi qiymatlar bo'yicha turli diogramma va grafiklarni qurish va bosmaga chiqarish ishlarini bajarish mumkin.

Elektron jadvallarni yaratish fikri AQSHning Garvard universiteti talabasi Den Briklandan chiqdi va uni 1979 yili yaratdi. U elektron jadval dasturini Visicalc deb nomladi. Keyingi elektron jadval 1982 yili Lotus firmasi tomonidan yaratildi va Lotus deb ataldi. Keyinchalik turli kompyuter va tizimlarda ishlay oladigan Supercalc, Multiplan, Framework, Symphony, Works, Master kabi jadval muharrirlari ishlab chiqarildi. 1987 yili Microsoft firmasi o'zining Excel deb atalgan elektron jadvalini yaratdi. MS Excel elektron jadvalida ishlash texnologiyasi xuddi Word matn protsessorida ishlash texnologiyasiga o'xshash (umuman olganda Microsoft firmasi dasturiy mahsulotlari ilovalari dizaynlari va ularda ishlash texnologiyalari bir-biriga o'xshash). Shuning uchun ham MS Excel dasturini elektron jadvallarning eng qulayi deyishimiz mumkin.

Hozirgi kunda Excel 2007, 2010, 2013 va 2016 versiyalari dunyo bozorida o'zining yangi imkoniyatlari bilan xaridorgir bo'lib qolmoqda. Bu dasturlar Visual Basic dasturlash tili bilan muvofiqlashuvi tufayli o'zining imkoniyatlarini yanada oshirib boradi.

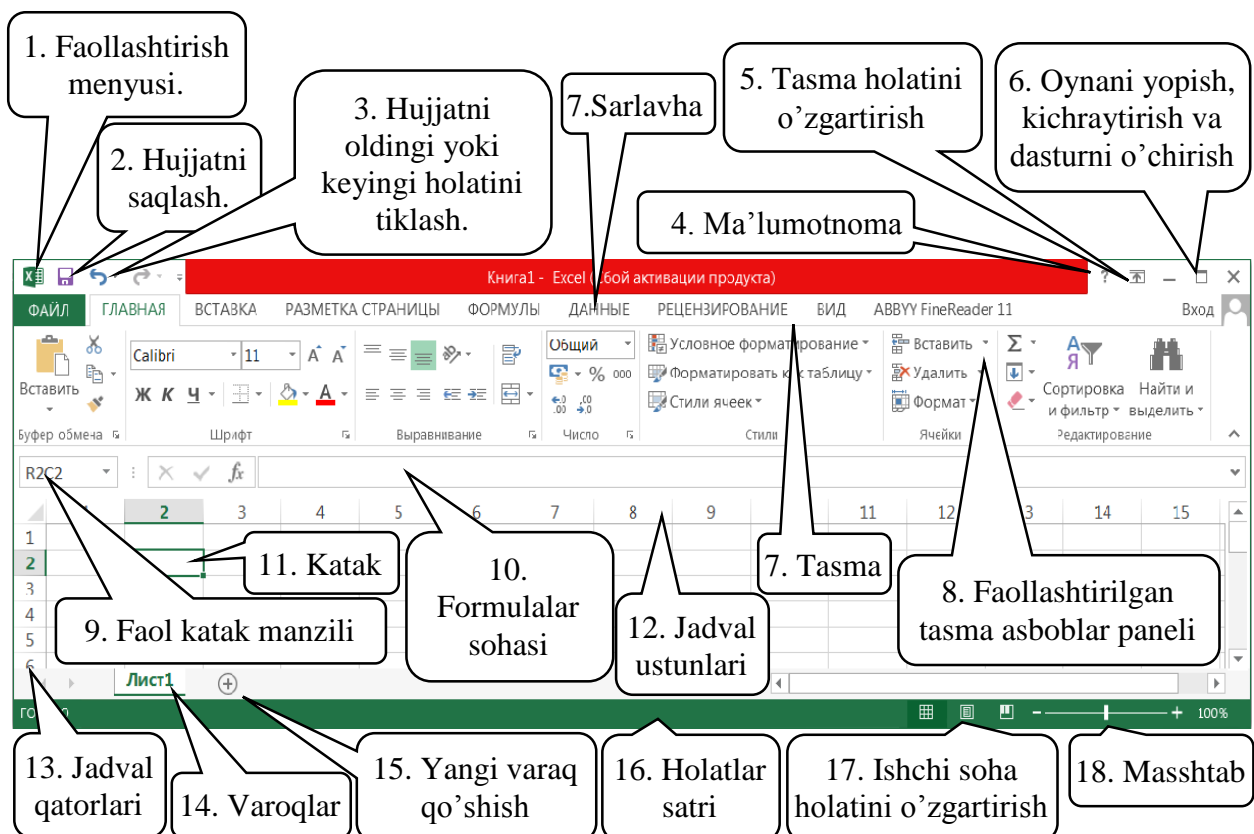
Excel 2007, 2010, 2013 va 2016 versiyalari ilova dizayni bir-biriga o'xshash, faqat har bir versiya o'zida oldingi versiyaga nisbatan ichki imkoniyatlari ancha oshirilgan va takomillashtirilgan. Biz Excel 2013 va Excel 2010 ni tahlil qilamiz. Uni ishga tushirish bir necha usullar bilan amalga oshirilishi mumkin.

1. “Пуск - Все программы - Microsoft Office 2013 - Excel 2013” buyruq orqali.
2. Ish stolidagi  yorliqni ishga tushirish (Agar yorliq ishchi stolda mavjud bo'lsa).
3. Ish stoli masalalar panelidagi  yorliqni ishga tushirish (Agar yorliq masalalar panelida mavjud bo'lsa).
4. “Пуск” menyusidan  yorliqni ishga tushirish (Agar Excel yaqin kunlarda ishlatilgan bo'lsa, u “Пуск” menyuning chap qismida turadi).
5. Excelda yaratilgan biror bir hujjatni ishga tushirish (Bu holda Excel oynasida, chaqirilgan hujjat ma'lumotlari tasvirlanadi).

Excel dasturi ishga tushirilsa, yangi yaratilgan hujjatga “Книга1”, ... “Книган” nomi beriladi. Excelda yaratilgan fayl kitob (kniga) deb ataladi.

Excel dasturini ishga tushirilganda quyidagi oyna ochiladi.





#### MS Excel elektron jadvali oynasining asosiy elementlari:

1. Sarlavha - dastur va fayl nomi ko'rsatilgan birinchi qator.

Bu qatorda tez-tez ishlatiluvchi, murojaatni osonlashtiruvchi panel mavjud bo'lib, ular yordamida xotiraga saqlash, oxirgi amalni bekor qilish kabi amallarni tez va oson bajarish mumkin. Ushbu panelga yangi buyruqlar kiritish orqali imkoniyatini yanada oshirsa bo'ladi. Qatorning chap qismida birinchi element faollashtirish yoki bosh menyu (Menyuda "Восстановить, Переместить, Размер, Свернуть, Развернуть, Заккрыть" buyruqlari mavjud) joylashgan. Keyingi element hujjatni qattik diskka yozish tugmasi hisoblanadi. Ixtiyoriy vaqtda ushbu tugmani bosib, hujjatni saqlash mumkin. Navbatdagi elementlar hujjatning oldingi va keyingi holatlariga o'tkazish tugmalaridir. Sarlavhaning markazida elektron jadval fayli va elektron jadval dasturi nomi yoziladi. Bu sohadan keyin so'roq belgisi, dastur haqida ma'lumotnoma beruvchi yordam tugmasi joylashgan. Agar ushbu tugma bosilsa, yordam oyna ochiladi va undan kerakli ma'lumotlarni o'qib olish mumkin. Keyingi tugma lenta holatini o'zgartiradi. Tugma bosilsa menyu меню ("Автоматически скрывать ленту, Показывать вкладки, Показывать вкладки и команды" buyruqlari mavjud) ochiladi va kerakli buyruqni tanlash orqali lenta holati o'zgartiriladi. Oxirgi uchta tugma dastur oynasi holatini o'zgartirish uchun ishlatiladi.

2. Lenta – elektron jadvalda muayyan vazifalarni bajarishga mo'ljallangan buyruqlar majmualari guruhi qatori.

MS Excel 2013 elektron jadvalida yangi interfeysi asosida lenta (ingliz tilida "Ribbon") deb nomlanuvchi va asosiy oynaning yuqori qismida joylashgan ko'p varaqli soha joylashgan. Lentaning har bir bo'lagida muayyan vazifalarni bajarishga mo'ljallangan tugmalar majmuasi jamlangan.

Lentalar vazifasi:

- "Главная" – jadvalga ma'lumotlarni kiritishga va tahrirlashda qo'llaniladigan amallar majmuasi;
- "Вставка" – jadvalga belgi, rasm, diagramma kabi obyektlarni joylashtirishda qo'llaniladigan amallar majmuasi;



- “Разметка страницы” - jadval varaqlari chegaralari va boshqa parametrlarini belgilashda qoʻllaniladigan amallar majmuasi;
- “Формулы” – jadvallardagi hisoblashlarni amalga oshiruvchi har xil formulalardan foydalanishda qoʻllaniladigan amallar majmuasi;
- “Данные” - jadvallarning ustun va satrlaridagi maʼlumotlarni filtrlash, saralash, nusxalarini koʻpaytirish, keraksizlarini yoʻqotish, tashqi axboratlarni kiritish kabi amallar majmuasi;
- “Рецензирование” – matn xatolarini tuzatish va hujjatlarni tahrirlash;
- “Вид” – jadvallarni ekranda turlicha namoyish qilishga moslashtirish.

3. Ustun (column) – jadval ustunlari koʻrsatilgan qator. Excel 2007 va Excel 2010 da ular nomi A dan XFD gacha boʻlgan lotin harflari bilan koʻrsatiladi. Z harfidan soʻng harflar kombinatsiyasi bilan koʻrsatiladi. Masalan: AA, AB, AC,.. IA, IB va h-zo. Excel 2010 dasturida ustunlar soni 16384 tadan iborat. Excel 2013 dasturida electron jadval ustunlari lotin alfavitining C harfi bilan boshlanib, undan keyin ustun raqami qoʻyiladi. Masalan, C1 – 1-ustun, C24 – 24-ustun. (Excel 2013 dasturida electron jadval ustunlarini A dan XFD gacha boʻlgan lotin harflari bilan belgilashni sozlash mumkin.)

4. Qator (row) – jadval qatorlari koʻrsatilgan ustun, ular nomi natural sonlar bilan koʻrsatiladi. Excel 2010 dasturida satrlar soni 1048576 ta.

Excel 2013 dasturida electron jadval qatorlari lotin alfavitining R harfi bilan boshlanib, undan keyin qator raqami qoʻyiladi. Masalan, R1 – 1-qator, C17 – 17-qator. (Excel 2013 dasturida electron jadval qatorlarini 1 dan 1048576 gacha boʻlgan sonlar bilan belgilashni sozlash mumkin.)

5. Katak - ustun va qatorlar kesishgan toʻrtburchak soha boʻlib, uning manzili ustun harfi va qator tartib sonlari bilan nomlanadi. Masalan A54 - A-ustun, 54-qator, V12 – V - ustun, 12-qator va h-zo. Excel 2010 dasturining bir sahifasida 17179869184 ta katakcha mavjud. Kursor faol katakda turadi va faol katakka 256 tagacha belgilar kiritish mumkin.

Excel 2013 dasturida electron jadval kataklari manzilini koʻrsatishda avval qator raqami, keyin esa ustun raqami beriladi. Bu belgilash avvalgi belgilashlardan farq qiladi. Masalan, R17C1 manzil – R17-qator (17-qator), C1-ustun (1-ustun) ekanligini anglatadi. Bu oʻzgarish avvalgi versiyalarda yaratilgan hujjatlardagi formulalarni xato ishlashiga olib keladi.

6. Katak manzili (nomi) sohasi – ustun qatori yuqorisida chap tomondagi soha boʻlib, unda faol katak manzili koʻrsatib boriladi. Elektron sahifaning birinchi (yuqori chap burchagidagi) katakchasining manzili A1 boʻlib, uning oxirgi katakchasi (pastki oʻng burchakda) ning manzili mos ravishda XFD1048576 ekanligi tushunarli.

*Izoh:* Oldingi Excel dasturlarida ishchi maydoni 65536 ta satr va 256 ta ustundan iborat edi. Excel 2007 dasturidan boshlab esa satrlar soni 1048576 ta boʻlib, ustunlar soni esa 16384 tadan iborat (A-XFD gacha). Demak Excel 2007 dasturida jami 17179869184 ta katakcha mavjud ekan. Katakcha – bu ustun va satr kesishgan joydagi birlik boʻlib, har bir katakcha oʻzining individual manzili (nomi)ga egadir. Bu nom shu katakchanning ustuni harfi (harflari) hamda uning satr tartib raqamining yonma-yon yozilishidan iboratdir. Elektron sahifaning birinchi (yuqori chap burchagidagi) katakchasining manzili A1 boʻlib, uning oxirgi katakchasi (pastki oʻng burchakda)ning manzili mos ravishda XFD1048576 ekanligi tushunarli.

7. Formulalar qatori  $f_x$  – katakka tegishli formulalarni kiritish sohasi. Faol katakka kiritilgan maʼlumotlar va kiritilgan formulalar formula qatorida koʻrsatiladi. Formula qatorida bu maʼlumotlarni tahrirlash ham mumkin.

8. Ishchi soha - kataklar joylashgan soha.

9. Sahifa nomlari - yorliq koʻrinishida boʻlib, ular “Лист” deyiladi. Standart holatida ular birta boʻlib, zaruriyat tugʻilganda ularni koʻpaytirilishi yoki kamaytirilishi mumkin. Bir kitobda 1 dan 255 tagacha sahifa joylashishi mumkin. Har bir sahifa jadval koʻrinishidadir.

10. Holatlar qatori - joriy sahifa holatini koʻrsatadi.

11. Siljitish (prokrutka) - qator va ustun boʻyicha oʻtish vositasi.

Bosh oynaning yuqori chap burchagida tez-tez ishlatiluvchi murojaatni osonlashtiruvchi panel mavjud bo'lib, ular xotiraga saqlash, oxirgi amalni bekor qilish kabi amallarni tez va oson bajarish mumkin. Ushbu panelga yangi buyruqlar kiritish orqali imkoniyatini oshirish mumkin.

Elektron jadval hujjati \*.xlsx kengaytmasi bilan saqlanadi. Fayl nomi o'zgartiriladi.

#### **Elektron jadvali fayllar formatlari**

Formati	Kengaytma	Berilishi
Excel kitobi	XLSX	Office Excel 2007-2010 va undan yuqori versiyalari.
Excel varag'i (kod)	XLSM	Makroslarni qo'llab-quvvatlovchi XML ga asoslangan Office Excel 2007 versiyasi.
Excel ikkilik kitobi	XLSB	Office Excel 2007 (BIFF12) ning ikkilik fayllari.
Shablon	XLTX	Excel uchun Office Excel 2007 ning standart formati.
Shablon (kod)	XLTM	Makroslarni qo'llab quvvatlovchi Excel 2007 fayl formati.

Excel bilan ishni tugatgandan keyin uning oynasini yopish kerak. Agar Excel dagi biror hujjatga o'zgartirish kiritilgan - u, saqlanmagan bo'lsa, dastur uni saqlash yoki uni saqlamaslikni so'raydi. Saqlashni lozim topsangiz “Да”, aks holda “Her” javobini berish lozim. Shu bilan Excelda ishlash tugatiladi.

#### **2. MS Excel elektron jadvali va sahifalarni boshqarish**

Excel dasturida tayyorlanadigan hujjat elektron kitoblar deb ataladi va bu elektron kitobda kompyuter xotirasi imkon beradigan darajada ko'p sahifa (list)larni joylashtirish mumkin. Bu sahifalar ishchi sahifalar (oddiy sahifalar – ularda jadvallarni joylashtirish mumkin) hamda diagramma sahifalari (ularda diagrammalar joylashtiriladi)ga bo'linadi. Excel kitobi (Kniga) sahifalardan tashkil topgan. Bir sahifadan boshqasiga o'tish uchun sichqoncha ko'rsatkichini shu sahifaga keltirib, chap tugmasini bir marta bosish kerak.

Excel dasturi oynasini boshqarish, Windows dasturlari oynalarini boshqarish singaridir. Oyna chegaralariga sichqoncha ko'rsatkichi olib borilsa, uning ko'rinishi ikki tomonlama ko'rsatkich ko'rinishiga o'tgach, chap tugmani bosib turgan holda chegarani keraklacha kattalashtirish yoki kichiklashtirish holatiga muvofiq surish mumkin. Dastur oynasining o'zini surish uchun shu dastur sarlavha satriga sichqoncha ko'rsatkichini keltirib, chap tugmani bosib turgan holda dastur oynasini siljitish mumkin. Dastur oynasini yopish uchun yuqori o'ng burchakdagi X harfi ko'rinishidagi tugma bosiladi (Alt+F4 kombinatsiyasi ham shu vazifani bajaradi). Bundan tashqari Excel dasturida chiqish uchun dastur yuqori chap tomonida joylashgan “Faollashtirish menyusi” dan “Закрыть” buyrug'i tanlanadi. Exceldan tezkor chiqish uchun shu tugmada sichqoncha chap tugmasini ikki marta tez bosish ham yetarlidir.

**Faol katakcha va sohalar.** Kursor turgan katak faol katak deyiladi va uning chegaralari boshqalarinikidan ko'ra qalinroq bo'ladi. Boshqa dasturlar bilan muvofiqlikni ta'minlash maqsadida biz bu holatni Excel kursori deb ataymiz va faol katakchada kursor mavjud deb hisoblaymiz. elektron jadvallar hosil qilish uchun, katakka matn, son yoki formula kiritish kerak va formula turgan katakda formula natijasi olinadi. Kursor turgan katakchanning ichidagi ma'lumot formulalar satrida ko'rinish turadi. Kerakli katakchani yoki bir necha katakchani faollashtirish uchun shu kerakli katakcha ustida sichqoncha chap tugmasini bosish yoki bir necha katakchani ko'rsatish uchun ulardan birinchisi ustida sichqoncha chap tugmasini bosib turgan holda kerakli katakchagacha chap tugmani qo'yib yubormasdan sichqoncha ko'rsatkichini olib borib, so'ng qo'yib yuboriladi. Bunda belgilangan katakchalar ustida keyinchalik kerakli amallarni bir vaqtning o'zida bajarish mumkin bo'ladi. Belgilangan katakchalar soha deb ataladi va ular ko'pincha yonma-yon joylashgan katakchalardan iborat bo'ladi. Lekin ba'zida bir-biridan uzoqda joylashgan katakchalarni ham birgalikda belgilash mumkin. Buning uchun belgilash amalini bajarish jarayonida CTRL tugmasidan foydalanish mumkin.

Ishchi jadvallarni qurib chiqishda yoki yacheykalarni bichimlashda ish olib borayotgan sohaning manzilini bilish shart emas, lekin formulalar bilan ishlayotganda bu narsa juda muhimdir.

Katakchalar ustida quyidagi amallarni bajarish mumkin:

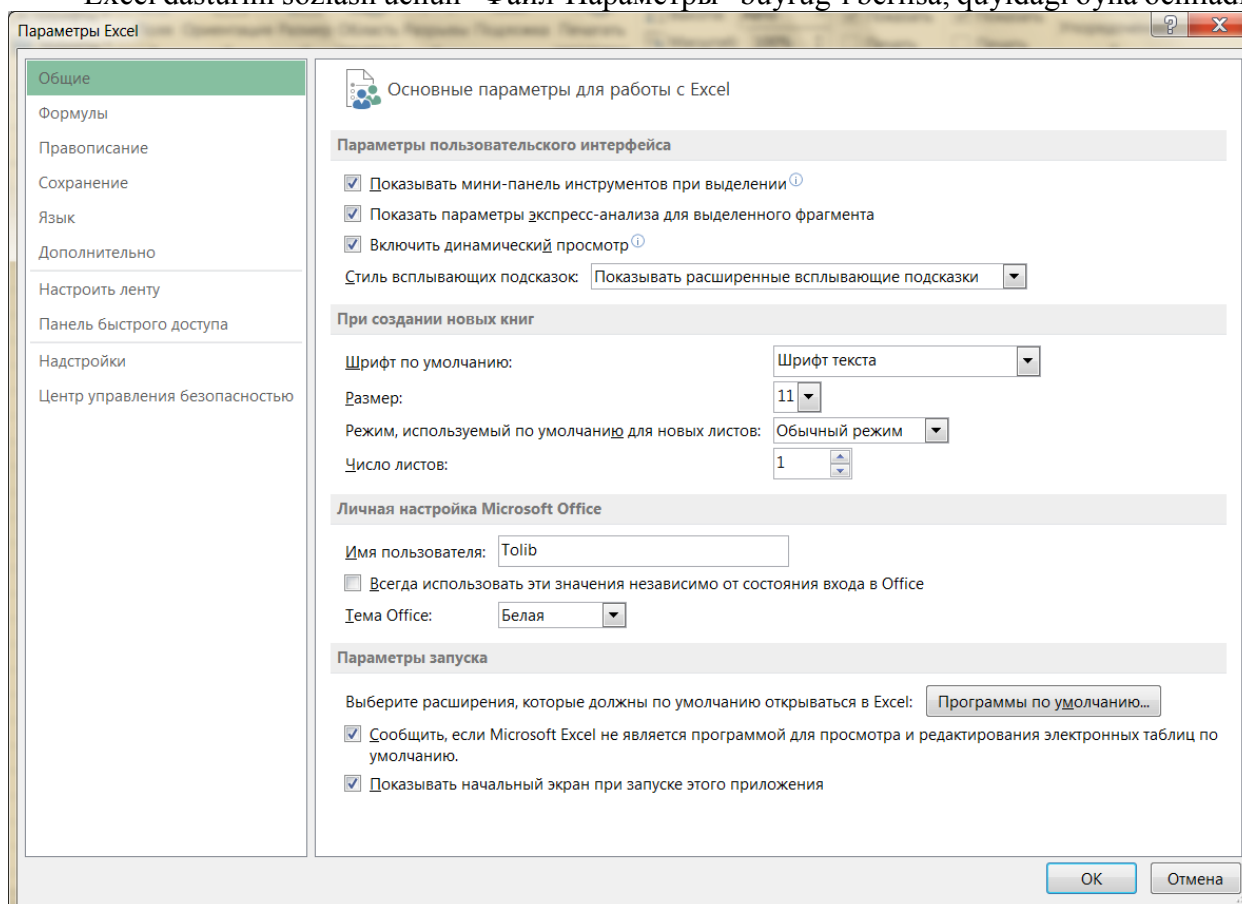
- katakchaga ma'lumotlarni kiritish;
- katakchadagi mavjud ma'lumotlarni tahrirlash;
- katakchalardagi ma'lumotlarni nusxalash;
- katakchalardagi ma'lumotlarni ko'chirish;
- katakchalarni formatlash;
- katakchalardagi ma'lumotlarni o'chirish.

Katakchada saqlanishi mumkin bo'lgan ma'lumotlar turi:

- doimiy qiymatlar (sana, vaqt, pulli miqdorlar, foizlar, kasrlar yoki matn);
- formulalar.

Katakchalardagi ma'lumotlarni formatlash ishi Wordda matnlarni formatlashga o'xshash.

Excel dasturini sozlash uchun "Файл-Параметры" buyrug'i berilsa, quyidagi oyna ochiladi.



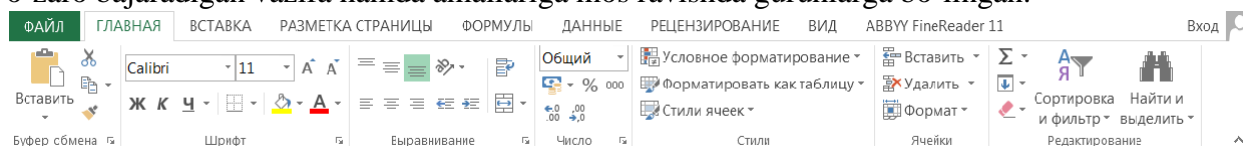
**Tezkor tugmalar.** Excel elektron kitobida 17 milliarddan ortiqroq katakchalar mavjudligini hisobga oladigan bo'lsak, albatta ular bo'ylab tezkor harakat qilish ehtiyoji paydo bo'ladi. Shu sababli Excel dasturiga ham tezkor tugmalar kombinatsiyalari kiritilgan. Tezkor tugmalar kombinatsiyalari deb bir yoki bir nechta tugmalarni bosish orqali chaqiriladigan amallarga aytiladi. Ularning ko'pchiliklari Ctrl tugmasi bilan birgalikda bajariladi. Ularni eslab qolish oson bo'lishi maqsadida ko'pgina kombinatsiyalar ularning ingliz tilidagi atamalariga mos holda mos harflarga bog'lab qo'yilgan. Masalan chop etish oynasini chaqirish uchun Ctrl+P(Print) tugmasi bosiladi. Bu amalni bajarish uchun birinchi yozilgan Ctrl tugmasini bosib turgan holda P tugmasi bosiladi va ikkala tugma birgalikda qo'yib yuboriladi. Bu vazifani bajarish uchun Excel elektron jadvaliga tezkor tugmalar kombinatsiyalari kiritilgan. Ko'pincha ishlatiladigan Excel kombinatsiyalari quyidagi jadvalda keltirilgan.

<i>Kombinatsiya</i>	<i>Bajariladigan amal nomi</i>
Ctrl+A	Hammasini belgilash.
Ctrl+B	Qalinlashtirish amalini qo`llash yoki bekor qilish.
Ctrl+C	Belgilangan qismni buferga nusxalash.
Ctrl+F	Qidiruv oynasini chaqirish.
Ctrl+G yoki F5	Ko`rsatilgan joyga o`tish.
Ctrl+H	Almashtirish.
Ctrl+I	Egri yozish amalini qo`llash yoki bekor qilish.
Ctrl+O yoki Ctrl+F12	Hujjatni ochish.
Ctrl+P	Chop etish oynasini chaqirish.
Ctrl+S yoki Shift+F12	Saqlash.
Ctrl+U	Matn tagiga chizish amalini qo`llash yoki bekor qilish.
Ctrl+V yoki Shift+Insert	Buferdagi ma`lumotni qo`yish.
Ctrl+W yoki Ctrl+F4	Faol elektron kitobni yopish
Ctrl+X	Belgilangan qismni buferga qirqib olish
Ctrl+Y yoki F4	Oxirgi amalni takrorlash
Ctrl+Z	Oxirgi amalni bekor qilish
F1	Yordamchi ma`lumotni ko`rish
Ctrl+F1	Uskunalar panelini ko`rsatish-yashirish
F2	Faol katakchani tahrirlash (o`zgartirish) rejimiga o`tish

(\* - belgisi ScrollLock tugmasi bosilganini bildiradi)

<i>Tugmalar</i>	<i>Bajariladigan amal nomi</i>
↑	Kursorni bir satr yuqoriga o`tkazish
↓	Kursorni bir satr pastga tushirish
←	Kursorni bir ustun chapga o`tkazish
→	Kursorni bir ustun o`ngga o`tkazish
PgUp	Kursorni bir ekran yuqoriga o`tkazish
PgDn	Kursorni bir ekran pastga o`tkazish
Alt+PgDn	Kursorni bir ekran o`ngga o`tkazish
Alt+PgUp	Kursorni bir ekran chapga o`tkazish
Home	Kursorni u turgan satrning birinchi katakchasiga o`tkazish
Ctrl+Home	Kursorni ishchi kitobning birinchi katakchasiga (A1 ga) o`tkazish
F5	O`tish muloqot oynasini chaqirish
Ctrl+BackSpase	Ekranni kursor ko`rinadigan holatha o`tkazish
↑*	Ekranni bir satr yuqoriga o`tkazish (faol katakcha o`zgarmaydi)
↓*	Ekranni bir satr pastga o`tkazish (faol katakcha o`zgarmaydi)
←*	Ekranni bir ustun chapga o`tkazish (faol katakcha o`zgarmaydi)
→*	Ekranni bir ustun o`ngga o`tkazish (faol katakcha o`zgarmaydi)

**Lenta bilan ishlash.** Excel 2007 versiyasidan boshlab oldingi versiyalardagi standart menyular satri hamda uskunalar paneli o`rnini bosadigan yangi qism kiritilgan bo`lib, u bir vaqtning o`zida bir qancha asboblar bilan ishlash imkonini berishi hamda keraksiz bo`lib qolayotgan menyular satri tomonidan egallangan joyning samarali ishlatilishiga olib keldi. Ushbu qism lenta deb ataladi hamda u bir nechta bandlardan iborat bo`lib, ulardagi buyruq va funksiyalar o`zaro bajaradigan vazifa hamda amallariga mos ravishda guruhlariga bo`lingan.



Lentadagi buyruqlar quyidagichadir:

**Tugma:** Eng keng tarqalgan boshqaruv elementi bo`lib, tugmalarning ko`pchiligi matnli izohlar bilan ta`minlangan. Tugmaning bajaradigan vazifasini bilish uchun uni sichqoncha orqali

ko`rsatish kifoya, izoh avtomatik tarzda beriladi. Tugmalar ishlatilsa, u to`g`ridan-to`g`ri vazifani bajariladi.

**Buyruqlar guruhi:** Lentaning har bir bandida o`zaro bog`liq buyruqlar mavjud bo`lib, ular guruhlarga birlashtiriladi. Masalan, “Главная” panelining “Шрифт” buyruqlar to`plamida matn shriftini boshqarish tugmalari joylashgan.

**Ochiladigan ro`yxat:** Ba`zi tugmalar yonida yoki alohida pastga qaratilgan ko`rsatkichlar mavjud. Ularni tanlanganda ro`yxat yoki kolleksiya hosil bo`ladi. Foydalanuvchi kerakli buyruqni tanlashi yetarli.

**Kolleksiya:** Excelda yaratilgan yangi element bo`lib, unda tanlanadigan imkoniyatlarning grafik ko`rinishlari hosil bo`ladi. Kolleksiyalardan foydalanish ko`p ishlatiladigan amallarga tezroq yetib boorish imkonini beradi hamda kolleksiyalarning uch xil turi mavjud: (Ochiladigan kolleksiya, Ochiladigan to`r va kolleksiya lentasi).

**Ma`lumotnoma:** Lentaning o`ng yuqori burchagida joylashgan so`roq belgisi ko`rinishidagi tugma bo`lib, uni tanlanganda Excel yordamchi ma`lumotlar oynasi hosil bo`ladi. Unda kerakli mavzuni tanlab kerakli ma`lumotlarni o`qish mumkin.

**Standart menyular:** Ko`plab pastga qaratilgan ko`rsatkichli tugmalarni tanlanganda xuddi shunday menyular bilan ishlash kerak bo`ladi. Ba`zi buyruqlarda mos tugmalar ham mavjud. Agar buyruqning oxirida uch tugma (...) ko`rinishidagi qo`shimcha bo`lsa, bu buyruqni tanlash muloqot oynasining ochilishiga olib keladi va u yerda kerakli qiymatlarni kiritish yoki tanlash kerak bo`ladi.

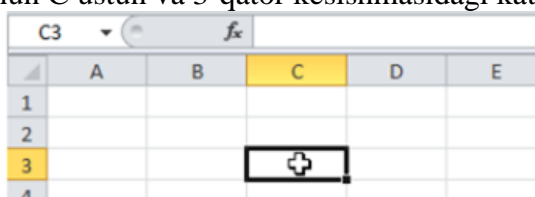
**Hisoblagich:** Yuqoriga va pastga qaratilgan ko`rsatkichli boshqaruv elementi bo`lib, uning qiymati sifatida sonlar qaraladi hamda tugmalarni tanlanganda qiymat mos tarzda o`sadi yoki kamayadi. Xuddi shunday bu qiymatlarni klaviatura orqali “qo`lda” kiritish ham mumkin. Kiritilayotgan qiymatlarning to`g`riligini Excel avtomatik tarzda tekshirib boradi.

**Kontekst lenta bandlari:** Bunday lenta bandlari biror yangi ob`yekt qo`yilgandan yoki tanlangandan so`ng hosil bo`ladi hamda ularning guruhlari nomlari Excel sarlavha satrida hosil bo`ladi. Bu kontekst lenta bandida shu tanlangan ob`yekt bilan bog`liq bo`lgan barcha xususiyat va parametrlar joylashgan bo`lib, u boshqa ob`yekt tanlanishi bilan yo`qoladi. Kontekst lenta bandini hosil qilish uchun shu ob`yekt ustida qo`sh chertish ham yetarli. Lekin barcha ob`yektlarda ham kontekst lenta bandi mavjud bo`lmasligini esda saqlang.

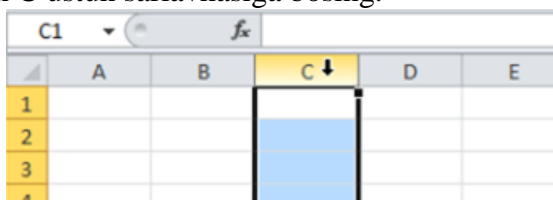
Excelda soha – bu ikki yoki undan ortiq kataklar to`plamidir. Endi soha bilan bog`liq ba`zi eng muhim amaliyotlarni ko`rib chiqamiz.

Ishni katak, qator va ustunni belgilashdan boshlaymiz.

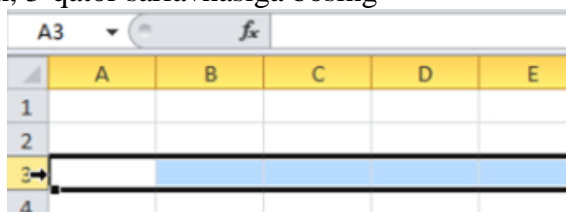
C3 katakni belgilash uchun C ustun va 3-qator kesishmasidagi katakni bosing.



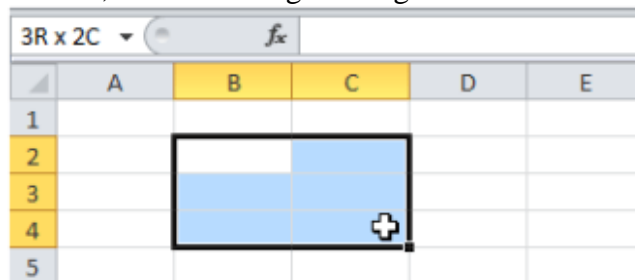
C ustunni tanlash uchun C ustun sarlavhasiga bosing.



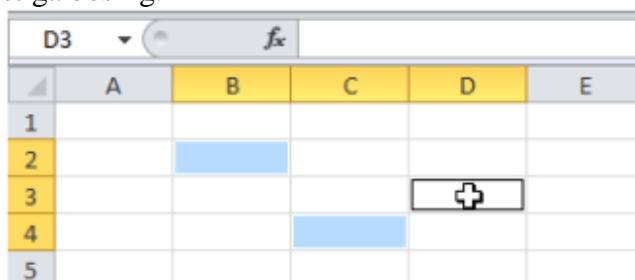
3-qatorni tanlash uchun, 3-qator sarlavhasiga bosing



B2:C4 sohani tanlash uchun, B2 katak ustiga bosing va C4 katakka torting.



Alohida kataklar sohasini belgilash uchun, CTRL ni bosib turgan holda sohaga kiritishni xohlagan har bir katakni ustiga bosing.



**Ma'lumotlarni kiritish va ma'lumotlarni kiritishda avtomatlashtirish vositalaridan foydalanish.** Elektron jadval kataklariga uch xil ma'lumotlarni kiritish mumkin:

- matnli;
- sonli ifodalar;
- formulalar.

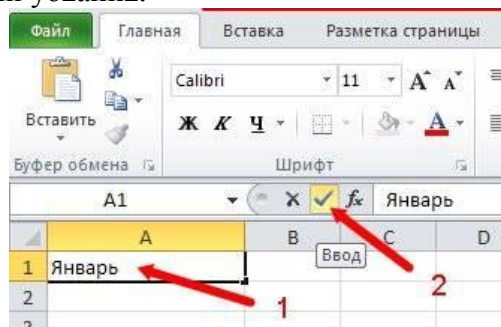
Excelda ayrim ishlarni bajarish judayam ko'p vaqtni olishi va qiyin bo'lishi mumkin. Jadvallar bilan ishlashda mavjud bo'lgan funkstiyalardan foydalanish ishni sezilarli darajada osonlashtiradi. Ayrim ishlarni qayta qayta bajarishga to'g'ri keladi. Bu esa ko'p vaqtni tatlab qiladi. Bunday ishlarni avtomatlashtirish uchun dastur yaratuvchilari qulay bo'lgan funkstiyani ishlab chiqishgan va u Excel kataklarini avtoto'ldirish deyiladi.

Ma'lumotlar kiritilayotganda ustun bo'yicha shu kiritilayotgan ma'lumotlarga o'xshash ma'lumot bo'lsa, shu ma'lumotning o'xshash harfi kiritilishi bilan qolgan ma'lumotlar avtomatik tarzda yuqoridagi o'xshash ma'lumot bilan to'ldiriladi va bu avtoto'ldirilgan ma'lumot qora ramkaga olinadi. Agar ma'lumot kiritish davom ettirilsa, qora ramka o'chirilib yangi ma'lumot kiritiladi. Bunday avtoto'ldirish usuli ustun bo'yicha bo'sh katak uchraguncha davom etadi.

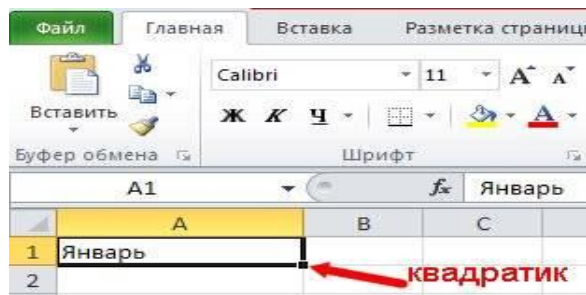
Agar faol katakka biror natural son kiritilgan bo'lsa va sichqoncha ko'rsatkichini shu son kiritilgan katakning o'ng pastki burchagiga keltirib + belgi paydo bo'lganda sichqonchanning o'ng tugmasi ustun yoki qator bo'yicha surilsa, qolgan kataklar shu son dan boshlab o'sish tartibida natural sonlar bilan to'ldiriladi. Chap tugma bosib surilsa, shu son bilan kataklar to'ldiriladi.

Excelda oy bo'yicha ma'lumotlar ro'yxatini hosil qilishni ko'ramiz. Bu ro'yxatdan kelib chiqqan holda keyinchalik o'z ro'yxatlarimizni yaratishimiz mumkin.

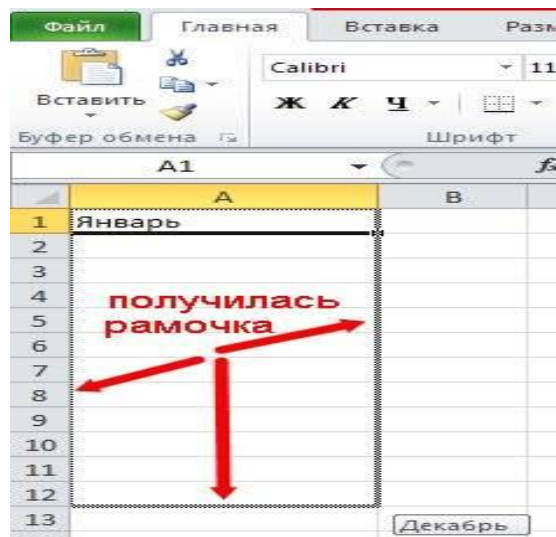
A1 katakka "Январь" (e'tibor bering, Yanvar emas, balki "Январь". Chunki baza rus tili so'zlari bilan to'ldirilgan) so'zini yozamiz.



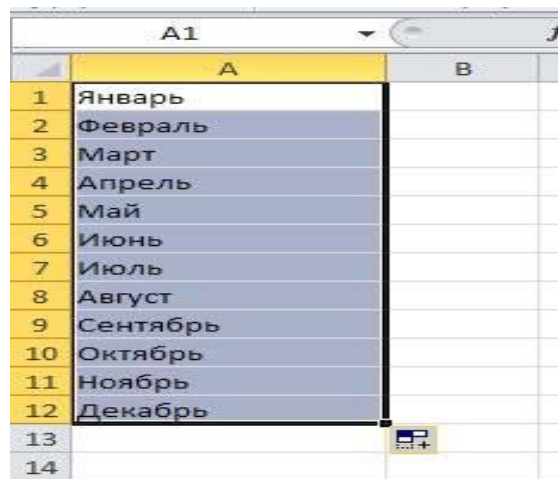
Endi sichqoncha kursorini A1 katakning o'ng pastki burchagidagi kichik kvadratchaga olib kelamiz.



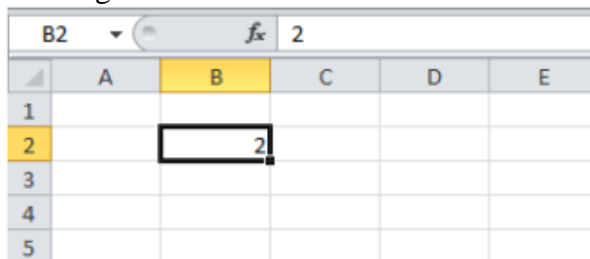
Kursor ko'rsatkichi oq qo'shuvdan kichik qora qo'shuv o'zgarganda sichqonchani o'ng tugmasini bosib va uni qo'yib yubormasdan sichqoncha ko'rsatkichini A12 katakka olib keling. Sizda soha hosil bo'ladi.



Siz sichqonchani tugmasini qo'yib yuborganingizdan keyin belgilangan kataklar oy nomlari bilan to'ldiriladi.

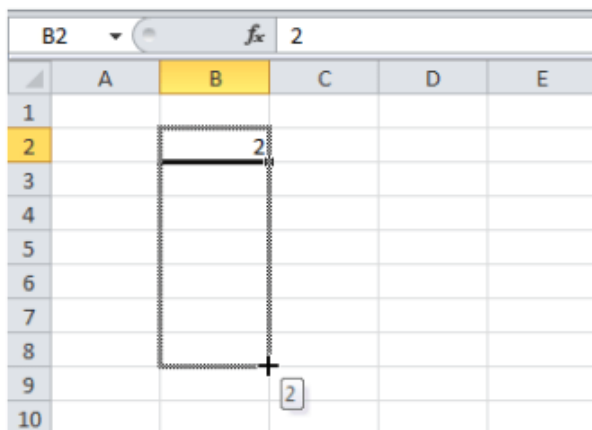


Sohani to'ldirish uchun, quyidagilarni ketma-ket bajaramiz. B2 katakka 2 qiymatni kiriting.

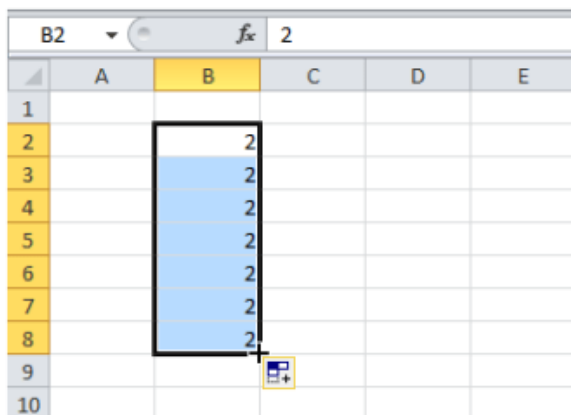




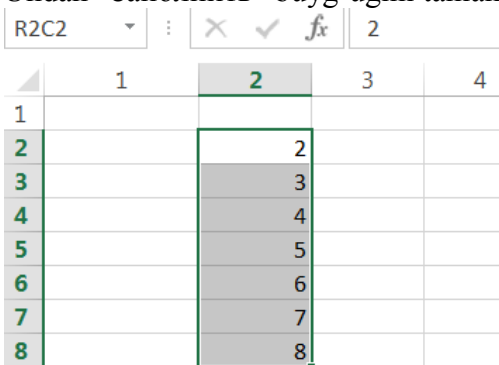
Belgilangan soha kataklarida kiritilgan son bir xilda takrorlanishi uchun B2 katakni belgilab, B2 katakning pastki o'ng burchagiga sichqoncha chap tugmasini bosning va uni pastga B8 katakka sudrang.



Natjajada hosil qilingan sohada bir xil ma'lumotlarni olamiz:



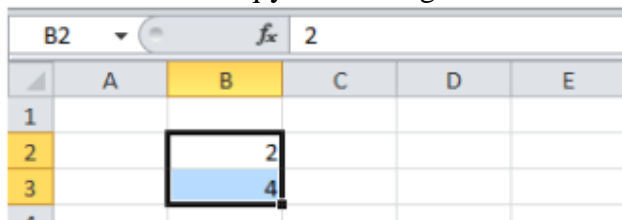
Belgilangan soha kataklarida kiritilgan son bir xil osib borish tartibida bo'lishi uchun B2 (R2C2) katakni belgilab, B2 (R2C2) katakning pastki o'ng burchagiga sichqoncha o'ng tugmasini bosning va uni pastga B8 (R8C2) katakka sudrang. Sichqoncha tugmasi qo'yib yuborilsa, nima qilish kerakligi oynasi ochiladi. Undan "Заполнить" buyg'ugini tanlang.



Bu sudrash (sudrab olib kelish) texnikasi juda ham muhim va siz uni Excelda tez-tez ishlatasiz.

Quyida boshqa bir misol:

B2 katakka 2 qiymatni va B3 katakka 4 qiymat kiriting.





B2 va B3 kataklarni belgilang, ushbu sohaning pastki o'ng burchagiga bosning va uni pastga sudrang.

	A	B	C	D	E
1					
2		2			
3		4			
4		6			
5		8			
6		10			
7		12			
8		14			
9					

Excel dastlabki ikki qiymatlarning namunasi asosida sohani avtomatik tarzda to'ldiradi. Quyida boshqa bir misol:

B2 katakka 6/13/2016 sanasini va B3 katakka 6/16/2016 sanasini kiriting.

	A	B	C	D	E
1					
2		6/13/2013			
3		6/16/2013			
4					

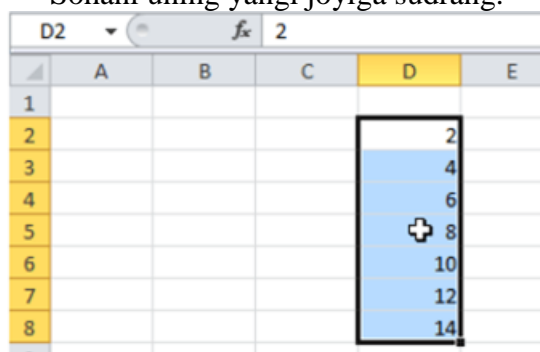
B2 va B3 kataklarni belgilang, bu sohaning pastki o'ng burchagiga bosning va uni pastga sudrang.

	A	B	C	D	E
1					
2		6/13/2013			
3		6/16/2013			
4		6/19/2013			
5		6/22/2013			
6		6/25/2013			
7		6/28/2013			
8		7/1/2013			
9					

**Sohani siljitish.** Sohani siljitish uchun quyidagilarni ketma-ket bajaring. Sohani belgilang va uning chegarasi ustiga bosning.

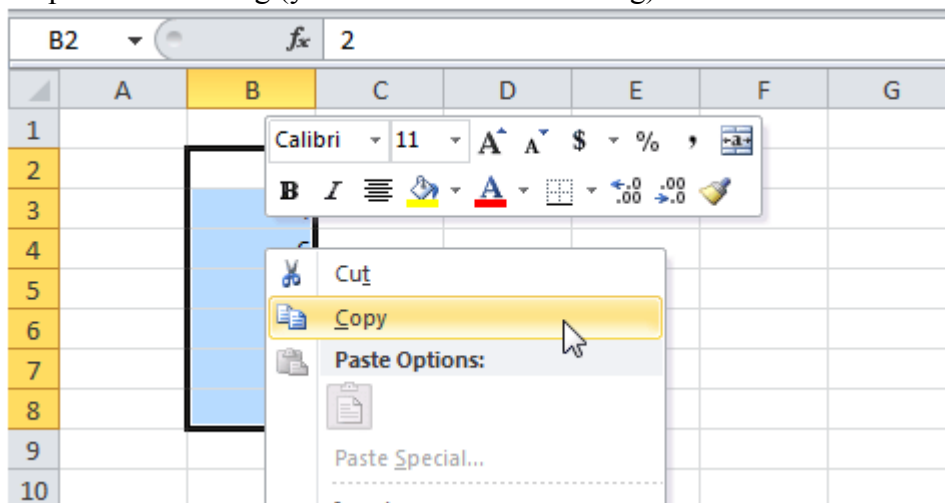
	A	B	C	D	E
1					
2		2			
3		4			
4		6			
5		8			
6		10			
7		12			
8		14			
9					

Sohani uning yangi joyiga sudrang.

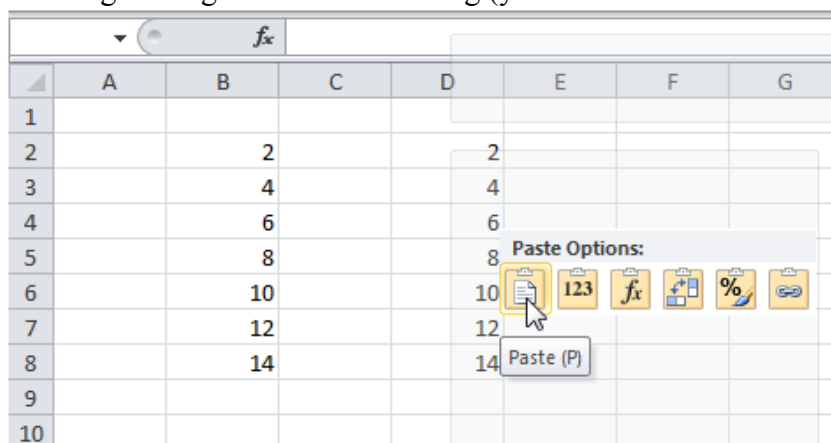


**Sohadan nusxa olish/qo'yish.** Sohadan nusxa olish va qo'yish uchun quyidagilarni ketma-ket bajaring.

Sohani belgilang, sichqonchani o'ng tomonini bosing (bundan keyin qisqacha: o'ng tomonni bosing) va "Копировать"ni bosing (yoki "CTRL + C" ni bosing).

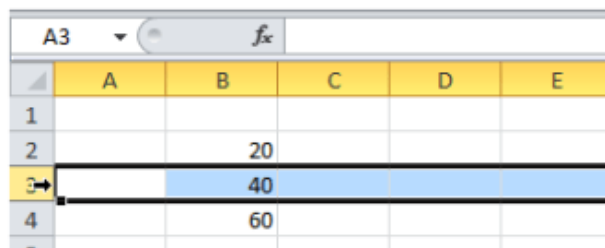


Sohaning birinchi katagi ko'rinishi kerak bo'lgan katakni belgilang, o'ng tomonni bosing va "Параметры вставки"ning ostidagi "Вставить" bosing (yoki "CTRL + V" ni bosing).

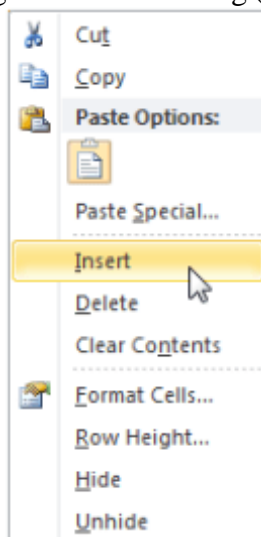


**Qator, ustun kiritish.** 20 va 40 qiymatlar orasiga qator kiritish uchun, quyidagilarni ketma-ket bajaring.

3-qatorni belgilang.



Sichqonchani o'ng tomonini bosing va Insert ni bosing (yoki "CTRL + Shift + =" ni bosing).



Natija:

	A	B	C	D	E
1					
2		20			
3					
4		40			
5		60			

Yangi qatordan quyidagi qatorlar pastga surilgan. Shunga o'xshash yo'l bilan, ustun ham kirita olasiz.

Endi yuqoridagi oylarni kiritgan misolimizda sana bo'yicha ma'lumotlar ro'yxatini hosil qilishni ko'ramiz.

B1 katakka 12.01.2012 sanani yozamiz va sichqoncha kursoriga to'ldirish markeriga olib kelib B1:B12 kataklar sohasini belgilaymiz. Avtomatik ravishda sanalar ketma ketligi hosil bo'ladi.

	A	B	C
1	Январь	12.01.2012	
2	Февраль	13.01.2012	
3	Март	14.01.2012	
4	Апрель	15.01.2012	
5	Май	16.01.2012	
6	Июнь	17.01.2012	
7	Июль	18.01.2012	
8	Август	19.01.2012	
9	Сентябрь	20.01.2012	
10	Октябрь	21.01.2012	
11	Ноябрь	22.01.2012	
12	Декабрь	23.01.2012	

**Excelda satrlarni raqamlash.** Agarda siz satr ustunlarini raqamlab chiqmoqchi bo'lsangiz, katakka 1, undan keyingga katakka 2 yozing. Keyin ikkala katakni belgilab avtoto'ldirish markerini bosib, kerakli sohani belgilagan holda avtomatik ravishda raqamlar ketma-ketligini hosil qilish mumkin. Bunda ketma-ketlik oshib boradi. Bu ishni faqat katakka birni kiritib, so'ngra uni avtoto'ldirish markerini belgilab klaviaturadagi Ctrl tugmasini bosib turgan holda ham hosil qilish mumkin.

Yuqoridagi ishni har xil oraliqdagi sonlar uchun, masalan, juft, toq yoki boshqa sonlar uchun ham amalga oshirish mumkin.

Masalan, D3 katakka 1, D4 katakka 3 kiritamiz va bu kataklarni belgilagan holda ularning avtoto'ldirish markerini ishlatamiz. Natijada toq sonlardan iborat bo'lgan ro'yxat hosil bo'ladi.

Bu ishni Ctrl tugmasini bosib turgan holda ham amalga oshirish mumkin.

	A	B	C	D	E
1	Январь	12.01.2012			
2	Февраль	13.01.2012			
3	Март	14.01.2012	1	1	
4	Апрель	15.01.2012	2	3	
5	Май	16.01.2012	3		
6	Июнь	17.01.2012	4		
7	Июль	18.01.2012	5		
8	Август	19.01.2012	6		
9	Сентябрь	20.01.2012	7		
10	Октябрь	21.01.2012	8		
11	Ноябрь	22.01.2012	9		
12	Декабрь	23.01.2012	10		
13		1	2	3	4

Natijada toq sonlardan iborat bo'lgan ro'yxat hosil bo'ladi.

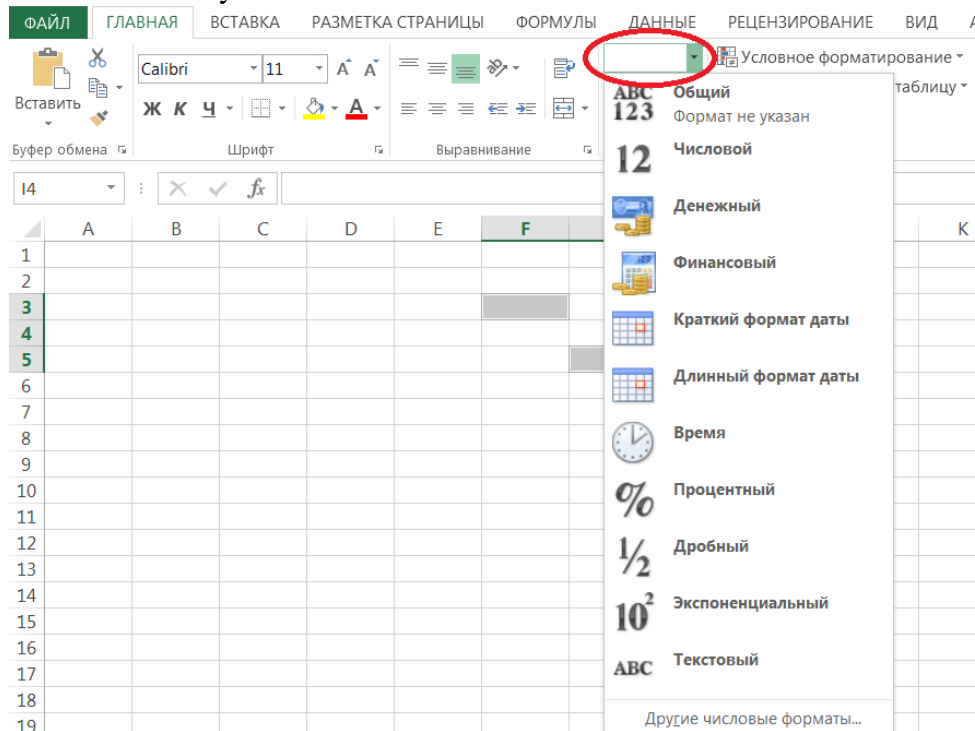
	A	B	C	D	E
1	Январь	12.01.2012			
2	Февраль	13.01.2012			
3	Март	14.01.2012	1	1	
4	Апрель	15.01.2012	2	3	
5	Май	16.01.2012	3	5	
6	Июнь	17.01.2012	4	7	
7	Июль	18.01.2012	5	9	
8	Август	19.01.2012	6	11	
9	Сентябрь	20.01.2012	7	13	
10	Октябрь	21.01.2012	8	15	
11	Ноябрь	22.01.2012	9	17	
12	Декабрь	23.01.2012	10	19	
13		1	2	3	4

**Kataklar formati.** Jadval kataklarini formatlash mumkin. Odatda jadvalning har bir ustuni bir xil formatdagi qiymatlarni qabul qiladi. Ular quyidagicha:

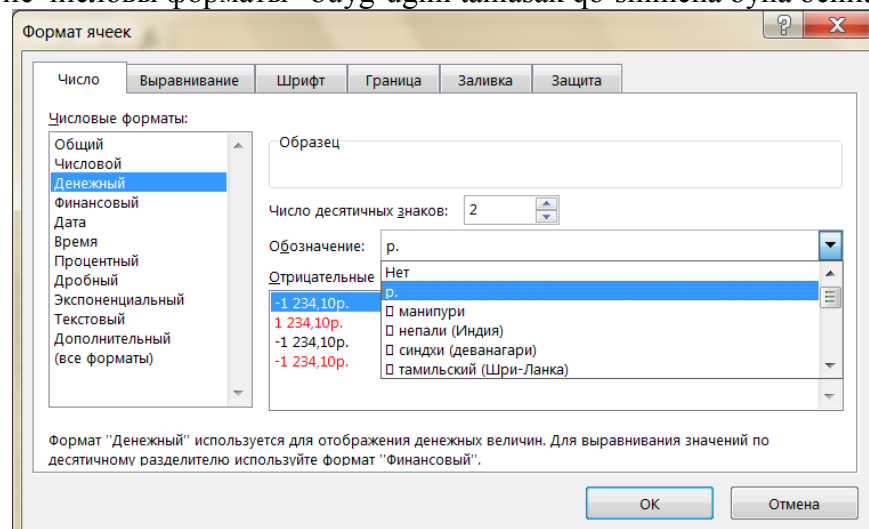
Format	Tavsifi	Saqlangan qiymat	Kiritilgan qiymat
Umumiy	Son va matn ma'lumotlarni tasvirlash shakli. Ma'lumotlar formatlanmaydi. Qanday kiritilgan bo'lsa shunday akslantiriladi.	123 Abvg	123 Abvg
Son	O'qli kasr va butun sonlarni tasvirlash shakli. Verguldan keyin beriladigan raqamlar soni, sonning butun va kasr qismini ajratuvchi belgi shaklini ko'rsatish mumkin.	1234,56 1234	1234,56 1234,6 1234,00 1.234
Pulli	Pul formati. Ma'lumotlar ustunda o'qli nuqta bo'yicha tekislanadi. Valyuta belgisi va o'qli kasr qismidagi raqamlar sonini ko'rsatish imkoni bor.	123 -123	123s. \$123 123 -123s
Moliyaviy	Pul formatiga o'xshash, o'qli nuqtagina emas sonning ishorasi va valyuta belgisi xam tekislanadi.	123	123s
Sana	Excel sanalarni XX asr boshidan boshlab ko'rsatilgan sanagacha saqlaydi.	35704	10.01.97 1-okt-97 oktyabr
Vaqt	Excel vaqtni 24 soatli kunni bo'laklarini ko'rsatuvchi kasr sifatida tasvirlaydi.	0,3004 35704,3004	1997 - 7:12 7:12AM 10.01.97 7:12:35
Foizli	Sonlarni yuzdan bir ulushida tasvirlash.	0,123	12,3%
Kasrli	Oddiy kasr. Talab qilingan aniqliqda tasvirlashni tanlash mumkin.	1,23	1x20/87 19/16 123/100

Eksponen- tsial	Sonlar darajali (ekspponentsial) shaklda tasvirlanadi.	123000	123+3 1,23e+5
Matnli	Ma`lumotlar qanday kiritilgan bo`lsa deyarli shunday saqlanadi.	F,DU 123	F,DU 123
Qo`shimcha	Telefon raqamlari.	5555555	555-5555
Hamma formatlar	Foydalanuvchi tomonidan aniqlanadigan formatlar		

Formatlash uchun kataklar yoki ustun belgilab olinadi va “Главная” lentadan formatlash tugmasi bosiladi. Formatlar oynasi ochiladi.



Belgilangan kataklarga kiritiladigan qiymatlarga muvofiq format tanlanadi. Shundan sung kiritiladigan qiymatlar tanlangan formatga muvofiq kiritilishi dastur tomonidan nazorat qilib boriladi. Masalan, ustunga son qiymat kiritish formati tanlangan bo`lsa, ushbu ustunning birorta ham katagiga harf qiymatni kirita olmaymiz. Sonli qiymatlar uchun yuqorida tavsiya qilingan formatlardan tashqari, yana qo`shimcha formatlar ham mavjud. Ularni tanlash uchun formatlar oynasidan “Другие числовые форматы” buyg`ugini tanlasak qo`shimcha oyna ochiladi.



Ustunga kiritiladigan qiymatlarga muvofiq formatlarni tanlash, qiymatlarni bir xilligini ta'minlaydi va ustunga muvofiq bulmagan qiymatni kiritishning oldini oladi.

**Formatlar maskalari.** Son formatlari maskalari kataklarda qiymatlarni aks ettirishni boshqarishga imkon beradi. Maskalarning ahamiyati quyidagicha:

Asosiy – sonlar kiritilgan paytdagi formatda chiqariladi.

- # – agar o`nli verguldan o`ngdagi son formatning o`ng qismida ko`rsatilgan # belgilarga qaraganda ko`proq miqdordagi raqamlardan iborat bo`lsa, u holda son verguldan keyingi o`nli belgilarning ko`rsatilgan miqdorigacha yaxlitlanadi. Agar son verguldan chapda formatning chap qismida ko`rsatilgan # belgilarga qaraganda ko`proq miqdordagi raqamlardan iborat bo`lsa, u holda raqamlarning katta miqdori chiqariladi.

- 0 (nol) – # ning o`zi, biroq agar sonda shablonda ko`rsatilgan 0 belgilardan kamroq raqamlar miqdori bo`lsa, yo`q bo`lgan nollar chiqariladi.

- ? – 0 ning o`zi, yagona farqi shundaki, Excel o`nli vergulning har ikkala tomonidan arziyas nollar o`rniga probel chiqaradi. Shu tarzda o`nli vergul bo`yicha to`g`rilangan bo`lib qoladi. Shuningdek, bu simvolni o`nli belgilarning turli soniga ega ratsional kasrlar uchun qo`llash mumkin.

- , (vergul) – bu simvol o`nli verguldan o`ngda va chapda qancha raqam (0 va # simvollarini miqdoriga muvofiq) chiqarilishini aniqlaydi. Verguldan shuningdek kun vaqtlari shablonlarida soniya qismlarini aks ettiruvchi kun vaqti formatini yaratish uchun foydalanish mumkin.

- %% – son 100 ga ko`paytiriladi va % belgisi qo`shiladi.

- (probel) – minglarni ajratuvchi. Agar shablon ikki tomondan # yoki 0 belgilari bilan o`rab olingan probellarni o`z ichiga olsa, minglar probellar bilan ajratiladi. Shablon belgisi ortidan keluvchi probel sonni minggacha yaxlitlab masshtablashtiradi. 0,0 simvollar ketidan ikkita probel keluvchi shablon 43 500 000 sonini 43,5 sifatida ko`rsatadi.

- E+00 – sonning eksponentsial formati.

- grn. – + / ( ) : – shu simvollar aks ettiriladi.

- \ – undan keyin keladigan format simvoli aks ettiriladi. "Teskari qiya chiziq" simvolining o`zi aks ettirilmaydi. Undan keyin keladigan simvol qo`shirnoq ichiga olingan bo`lgandagi singari amal qiladi.

- \* uning ketidan keluvchi format simvolini ustun enini to`ldirish uchun necha marta talab qilinsa shuncha marta takrorlaydi. Formatning har bir format sektsiyasida bittadan ortiq " \* " simvoli bo`lishi mumkin emas.

- tagiga chizish ( ) – probelni keyingi shablon simvoliga mos keluvchi kenglikka aylantiradi. Masalan, Microsoft Excel qavslar simvoli eniga teng probel qilishi va qavslar ichiga solingan musbat sonlar manfiy sonlar singari to`g`rilanib qolishi uchun musbat sonlar uchun format sektsiyasi oxirida ) kiritish lozim.

- "mekm" – qo`shirnoq ichiga olingan har bir matnni aks ettiradi.

- @ – agar yacheykada matn bo`lsa, u holda bu matn formatning @ simvoli joylashgan joyida ko`rsatiladi.

Maskalardan foydalanish misollari

Kiritilgan son	Maska	Sonning tasvirlanishi
76, 364	000, 000	076, 364
44, 894	###, ###	044, 894
7636, 4	#####, #	7636, 4
781,7	\$###,	\$782,
7,6	#,000	7,600
2,4	000,#	002,4
3,25	#??? /???	13/4
0,8	?,#	,8
0,2	0,00	0,20

0,7	#,00	,70
0,4	?,#%	40%

### 3. Funktsiyalar masteri bilan ishlash

Excel EJ kataklariga matnlar, sonlar va formulalar kiritiladi. Baʼzan katakdagi qiymatlar ustida ayrim hisoblashlarni bajarish zaruriyati tugʻiladi. Bunday hollarda formulalardan foydalaniladi. Umuman ixtiyoriy katakdagi sonlarni hisoblash yoki formulalarni hisoblash uchun “=” (tenglik) belgisidan foydalanadi.

Masalan, A5 katakdagi sonni D7 katakdagi songa koʻpaytirish uchun kerakli kattakka quyidagi yoziladi;

=A5\*D7

Excelda ishlatiladigan arifmetik amallar belgilari quyidagilar:

+ – qoʻshish;

- - ayirish;

\* – koʻpaytirish;

/ – bulish;

^ – darajaga kutarish

& - bir necha katakdagi matnlarni

Formula – bu bir katakning qiymatini hisoblaydigan ifodadir. Funktsiyalar esa, oldindan belgilangan formulalar va ular Excelda allaqachon mavjud.

Misol uchun, quyidagi A3 katak A2 va A1 kataklarning qiymatlarini qoʻshadigan formulani oʻz ichiga olgan.

	A	B	C	D	E
1	2				
2	3				
3	5				

Misol uchun, quyidagi A3 katak A1:A2 sohaning yigʻindisini hisoblaydigan SUM funksiyasini oʻz ichiga olgan.

	A	B	C	D	E
1	2				
2	3				
3	5				

**Formula kiritish** uchun quyidagilarni ketma-ket bajaring.

1. Bir katakni belgilang.
2. Excel siz formula kiritmoqchi ekanligingizni bilishi uchun barobar (=) belgisini kiriting.
3. Misol uchun, A1+A2 formulani yozing.

	A	B	C	D	E
1	2				
2	3				
3	5				

Maslahat: A1 va A2 deb yozish oʻrniga, oddiygina A1 va A2 kataklarni tanlang.

4. A1 katakning qiymatini 3 ga oʻzgartiring.

	A	B	C	D	E
1	3				
2	3				
3	6				

Excel avtomatik tarzda A3 katakning qiymatini qayta hisoblaydi. Bu Excelning eng kuchli xususiyatlaridan biri!

**Formulani tahrirlash.** Siz bir katakni tanlaganingizda, Excel o'sha katakning qiymati yoki formulasini formulalar qatorida ko'rsatadi.

A3		fx =A1+A2			
	A	B	C	D	E
1	2				
2	3				
3	5				
4					

1. Formulani tahrirlash uchun, formulalar qatoriga bosib va formulani o'zgartiring.

SUMIF		fx =A1-A2			
	A	B	C	D	E
1	2				
2	3				
3	=A1-A2				
4					

A4		fx			
	A	B	C	D	E
1	2				
2	3				
3	-1				
4					
5					

2. Enter tugmasini bosib.

**Amallar tartibi.** Excel odatdagi standart tartib bo'yicha hisob-kitoblarni amalga oshiradi. Agar formulaning biror qismi qavs ichida bo'lsa, o'sha qism avval hisoblanadi. Keyin ko'paytirish va bo'lish amallarini bajaradi. Ular tugagandan so'ng, formuladagi qo'shish va ayirishni bajaradi. Quyidagi misolga qarang.

A4		fx =A1*A2+A3			
	A	B	C	D	E
1	2				
2	2				
3	1				
4	5				
5					

Dastlab, Excel ko'paytirishni bajaradi ( $A1 * A2$ ). Keyin, A3 ning qiymatini chiqqan natijaga qo'shadi.

Boshqa bir misol:

A4		fx =A1*(A2+A3)			
	A	B	C	D	E
1	2				
2	2				
3	1				
4	6				
5					

Excel avval, qavs ichidagilarni hisoblaydi. Keyin, chiqqan natijani A1 katakning qiymatiga ko'paytiradi.

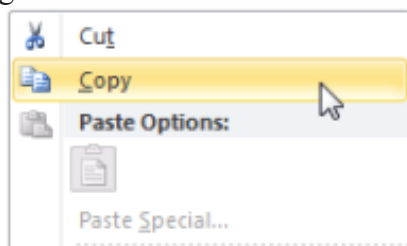
**Formuladan nusxa olish/qo'yish.** Siz formuladan nusxa ko'chirganingizda, Excel katak raqamlarini nusxa ko'chirilgan katakka mos yangi katak raqamlariga avtomatik tarzda o'zgartirib chiqadi. Buni tushunish uchun quyidagilarni bajaring.

1. Quyida ko'rsatilgan formulani A4 katakka kiriting.

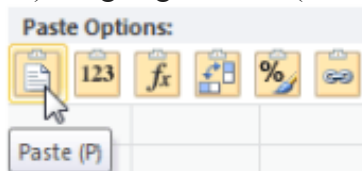


	A	B	C	D	E
1	2	5			
2	2	6			
3	1	4			
4	6				
5					

2a. A4 katakni belgilang, o'ng tomonni bosning, va Copy (Копировать) (yoki CTRL + c)ni bosning...



...so'ng, B4 katakni belgilang, o'ng tomonni bosning va 'Paste Options:' (Параметры вставки)ni tagidagi "Paste" (Вставить)ni (yoki CTRL + v ni) bosning.



2b. Shuningdek, siz formulani B4 katakka sudray olasiz. A4 katakni tanlang, uning pastki o'ng burchagiga bosning va uni B4 katakka sudrang. Bu ancha oson va aynan bir xil natijani beradi!

	A	B	C	D	E
1	2	5			
2	2	6			
3	1	4			
4	6				
5					

Natija. B4 katakdagi formula B ustundagi qiymatlar bo'yicha hisoblaydi.

	A	B	C	D	E
1	2	5			
2	2	6			
3	1	4			
4	6	50			
5					

**Hujjatni himoyalash.** MS Excel da yaratilgan faylni parol orqali himoyalash mumkin. Bundan tashqari hujjat sahifalarini, hattoki katakchalarini ruxsatsiz kirishdan himoyalash mumkin.

Excel dasturi kitobni bir nechta usul bilan himoyalashni taklif etadi. Bu hujjatni ochish, ma'lumotlarni o'zgartirish va sahifalarni o'chirishdan himoyalash imkonini beradi. Quyidagicha himoyalash usullari mavjud:

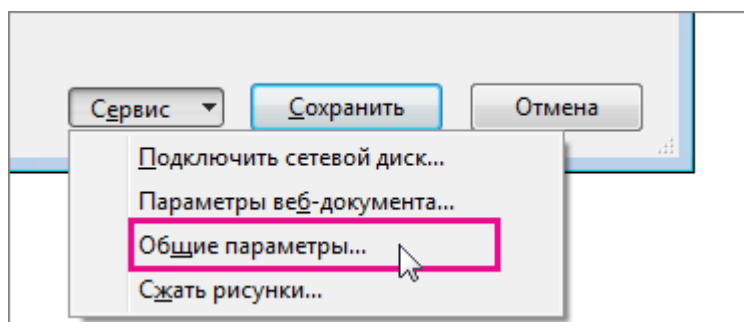
1. Faylga ruxsatni chegaralash.
2. Kitob elementlarini himoya qilishni tadbiiq etish.
3. Sahifa elementlarini himoyalash.

Har bir holda ham paroldan foydalaniladi. Parolda lotin va rus alfavitinging har xil katta va kichik harflaridan foydalanish mumkin.

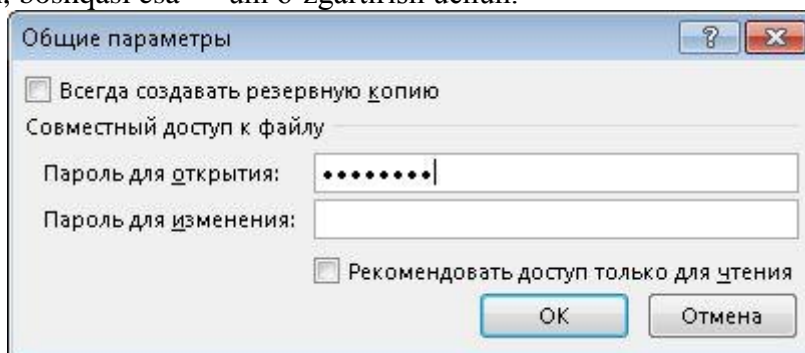
Har bir parolni kiritganda u yana qaytadan tasdiqlanishi lozim. Bu parolni kiritishda adashib ketmaslik uchun zarur.

*Faylni ochishda parol so'ralishini o'rnatishni ko'rib chiqamiz.*

1. “Файл-Сохранить как” ni tanlang.
2. Oynadagi “Обзор” tugmasini bosing.
3. Ochilgan “Сохранения документа” muloqot oynasidan faylni saqlanish joyi va parkani tanlang.
4. “Сервис” tugmasini bosing va ochilgan sohadan “Общие параметры” buyrug'ini ishga tushiring.

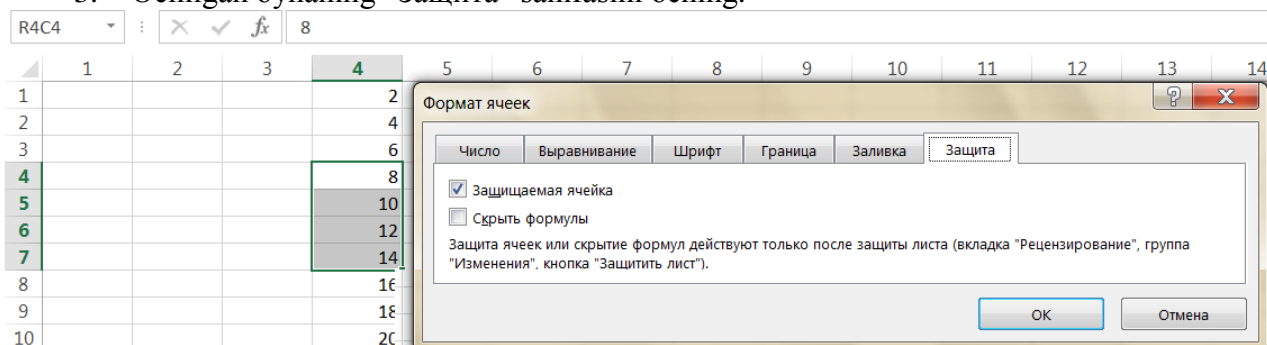


5. Ochilgan “Общие параметры” muloqot oynasida ikkita parolni kiritish lozim: bittasi— faylni ochish uchun, boshqasi esa — uni o'zgartirish uchun.



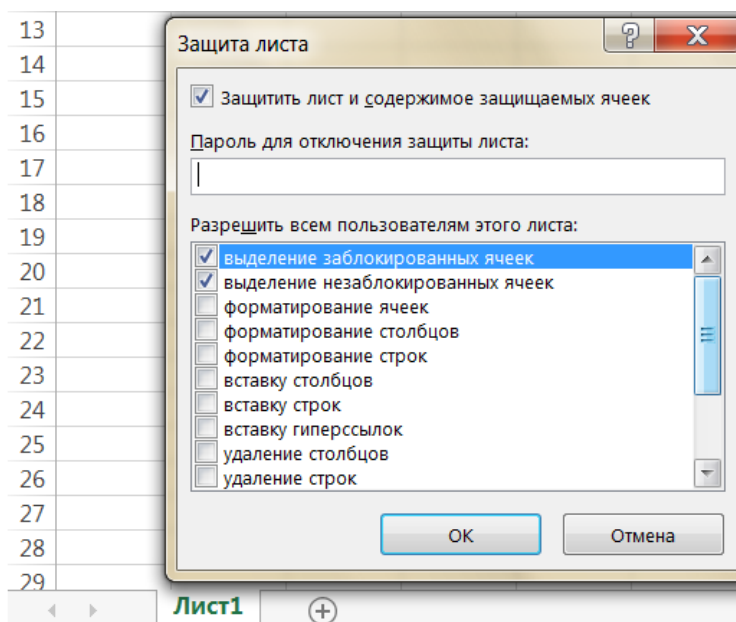
Parolni kiriting, keyin uni tasdiqlash uchun kiriting va OK tugmasini bosing.

6. OK tugmasini bosing.
  7. Oynadan “Сохранить” tugmasini bosing.
- Kataklarni himoylashni ko'rib chiqamiz.*
1. Himoyalani kerak bo'lgan kataklarni belgilab oling.
  2. Kontekst menyuni chaqiring (Belgilangan katak ichida sichqoncha o'ng tugmasi bosiladi).
  3. Ochilgan oynaning “Защита” sahifasini oching.



4. “Защищаемая ячейка” ga qayd qo'yung.
  5. OK tugmasini bosing.
- Katakka qo'yilgan himoy sahifa himoyalangandan so'ng kuchga kiradi.
- Kataklarni himoylashni ko'rib chiqamiz.*

1. Himoyalanihi kerak bo'lgan sahifani belgilab oling.
2. Kontekst menyuni chaqiring (Belgilangan sahifa ichida sichqoncha o'ng tugmasi bosiladi).



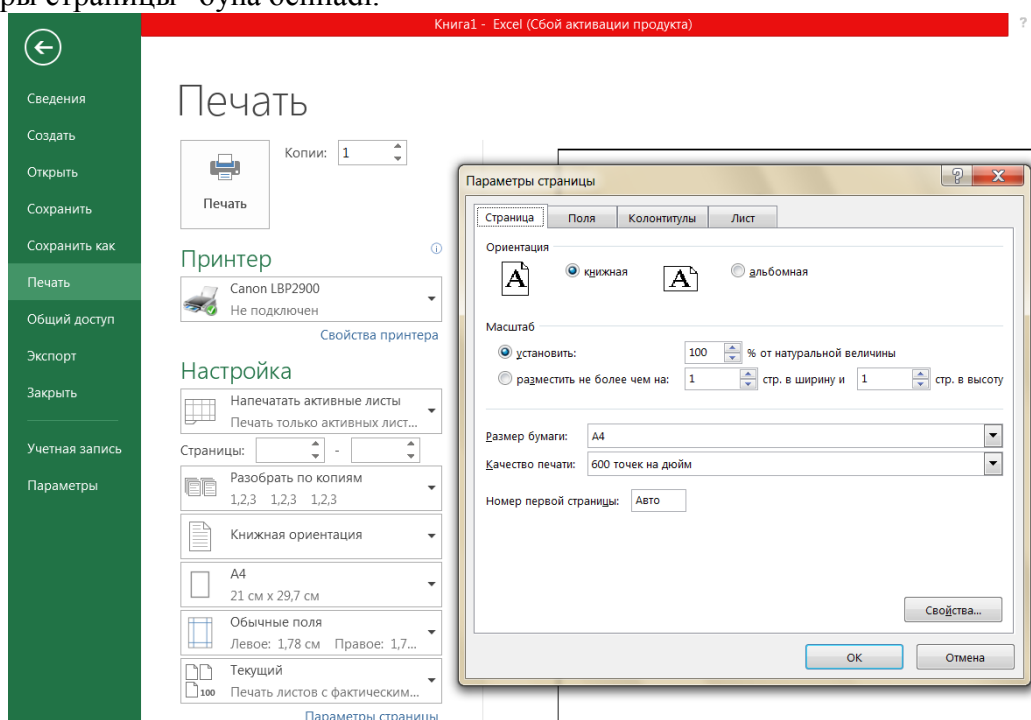
3. Ochilgan “Защита листа” оунadagi “Парол для отключения защиты листа:” so'rov satriga yashirin parol beriladi.

4. Ushbu sahifadan foydalanish ruxsatlari “Разрешить всем пользователям этого листа” sohada qayd qilinadi.

5. OK tugmasini bosing.

Kiritilgan himoya saqlanishi uchun fayl xotirada qayta saqlanishi lozim. Himoyalangan sahifani faqat parolni bilgan foydalanuvchi ocha oladi, agar parol to'g'ri kiritilmasa, ushbu sahifa ochilmaydi.

**Hujjatni chop etish va sahifa parametrlarini sozlash.** Excel dasturida ham MS Word dasturidek ma'lumotlarni qog'ozga chiqarish va ekranga chiqarish parametrlarini berish mumkin. Buning uchun “Файл-Печать” ochilib “Параметры страницы” buyrug'i ishga tushirilsa, “Параметры страницы” оуна ochiladi.



Bu oynaning: “Страница” varag’ida qog’ozda ma`lumotni tasvirlash holati paramerlari o`rnatiladi.

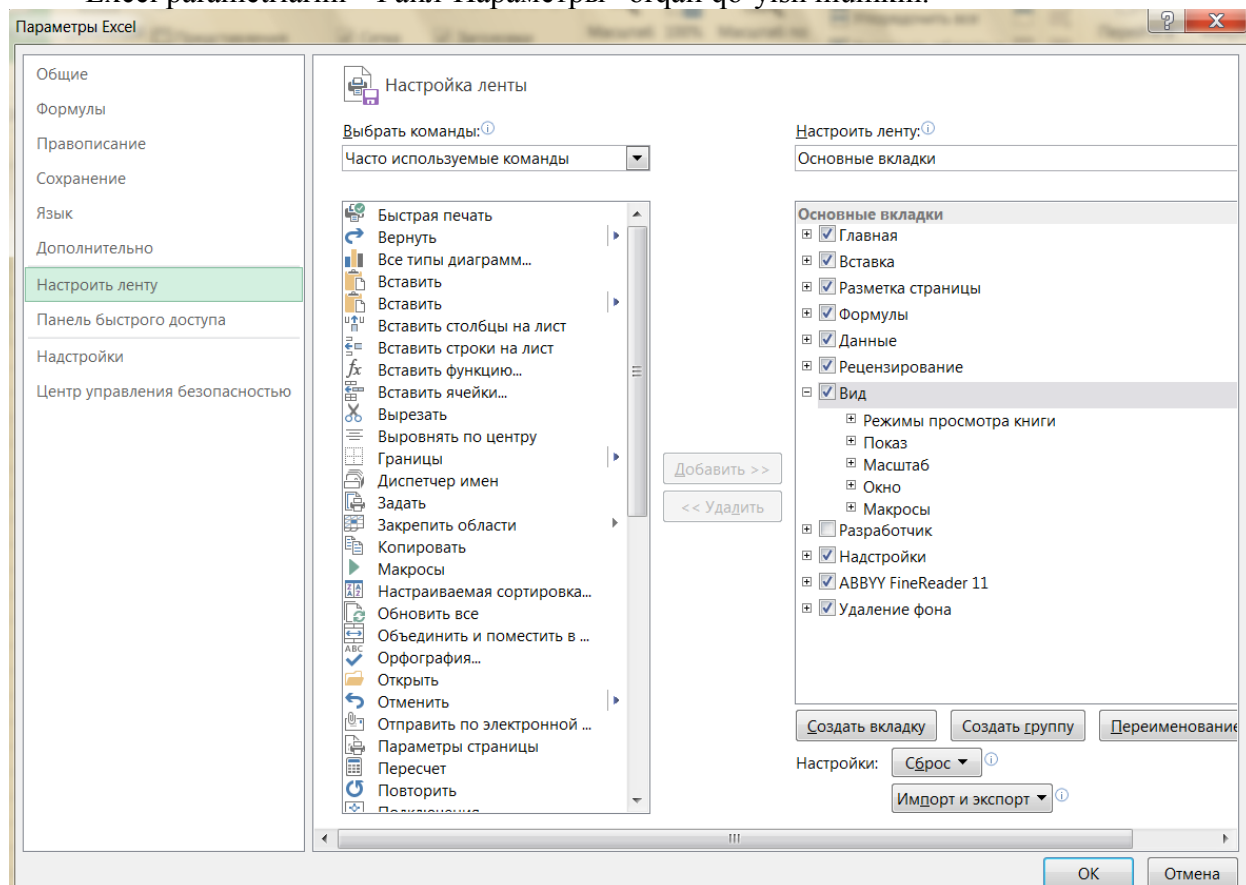
“Поля” sahifasida ma`lumotni qog’ozga chiqarish chegaralari paramerlari o`rnatiladi.

“Колонтитулы” sahifasida har bir betda beriladigan kolontitula so`zi va uning paramerlari o`rnatiladi.

“Лист” sahifasida varaqlar ma`lumotlarini qog’ozga chiqarish chegaralari paramerlari o`rnatiladi.

Ekran ishchi sahifasi masshtabi va ko`rinishi “Вид” lentasi orqali o`rnatiladi. Bu erda ekranning odatdagi ko`rinishi, sahifalarni belgilash, masala sohasi, asboblar paneli, formulalar qatorini berish, holatlar qatorini berish, kolontitula, eslatmalar, taqdimot, to`liq ekran, ma`lumotlarni tasvirlash masshtablari parametrlari o`rnatiladi.

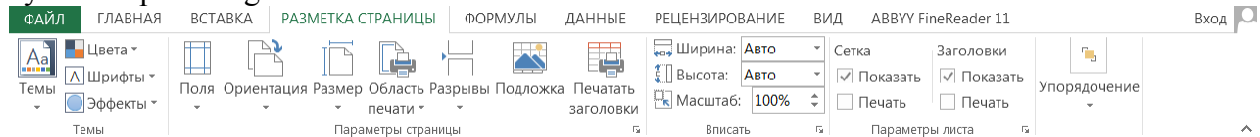
Excel parametrlarini “Файл-Параметры” orqali qo`yish mumkin.



**Хужатларни oldindan ko`rib chiqish, sozlash va chop etish.** Chop etish oynasida siz kitobning hammasini, ajratilgan varaqlarni yoki ajratilgan sohani chop etishga berishingiz, boshlang`ich va so`nggi sahifalarning tartib raqamini ko`rsatib, sahifalarning aniq sohasini chop etish, bir nechta nusxa bosish va ularni nusxalarga qarab saralashingiz mumkin. Bundan tashqari agar siz oldindan ko`rib chiqish rejimiga qaytmoqchi bo`lsangiz, buning uchun ko`rib chiqish tugmachasini ishlatishingiz lozim.

Agar sizning kompyuteringizda boshqa printerni ulash imkoniyati bo`lsa, uni nom darchasidan tanlash mumkin.



Chop etishda qog`ozning o`lchamini, mo`ljalinini va maydon o`lchamini belgilash mumkin. Jadvalni qoida bo`yicha chop etishga tayyorlash zarur. “Разметка страницы” lentasi hujjatni aynan chop etiladigan ko`rinishida ko`rish imkonini beradi.

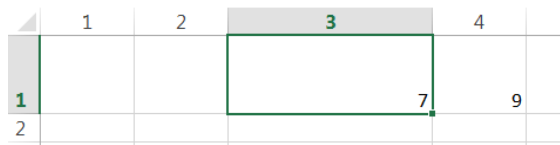



Bu rejimda sahifalarning parametrlarini sozlash, kolontitullar yaratish va o`zgartirish, hujjatni chop etishni belgilash mumkin. Bundan tashqari, hujjatni butunlay emas, balki qisman

chop etish yoki bir necha nusxada chop etish imkoniyatini ham beradi. Bunda chop etish parametrlarini sozlash lozim.

Agar jadval chop etish jarayonida bitta sahifaga sig'masa, Excel avtomatik ravishda uni sahifalarga bo'lib yuboradi.

Kataklarni birlashtirish uchun “Главная” lentadan  tugmani, ma'lumotlarni jadval holiga keltirish uchun esa ma'lumotlar belgilangach  tugmani bosish kerak



*Jadval ustun kengligi va satr balandligini o'zgartirish, ishchi kitob sahifalari bilan ishlash.* Jadvalning  ixtiyoriy katagi o'lchamini o'zgartirish mumkin. Katak o'lchami o'zgartirilsa, shu katak joylashgan ustunga tegishli barcha kataklar ustuni kengligi bir xil bo'ladi. Xuddi shuningdek, shu katak joylashgan qatorga tegishli barcha kataklar qator kengligi bir xil bo'ladi. Katakni eni va bo'yiga o'lchamini o'zgartirish uchun, uning ustuni va qatori o'lchamini o'zgartirish kerak. Masalan C3 katak o'lchamini o'zgartirishni ko'raylik. Buning uchun ustunning o'ng chegarasiga sichqoncha ko'rsatkichi olib borilsa belgi paydo bo'ladi. Bu holatda sichqoncha chap tugmasi bosilgan holda ustun chegarasi oshiriladi yoki kamaytiriladi. Qatorida ham shu ish amalga oshiriladi.



Ishchi sahifalar boshlang'ich holda “Лист1”, “Лист2” va h-zo deb nomlanadi. Yangi sahifa qo'shish uchun “+” tugma bosiladi. Ishchi sahifaning nomini o'zgartirish uchun kerakli ishchi sahifa varag'i nomi ustiga sichqoncha ko'rsatkichi olib borilib, ikki marta sichqoncha chap tugmasi chiqillatilsa yoki kontekst menyusidan “Переименовать” buyrug'i tanlansa, u qora fonda belgilanadi. Shundan so'ng nomni kiritish mumkin.

Ishchi sahifani o'chirish uchun kerakli ishchi sahifa nomi ustiga sichqoncha ko'rsatkichi olib borilib, kontekst menyusidan “Удалить” buyrug'i tanlanadi. Agar ishchi sahifada ma'lumot mavjud bo'lsa, ogohlantirish beriladi va shunda ham “Удалить” tugma bosilsa, ishchi sahifa o'chiriladi. Ishchi sahifa o'chirilsa, undagi ma'lumotlar ham o'chadi.

Yangi ishchi sahifani qo'yish uchun sahifa yonidagi + “Новый лист” tugma bosilsa yangi ishchi sahifa qo'yiladi. Bu ishni kontekst menyuning “Добавить” buyrug'i orqali ham bajarish mumkin.

Ishchi sahifalarning o'rmini o'zgartirish uchun kerakli ishchi sahifa varag'i nomi ustiga sichqoncha ko'rsatkichi olib boriladi, sichqoncha chap tugmasi bosilgan holda kerakli sahifadan keyinga suriladi.

Elektron jadvalda kataklar va soha bilan ish olib boriladi. Soha kataklar guruhidan taskil topadi va unda murojaat qilish uchun maxsus belgilardan foydalaniladi:

- : (ikki nuqta) – kataklar blokiga murojaat qilishni shakllantiradi. Ikki nuqta orqali blokning chap yuqori va o'ng pastki kataklari ko'rsatiladi. Masalan: S4:D6 – S4, S5, S6, D4, D5, D6 kataklariga murojaat qilishni anglatadi.

- ; (nuqtali vergul) – kataklar birlashmasini ifodalaydi. Misol uchun, D2:D4; D6:D8 – D2, D3, D4, D6, D7, D8 kataklariga murojaat qilishni anglatadi.

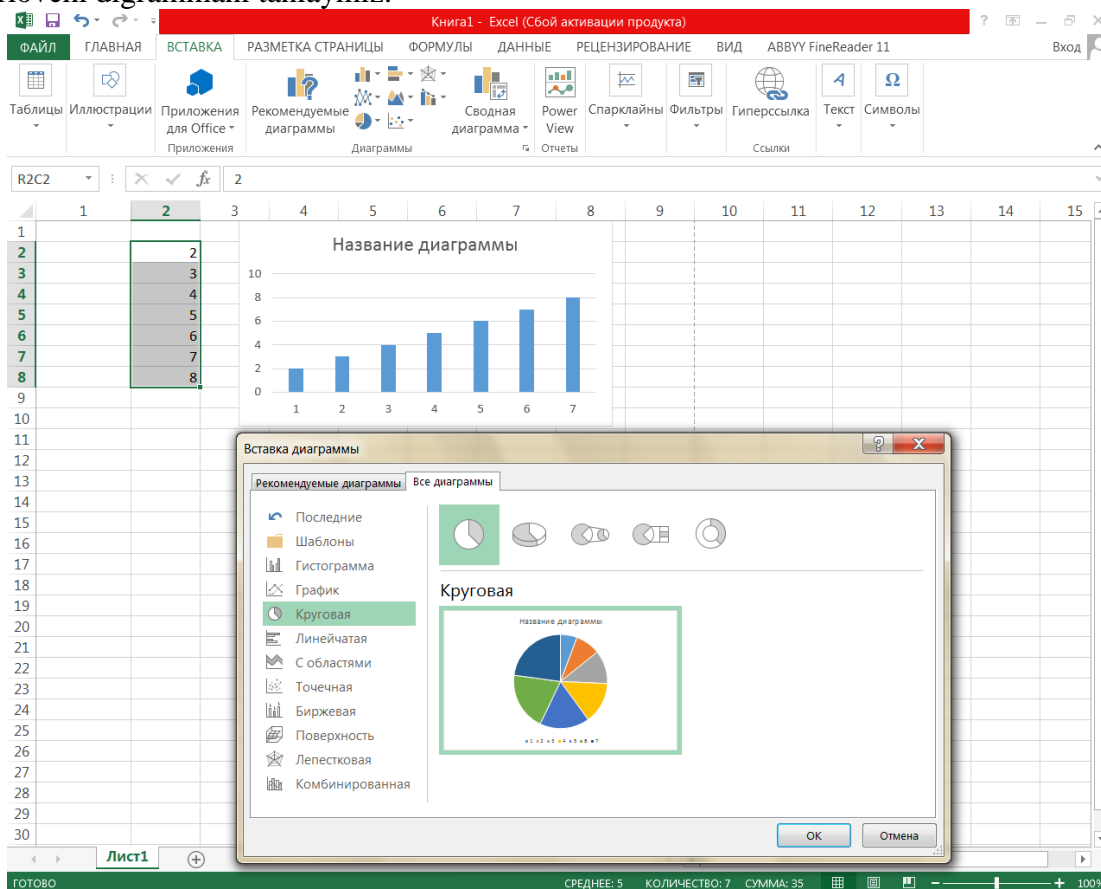
#### **4. Diogrammalar, grafiklar tuzish va boshqa shu kabi imkoniyatlardan foydalanish**

*Diagrammalar yaratish.* Excel dasturida jadval ma'lumotlarga qarab har xil diagrammalar yaratish imkoniyati bor. Diagrammalar bizga ma'lumotlar bilan ishlashni osonlashtiradi.

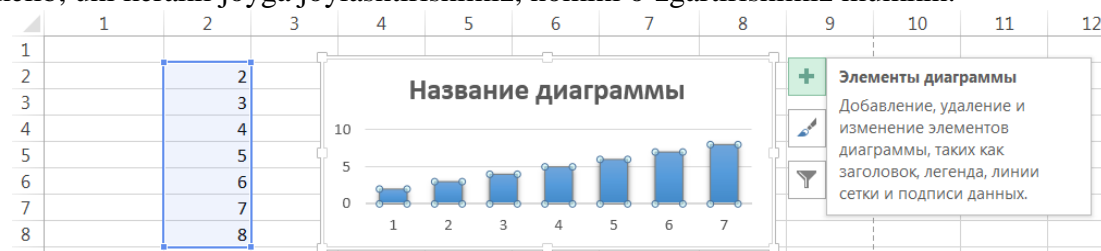
Diagramma — bu Exsel hosil qiladigan ob'ektlardan biri hisoblanadi. Bu ob'ekt bitta yoki bir nechta ma'lumotlar to'plamining grafik tasviridan iboratdir (ma'lumotlarning qaysi ko'rinishda tasvirlanishi diagramma turiga bog'liq).

Diagrammani yaratishdan oldin jadvalni tayyorlashimiz, belgilangan sohaga kerakli qiymatlarni kiritishimiz kerak. Jadval tayyor bo'lgandan keyin, jadval ma'lumotlari sohasini sichqoncha bilan belgilab olamiz, aks holda diagramma yoki grafikni yasay olmaymiz. So'ngra

“Вставка” lentasining “Диограммы” guruhidan kerakli diogrammalardan birini tanlaymiz. Masalan, “Рекомендуемые диограммы” ni tanladik. Natijada diogrammalar turlari tasvirlangan “Вставка диограммы” oynasi ochiladi. Oynada gistogrammala, grafiklar hamda aylana, chiziqli, sohaviy, nuqtali, birja, yuzadagi, yaproqli diogrammalar singari guruhlari mavjud. Har bir guruhda diogrammalarning turli ko`rinishlari tavsiya qilinadi. Oynadan ma`lumotlarni qulay shaklda tasvirlovchi digrammani tanlaymiz.



Hosil bo`lgan diagrammani tahrirlash uchun sichqoncha ko`rsatkichini diagramma ustiga olib kelib, uni kerakli joyga joylashtirishimiz, nomini o`zgartirishimiz mumkin.



Diagrammani belgilab, bir joydan ikkinchi joyga surish, o`lchamini o`zgartirish yoki asboblari panelidagi “Вырезать”, “Копировать”, “Вставить” tugmalari yordamida diogrammalarni boshqa fayl va sahifalarga o`tkazish mumkin.

Eng avvalo yaratgan diogrammamiz nomini qo`yishimiz lozim. Buning uchun “Название диограммы” so`zi ustida sichqoncha tugmasi ikki marta chiqillatilsa, so`zni tahrirlash rejimiga o`tamiz. Kerakli so`z kiritilgach, uning shriftini, o`lchamini va rangini o`zgartirishimiz mumkin. Shuningdek, diogramma sohasi rangi, chiziqlari va diogramma ob`ektlari rangini ham o`zgartirish mumkin.

Excel diagrammasi ishchi jadval sahifasining ma`lumotlari bilan bevosita bog`liq. Ishchi jadval sahifasidagi ma`lumotlar o`zgartirilsa, tezda ularga bog`liq bo`lgan diagramma chiziqlari o`zgaradi. Diagrammalarning bir necha xil turlari mavjud:

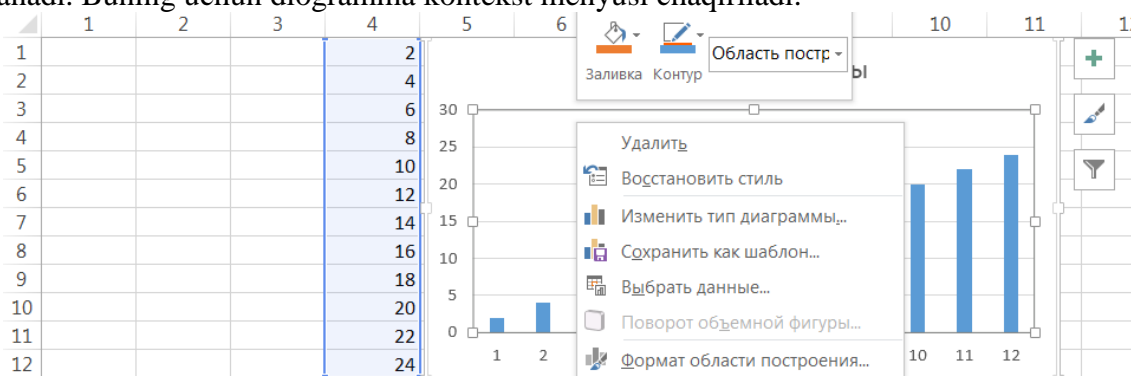
- chiziqli;
- doiraviy;



- grafikli;
- halqasimon;
- dumaloq;
- va boshqalar.

Excelda diagrammalarning ixtiyoriy turini tuzish mumkin. Ayrim diagrammalar juda murakkab shakllarni aks ettiradi. Masalan, konusli, tsilindrli piramidali va h-zo.

Diagrammalar hosil qilingandan keyin u o'zgaras holatda bo'lmaydi, balki har doim uning shaklini o'zgartirib turish va maxsus bichimlash atributlarini qo'shish, yangi ma'lumotlar to'plami bilan to'ldirish, mavjud ma'lumotlar to'plamini boshqa soha kataklardan foydalaniladigan qilib ma'lumotlarni o'zgartirish mumkin. Alohida varaqda joylashtirilgan diagrammalar butun varaqni egallaydi. Agar bir necha diagramma tuzmoqchi bo'lsangiz, unda har birini alohida sahifalarga joylashtirish maqsadga muvofiq. Shunda varaqdagi diagrammalarning "ko'rinarlilik darajasi" saqlanadi. Buning uchun diagramma kontekst menyusi chaqiriladi.



*Diagramma sahifalari.* Alohida sahifada joylashgan diagramma sahifani to'lig'icha egallaydi. Agar siz diagrammaning o'zini alohida sahifada chop etmoqchi bo'lsangiz, uni alohida sahifaga joylashtirgan ma'qul. Agar siz ko'p diagramma yasamoqchi bo'lsangiz, ularning har birini alohida sahifada joylashtirgan ma'qul. Bu holda diagrammalar bir-biriga kirishib ketmaydi. Bundan tashqari sahifa yorliqlariga diagramma nomlarini bersangiz, ularni qidirish ham osonlashadi.

Har bir diagrammaning muhim tomonlaridan biri, unda foydalaniladigan o'qlar sonidir. Oddiy diagrammalar: gistogramma, grafiklar va boshqalarda ikkita o'q, mavjud bo'lib, ulardan biri kategoriya, ikkinchisi esa qiymatlar o'qi deb ataladi. Odatda kategoriya o'qi gorizontal, qiymatlar o'qi esa vertikal yo'nalgan bo'ladi.

Ixtiyoriy vaqtda diagramma o'lchamini, sarlavhasini, o'qlar nomini hamda izohni tahrirlash mumkin. So'zlar shrifti o'lchami, turi va rangini ham o'zgartirish mumkin. Buning uchun kerakli ob'ektga sichqoncha ko'rsatkichi olib borilib, chap tugma bir marta bosilsa, tahrirlash rejimi o'rnatiladi.

Doiraviy va halqali diagrammalarda o'qlar bo'lmaydi. Yaproqli diagrammalar alohida xususiyatga ega bo'lib, ma'lumotlar to'plamining har bir nuqtasi uchun alohida o'q mavjud. Bu o'qlar diagrammaning markazidan chiqqan bo'ladi.

Uch o'lchovli (fazoviy) diagrammalarda uchta o'q bo'lib, ular: kategoriya o'qi, qiymatlar o'qi va ma'lumotlar to'plami o'qi deb ataladi.

Boshqa grafik ob'ektlardek (matnli oyna, to'g'ri to'rtburchak) diagrammalarni siljitish, o'lchamlari va proporsiyalarini o'zgartirish, chegaralarini surish va bir qator boshqa o'zgarishlar kiritish mumkin. Boshqa grafik ob'ektlar kabi diagrammalarni bir joydan ikkinchi joyga ko'chirish ham mumkin. Uning o'lchamlarini, nisbatlarini o'zgartirish, chegaralarini to'g'rilash va ular ustida boshqa amallarni bajarish ham mumkin. Joriy qilingan diagrammaga o'zgartirishlar kiritish uchun sichqonchani chap tugmasini ikki marta bosish kerak bo'ladi. Bunda diagrammalar faollashib, diagramma kontekst menyusi diagrammalar bilan ishlash uchun kerakli buyruqlarni ko'rsatadi. Joriy qilingan diagrammalarning asosiy afzalligi shundaki, ularni diagramma tuzish uchun foydalanilgan ma'lumotlar yoniga joylashtirish mumkin.

*Diagramma ko`rinishini o`zgartirishning asosiy turlari.* Diagramma yasalgandan so`ng, xohlagan paytingizda siz uning ko`rinishini o`zgartirishingiz mumkin. Buning uchun u faollashgan bo`lishi shart. Tatbiq etilgan diagrammani faollashtirish uchun unda sichqoncha tugmasini bosing. Diagramma sahifasini faollashtirish uchun esa sahifasi yorlig`ini bosing.

Diagrammani o`zgartirishning usullari ko`p. Bu erda ulardan eng ko`p tarqalganlarini keltiramiz:

- Diagramma turini o`zgartirish.
- Diagramma elementlarini almashtirish.
- Diagramma elementlarini yo`qotish.

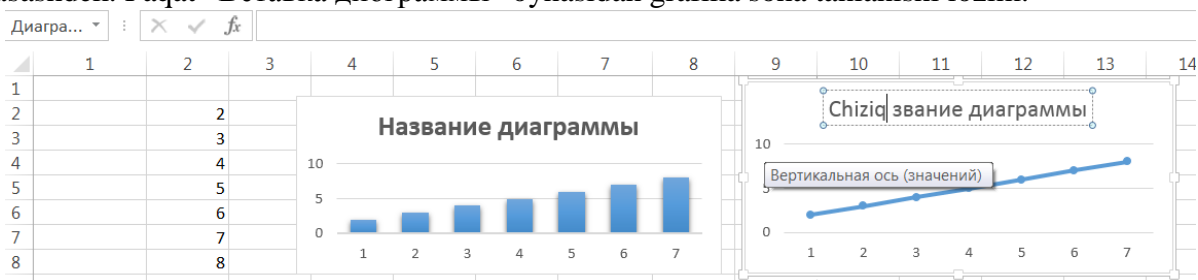
Bu ishlar moc elementlarni tanlash yo`li bilan amalga oshiriladi, sichqoncha o`ng tugmasini bosib, hosil bo`lgan kontekst menyudan kerakli bo`lim tanlanadi.

*Diagrammalarda matn.* Diagrammalar tuzishda ikki turdagi, bog`liq va erkin matnlar tavsiya etiladi. Bog`liq matn - bu diagramma elementlarining sarlavhasi, erkin matnlarni biz o`z xohishimiz bilan kiritishimiz mumkin.

Biz butun diagrammaga yoki uning o`qlariga sarlavha qo`yishni ko`rib o`tdik. Bu matnlar bog`liq matnlarni chaqiradi. Ularni tahrirlash mumkin.

Erkin matnni siz qayta ishlash qatoriga kiritishingiz mumkin. Lekin bunda diagrammada bitta ob`ekt belgilanmagan bo`lishi kerak, bo`lmasa bu matn sarlavha bo`lishi mumkin. Matn kiritilgach Enter tugmasi bosilsa, so`z diagramma o`rtasida paydo bo`ladi. Matn maydonidagi ramka o`lchamini hamda shrift o`lchamini, uning qalin yoki kursiv ko`rinishlarini tanlash va rangini o`zgartirish mumkin.

*Funksiyalarning grafiklarini qurish va tabulyatsiyalash.* Grafiklarni yasash ham diagramma yasashdek. Faqat “Вставка диаграммы” oynasidan grafika soha tanlanishi lozim.

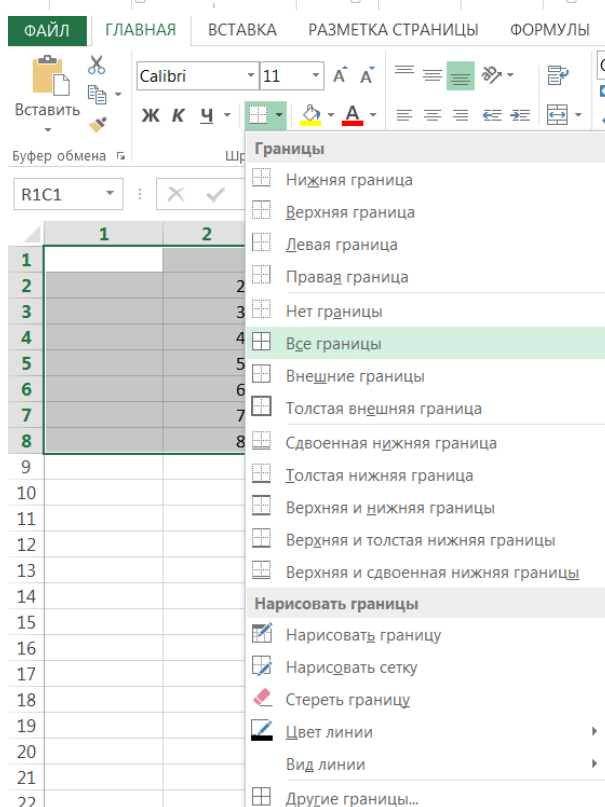


Grafikni yanada ko`rkamliroq qilish uchun tur ishlatish mumkin. Turlarning vertikal va gorizontal chiziqlarini faollashtirish hamda o`chirish mumkin.

Diagrammani ixtiyoriy vaqtda parametrlari o`zgartirgandek, grafik parametrlarini u yasalgandan so`ng, xohlagan paytda o`zgartirish mumkin.

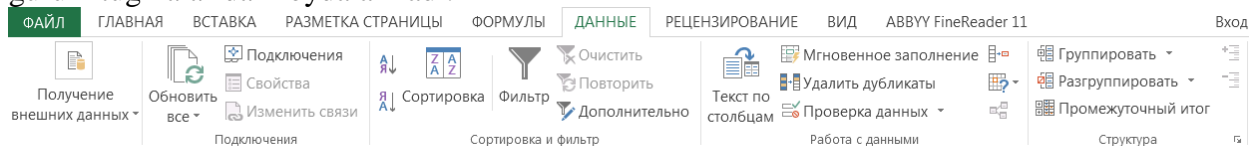
Excel electron jadvali sohasi jadval chiziqlari qogozda tasvirlanmaydi. Ularni tasvirlash uchun soha belgilanadi va “Главная” lentadan “Границы” tugma bosiladi va ochilgan oynadan “Все границы” buyrug`i ishga tushiriladi.

**Jadval ma`lumotlarini guruhlash, tartiblash, saralash va fil`trlash. Ma`lumotlarni saralash.** Excel da kiritilgan ma`lumotlarni saralash mumkin. Saralashda sonlar o`sib borish yoki kamayib borish tartibida, matnlar esa alfavit harflarining boshlanishidan yoki oxiridan boshlab





saralanadi. Buning uchun ustunlar belgilanadi va “Данные” lentasining “Сортировка и фильтр” guruhi tugmalaridan foydalaniladi.

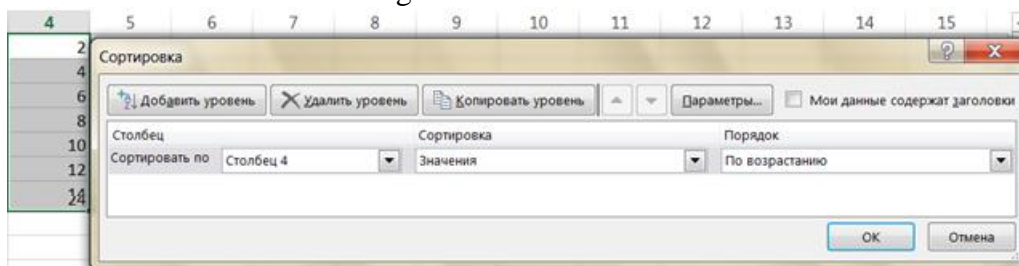


Excel da saralash quyidagicha amalga oshiriladi:

- Agar saralash bir ustun bo'yicha qilinayotgan bo'lsa, bu ustunning bir xil ma'nodagi qatorlarining avvalgi tartibi saqlanadi.

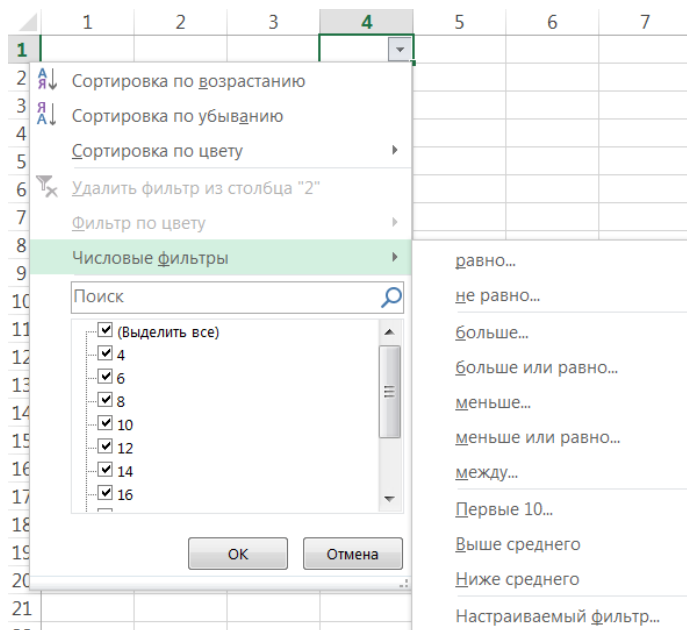
- Ustundagi tartiblashtirishda hosil bo'layotgan qatorning bo'sh kataklari tartiblashtirilgan ro'yxatning oxirida joylashadi.

Saralash guruhidan “Сортировка” tugma bosilsa saralash parametrlarini o'rnatish oynasi ochiladi va unda kerakli tanlovlar amalga oshiriladi.



Saralashni asboblar panelidagi tugmalaridan birini bosish orqali ham amalga oshiriladi.

*Ro'yxatdagi ma'lumotlarni fil'trlash.* Bir necha sonli yozuvlarni qayta ishlash, ro'yxatda ma'lumotlarni qandaydir shart bo'yicha topish imkonini beradi. Fil'tr qo'llanilgach, ekranda ma'lum shartlarga javob beruvchi qatorlarga qoladi, kerakmas qatorlar esa yashiriladi. Filtr qo'yish uchun ustunning birinchi katagi faollashtirilib “Фильтр” tugma bosilsa, katakda ko'rsatkich paydo bo'ladi va u bosilsa, fil'trlanadigan tartib tanlanadi.

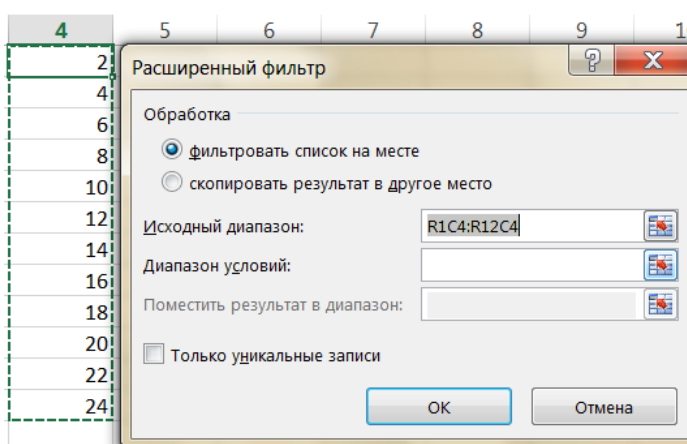


Bunda mos qatorlar bo'yicha chiqarish kerak bo'lgan elementni ajratish lozim. Agar amallar to'g'ri bajarilsa, ekranda faqat fil'trda ko'rsatilgan qatorlar qoladi.

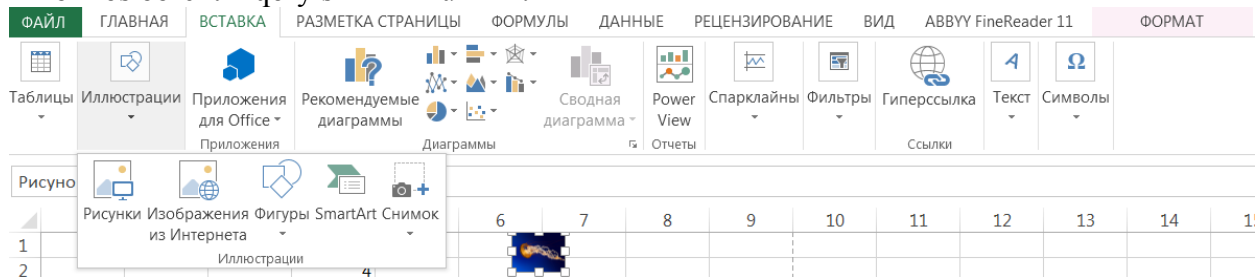
Fil'trlashni bekor qilish uchun fil'trlash belgilangan katak belgilanadi va “Фильтр” tugma bosiladi.

Fil'trlashni qo'shimcha imkoniyatlarini ishlatish uchun “Дополнительно” tugma bosilib, kerakli shartlar o'rnatiladi.

**Excel da ob'ektlarni joylashtirish.** Excel da Windows qo'llab qo'vvatlagan har qanday ob'ektni qo'yish mumkin. Buning uchun “Вставка” lentasining “Иллюстрация” tugmasi



ishlatiladi. Bu tugma bosilsa ob'ektlar turini tanlash sohasi ochiladi. Undan kerakli ob'ekt turiga kirib mos ob'ektni qo'yishimiz mumkin.



Qo'yilgan ob'ektning o'lchamini o'zgartirish, siljitish va tahrirlash mumkin.

**Kataklarni manzili turlari.** Excel dasturi katakchasi yoki yacheykalar diapazoniga murojaatda uch turdagi manzillash usulini ishlatish imkoniyatini beradi:

- nisbiy manzil;
- absolyut manzil;
- nomlangan manzil.

Alohida ta'kidlanmagan holda formulalarda katakka murojaat nisbiy manzil orqali bajariladi.

Bu holda nusxa olish amalida formuladagi kataklarga murojaat manzillari avtomatik ravishda boshlang'ich katakka nisbatan o'zgaradi.

Quyidagi misolni ko'raylik:

	A	B	C	D
1	Blister №	Qutidagi soni	Qutidagi narxi	Umumiy qiymati
2	2	24	50	1200 ← = B2*C2
3	5	24	125	3000 ← = B3*C3
4	10	24	75	1800 ← = B4*C4
5	25	24	200	4800 ← = B5*C5

D2 katakka blisterdagi dori soni ega bo'lgan qutilardagilarning umumiy qiymatini hisoblash formulasini yozish, ya'ni B2 va S2 kataklardagi qiymatlarni ko'paytirish kerak. U xolda  $D2 \leftarrow = B2 * C2$ . So'ngra Blister №5 ega bo'lgan blisterlarning umumiy qiymatini hisoblash uchun shu formulaning nusxasini D3 katakka olinganda, biz  $B3 * C3$  formulani hosil qilamiz.

Shu tarzda, navbatdagi kataklarga nusxa olinganda:

$D4 \leftarrow = B4 * C4$ , va  $D5 \leftarrow = B5 * C5$  hosil qilamiz.

Shunday qilib, nusxa olish jarayonida formuladagi ishlatilgan nisbiy manzillar satr nomeriga nisbatan mos ravishda o'zgaradi.

Yuqoridagi jadvalning **B** ustunidagi takrorlanuvchi qiymatlarga e'tibor qilaylik. Takrorlanuvchi qiymatni bir marta katakka yozib, barcha formulalarda shu yagona ko'rsatilgan songa murojaat qilish maqsadga muvofiq bo'lar edi. U holda bizning jadvalimiz quyidagicha ko'rinishga ega bo'ladi:

	A	B	C	D
1	Blister №	Qutidagisoni	Qutidagi narxi	Umumiy qiymati
2	2	24	50	1200= B2*C2
3	5		125	= B3*C3
4	10		75	= B4*C4
5	25		200	= B5*C5

Agar nusxa olish jarayonida biz nisbiy manzillardan foydalansak, jadvalda ko'rsatilgandek, D2 katakdan tashqari, barcha kataklarda noto'g'ri natija olamiz. To'g'ri natija olish uchun biz barcha formulalarda B2 manzilni o'zgartirmasdan ushlab turishimiz kerak, ya'ni navbatdagi satrga o'tganda uning manzilini o'zgarimasdan qolishi kerak. Nusxa olish jarayonida manzili o'zgarmaydigan manzillash usuli absolyut manzillash deyiladi. Absolyut manzilning elementlari \$ belgisi bilan ajralib turadi. Bizning misolimizda D ustundagi barcha formulalar quyidagi ko'rinishga ega bo'lislari kerak:

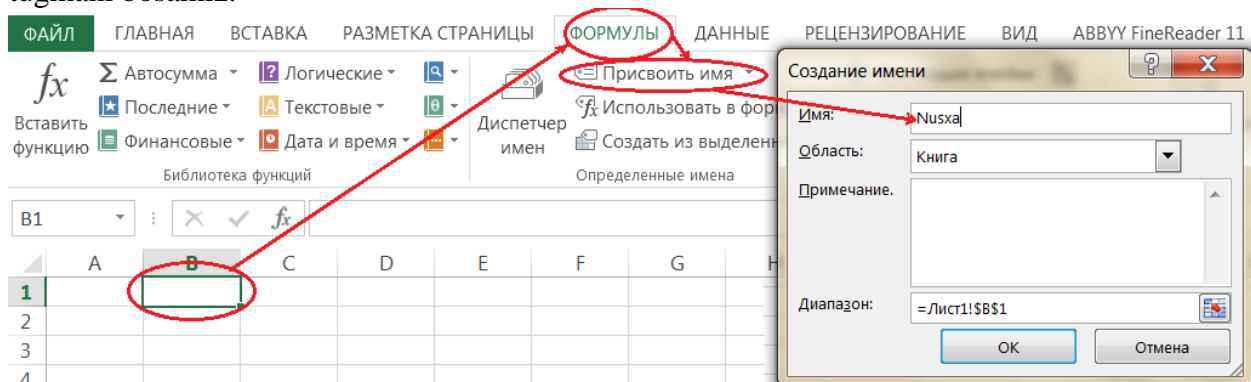
\$ B \$2\*C2;  
 \$ B \$2\*C3;  
 \$ B \$2\*C4;  
 \$ B \$2\*C5.

Buning uchun D2 yacheykaga  $B2*C2$  formula yoziladi va barcha quyidagi kataklarga nusxa ko'chiriladi. Formulani yozish jarayonida \$ belgisini F4 klavishasini bosib hosil qilish mumkin. Bizning misolimizda B2 yacheykasini faollashtirib F4 klavishasini:

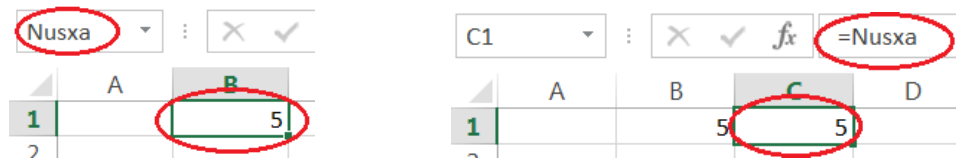
- birinchi bor ishlatilsa →\$B\$2; (ustun ham, satr ham siljmaydi)
- ikkinchi bor →B\$2; (ustun bo'yicha siljitish mumkin, satr bo'yicha siljmaydi)
- uchunchi bor →\$B2; (ustun bo'yicha siljmaydi, satr bo'yicha siljitish mumkin)
- to'rtinchi bor →yana B2. (ustun hamda satr bo'yicha siljitish mumkin)

Manzillashning yana bir usuli – bu yacheykaga nom berish. Yacheykaning nomi albatta harf yoki maxsus “\_” belgi bilan boshlanib, qolgan belgilar harflar, sonlar bo'lishi mumkin. Kataikka yoki diapazonga nom berish uchun,

1. Nom beriladigan katak faollashtiriladi;
2. “Формулы” sahifa ochiladi;
3. “Формулы” sahifadan “Присвоить имя” buyruq ishga tushiriladi;
4. Ochilgan “Создание имени” sohaning “Имя” qatoriga “Nusxa” nomni kiritgach, “OK” tugmani bosamiz.



5. Nom kiritilgan katak nomi “Nusxa” tarzida beriladi. Agar boshqa katakka ushbu nomni formula shaklida bersak, unda nomlangan katak qiymati beriladi.



Bir ishchi kitob ichida nomlar qaytarilishi mumkin emas. Ishchi kitobning nomlangan yacheykalari ro'yxatini ko'rish uchun “Имя” maydonini faollashtirish mumkin.

C1 katakka  $=A1*Nusxa$  formulani kiritgach, ushbu katakni faollashtiramiz. Sichqoncha ko'rsatkichini katakning quyi o'ng burchagiga olib borsak, “+” belgi paydo bo'ladi. Shu holatda sichqonchani C4 katakkacha siljitamiz. Formula mos holda ushbu soha kataklariga ko'chiriladi, ya'ni:

$C1=A1*Nusxa$   
 $C2=A2*Nusxa$   
 $C2=A2*Nusxa$   
 $C2=A2*Nusxa$

	A	B	C	D
1	1	5	5	
2	2		10	
3	3		15	
4	4		20	

### Nazorat savollari:

1. Elektron jadval nima ekanligini tushuntirib bering.
2. Elektron jadval dasturlari qanday maqsadlarda foydalaniladi?
3. Excel lentasi qanday buyruqlar guruhlaridan iborat?
4. Uskunalar panelining vazifasini tushuntirib bering.
5. Excel oynasi qanday sozlanadi?
6. Excel dasturida yaratilgan fayl qanday nomlanadi va u qanday kengaytmaga ega?
7. MS Excel elektron jadvalni sozlashdan maqsad nima?
8. MS Excel elektron jadvalni ekranga chiqishi bo'yicha o'rnatishni sozlash qanday amalga oshiriladi?
9. MS Excel elektron jadvalini chop etishni sozlash qanday amalga oshiriladi?
10. MS Excel asboblar panelini sozlash qanday amalga oshiriladi?
11. Elektron jadvallarda axborotlarni himoyalashdan maqsad nima?
12. Parolning vazifasi nimadan iborat?
13. Yaratilgan faylga ochish uchun parol qanday qo'yiladi?
14. Alohida kataklarga himoya qanday qo'yiladi?
15. Alohida varaqlarni ochish uchun parol qanday qo'yiladi?
16. Diagramma nima?
17. Diagramma chizish uchun qiymatlar qanday belgilanadi?
18. Belgilangan qiymatlar uchun diagrammalar qanday tanlanadi?
19. Diagrammalarni tahrirlashni tushuntirib bering.
20. Diagrammalarni qanday qilib bir joydan boshqa joyga suriladi?
21. Diagramma turi qanday tanlanadi?
22. Diagramma ko'rinishi qanday o'zgartiriladi?
23. Funktsiyalarning grafiklarini qurishning eng sodda yo'li qaysi?
24. Ma'lumotlar bilan ishlashda saralash qanday amalga oshiriladi?
25. Ma'lumotlar bilan ishlashda guruhlash qanday amalga oshiriladi?
26. Ma'lumotlar bilan ishlashda filtrlash qanday amalga oshiriladi?
27. Avtoto'ldirish deganda nimani tusghunasiz?

### 3 mavzu. MS Excel asoslari

#### Reja:

1. Funktshiya masteri  $f_x$  dan foydalanib formulalar kiritish va undan foydalanish.
2. Trigonometric, statistic, arifmetik, mantiqiy funktsiyalar bilan ishlash.
3. VBA (Visual Basic For Applications) imkoniyatlaridan foydalanish.

#### Tayanch iboralar: Funktshiya masteri $f_x$ .

#### 1. Funktshiya masteri $f_x$ dan foydalanib formulalar kiritish va undan foydalanish

Formula — bu mavjud qiymatlar asosida yangi qiymatlarni hisoblovchi ifodadir. Formulalar yordamida elektron jadvalda ko`pgina foydali ishlarni amalga oshirish mumkin. Elektron jadvallar formulalarsiz oddiy matn muharririga aylanib qoladi va uni formulalarsiz tasavvur qilish qiyin.<sup>12</sup>

Formulalar har doim “=” belgisi bilan boshlanadi. Ushbu belgi yordamida Excel qiymatlarni va formulalarni farqlaydi. Excel da boshlang`ich ma`lumotlar o`zgartirilsa, formulalar qiymati qayta hisoblab chiqiladi.

Jadvalga formulani qo`yish uchun uni kerakli katakka kiritish kerak. Formulalarni ham boshqa ma`lumotlar singari o`zgartirish, saralash, ulardan nusxa ko`chirish va o`chirish mumkin. Formuladagi arifmetik amallar sonli qiymatlarni hisoblashda, maxsus funktsiyalar matnlarni qayta ishlashda hamda katakdagi boshqa qiymatlarni hisoblashda ishlatiladi.

Formulalarda kerakli ma`lumotlarni saqlovchi soha va kataklar nomi yoki manzili ko`rsatiladi. Formulalarda kataklar nomi faqat lotin harflar bilan berilishi shart. Masalan: D10 yoki A1:E8. Shuningdek,  $c=a+4b$  ifodani xisoblash talab qilingan bo`lsin. Ifodada ishtirok etayotgan a o`zgaruvchini A1 katakka va b o`zgaruvchini B1 katakka joylashtirsak, c natija C1 katakda yozilishi kerak. Bu holda formula C1 katakda  $=A5 + 4 * B6$  shaklida yoziladi.

Agar katakdagi formula qiymatlari boshqa katakdagi qiymatlarga bog`liq bo`lsa, katakdagi qiymatlarning o`zgarishi bilan, formula natijasi avtomatik tarzda o`zgaradi.

Formula quyidagi elementlardan ixtiyoriy birortasini o`z ichiga olishi mumkin:

*Sonlar va matnlar.* Formuladagi hisoblashlarda qatnashayotgan sonlar va matnlar boshqa kataklarda joylashgan bo`lishi mumkin bo`lsada, ularning ma`lumotlarini oson almashtirish mumkin.

*Operatorlar.* Bittadan oshiq operatoridan tuzilgan formulani tuzishda Excel bu operatorlarni tahlil qiladi. Bunda standart matematik qoidalarga asoslanadi. (Arifmetik amallarni bajarish tartibi saqlanib qoladi.)

Excelda formulalarni hisoblash va bajarish quyidagi tartib asosida amalga oshiriladi:

Birinchi bo`lib qavs ichidagi ifodalar qarab chiqiladi. Undan keyin amallar bajarish tartibi saqlangan holda operatorlar bajariladi. Agar formulalarda bir xil tartibli bir necha operatorlar bo`lsa, ular ketma-ket chapdan o`ngga qarab bajariladi.

Quyida, formulalarda qo`llaniladigan operatorlarning bajarilish tartibi keltirilgan.

Belgilar	Operatorlar	Bajarilish tartibi
^	Darajaga ko`tarish	1
*	Ko`paytirish	2
/	Bo`lish	2
+	Qo`shish	3
-	Ayirish	3
&	Konkatenatsiya	4
=	Tenglik	5
>	Dan katta	5
<	Dan kichik	5

<sup>12</sup> Минько А.А. Принятие решений с помощью Excel. Просто так дважды два. – М.: Эксмо, 2009.

Agar formula katakka kiritilsa, unda katakda kiritilgan formula asosidagi hisob-kitob natijasi ko`rinadi. Lekin formulaning o`zi tegishli katak faollashtirilsa formulalar qatorida paydo bo`ladi.

Katakka formulalarni kiritishning ikkita usuli mavjud:

1. *Formulani klaviatura orqali kiritish*: “=” belgisini qo`yib, keyin formulalar kiritiladi. Kiritish paytida belgilar formulalar qatorida hamda faollashgan katakda paydo bo`ladi. Formulalarni kiritishda odatdagi tahrirlash tugmalaridan foydalanish mumkin. Masalan: =Sin(A1)+Cos(B2) teriladi va Enter tugmasi bosiladi. Agar A1 va B1 kataklardagi sonlarni bir biriga qo`shib, natijani C1 katakka yozish kerak bo`lsa, u holda C1 katakdagi formulamiz quyidagicha bo`ladi: =A1+B1

2. *Kataklar manzilini ko`rsatish yo`li bilan formulalar kiritish*: Bu usulda ham formulalar klaviaturadan kiritish orqali, lekin kamroq foydalangan holda amalga oshiriladi. Ushbu usulda kataklar manzilini kiritish o`rniga ular ko`rsatiladi, xolos. Masalan, A3 katakga =A1+A2 formulasini kiritish uchun quyidagilarni bajarish kerak.

- o jadval kursori A3 katakka o`tkaziladi;
- o “=” belgisi kiritiladi. Formulalar qatori yonida “Ввод” yozuvi paydo bo`ladi;
- o sichqoncha ko`rsatkichi A1 katakra olib boriladi va chap tugmachasi bosiladi. Natijada katak ajratib ko`rsatiladi, ya`ni uning atrofida harakatlanuvchi ramka (rom) paydo bo`ladi. A3 katakagi formulalar qatorida A1 katak manzili ko`rinadi. Holat qatorida esa “Укажите” yozuvi paydo bo`ladi:

- o “+” belgisi kiritiladi. Natijada harakatlanuvchi rom yo`qolib, yana “Vvod” so`zi chiqadi;

- o sichqoncha ko`rsatkichi A2 katakka o`tkaziladi va tugmachasi bosiladi. Formulaga A2 katak qo`shiladi;

- o Enter tugmasini bosish bilan formulani kiritish yakunlanadi.

Katak manzilini ko`rsatish usuli klaviatura yordamida kiritish usulidan oson va tez bajariladi.

Formula yozishga doir misollar:

- ✓ C7 katagi B1 katagida yozilgan qiymatga teng bo`lsa, uni ikkinchi marta kiritib o`tirmasdan C7 katagiga kursorni o`rnatib “=B1” deb kiritsak, Excel C7 katagi B1 katagi qiymatiga tengligini tushunadi va B1 katagidagi yozuvni takrorlaydi.

- ✓ C10 katagida B10 katagining 25 foizini yozish kerak. C10 katagiga kursorni o`rnatib “=B10\*25/100” ni kiritsak, u B10 ning 25 foizini C10 katagiga yozadi. Keyinchalik B10 ning qiymatini o`zgartirganda, avtomatik C10 ning qiymati ham mos ravishda o`zgaradi. Shu tartibda ixtiyoriy matematik amallarni bajarish mumkin.

- ✓ Yuqoridagi misolda C10 katagida B10 katagining 25 foizini yozdik deylik. endi, C11 ga B11 ning, C12 ga B12 ning to C20 ga B20 ning 25 foizini yozish kerak bo`lsa, yuqorida bir marta birinchisi uchun “=B10\*25/100” deb yozish etarli. Keyingilarini shu tartibda davom ettirish uchun C10 katagining tagidagi + belgichasini sichqoncha orqali ushlab C20 gacha tortish etarli.

Formulalarni boshqa ishchi jadvallar kataklariga ham yuborish mumkin, boshqacha aytganda, formulalar bir necha joyda takrorlanishi mumkin. U hattoki, boshqa ishchi kitobdagi ishchi jadvallarda ham. Buning uchun Excel da maxsus yozuv ishlatiladi.

*Kataklardagi ma`lumotlarni boshqa ishchi jadvallarga yuborish (o`tkazish)*. Joriy ishchi kitobdagi ma`lumotlarni boshqa ishchi kitobdagi katakka yuborish quyidagi usullardan foydalanib hal qilinadi:

- ✓ *Joy nomi. Katak manzili*. Boshqacha qilib aytganda, katak manzili oldiga joyning nomi undov belgisi bilan qo`yiladi. Masalan, “=A1\*Лист1!A2”. Bu formulada joriy ishchi jadvaldagi A1 katak qiymati A2 katak qiymatiga ko`paytiriladi va “List2” ishchi sahifasida joylashadi. Agar jo`natishda ishchi jadvalning nomi bir yoki bir nechta bo`shliqni o`z ichiga olsa, jadvalning nomi bittali qo`shirnoq ichiga olinib ko`rsatiladi.

Masalan, =A1 ‘Barcha bo`limlar’!A2.

- ✓ *Boshqa ishchi kitob kataklariga ma`lumotlarni o`tkazish*. Boshqa ishchi kitob kataklariga ma`lumotlarni o`tkazish uchun quyidagi bichimlardan foydalaniladi:

=[Ishchi kitob nomi] Varaq nomi! Katak manzili

Katak manzili oldiga ishchi kitob nomi yozilib, kvadrat qavslarga olinadi va ishchi jadval nomi undov belgisi yordamida ko`rsatiladi. Masalan, =[Byudjet.xlsx] Лист 1!A1

Agar ishchi kitob nomida bir yoki bir nechta bo`shliq bo`lsa, u holda uning nomi bittali qo`shtrinoq ichiga olinishi kerak. Masalan, =A1\*'[Byudjet na 1999] Лист1!'A1

### **Funktshiyalar bilan ishlash.**

Funksiya - bu formulalarda qo`llaniladigan, kiritib qo`yilgan tayyor uskunalar qolipidir. Ular murakkab bo`lgan matematik va mantiqiy amallarni bajaradi.

Funksiyalar quyidagi ishlarni bajarish imkonini beradi.

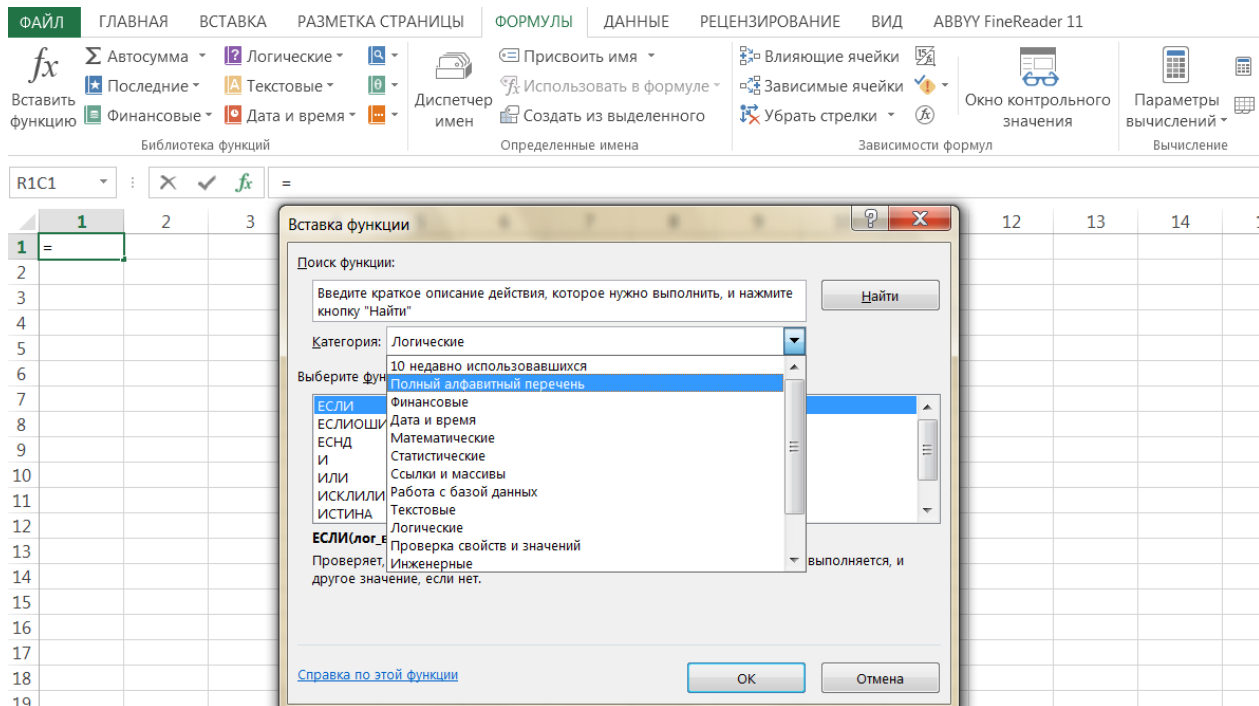
1. Formulalarni qisqartirish.
2. Formulalar bo`yicha boshqa qilib bo`lmaydigan hisob ishlarini bajarish.
3. Ayrim muharrirlik masalalarini hal qilishni tezlashtirish.

Barcha funksiyalarda oddiy () qavslar ishlatiladi. Qavs ichidagi ma`lumotlar argumentlar deb ataladi. Funksiyalar qanday argumentlar ishlatilayotganligiga ko`ra bir-biridan farq qiladi. Funksiyaning turlariga qarab ular quyidagicha ishlatilishi mumkin:

- argumentsiz;
- bir argumentli;
- qayd qilingan cheklangan argumentlar soni bilan;
- noma`lum sondagi argumentlar soni bilan;
- shart bo`lmagan argumentlar bilan.

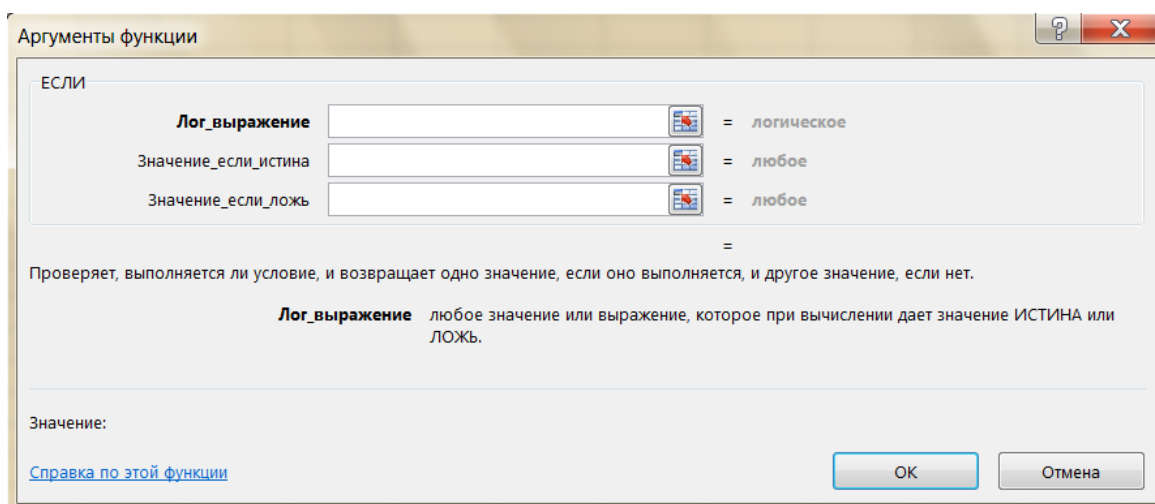
Klaviaturadan qo`lda funksiya nomi va argumentlar ro`yxati kiritiladi. Excel funksiya nomidagi belgilarni yuqori registrga o`zgartiradi, chunki formula va funksiyalarda katta harflar ishlatiladi. Agar dastur kiritilgan matnni yuqori registrga o`zgartirmagan bo`lsa, demak, u yozuvni funksiya deb qabul qilmagan, ya`ni funksiya noto`g`ri kiritilgan bo`ladi.

Excel dagi funksiya ustasi funksiya va uning argumentini yarim avtomatik tartibda kiritishga imkon yaratadi. Funksiyalar “Формулы” lentasining “Библиотека функций” guruhidan tanlanadi. Agar “Вставить функции” tugma bosilsa Excel dasturining barcha funksiyalari ro`yxati oynasi ochiladi.



Funksiyalar ustasini qo`llash funksiyaning yozilishi va uning hamma argumentlarini sintaktik to`g`ri tartibda kiritilishini ta`minlaydi. Bu oyna ikkita muloqot shaklidagi sohaga ega: “Категория” va “Выберите функцию”. Eng avvalo “Категория” sohasidan funksiy guruhi tanlanadi. “Выберите функцию” sohada shu guruhga tegishli funksiyalar guruhi chiqadi. Guruhdagi funksiyalardan biri tanlab olinsa, navbatdagi argumentlarni ko`rsatish muloqot oynasi ochiladi.





Har bir funktsiya uchun mos oyna ochiladi. Argumentlar ko'rsatilgach OK tugma bosilsa, funktsiya belgilangan katakka yoziladi.

Endi funktsiyalar ustasi bilan ishlash haqida tavsiyanomalarni beramiz:

1. Agar tanlangan funktsiya haqida qo'shimcha ma'lumot olmoqchi bo'lsangiz, unda sichqoncha ko'rsatkichini "Справка по этой функции" tugmasiga olib borib bosing.
2. Yangi funktsiyani kiritishda funktsiyalar ustasi avtomatik ravishda qator boshiga "=" (teng) belgisini qo'yadi.
3. Funktsiyalar ustasini chaqirishda katak bo'sh bo'lmasa, unda katakdagi ma'lumotlar o'chirib tashlanadi.

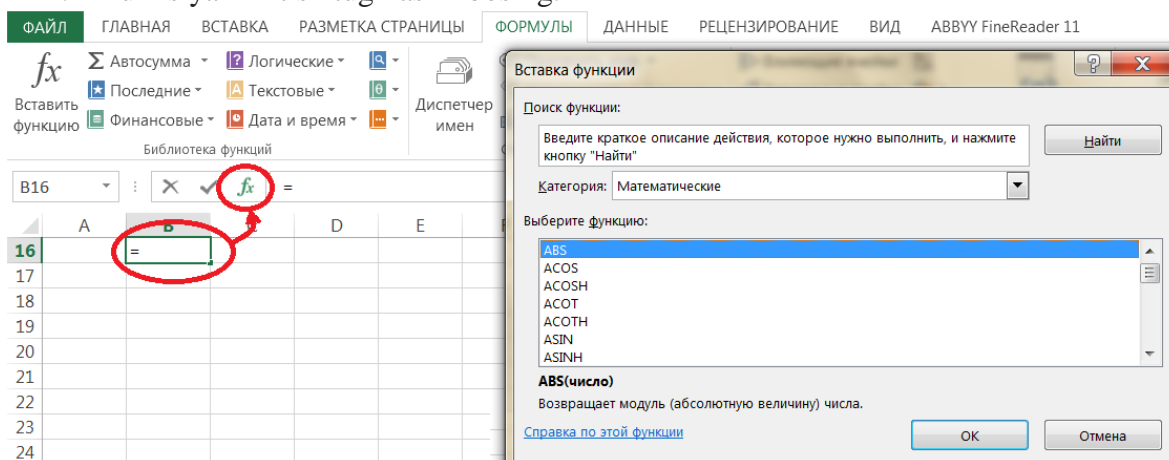
4. Funktsiyalar ustasini mavjud bo'lgan formulaga yangi funktsiyani kiritishda qo'llash mumkin. Buning uchun formulani tahrirlashda funktsiya kiritilishi kerak bo'lgan joyga kursorni qo'yish, keyin esa bu kiritishni amalga oshirish uchun funktsiyalar ustasini ni ishga tushirish kerak.

## 2. Trigonometric, statistic, arifmetik, mantiqiy funktsiyalar bilan ishlash

**Функция кiritish.** Har bir funktsiya bir xil tuzilishga ega. Misol uchun, SUM(A1:A4). Bu funktsiyaning nomi SUM (yig'indi). Qavs ichidagi qism (argument) shuni anglatadiki, biz Excelga qiymat kiritiladigan A1:A4 sohani beramiz. Bu funktsiya A1, A2, A3 va A4 kataklardagi qiymatlarni qo'shadi. Har bir ish uchun qaysi funktsiya va qaysi argumentlarni ishlatishni eslab qolish oson emas. Hayriyatki, Exceldagi Formula kiritish funktsiyasi bu borada sizga yordam beradi.

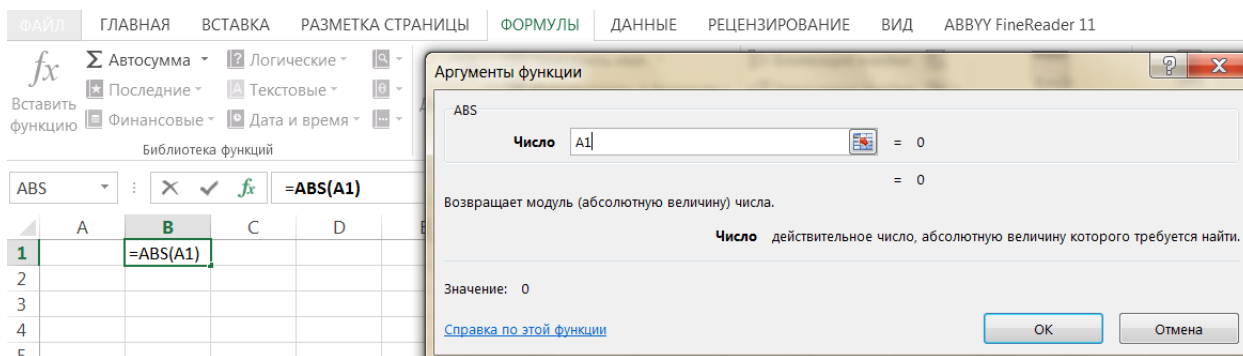
Функция kiritish uchun quyidagi qadamlarni bajaring.

1. Bir katakni tanlang.
2. Функция Kiritish tugmasini bosing.

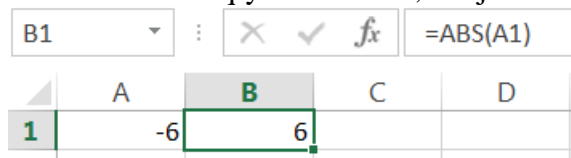


3. Ochilgan "Вставка функции" sohadan kerakli funktsiyalar tanlanadi va "OK" tugmani bosing.
4. "Аргументы функции" sohadan funktsiya argumentini kiriting va "OK" tugmani bosing.

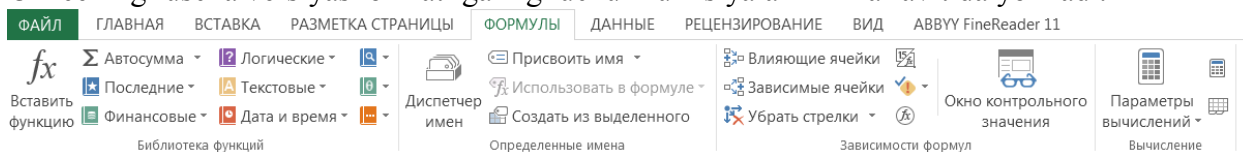




5. A1 katakka qiymat kiritilsa, natija B1 katakka chiqadi.



**Excel funksiyalar kutubxonasi.** “Формулы” lentasining “Библиотека функций” guruhidan “Вставить функции” tugma bosilsa Excel dasturining barcha funksiyalari ro`yxati oynasi ochiladi. “Категория” sohasida 13 ta turli xil yo`nalishlarga tegishli bo`lgan funksiyalar kategoriyalari berilgan. Agar foydalanuvchining maxsus funksiyalari ham qo`llanilsa, bu kategoriyalar soni undan ham ko`p bo`lishi mumkin. Oynaning “Выберите функцию” sohasida shu guruhga tegishli funksiyalar guruhi chiqadi. Excel dasturida 800 dan ortiq funksiyalar mavjud. Biz ushbu ro`yxatdagi ba`zi bir, ko`p ishlatiladigan funksiyalarni tavsiflaymiz. Kompyuterga MS Officening ruscha versiyasi o`rnatilganligi uchun funksiyalar kirill alfavitida yoziladi.



### Математик funksiyalar:

- RRODUST (<argumentlar ro`yxati> (PROIZVED) — argument qiymatlarining ko`paytmasini hisoblaydi;
- SQRT (son) (ildiz) — sonning kvadrat ildizini hisoblaydi;
- FAST (son) (FAKTOR) — argument sifatida berilgan butun songacha bo`lgan natural sonlar ko`paytmasini hisoblaydi;
- RAND (tisodifiy son) — 0 va 1 oraliq`idagi tasodifiy sonni hisoblaydi;
- AVS (son) — argument qiymatining modulini hisoblaydi;
- LN (son) — sonning natural logarifmini aniqlaydi;
- EXP (son) — sonning eksponentasini hisoblaydi;
- SIN (son) — sonning sinusini hisoblaydi;
- COS (son) — sonning kosinusini hisoblaydi;
- TAN (son) — sonning tangensini hisoblaydi (radianda);
- СУММ (<argumentlar ro`yxati>) — barcha argumentlar qiymatining yig`indisini hisoblaydi.
- ЗНАК(katak) - ko`rsatilgan katak ichidagi ma`lumotlar ishorasini aniqlash.
- СТЕПЕНЬ(son, daraja) - ko`rsatilgan sonni kerakli darajaga ko`tarish.

### Statistik funksiyalar:

- AVERAGE (<argumentlar ro`yxati>) — barcha argumentlar qiymatining o`rta arifmetigini hisoblaydi;
- МАКС (<argumentlar ro`yxati>) — argumentlar ro`yxatidan eng kattasi (maksimal son)ni topadi;
- МИН (<argumentlar ro`yxati>) — argumentlar ro`yxatidan eng kichigi (minimal son)ni topadi;
- ДИСП (<argumentlar ro`yxati>) — barcha argumentlar uchun dispersiyasini hisoblaydi.

- ДОВЕРИТ (a;b;n)

*a* - ishonchlilik darajasi uchun tanlab olingan qiymat. Masalan, *a* 0 ga teng bo'lsa, ishonchlilik 100% ni tashkil qiladi, agar *a* 0,05 bo'lsa, ishonchlilik darajasi 95% ni tashkil qiladi.

*b* — tanlab olingan tajriba natija to'plami uchun o'rtacha farqlanish bo'lib, oldindan ma'lum deb faraz qilinadi.

*p* — tanlanmadagi elementlar soni.

- КВАДРОТКЛ (<argumentlar ro'yxati>)— barcha argumentlar uchun kvadrat farqlanishni aniqlaydi.

- СРЗНАЧ(kataklar) - kataklardagi sonlarni o'rtachasini topish.

- СЧЁТЗ(kataklar) - bo'sh bo'lmagan kataklar sonini aniqlash.

- СЧЁТЕСЛИ(kataklar, shart) - shartga javob beruvchi kataklar sonini aniqlash.

### Mantiqiy funksiyalar:

Ayrim amaliy masalalarni echishda hisoblashlar u yoki bu shartlarga bog'liq bo'lishi mumkin. Bunday holatda “ЕСЛИ” shartli funksiyasidan foydalanish mumkin. Bu funksiyaning formati quyidagicha:

- ЕСЛИ (shart, to'g'ri, noto'g'ri) - shart bo'yicha amalni bajarish.<sup>13</sup>

Rus            Ingliz

И            AND            Выдаёт значение ИСТИНА, если все аргументы имеют значение ИСТИНА.

ЛОЖЬ        FALSE        Вставляет логическое значение ЛОЖЬ.

ЕСЛИ        IF            Выполняет проверку условия.

НЕ            NOT            Меняет на противоположное логическое значение своего аргумента.

ИЛИ        OR            Выдаёт значение ИСТИНА, если хотя бы один аргумент имеет значение ИСТИНА.

ИСТИНА    TRUE        Вставляет логическое значение ИСТИНА.

Excelning shart (IF, ЕСЛИ), “va” (AND, И) hamda “yoki” (OR, ИЛИ) kabi mantiqiy funksiyalarini qanday ishlatishni o'rganing.

**Shart (If, Если) funksiyasi.** ЕСЛИ (IF) funksiyasi biror shart bajarilishini tekshiradi, va agar rost (TRUE, ИСТИНА) bo'lsa bir qiymatni va agar yolg'on (FALSE, ЛОЖЬ) bo'lsa boshqa bir qiymatni qaytaradi.

C1 katakni tanlang, va quyidagi funksiyani kiriting.

=IF(A1>10,"To'g'ri","Noto'g'ri") yoki

=если(A1>10,"To'g'ri","Noto'g'ri")

C1		fx =ЕСЛИ(A1>10,"To'g'ri","Noto'g'ri")								
	A	B	C	D	E	F	G	H	I	
1	12	3	To'g'ri							
2										
3										
4										
5										
6										
7										

ЕСЛИ (IF) funksiyasi “To'g'ri” qaytaradi chunki A1 katakdagi qiymat 10 dan kattaroq.

**“VA” (And, И) Funksiyasi.** И (AND) funksiyasi agar barcha shartlar rost bo'lsa, rost (TRUE, ИСТИНА) qiymat qaytaradi, va agar shartlardan ixtiyoriy bittasi yolg'on bo'lsa, yolg'on (FALSE, ЛОЖЬ) qaytaradi.

D1 katakni tanlang va quyidagi formulani kiriting.

=IF(AND(A1>10,B1>5),"To'g'ri","Noto'g'ri") yoki

<sup>13</sup> Минько А.А. Принятие решений с помощью Excel. Просто так дважды два. – М.: Эксмо, 2009.

=ЕСЛИ(И(A1>10,B1>5),"То'g'ri","Noto'g'ri")

	A	B	C	D	E	F	G	H	I
1	12	3	To'g'ri	Noto'g'ri					
2									
3									
4									

И (AND) funksiyasi yolg'on (FALSE, ЛОЖЬ) qaytaradi chunki B1 katakdagi qiymat 5 dan kata emas. Natijada ЕСЛИ (IF) funksiyasi "Noto'g'ri" qaytaradi.

**"Yoki" (Or, Или) Funksiyasi.** ИЛИ (OR) funksiyasi agar shartlardan ixtiyoriy bittasi rost (TRUE, ИСТИНА) bo'lsa, rost (TRUE, ИСТИНА) qaytaradi va barcha shartlar yolg'on bo'lsa, yolg'on (FALSE, ЛОЖЬ) qaytaradi.

E1 katakni tanlang va quyidagi formulani kiriting.

=IF(OR(A1>10,B1>5),"То'g'ri","Noto'g'ri") yoki

= ЕСЛИ(ИЛИ(A1>10,B1>5),"То'g'ri","Noto'g'ri")

E1      fx      =ЕСЛИ(ИЛИ(A1>10,B1>5),"То'g'ri","Noto'g'ri")

	A	B	C	D	E	F	G	H
1	12	3	To'g'ri	Noto'g'ri	To'g'ri			
2								
3								
4								
5								

ИЛИ (OR) funksiyasi rost (TRUE, ИСТИНА) qaytaradi, chunki A1 katakdagi qiymat 10 dan kattaroq. Natijada ЕСЛИ (IF) funksiyasi "To'g'ri" qaytaradi.

Umumiy izoh: И (AND) va ИЛИ (OR) funksiyalari 255 tagacha shartlarni tekshira oladi.

**Funksiyalarni qo'llashga doir tavsiyalar.** Funksiyalarda bir nechta kataklar ko'rsatilganda, ular o'rtasidagi ":" belgi "dan gacha" ma'noni bildiradi. Shu bilan birga, ";" belgisi esa "bilan" ma'noni bildiradi. Masalan:

=СУММ(A1:F11) - bu A1 dan F11 gacha to'rtburchak sohadadagi ma'lumotlar yig'indisini topish ma'nosini bildiradi.

= СУММ(A11;C15) bu A11 va C15 kataklardagi sonlar yig'indisini topish ma'nosini bildiradi.

Kataklarni sichqoncha yordamida ko'rganimizda Shift tugma bilan tanlanganlar "dan gacha", Ctrl bilan esa "va" ma'nosini bildiradi. Formulalarda qo'shtirnoq ' ' ichida yozilgan mantlar o'zgartirilmasdan ekranga chiqariladi. Masalan:

= ЕСЛИ(A1=5, 'A'lo', 'A'lo emas' ) - bu agar A1 katakdagi son 5 ga teng bo'lsa u holda formula katagida 'a'lo' so'zi yoziladi, aks holda esa 'a'lo emas' so'zi yozilish ma'nosini bildiradi.

Formula bor katakni avtoro'xat bilan to'ldirganimizda ichidagi formula o'zgarish qatori yoki ustunga qarab o'zgaradi. Masalan: Agar = ЕСЛИ(A1=5, 'A'lo', 'A'lo emas' ) formulasi bor katakni pastki avtoro'xat bilan yo'li bilan to'ldirsak, u holda pastki katakda bu formula = ЕСЛИ(A2=5, 'A'lo', 'A'lo emas' ) ga o'zgaradi. Agar uni o'ng tomonga to'ldirsak u holda o'ng tomondagi katak formulasi = ЕСЛИ(B1=5, 'A'lo', 'A'lo emas' ) ga o'zgaradi.

Agar sizga avtoto'ldirish yoki nusxa olish natijasida hosil bo'lgan formulada biror bir katak nomi o'zgarmas bo'lishi kerak bo'lsa, u holda uning nomi oldida \$ belgisini qo'yishingiz kerak. Masalan: = ЕСЛИ(\$A\$2=5, 'A'lo', 'A'lo emas' ) formulada A2 katak nomi o'zgarmas. Agar \$A\$2 o'rniga \$A2 u holda faqat satr nomeri o'zgaruvchan, agar esa \$A\$2 o'rniga A\$2 bo'lsa u holda faqat ustun nomi o'zgaruvchan bo'ladi.

Funksiyada argumentlar ishlatilmasa ham, bo'sh qavslar ko'rsatilishi lozim. Masalan, =RAND(). Agar funksiyada bittadan ortiq argument ishlatilsa, ular orasiga nuqtali vergul (;) qo'yiladi. Funksiyani kiritishning ikkita usuli mavjud: klaviatura yordamida qo'lda kiritish va Excel da funksiyalar ustasi orqali kiritish.

**Xatolar haqida xabarlar.** Agar katakdagi formula to'g'ri hisoblanmasa, Excel katakda xato haqida xabar chiqaradi, ya'ni katakda # belgisidan boshlangan yozuv paydo bo'ladi. Bu xatolik kodi. Agar xato kodi paydo bo'lgan katakga sichqoncha bilan bossak, u holda formulalar qatorida xatolik paydo bo'lganligi sababini ko'rsatuvchi fragment so'z hosil bo'ladi. Quyidagi jadvalda ba'zi hatoliklar kodlari va ularning kelib chiqish sabablari keltirilgan:

- ##### –katakning eni sonni berilgan formatda aks ettirishga imkon bermaydi;
- #ИМЯ? –Excel formulada ishlatilgan nomni aniqlay olmadi;
- #ДЕЛ/0! – formulada nolga bo'lishga harakat qilinmoqda;
- #ЧИСЛО! – matematikada qabul qilingan operatorlarni ko'rsatish qoidalari buzilgan;
- #Н/Д – argument sifatida bo'sh katakka tayanish ko'rsatilgan bo'lsa, shunday xabar chiqishi mumkin;
- #ПУСТО! – umumiy kataklarga ega bo'lgan ikki soha kesishishi noto'g'ri ko'rsatilgan bo'lsa;
- #ССЫЛКА! –formulada mavjud bo'lmagan tayanish berilgan;
- #ЗНАЧ! – qabul qilib bo'lmaydigan argument turidan foydalanilgan.

**Formulani aynan o'zini ko'chirish.** Siz biror formulani boshqa katakka ko'chirib o'tganingizda, Excel katak bog'lanishlarini avtomatik tarzda formula ko'chirib o'tkazilgan yangi katakka to'g'rilab chiqadi.

Misol uchun, quyidagi A3 katakda A1 katakning qiymatini A2 katakning qiymatiga qo'shadigan formula bor.

	A	B	C	D	E	F	G	H	I
1	4								
2	5								
3	9								
4									

Siz bu formulani B3 katakka ko'chirib o'tkaningizda (A3 katakni belgilang, "CTRL + c"ni bosib, B3 katakni belgilang, "CTRL + v"ni bosib), formula avtomatik tarzda B ustundagi qiymatlarga ishora qiladi (bog'lanadi).

	A	B	C	D	E	F	G	H	I
1	4								
2	5								
3	9	=B1+B2							
4									

Izoh: B3 katakdagi formula B1+B2 bo'ldi.

Agar siz buni xohlamasangiz, balki formulaning aynan o'zini ko'chirib o'tishni xohlasangiz (katak bog'lanishlarini o'zgartirishsiz), quyidagilarni amalga oshiring.

1. Formulalar qatoriga bosib va formulani belgilang.
2. "CTRL + c" ni bosib va "Enter"ni bosib.

	A	B	C	D	E	F	G	H	I
1	4								
2	5								
3	=A1+A2								
4									

3. B3 katakni tanlang va yana formulalar qatoriga bosing.
4. “CTRL + v”ni bosing va “Enter”ni bosing.

Natija:

	A	B	C	D	E	F	G	H	I
1	4								
2	5								
3	9	9							
4									

Endi A3 va B3 kataklarda aynan bir xil formula joylashgan (A1+A2).

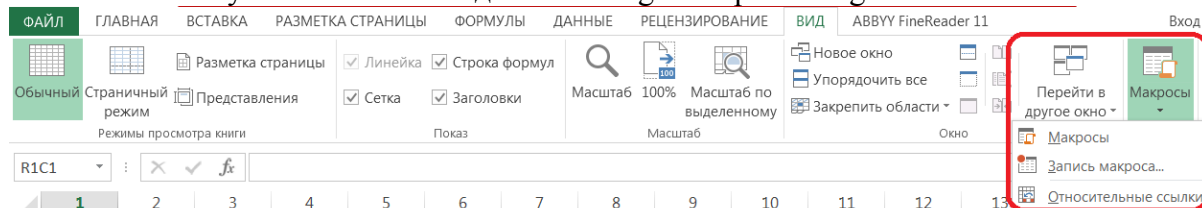
### 3. VBA (Visual Basic For Applications) imkoniyatlaridan foydalanish

Ish jarayoni davomida bir xil amal va hisoblashlarni Excelda qayta-qayta bajarishimizga to`g`ri keladi. Bu ba`zan ko`p vaqtimizni olsa, ba`zan medamizga tegib ketadi. Shuning uchun, ba`zi bir xillik talab qilinadigan ishlarni makroslarni yozish orqali Excel zimmasiga yuklashimiz mumkin yoki Excelda mavjud bo`lmagan imkoniyatlarni makroslarni qo`llash orqali hal qilish mumkin.

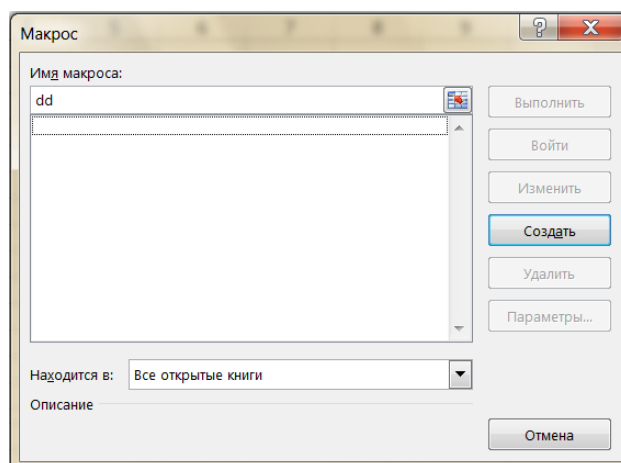
Makros - bu Excel dasturida ishlatish uchun Visual Basic for Applications (VBA) dasturlash tilida yozilgan dastur matni.

Dasturlash tillari (C++, C#, Java, ASP, PHP...) juda ko`p, ammo VBA tili bevosita Microsoft Office paketi uchun standart til bo`lib hisoblanadi. U Excel, Word, Outlook yoki Access dasturlari bilan do`stona hamkorlikda ishlash qobiliyatiga ega yoki Microsoft Office paketi aynan VBA tili bilan ishlashga moslashtirilgan.

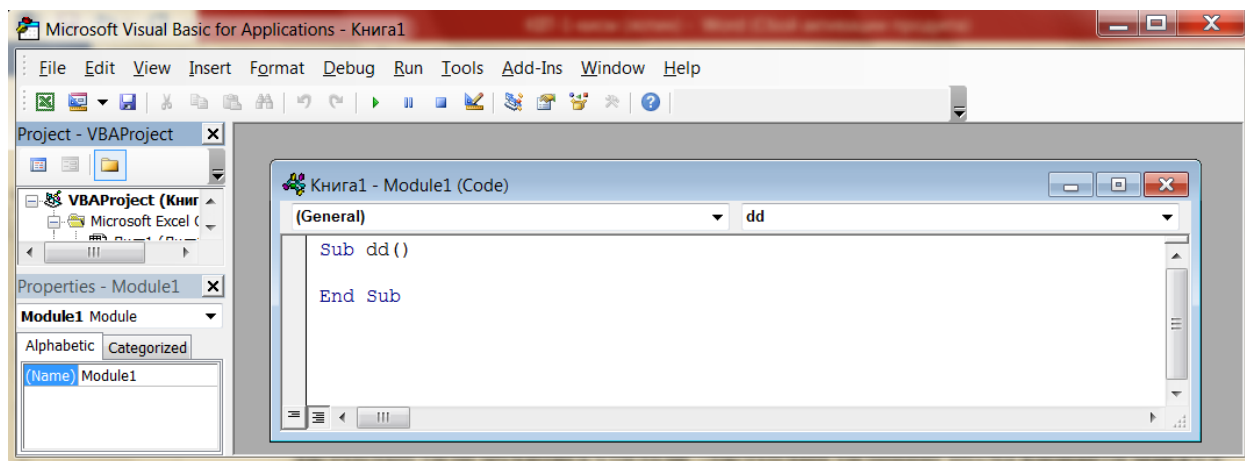
Makroslarni yaratish uchun “Вид” lentasining “Макросы” tugmasi ishlatiladi.



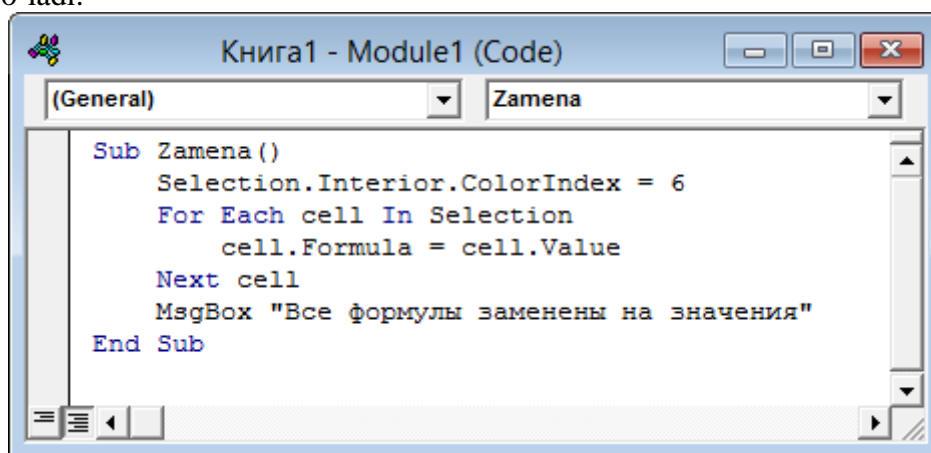
“Макросы” tugmasi bosilsa, makroslar bilan ishlash soha ochiladi. Bu sohadan “Макросы” buyrug`ini ishga tushirsak, quyidagi oyna ochiladi.



Ochilgan oynadan “Создать” tugma bosilsa, Visual Basic for Applications (VBA) dasturlash tili muhitiga o`tiladi. Dasturlash tilining ruscha varianti mavjud emas, shuning uchun muhit interfeysi ingliz tilida beriladi.



Makros yozish uchun VBA dasturlash tilini yaxshi bilish lozim. Agar siz VBA dasturlash tilini bilsangiz dasturni yarata olishingiz mumkin. Tayyorlangan dastur matni “Книга1-Module1” oyna ishchi sohasiga kiritiladi. Eng oddiy dastur “Книга1-Module1” oyna ishchi sohasida quyidagicha bo`ladi.



Dastur kompilyatsiya qilinadi va xatolar bo`lmasa, u Excel dasturida saqlanadi. Bu dastur siz ko`zlagan ishni bajarib beradi.

Misol. Tanlangan katakdagi kasr sonlar “=okrugl(son;yaxl-xona)” funktsiya orqali berilgan xona bo`yicha yaxlitlansin. Bu ishni amalga oshiruvchi quyidagi funktsiya dasturini VBA dasturlash tilida tuzamiz.

```

Function ZVI_Round(V As Double, Optional DecPlaces As Integer = 0) As Double
    If DecPlaces < 0 Then
        ZVI_Round = Round(V / 10 ^ -DecPlaces + V * 2E-16, 0) * 10 ^ -DecPlaces
    Else
        ZVI_Round = Round(V + V * 2E-16, DecPlaces)
    End If
    If Abs(ZVI_Round) = 0 Then
        ZVI_Round = 0
        Exclude -0
    return value
End Function

```

### **Nazorat savollari:**

1. Formulalar nima va ular qayerda yoziladi?
2. Formulalarda ishlatiladigan belgilarni aytib bering.
3. Formula qanday belgi bilan boshlanadi?
4. “:”, “;” belgilari qachon ishlatiladi?
5. Formulalar Excel da qanday hisoblanadi?
6. Funksiya deganda nimani tushinasiz?,
7. Funksiyalar Excel ga qanday kiritiladi?
8. Funksiyalar Excel da qanday hisoblanadi?
9. Funksiyalar kategoriyalari qanday tanlanadi?
10. Excel dasturining funktsiyalari haqida nima bilasiz ?
11. Excel dasturining funktsiyalar kutubxonasi haqida nima bilasiz?
12. Funktsiyalar bilan ishlash uchun qanday tugmadan foydalaniladi?
13. Funktsiyalar bir necha qiymatlar uchun qanday ishlatish mumkin?
14. =SUMM(A1:F11) ifodani izohlang.
15. Makros deganda nimani tushunasiz?
16. Makroslar qaysi dasturlash tilida yoziladi?

#### 4 mavzu. MS Excel elektron jadvali yordamida misollar echish

##### Reja:

1. MS Excelda Solver imkoniyatlarini o'rganish.
2. Kimyo texnologiyasi masalalarini yechishda MS Excelda Solver imkoniyatlarini o'rganish.

**Tayanch iboralar:** optimal, optimallashtirish, Solver, yechimlarni qidirish, maqsad funksiyasi, kommiyoyajer masalasi

##### 1. MS Excelda Solver imkoniyatlarini o'rganish

Ishlab chiqarish ish jarayoni faoliyati natijalarini nazorat qilish va hisobga olish bilan bog'liq ko'plab turli masalalarni echishda elektron jadvallardan keng foydalaniladi. Mutaxassis o'z ish faoliyati davomida ishlab chiqarish jarayonidagi juda ko'plab tahliliy masalalarni hal qilishga to'g'ri keladi va ular tahlili natijasida bir qancha yechimlar vujudga kelishi mumkin. Bu yechimlar orasidan eng yaxshisi va eng samaralisini tanlab olish maqsadga muvofiq hisoblanadi. Odatda, bunday yechim optimal deb ataladi. "Optimal" - lotincha "optimus" so'zidan olingan bo'lib, eng yaxshi (mavjud imkoniyatlar doirasida undan yaxshisi yo'q), eng ma'qul, eng samarali kabi ma'nolarni anglatadi. Optimal yechimni izlash masalasi keng ko'lamli bo'lib, matematikada avvalo ekstremal (maksimal yoki minimal) masalalar deb ataluvchi sohani qamrab oladi. Bunday masalalarning optimal yechimlarini aniqlashda Excel elektron jadvali qo'llanilib, undagi Solver – yechimlarni qidirish dasturlari paketidan foydalaniladi. Bu paket amallarni tadqiq qilish nazariyasining asosiy yo'nalishlaridan bo'lgan matematik dasturlash usullari va algoritmlari asosida yaratilgan.<sup>14</sup>

Tizimni elektron jadvallar yordamida tadqiq qilish jarayonini kundalik amaliy faoliyatda jadvallar orqali bajariladigan hisob-kitoblarning tabiiy davomi deb qarash mumkin. Bu faoliyatga "Nima bo'ladi, agar .....,?", "Bu qanday o'zgarishlarga olib keladi?" degan nuqtai nazardan yondoshish mumkin. Bu savollarning javoblari tabiiy resurslardan foydalaninshning optimal taktikasi va strategiyasi, "optimal nuqta"ga erishish va shu kabi masalalarning yechimlariga bog'liq holda topiladi. Javoblarni topish ixtiyoriy tizim tahlilining asosiy maqsadini tashkil etadi. Demak, elektron jadvallarda amalga oshiriladigan oddiy hisob kitoblar bilan bunday jadvallarda ishlab chiqarish jarayonining optimal yechimlarini qidirish tabiiy bog'langandir.

Excel dasturida ma'lumotlarni tahlil qilishning kuchli vositasi – Solver (yechimlarni qidirish) hisoblanadi.

Yechimlarni qidirish (Excel Solver – Poisk resheniy) dasturi Excel elektron jadvali dasturining qo'shimcha vositasi bo'lib, undan 1991 yildan buyon chiziqli tenglamalar tizimini optimallashtirishning chiziqli va chiziqli bo'lmagan masalalarini echishda foydalaniladi.

Dasturning bazaviy versiyasida echiladigan masala o'lchamlariga quyidagi cheklovlar qo'yilgan:

- o Noma'lumlar soni 200 tagacha;
- o noma'lumlarga qo'yilgan cheklov formulalari soni 100 tagacha;
- o noma'lumlarga qo'yilgan eng ko'p shartlarning soni 400 tagacha.

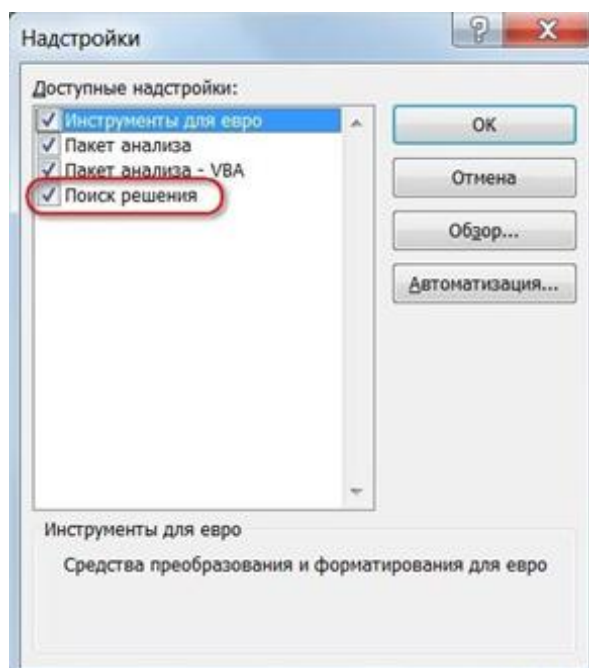
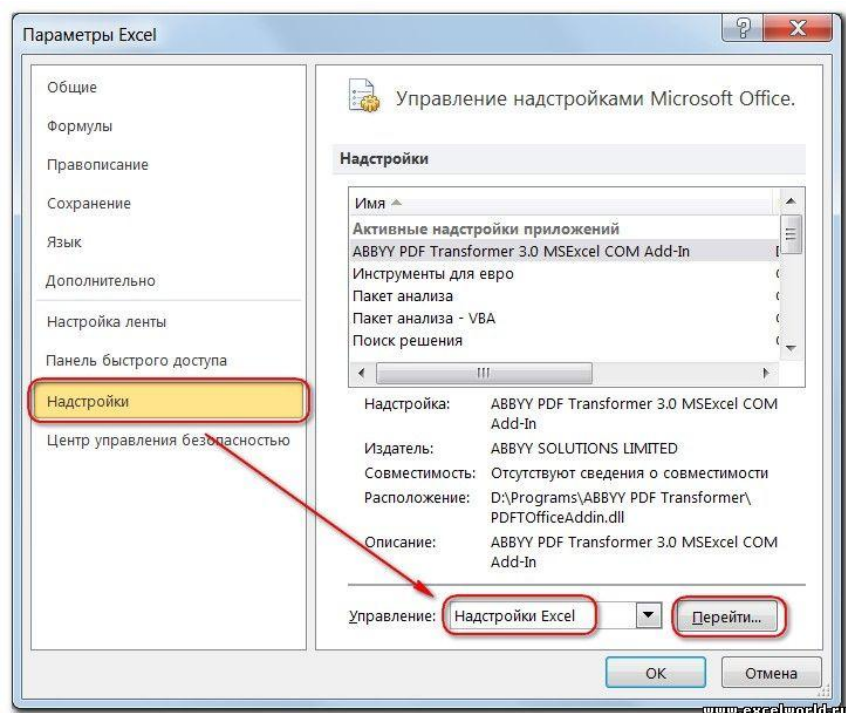
Excelga o'rnatilgan Solver dasturi optimallashtirishning kuchli va qulay usullarini yaratishga ixtisoslashgan Frontline System kompaniyasi tomonidan ishlab chiqilgan.

Odatda Excel dasturining "Поиск решения" (yechimni qidirish) vositasi faollashtirilmagan holatda bo'ladi. Uni faollashtirish uchun "Файл-Параметры" buyrug'i ishga tushirilsa, ochilgan oynaning chap qismida joylashgan "Настройка" tugma bosiladi.

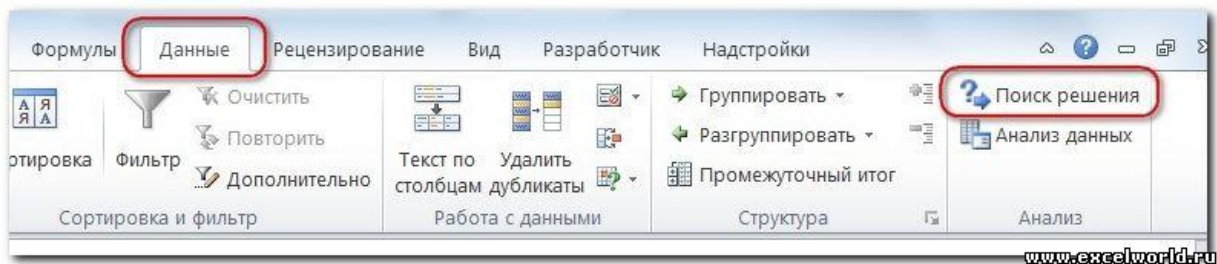
Ochilgan sohaning "Управление:" maydonidan "Надстройка Excel" tanlanib, "Перейти" tugmasi bosiladi. Ochilgan oynadan "Поиск решения" tanlangach, unga belgi qo'yiladi va ОК tugma bosiladi.

<sup>14</sup> Соловьев В.И. Методы оптимальных решений: Учебное пособие. –М.: Финансовый университет, 2012.





Natijada Excel dasturining “Данные” lentasida “Анализ” guruhi va unda “Поиск решения” buyrug’i paydo bo’ladi<sup>15</sup>.



<sup>15</sup> Чернов.В.П., Эйсснер Ю.Н., Чернов.А.В.П Модели и методы разработки управленческих решений: прогнозирование и планирование: Учебное пособие. – СПб.: Изд-во Санкт-Петербургского государственного университета экономики и финансов, 2009.

Endi bevosita yechimlarni qidirish amalini bajarishga qirishishimiz mumkin. Yechimlarni qidirish amalining bajarilish ketma-ketligi quyidagicha:

1. Katakalar orasida bog'lanishlarni o'rnatadigan formulalarni ifodalagan jadvalni hosil qiling.

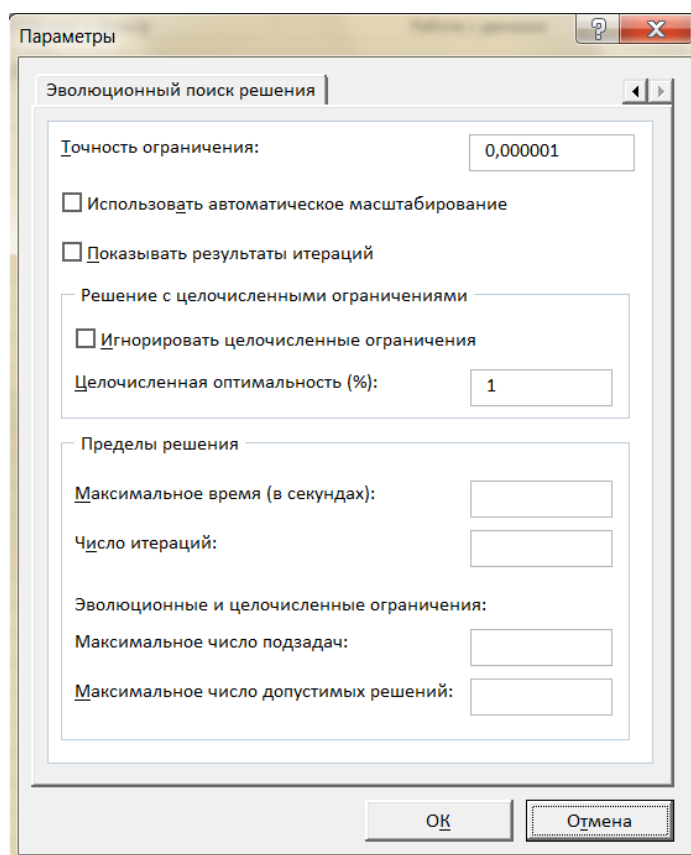
2. Kerakli qiymatni qabul qilishi lozim bo'lgan maqsad funktsiyasi joylashgan katakni ajrating va "Данные" lentaning "Анализ" guruhidan "Поиск решения" buyrug'ini bajaring. Natijada "Параметры поиска решения" nomli muloqot oynasi ochiladi. Undagi "Оптимизировать целевую функцию:" maydonida maqsad funktsiyasi joylashgan katakning manzili yoziladi.

3. Maqsad funktsiyasi qabul qilishi kerak bo'lgan holatlardan ("Максимум, Минимум, Значение") biri beriladi. Agar oxirgisi tanlansa, uning o'ng tomonidagi maydonga qaysi qiymatga teng bo'lishi kiritiladi.

4. Optimal natijani qidirishda, dastur qaysi kataklardagi qiymatlarni o'zgartirishi kerakligi "Изменяя ячейки переменных:" maydonida ko'rsatiladi.

5. Mavjud cheklovlar "В соответствии с ограничениями:" ro'xatida akslantiriladi. Buning uchun "Добавить" tugmasi bosiladi va hosil bo'lgan "Добавление ограничения" muloqot oynada cheklovlar beriladi.

6. "Выберите метод решения:" ro'xatidan yechimni ("Поиск решения нелинейных задач методом ОПГ, Поиск решения лин. задач симплекс методом, Эволюционных поиск решения") usullaridan biri tanlanadi. Agar "Эволюционных поиск решения" usuli tanlansa, "Параметры" tugmasini bosish orqali, ushbu usul kattaliklarini kiritish muloqot oynasi ochiladi va unda kerakli kattaliklar kiritiladi.



7. Yechimni kidirish jarayonini ishga tushirish “Найти решение” tugmasini bosish bilan amalga oshiriladi.

8. Yechimni qidirish jarayoni tugagandan so`ng, “Результаты поиска решения” muloqot oynasi ochiladi. Undagi “Сохранить найденное решение” yoki “Восстановить исходное значения” buyruqlaridan keraklisi tanlanadi va OK tugmasi bosiladi.

Ekkranda “Параметры поиска решения” kattaliklari haqida ma`lumot beriladi.<sup>16</sup>

Excel 2007 versiyasidan boshlab yechimni qidirish vositasini ishlatish bir xil tamoyilga asoslangan, faqat keying versiyalarda oyna ko`rinishi ozgina o`zgartirilgan.

## 2. Kimyo texnologiyasi masalalarini yechishda MS Excelda Solver imkoniyatlarini o`rganish

**Marshrutlash masalasi:** Qaror qabul qilishning matematik modeli bo`lgan optimallashtirishning eng tarqalgan masalasi – marshrutlashni o`rganish muhimdir. Kommivoyajer masalasi marshrutlashning bazaviy modeli deb qabul qilingan.

Kommivoyajer masalasi dastlab marketing sohasida qo`llanilgan. Keyinchalik boshqaruv faoliyatining boshqa sohalariga, ayniqsa hududlardagi tarqoq ob`ektlarning boshqaruviga tadbiq etila boshlandi.

Bunday amaliy qo`llanilishga misol sifatida:

Sanoatning har xil sohalaridagi tarqoq ob`ektlarga xizmat ko`rsatishda;

- ekspeditsiyalarni uchib aylanib chiqish va mos yuklarni ularga tashlashda;

- sanoat sohalaridagi texnik ob`ektlarga xizmat ko`rsatishda optimal marshrutlarni aniqlashda.

Turli hududlarda joylashgan sanoat korxonalarida marshrutlash masalasi alohida o`rin tutadi. Masalan, korxonada muhandisi xizmat faoliyati chog`ida turli hududlarda joylashgan ob`ektlarga turli zaruriy jihozlarni etkazishi, ob`ektlarga borishiga va ulardagi ish jarayonlarini kuzatishiga to`g`ri keladi. Muhandisning xizmati yo`nalishini optimallashtirish ushbu xizmat

<sup>16</sup> Амелин С.В. Математические методы в теории управления и исследование операций: Учебное пособие. – Воронеж.: Воронежский государственный технический университет, 2015.

samaradorligini oshirib, yo`nalishdagi bir ob`ektdan boshqasiga qisqa vaqt ichida o`tilishini ta`minlaydi.

**Masalaning qo`yilishi.** Muhandis tomonidan  $m$  ta ob`ekt tekshirilishi kerak. Bu ob`ektlarning hududdagi joylashishi ma`lum va  $C = /c_{ij}/, i = \overline{1, m}, j = \overline{1, m}$  matritsa bilan aniqlanadi. ( $C$  – masofalar matritsasi,  $c_{ij}$  – ob`ektdan  $j$  – ob`ektgacha bo`lgan masofa). Muhandis joylashgan punkt  $i = 1$  deb qabul qilinadi. Harakat shu punktdan boshlanadi. Muhandis eng qisqa vaqt orasida har qaysi ob`ektni bir martadan tekshirishi, so`ngra boshlang`ich punktga qaytishi talab etiladi, ya`ni optimal yo`nalishni tuzishi kerak. Ayniqsa, ob`ektlarga turli xil jihozlarni etkazganda, jihozning turi va boriladigan ob`ektlarning o`rniga muvofiq optimal marshrutlash muhim hisoblanadi.

Umumiy holda, masalaning aniq yechimi  $m$  kattalik bilan tavsiflanuvchi masala o`lchamiga hamda qo`llanilayotgan algoritm va kompyuterning samaradorligiga bog`liq.

**Matematik model.** Ob`ektlarni aylanib o`tish rejasi  $|x_{ij}|$  - matritsa orqali berilgan bo`lsin. Matritsa elementlari ikkilik sonlardan iborat, ya`ni  $x_i = 1$ , agar  $i$  - ob`ektdan  $j$  – ob`ektga o`tsa va  $x_i = 0$ , agar  $i$  - ob`ektdan  $j$  – ob`ektga bormasa.

Maqsad funksiyasining

$$F = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij}$$

Minimumini quyidagi cheklovlarda aniqlashni talab etiladi:

$$\sum_{i=1}^m x_{ij} = 1, \quad j = \overline{1, m}$$

faqat bir marta  $j$  – ob`ektga kirish;

$$\sum_{j=1}^m x_{ij} = 1, \quad i = \overline{1, m}$$

faqat bir marta  $i$  – ob`ektdan chiqish.

Bu turdagi masalalarda yana bir qo`shimcha shart qo`yiladi:

$$u_i - u_j + (m-1)x_{ij} \leq m-2, \quad i \neq j, \quad i, j = 2, \dots, m$$

Bu maxsus shart o`zaro bog`lanmagan bir qancha yo`nalish va takrorlanishlarni bartaraf etadi, ya`ni muhandisning ob`ektlarni aylanib chiqishi yopiq marshrut bo`ylab amalga oshadi.

Demak, muhandisning ob`ektlarni aylanib chiqishining shunday rejasi tuzilishi kerakki, “Umumiy masofa” = “Masofalar matritsatsi” \* “Aylanib o`tish rejasi” minimal qiymatni quyidagi cheklovlarda qabul qilishi kerak:

- o “Kirish” = 1 (muhandisning ob`ektga kirishi);
- o “Chiqish” = 1 (muhandisning ob`ektdan chiqishi);
- o o`zaro bog`lanmagan yo`nalishlar mavjudligini istisno qilish.

Excel dasturida masalani echish: beshta ob`ekt berilgan bo`lib, ular orasidagi  $c_{ij}$  masofa (km da) quyidagi jadvalda berilgan.

Ob`ekt raqami	1	2	3	4	5
1	$\infty$	11	10	6	12
2	8	$\infty$	6	7	9
3	7	5		8	4
4	3	9	4	$\infty$	5
5	4	6	7	4	$\infty$

Jadvalning diognal kataklarida cheksiz belgisi ( $\infty$ ) yozilgan. Ob`ektning o`zidan o`ziga marshrut bo`lmagani uchun bu belgi o`rniga amaldagi jadvaldagi sonlarga nisbatan katta son ko`rsatiladi.

Excel dasturining ish sahifasiga boshlang`ich ma`lumotlarni, cheklovlar va maqsad funksiyasining hisoblash formulalarini kiritamiz.

1) “O`zgaruvchilar matritsasi” va “cheklovlar” matnini mos ravishda B2 va G2 kataklarga yozamiz.

2) Qulaylik uchun birdan beshgacha bo`lgan raqamlarni B3:F3 va A4:A8 kataklar sohasiga kiritamiz.

3) O`zgaruvchilar matritsasi uchun B4:F8 kataklar sohasini ajratamiz va ularning qiymati sifatida nolni kiritamiz.

4) Cheklovlar formulalarini (ob`ektga faqat bir marta kirish) G4:G9 kataklar sohasiga yozamiz:  $\text{СУММ(B4:F4)}$ ;  $\text{=СУММ(B5:F5)}$ ;  $\text{=СУММ(B6:F6)}$ ;  $\text{=СУММ(B7:F7)}$ ;  $\text{=СУММ(B8:F8)}$ ;

5) Cheklovlar so`zini A9 katakka kiritamiz;

6) Cheklovlar formulalarini (ob`ektdan faqat bir marta chiqish) B4:F9 kataklar sohasiga yozamiz:  $\text{=СУММ(B4:B8)}$ ;  $\text{=СУММ(C4:C8)}$ ;  $\text{=СУММ(D4:D8)}$ ;  $\text{=СУММ(E4:E8)}$ ;  $\text{=СУММ(F4:F8)}$ ;

	A	B	C	D	E	F	G
1							
2		<b>Ўзгарувчилар матрицаси</b>					<b>чекловлар</b>
3		1	2	3	4	5	
4	1	0	0	0	0	0	$\text{=СУММ(B4:F4)}$
5	2	0	0	0	0	0	$\text{=СУММ(B5:F5)}$
6	3	0	0	0	0	0	$\text{=СУММ(B6:F6)}$
7	4	0	0	0	0	0	$\text{=СУММ(B7:F7)}$
8	5	0	0	0	0	0	$\text{=СУММ(B8:F8)}$
9	<b>чекловлар</b>	$\text{=СУММ(B4:B8)}$	$\text{=СУММ(C4:C8)}$	$\text{=СУММ(D4:D8)}$	$\text{=СУММ(E4:E8)}$	$\text{=СУММ(F4:F8)}$	
10	<b>В10да мақсад функцияси</b>	$\text{=СУММПРОИЗВ(B4:F8;B14:F18)}$					
11	<b>С11:F11 блокда u ўзгарувчилар</b>						

7) “B10da maqsad funktsiyasi” va “C11:F11 sohada u o`zgaruvchilar matnini A10 va A11 kataklarga kiritamiz.

8) B10 katakka – –  $\text{СУММПРОИЗВ(B4:F8;B14:F18)}$  formulani yozamiz.

9) “Masofalar matritsasi” matnini, birdan beshgacha raqamlarni mos ravishda B12, B13:F13, A14:A18 kataklarga kiritamiz.

10) Masofalar matritsasi qiymatini B14:F18 kataklar sohasiga yozamiz.

11) “Qo`shimcha u o`zgaruvchilar bo`yicha cheklovlar uchun formulalar” matnini A19 katakka yozamiz.

12) A21:A24, B20:E20 kataklar sohasiga  $u_2, u_3, u_4, u_5$  belgilarni kiritamiz.

13) A21:E24 kataklar sohasiga “o`zaro bog`lanmagan yo`nalishlar mavjudlik holatini istisno qiluvchi” cheklovlarni, ya`ni:

B21:E21 kataklarga:

$$\text{=C11-C11+4*C5}; \text{=C11-D11+4*D5}; \text{=C11-E11+4*E5}; \text{=C11-F11+4*F5}.$$

B22:E22 kataklarga:

$$\text{=D11-C11+4*C6}; \text{=D11-D11+4*D6}; \text{=D11-E11+4*E6}; \text{=D11-F11+4*F6}.$$

B23:E23 kataklarga:

$$\text{=E11-C11+4*C7}; \text{=E11-D11+4*D7}; \text{=E11-E11+4*E7}; \text{=E11-F11+4*F7}.$$

B24:E24 kataklarga:

$$\text{=F11-C11+4*C8}; \text{=F11-D11+4*D8}; \text{=F11-E11+4*E8}; \text{=F11-F11+4*F8}.$$

formulalarni yozamiz.

	A	B	C	D	E	F
12		<b>Масофалар матрицаси</b>				
13		1	2	3	4	5
14	1	9999	11	10	6	12
15	2	8	9999	6	7	9
16	3	7	5	9999	8	4
17	4	3	9	4	9999	5
18	5	4	6	7	4	9999
19	<b>Қўшимча u ўзгарувчилар бўйича чекловлар учун формулалар</b>					
20		$u_2$	$u_3$	$u_4$	$u_5$	
21	$u_2$	$\text{=C11-C11+4*C5}$	$\text{=C11-D11+4*D5}$	$\text{=C11-E11+4*E5}$	$\text{=C11-F11+4*F5}$	
22	$u_3$	$\text{=D11-C11+4*C6}$	$\text{=D11-D11+4*D6}$	$\text{=D11-E11+4*E6}$	$\text{=D11-F11+4*F6}$	
23	$u_4$	$\text{=E11-C11+4*C7}$	$\text{=E11-D11+4*D7}$	$\text{=E11-E11+4*E7}$	$\text{=E11-F11+4*F7}$	
24	$u_5$	$\text{=F11-C11+4*C8}$	$\text{=F11-D11+4*D8}$	$\text{=F11-E11+4*E8}$	$\text{=F11-F11+4*F8}$	

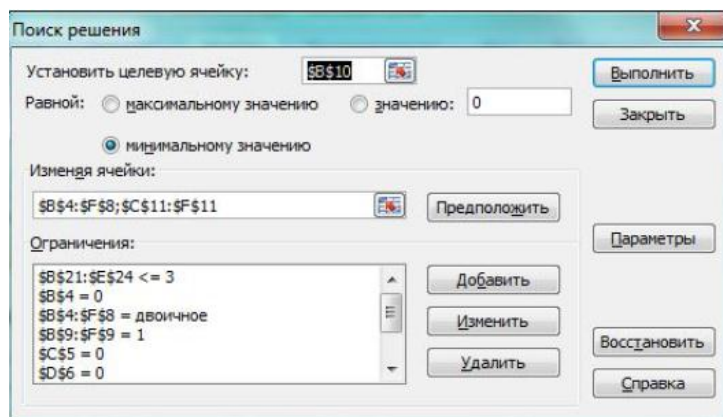
14) “Поиск решения” elementini ishga tushirib, unga quyidagi kattaliklarni kiritamiz:



Maqsad katagi - \$B\$10.

U kichik qiymatga teng (Равной минимальному значению).

Qiymatlari o`zgaruvchi kataklar (Изменяя ячейки): \$B\$4:\$F\$8; \$B\$11:\$F\$11. Bunda faqat qiymatlari o`zgaruvchi va masalaning yechimi yoziluvchi kataklar (\$B\$4:\$F\$8) bilan bir qatorda u<sub>i</sub> o`zgaruvchini o`zida saqlovchi va qiymatlari o`zgaruvchi \$C\$11:\$F\$11 kataklar ham kiritiladi.



Чекловлар:

$$B_{21}:E_{24} \leq 3;$$

$$B_4:F_8 = \text{ikkilik son};$$

$$B_9:F_9 = 1;$$

$$G_4:G_8 = 1;$$

$$B_4 = 0;$$

$$C_5 = 0;$$

$$D_6 = 0;$$

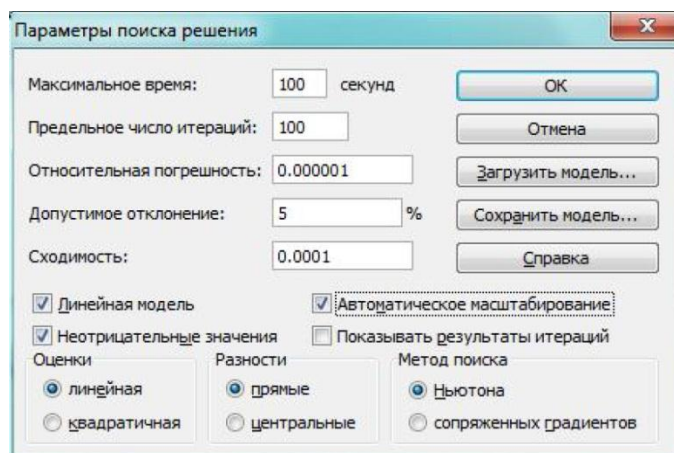
$$E_7 = 0;$$

$$F_8 = 0;$$

Kattaliklar: chiziqli model

(линейная модель), manfiy bo`lmagan

qiymatlar (неотрицательные значения), to`g`ridan-to`g`ri mashtablash (автоматическое масштабирование).



“Поиск решения” muloqot oynada kursorni “Выполнить” tugmasiga keltirib sichqonchanning chap tugmasi bosilsa, dasturning ish sahifasida masalaning yechimi hosil bo`ladi.

Optimal yechim: maqsad funksiyasining qiymati  $F=28$  bo`lib, ob`ektlarni aylanib chiqish 1, 4, 3, 5, 2, 1 ketma-ketlikda amalga oshiriladi.

	A	B	C	D	E	F	G
1							
2		Ўзгарувчилар матричаси					чекловлар
3		1	2	3	4	5	
4	1	0	0	0	1	0	1
5	2	1	0	0	0	0	1
6	3	0	0	0	0	1	1
7	4	0	0	1	0	0	1
8	5	0	1	0	0	0	1
9	чекловлар	1	1	1	1	1	
10	В10да мақсад функцияси	28					
11	C11:F11 блокада и ўзгарувчилар						
12		Масофалар матричаси					
13		1	2	3	4	5	
14	1	9999	11	10	6	12	
15	2	8	9999	6	7	9	
16	3	7	5	9999	8	4	
17	4	3	9	4	9999	5	
18	5	4	6	7	4	9999	
19	Қўшимча и ўзгарувчилар бўйича чекловлар учун формулалар						
20		u2	u3	u4	u5		
21	u2	0	2	3	1		
22	u3	-2	0	1	3		
23	u4	-3	3	0	-2		
24	u5	3	1	2	0		

### **Nazorat savollari:**

1. MS Excelda Solver vositasi vazifasi nimadan iborat?
2. Optimal yechim deganda nimani tushunasiz?
3. “Поиск решения” (yechimni qidirish) vositasi qanday faollashtiriladi?
4. Yechimlarni qidirish amalining bajarilish ketma-ketligini tavsiflang.
5. Marshrutlashning bazaviy modeli, kommivoyajer masalasining mohiyati nima?
6. Kommivoyajer masalasining modelini tavsiflang.
7. Kommivoyajer masalasining matematik modeli qanday shakllantiriladi?
8. Kommivoyajer masalasiga qanday qo'shimchs cheklovlar qo'yiladi?
9. “Параметры поиска решения” muloqot oynasidagi “Автоматическое масштабирование” kattaligi nima sababdan faollashtiriladi?
10. Kommivoyajer masalasiga o'xshash masalaga misol keltiring?

### 3 modul. C++ ALGORITMIK TILI

#### 5 mavzu. Algoritm lash asoslari. Algoritm turlari va xossalari

##### Reja:

1. Kompyuterda masalalarni yechish bosqichlari.
2. Algoritm tushunchasi va uning xossalari.
3. Algoritm turlari.
4. Chiziqli hisoblash jarayonlarini algoritm lash.
5. Algoritm larqa qo'yiladigan talablar.
6. Algoritm larni ta'riflash usullari.
7. Masalalarni algoritm lash va dasturlash asoslari.

**Tayanch iboralar:** Algoritm, matematik model, dastur, dasturlash tili, translyator, kompilyator, interpretator

##### 1. Kompyuterda masalalarni yechish bosqichlari

Kompyuterda masalalarni yechish dasturlarini yaratish jarayonida nazariy va amaliy nuqtai nazardan algoritm lash, dasturlash va kompyuterlarni qo'llash bilan bog'liq bo'lgan bilimlar kerak bo'ladi. Bu yerda asosiy maqsad masalani qo'yish, masalaning yechish algoritmini tuzish, algoritmi mashina dasturi ko'rinishida amalga oshirish, dasturning to'g'riligini tekshirish (algoritmni samaradorligini ko'rsatish) va dasturni foydalanishga topshirishdan iborat. Bu jarayonlar dasturni to'liq yaratish tushunchasiga olib keladi va quyidagi bosqichlarni belgilaydi:

1. Masalaning qo'yilishi.
2. Modelni yaratish.
3. Algoritmni ishlab chiqish.
4. Algoritmni tahlil qilish va to'g'riligini tekshirish.
5. Dastur tuzish.
6. Dasturni ishga tushirish va tekshirish.
7. Dasturni foydalanishga topshirish.

*1. Masalaning qo'yilishi.* Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to'g'ri savollarni aniqlash bo'lib, savollar quyidagicha bo'lishi mumkin:

- 1.1. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi?
- 1.2. Nima berilgan?
- 1.3. Nimani topish kerak?
- 1.4. Yechimni qanday ta'riflash kerak?
- 1.5. Qaysi ma'lumotlar yetarli emas va hammasi kerakmi?
- 1.6. Qanaqa shartlar qabul qilingan?

Albatta, bulardan tashqari boshqa savollar ham tug'ilishi mumkin, yoki ayrim savollarni bir necha bor takror ishlatishga to'g'ri keladi.

Bu bosqichda masalaga tegishli barcha ma'lumotlar yig'iladi, o'rganib chiqiladi va tahlil qilinadi. So'ngra yechishning maqsadi, ya'ni masalani echish natijasida ayni nimani yoki nimalarni aniqlash zarurligi ko'rsatiladi hamda boshlang'ich ma'lumotlar yoki argumentlar hamda qiymatlari aniqlanishi kerak bo'lgan kattaliklar, ya'ni natijalar ajratiladi.

*2. Modelni yaratish.* Matematik model tashqi dunyoning hodisalar turkumini matematik belgilar yordamida taxminiy tavsiflashdir. Hodisani tavsiflash uchun uning muhim xususiyatlarini, qonuniyliklarini, ichki aloqalarini, ayrim xossalarning ahamiyatini aniqlash zarur. Eng muhim faktorlari aniqlanganda, ahamiyatlari kamroq bo'lganlarini hisobdan chiqarish mumkin. Umuman, modelni tanlash fandan ko'ra, ko'proq san'at ishi deb hisoblanadi. Yahshi tuzilgan modellarni o'rganish modellashtirishda tajriba orttirishning eng yahshi usuli hisoblanadi. Modelni yaratishda quyidagi savollarni aniqlash maqsadga muvofiq:

- 2.1. Masalani yechish uchun qaysi matematik uslub ko'proq mos keladi?
- 2.2. O'xshash masalaning yechimi bormi?
- 2.3. Masalaning barcha muhim ma'lumotlari matematik ob'yektlar orqali tavsiflanadimi?
- 2.4. Izlanayotgan natija biron bir matematik o'lchamga mos keladimi?



2.5. Modelning ob'yektlari orasidagi bog'lanishlar aniqlanganmi?

2.6. Tuzilgan model bilan ishlash qulaymi?

Bu yerda tekshirilayotgan ob'ektni matematik atamalarda tavsiflash, ya'ni uning matematik modelini qurish kerak. (Misol,  $S=ab$  formula to'g'ri to'rtburchakning yuzini hisoblaydi).

3. *Algoritmni ishlab chiqish.* Masalaning matematik modeli yaratilgandan so'ng, echish usulini izlash boshlanadi. Ayrim hollarda, masalaning qo'yilishidan keyin to'g'ridan-to'g'ri, masalani echish usuliga ham o'tish kerak bo'ladi. Bunday masala oshkor ko'rinishdagi matematik model bilan ifodalanmasligi mumkin. Bu holda uni hal qilish usullari bayon qilinadi. Bu bosqichda masalani kompyuterdan foydalanib echishning algoritmi tuziladi. Algoritm tuzish jarayoniga algoritmlashtirish deyiladi. Har bir algoritmni ishlab chiqish bevosita o'ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba'zan dasturlarni tezroq tuzish uchun algoritmni tuzmasdan, to'g'ridan to'g'ri dastur yaratishga kirishuvchilar ham bor. Bu sodda masalalar uchun o'rinli bo'lishi mumkin, ammo murakkab masalalar uchun bu xato uslub hisoblanadi. Chunki, aynan algoritmni ishlab chiqish bosqichida eng optimal va samarali echim ishlab chiqiladi. Algoritm turli xil ko'rinishda yozilishi mumkin. Algoritm so'z, formula, jadval yoki blok-sxema ko'rinishlarida beriladi..

4. *Algoritmni tahlil qilish va to'g'riligini tekshirish.* Algoritmni tahlil qilishdan maqsad uning to'g'ri tuzilganligini aniqlashdir. To'g'ri deb shunday algoritmgga aytiladiki, u masalaning qo'yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang'ich ma'lumotlar bilan ham tez bera oladi. Bundan tashqari, algoritmning samaradorligi ham tahlil qilinadi, ya'ni u kam qadamlar bilan natijani tez bera olsin.

5. *Dastur tuzish.* Biror-bir usulda yozilgan algoritmning ma'lum bir dasturlash tiliga dasturi tuziladi. Dastur tuzish deganda, kompyuter uchun dasturni yozish tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

5.1. Asosiy o'zgaruvchilarni aniqlash.

5.2. O'zgaruvchilarning turlarini aniqlash.

5.3. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo'ladi?

5.4. Bog'lanilgan ro'yxatlardan foydalanish ma'nolimi?

5.5. Qanday dasturiy qismlar kerak bo'lishi mumkin (tayyor bo'lsa ham)?

5.6. Qaysi dasturlash tilini tanlash kerak?

Bu bosqichni 3 bosqich natijasini kompyuter tiliga "tarjima" qilish deyish mumkin. Bu yerda oldingi bosqichda amalga oshirilgan ishlar tahlil qilinadi va lozim topilganda takomillashtiriladi. Ba'zi bir hollarda dasturlash tili imkoniyatlaridan kelib chiqqan holda jiddiy o'zgartirishlar ham kiritish mumkin.

6. *Dasturni ishga tushirish va tekshirish.* Dastur ko'rinishida yozilgan algoritmni kompyuter yordamida bajarilishi. Dasturda xatolik bo'lmasa, u natija olish bilan tugalanadi. Biz dasturni har bir qismini tekshiradigan kirituvchi ma'lumotlar to'plamini tanlashimiz kerak. Ushbu ma'lumotlarni dasturga kiritib, olingan natijani, qo'lda olingan natija bilan solishtirib ko'rish lozim. Odatda, dastur bergan natijalar ma'lum bo'lgan yoki qo'lda hisoblangan ma'lumotlar bilan taqqoslanadi, va ular to'g'riligi aniqlansa, dastur to'g'ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo'lmaydi, ya'ni dastur ishlaymaydigan hamma holatlarni hisobga olish qiyin.

7. *Dasturni foydalanishga topshirish.* Dasturda to'g'ri ishlasa va kutilgan natijani berayotgan bo'lsa, un kelgusida foydalanishga topshiriladi.

## **2. Algoritm tushunchasi va uning xossalari**

Har bir inson kundalik hayotida eng sodda yoki murakkab ishlarni bajarishiga to'g'ri keladi. Ko'pchilik hollarda biror ishni qanday bajarish kerakligini tushuntiruvchi ma'lum qoidalar (ko'rsatmalar) berilgan bo'ladi. Inson ba'zi bir ishlarni bajarishda hayotiy tajribasiga suyanadi. Chunki u, o'ziga tanish bo'lgan ishlarni yoki masalalarni bajarish uchun zarur bo'ladigan ko'rsatmalarni qachonlardir, qaerlardandir olgan yoki o'rgangan bo'ladi. Agar undan, bu ishni qanday bajarish lozimligi so'ralsa, uni og'zaki yoki yozma ravishda ketma-ket bajariladigan harakatlar tarzida bayon qilib beradi. Ishni bajarish ko'rsatmasi qanchaliq aniq va tushunarli

tavsiflansa, boshqa kishi uni shunchalik tez o'zlashtiradi va undan samarali foydalanadi. Biror maqsadga erishish uchun tavsiya qilingan ko'rsatma (buyruq, qoida, tartib, ketma-ketlik)lar algoritm atamasi bilan ifodalanadi. Demak, inson har kuni bajaradigan ishlarida o'zi bilmagan holda ma'lum bir algoritmlarga rioya qiladi. Masalan, ovqat tayyorlash, choy damlash, qahva tayyorlash, telefondan foydalanish va h-zo.

Algoritmning aniq ta'rifi yo'q va unga turlicha ta'riflar berilgan. Masalan:

Algoritm - bu chekli qadamlar orqali ko'zlangan natijani olishga qaratilgan ketma-ket harakatdir.

Algoritm - qo'yilgan masalaning echimini chekli qadamlardan so'ng olish uchun ijrochiga berilgan aniq va tushunarli ko'rsatmalardir.

Algoritm ijrochisi — algoritmda ko'rsatilgan buyruqlarni bajara oladigan shaxs yoki texnikadir.

Algoritmlarga xos xususiyatlar:

- o oddiy harakatlar;
- o buyruqlar tizimi.

Har bir ijrochi faqatgina o'zi tushunadigan buyruqlarni (ya'ni, buyruqlar tizimini) bajaradi. Ijrochi buyruqlarni bajarish jarayonida oddiy harakatlarni bajaradi.

Odatda ijrochiga algoritmning maqsadi ma'lum bo'lmaydi. Shuning uchun ijrochi "nimaga" va "nima uchun" degan savollarni bermaydi.

Kundalik hayotimizda ko'pincha bir necha harakatlarni bajarib, kutilgan natijaga erishamiz. Bu harakatlarni tartib bilan yozsak ko'rsatma hosil bo'ladi.

Misol 1. Choy damlash uchun quyidagi amallarni bajarish zarur:

1. Choy qaynatadigan idishga kerakli miqdorda ichimlik suvi solinsin;
2. Suv qaynatilsin (elektr toki yoki gaz plitasi yordamida);
3. Choy damlanadigan choynakka kerakli miqdorda quruq choy solinsin;
4. Suv qaynagach u choynakka solinsin.

Shunga o'xshash ko'rsatmalar matematik masalalarni echishda ham tuziladi.

Misol 2.  $a$  va  $v$  sonlarning o'rta arifmetik qiymatini topish uchun quyidagi ko'rsatmani keltirish mumkin:

1.  $a$  va  $b$  sonlar qo'shilsin.
2. Hosil bo'lgan yig'indi 2 ga bo'linsin.

Demak har xil hayotiy voqealar ham algoritm tarzida tavsiflanar ekan.

Algoritmni yaratish deganda, qo'yilgan masalani echish jarayonini ketma-ket qadamlarga (bosqichlarga) bo'lib chiqish tushuniladi. Bunda, oldingi qadamda olingan natijalar keyingi qadamlarda ishlatiladi. Algoritmda har bir qadami (bosqich)ning mazmuni va ularning bajarilish tartibi aniq ko'rsatilishi kerak. Algoritmning alohida qadami (bosqichi), echilish algoritmi oldin ishlab chiqilgan, boshqa, ancha oddiy masalani bildirishi yoki etarli darajada oddiy va izohlarsiz tushunarli bo'lishi kerak. Algoritmlarni tuzishda, insondan maxsus algoritmik fikrlash uslubi malakasiga ega bo'lishlik talab qilinadi. Algoritmik fikrlash uslubining asosini amallarning uzundan-uzoq ketma-ketligini rejalashtira olish, ularni algoritm ko'rinishida yozish, ularning oqibatlarini ko'ra bilish va algoritmni bajarishda kelib chiqishi mumkin bo'lgan barcha shart-sharoitlarni ko'zda tuta bilish tashkil etadi. Algoritmga arifmetik amallarni bajarish qoidalari, eng katta umumiy bo'luvchini topish, kvadrat tenglamaning ildizlarini topish, ko'phadning hosilasini topish qoidalari va h-zolar misol bo'ladi. Yuqorida keltirilgan misollardan ko'rinadi-ki, bir xil tipli masalalar to'plami uchun yagona algoritm tuzilsa, ushbu algoritm bir xil tipli masalalar bo'yicha umumiy hisoblanadi va bunday masalalarni echish uchun undan foydalanishimiz mumkin ekan. Masalan,  $ax^2+bx+c=0$  ko'rinishdagi kvadrat tenglamalarni echish masalasi ommaviy muammodir. Chunki biz  $a$ ,  $b$ ,  $c$  larni o'zgartirib bir xil tipli masalalar sinfini hosil qilamiz.

Algoritm so'zi buyuk o'zbek mutafakkiri Abu Abdulloh Muhammad ibn Muso al-Xorazmiyning (783-855) nomidan kelib chiqqan. 825 yilda u, Hindistonda qo'llanilgan o'nli sanoq tizimini o'z risolasida batafsil bayon qilib berdi. Al-Xorazmiy risolasida yangi sanoq tizimida amallarni bajarish qoidalarini keltirib o'tgan va birinchi marta 0 sonini qo'llagan. (U sonlarning

hindcha nomini arab tilida as-sifr yoki oddiygina qilib sifr deb ishlatgan, shu tufayli «stifr» va «shifr» soʻzlari vujudga kelgan) Shundan soʻng islom olami olimlari hind sonlarini qoʻllay boshlashgan. XII asrning birinchi yarmida al-Xorazmiyning “Kitab al - jabr val – muqabala” (“Qoʻshish va ayirishlar haqida kitob”) kitobi lotin tiliga tarjima qilingan va u Evropaga tarqaldi. Bu kitob gʻarbda juda mashhur boʻlgan va undan “algebra” soʻzi vujudga kelgan. Bu asar sababli hisoblash ishlarini ixchamlash va osonlashtirish imkoni yaratildi, chunki bu bilan oʻsha davrda qoʻllash rasm boʻlgan rim raqamlari va sonlarni soʻz orqali yozib bajarishdagi noqulayliklar bartaraf etildi.

Al-Xorazmiyning ilmiy risolalari evropalik olimlarga juda chuqur taʼsurot qoldirib algoritmi (algoritmi) soʻzining kelib chiqishiga sabab boʻldi. Chunki, Al-Xorazmiyning lotin tiliga tarjima qilingan kitobi sarlavhasida Abu Abd Allah Muhammad ibn Musa al-Khwarizmi nomi keltirilgan. Bundan tashqari kitobning dastlabki beti “Dixit Algoritmi” (“Dediki Al-Xorazmiy” ning lotincha ifodasi) degan jumlar bilan boshlangan. Keyinchalik al-Xorazmiy soʻzi Evropada “algitm” deb talqin qilingan. Al-Xorazmiy ushbu kitobida matematikada koʻp uchraydigan bir necha amaldan iborat misollarni echish tartibini birinchi boʻlib qoʻllagan (avval qavslar ichidagi va darajaga koʻtarish, keyin koʻpaytirish va boʻlish, va nihoyat qoʻshish va ayirish amallari bajariladi). U birinchi boʻlib oʻnli sanoq sistemasining tamoyillari va unda turli amallar bajarish qoidalarini asoslab berdi hamda kvadrat tenglamalarni echish usullarini keltirib oʻtgan.

*Algoritm xossalari.* Biror masalani kompyuterda yechish uchun avval uning matematik modelini, keyin algoritmini va dasturini tuzish kerak boʻladi. Bu uchlikda, algoritmi tuzish muhim ahamiyatga ega hisoblanadi. Algoritmni tuzish algoritmlashtirish deyiladi. Har qanday algoritmi quyidagi xususiyatlarga ega boʻlishi kerak:

1. Uzluksizlik (diskretlilik).
2. Aniqlilik (determinantlik).
3. Natijaviylik (yoki cheklanganlik).
4. Umumiylik. (ommaviylik).
5. Tushunarlilik.

Har bir xossani alohida koʻrib oʻtamiz.

1. *Uzluksizlik (diskretlilik).* Masalani echish jarayoni algoritmi aniq va oddiy qadamlarning ketma-ket bajarilishi koʻrinishida ifodalanishi shart.

Algitmning har bir qadami toʻla bajarilgandan soʻng navbatdagi qadam bajarilishi lozim. Ketma-ket bajariladigan buyruqlar, amallar algoritmning qadamlari deyiladi. Demak, algoritmlarni doimo chekli qadamlarga boʻlaklash orqali uni chekli sondagi oddiy koʻrsatmalar ketma – ketligi shaklida ifodalash mumkin ekan. Bu qadamlar ketma-ket tartibda bajarilishi lozim. Agar jarayonni ketma-ket bajariluvchi chekli qadamlarga boʻlaklay olmasak, uni algoritmi deb boʻlmaydi.

2. *Aniqlilik (determinantlik).* Algitmning har bir qadami aniq yozilishi va faqat bitta maʼnoga ega boʻlishi shart.

Bu xossa natijasida algoritmning bajarilishi mexanik tavsifga ega boʻladi va yechilayotgan masala haqida hech qanday qoʻshimcha koʻrsatmalar talab qilinmaydi.

Algitmdagi qiymatlar toʻplami oʻzidan oldingi qiymatlar orqali toʻliq aniqlanadi. Har bir qadam bir xil maʼnoda tushunilib, keyingi qadam oʻzidan oldingi qadamlar bilan toʻlaligicha aniqlanishi lozim. Bundan tashqari ijrochiga berilayotgan koʻrsatmalar mazmuni aniq boʻlishi zarur, chunki koʻrsatmadagi noaniqliqlar moʻljallangan maqsadga erishishga xalal beradi. Masalan, inson uchun tushunarli boʻlgan “3 - 4 marta silkitilsin”, “5 – 10 daqiqa qizitilsin”, “1 – 2 qoshiq solinsin”, “tenglamalardan bittasi echilsin” kabi noaniq koʻrsatmalar robot yoki kompyuterni qiyin ahvolga solib qoʻyadi. Shuningdek, koʻrsatmalarning qaysi ketma – ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, koʻrsatmalar aniq berilishi va faqat algoritmda koʻrsatilgan tartibda bajarilishi shart.

3. *Natijaviylik (yoki cheklanganlik).* Algitmi chekli miqdordagi qadamlardan soʻng masalaning echimiga olib kelishi kerak.

Har bir algoritm chekli sondagi qadamlardan so'ng albatta natija berishi shart. Bajariladigan amallar ko'p bo'lsa ham baribir natijaga olib kelishi kerak. Chekli qadamdan so'ng qo'yilgan masala echimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko'rilayotgan jarayon cheksiz davom etib natija bermasa, uni algoritm deb atay olmaymiz.

4. *Umumiylik. (omnaviylik).* Algoritmni, bir-biridan faqat boshlang'ich qiymatlar bilan farq qiluvchi ma'lum sinf masalalarining hammasiga qo'llash mumkin bo'lsin.

Bunda boshlang'ich qiymatlar algoritmning qo'llanilish sohasi deb atalgan sohasidan tanlanishi mumkin. Masalan,  $ax^2 + bx + c = 0$  kvadrat tenglamani haqiqiy sonlar to'plamida echish algoritmini misol sifatida oladigan bo'lsak,  $D = b^2 - 4ac \geq 0$ , bo'lganda ikkita echim hosil qilamiz. Bu echimlar algoritmning natijasiga aylanadi. Agar  $D < 0$  bo'lsa, tenglamaning haqiqiy ildizlari yo'q va algoritmning natijasi sifatida "tenglama haqiqiy ildizlarga ega emas" degan jumla olinadi. Demak, har bir algoritm mazmuniga ko'ra bir turdagi masalalarning barchasi uchun ham o'rinli va masaladagi boshlang'ich ma'lumotlar qanday bo'lishidan qat'iy nazar algoritm shu xildagi har qanday masalani echishga yaroqli bo'lishi lozim. Masalan, ikki oddiy kasrning umumiy maxrajini topish algoritmi kasrlarni turlicha o'zgartirib bersangiz ham ularning umumiy maxrajlarini aniqlab beradi. Yoki uchburchakning yuzini topish algoritmi uchburchakning qanday bo'lishidan qat'i nazar uning yuzini hisoblab beradi.

5. *Tushunarlilik.* Algoritmilar ma'lum bir ijrochiga mo'ljallab tuziladi. Chunki, har bir ijrochi o'ziga xos ko'rgazmaga ega va bu ko'rgazmalar faqat shu ijrochiga tushunarlidir.

Algoritmning ijrochisi hamma vaqt ham inson bo'lavermaydi. Chunki, kundalik hayotimizda ma'lum bir algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, komyuterlar, turli avtomatik va mexanik qurilmalarni ko'ramiz. Bundan kelib chiqadiki, ijrochiga tavsiya etilayotgan ko'rsatmalar, uning uchun tushunarli mazmunda bo'lishi shart, aks holda ijrochi oddiygina amalni ham bajara olmaydi. Har bir ijrochining bajara olishi mumkin bo'lgan ko'rsatmalar yoki buyruqlar majmui mavjud bo'lib, u ijrochining ko'rsatmalar tizimi deyiladi. Demak, ijrochi uchun berilayotgan har bir ko'rsatma ijrochining ko'rsatmalar tizimiga mansub bo'lishi lozim. Ko'rsatmalarni ijrochining ko'rsatmalar tizimiga tegishli bo'ladigan qilib ifodalay bilishimiz muhim ahamiyatga ega. Masalan, quyi sinfnig a'lochi o'quvchisi «sonni kubga oshirilsin» degan ko'rsatmani tushunmaganligi uchun uni bajara olmaydi. Ammo, bu ko'rsatmani "sonni o'zini o'ziga uch marta ko'paytirilsin" shaklida ifodalasak, u bu ko'rsatmani bemalol bajara oladi. Chunki, u ko'rsatma mazmunidan ko'paytirish amalini bajarishi kerakligini anglaydi.

*Algoritmga qo'yiladigan talablar.* Algoritm yuqorida bayon qilingan xossalardan tashqari bir qancha talablarga javob berishi lozim bo'ladi.

*Soddalik.* Har bir tuzilgan algoritm, talab qilingan xossalarni o'zida mujassamlashtirgan holda, tabiatiga ko'ra imkon darajasida, ishlash jarayoni sodda bo'lgan qadamlardan iborat bo'lishi lozim.

Bundan tashqari, algoritm tuzishda quyidagilarga amal qilinsa, quyilgan masalaning natijasini tez va to'g'ri olish mumkin:

- qo'yilgan masalani to'g'ri o'qish va tushunib olish, masalaning asosiy maqsadini ajrata bilish;
- ishga daxldor qiyinchiliklarni aniq ko'ra olish va ortiqcha, masala yechimiga ta'siri bo'lmagan parametrlarni yo'qota bilish;
- qo'yilgan masalani bir-biriga bog'liq bo'lmagan mustaqil bo'laklarga ajrata olish va ular orasidagi bog'liqlikni to'g'ri tashkil etish;
- qo'yilgan masalaning yechimini olishda har bir bo'lak yechimlarni to'plamini bir butun holga keltirish;
- masala yechimini sodda va tushunarli tilda bayon eta olish.

Algoritmni ishlab chiqish uchun qo'yilgan masala maqsadi va mohiyatini aniq tushunish, masala tegishli bo'lgan soha bo'yicha chuqur bilimga ega bo'lish hamda masalani echish yo'lini yaxshi tasavvur qilish talab qilinadi. Bunda masalani avval qog'ozda, katta hisoblashlarni talab qilmaydigan qiymatlarning qandaydir bir to'plami bilan yechib ko'rish, keyin esa uni rasmiylashtirish, ya'ni aniq qoidalar ketma-ketligi ko'rinishida yozish kerak.

### 3. Algoritm turlari

*Algoritmni ifodalash shakllari.* Algoritmni ifodalaganda, uni ko'rgazmali, aniq va ixcham ko'rinishda keltirish va amaliy qo'llash uchun qulay holda ifodalash talab qilinadi. Amaliyotda ishlab chiqilgan algoritmni bir necha shaklda bayon etish qo'llaniladi. Biz ularning eng ko'p uchraydiganlari bilan tanishamiz:

1. So'z orqali, ya'ni tabiiy inson tilida (matn ko'rinishida) ifodalash.
2. Formulalar orqali ifodalash.
3. Jadvallar ko'rinishida ifodalash.
4. Grafik ko'rinishda, ya'ni blok-sxemalar ko'rinishida ifodalash.
5. Maxsus algoritmik tilda ifodalash.
6. Dastur shaklida ifodalanishi.

*1. So'z orqali, ya'ni tabiiy inson tilida (matn ko'rinishida) ifodalash.* Bu holda ijrochi uchun beriladigan har bir ko'rsatma jumlar orqali buyruq mazmunida beriladi.

1-misol. A,B,C,D va x sonlari berilgan. Agar  $Cx - D \neq 0$  shart berilgan sonlar uchun o'rinli bo'lsa, quyidagi ifoda hisoblansin:

$$y = (Ax+B):(Cx-D);$$

Echish: Ifodani hisoblash algoritmini so'z orqali keltiramiz:

- 1) A ni X ga ko'paytirib, natija R bilan belgilansin;
- 2) R ni V ga qo'shib, natija R bilan belgilansin;
- 3) S ni x ga ko'paytirib, natija R1 bilan belgilansin;
- 4) R1 dan D ni ayirib, natija R1 bilan belgilansin;
- 5) R ni R1 ga bo'lib, natija Y ning qiymati deb hisoblansin

*2. Formulalar orqali ifodalash.* Algoritmni formulalar bilan berish shaklidan matematika, fizika, kimyo kabi aniq fanlarni o'rganishda ko'proq foydalaniladi. Bu usulni ba'zan analitik ifodalash deyiladi. Masalan:

- 1)  $A * x = R$
- 2)  $R + B = R$
- 3)  $C * X = R1$
- 4)  $R1 - D = R1$
- 5)  $R / R1 = Y$

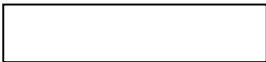
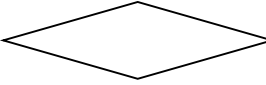





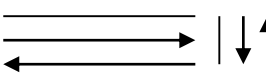

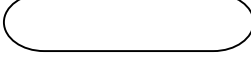

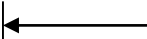
*3. Jadvallar ko'rinishida ifodalash.* Boshlang'ich qiymatlar va natijalar jadval shaklida tasvirlanadi. Masalan, maktabda qo'llanib kelinayotgan to'rt xonali matematik jadvallar yoki turli lotereya, zayomlarning yutuq jadvallari bunga misol bo'la oladi. Funkstiyalarning grafiklarini chizishda ham algoritmning qiymatlari jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmli sodda bo'lganli sababli ularni o'zlashtirib olish oson.

*4. Grafik ko'rinishda, ya'ni blok-sxemalar ko'rinishida ifodalash.* Algoritmning so'zlar orqali bayoni unchalik ko'rgazmali va qulay bo'lmaganligi sababli uni ifodalashda asosan grafik usulda yoki blok-sxema ko'rinishida ifodalash keng tarqalgan.

Matematika kursida chizilgan grafiklarning ko'pchiligi algoritmning grafik usulda berilishiga misol bo'ladi. Bundan tashqari shahar yoki turar joy mavzellarida joylashgan biror uy hamda inshootlarini izlash va harakatlanish bo'yicha berilgan karta – sxemalar ham shunga misol bo'la oladi.

Algoritmni blok sxemalar ko'rinishida ifodalashda alohida bosqichlar har xil geometrik figuralar (bloklar) yordamida tasvirlanadi, ular o'rtasidagi bog'liqliklar esa bloklarni bog'lovchi yo'nalishlar bilan ko'rsatiladi. Bloklarni ulashda oqimlarning faqat vertikal va gorizontol chiziqlaridan foydalanish kerak. O'ngdan chapga tomon yo'nalishga ega bo'lgan gorizontol oqimlar va pastdan yuqoriga yo'naltirilgan vertikal oqimlar, albatta, yo'nalishlar bilan belgilanishi kerak. Oqim chiziqlari tashqi ramka yoki sahifa chegaralariga parallel bo'lishi kerak.

Quyida blok-sxemaning asosiy belgilashlarini keltiramiz.

Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarilishi natijasida ma'lumotlarning o'zgarishi
Qaror		Biror shartni kiritish va unga bog'liq ravishda algoritmning bajarilish yo'nali-shini tanlash
Shakl o'zgartirish, takrorlash		Takrorlash sarlavhasi, takrorlash bloki
Avval aniqlangan jarayon		Oldindan ishlab chiqilgan dastur yoki algoritmdan foydalanish
Kiritish Chiqarish		Axborotlarni qayta ishlash mumkin bo'lgan shaklga o'tkazish yoki olingan natijani tasvirlash
Display		EHM ga ulangan displeydan axborotlarni kiritish yoki chiqarish
Hujjat		Axborotlarni qog'ozga chiqarish yoki qog'ozdan kiritish
Axborotlar oqimi chizig'i		Bloklar orasidagi bog'lanishlarni tasvirlash
Bog'lagich		Uzilib qolgan axborot oqimlarini ulash belgisi
Boshlash Tugatish		Algoritmning boshlanishi yoki tugallanishini belgilaydi
Izoh		Bloklarga tegishli turli xildagi tushuntirishlar
Birlashish		Oqim chiziqlarining birlashishi

Algoritm blok-sxemalari uchun Davlat Standarti qabul qilingan va unga rioya qilish talab qilinadi.

Har bir blok ichida bajariladigan amal yoziladi. Bloklar o'zaro chiziqlar orqali birlashtiriladi va ular hisoblash yo'nalishini belgilaydi. Ketma-ketlikni aniq ko'rsatish maqsadida har bir blokning yuqori chap chizig'ida raqam qo'yish mumkin.

Oqimlarning parallel chiziqlari orasidagi masofa 3 mmdan, sxemaning qolgan elementlari orasidagi masofa 5 mmdan kam bo'lmasligi kerak.

Bloklarni joylashtirish uchun sahifa maydonini gorizonta va vertikal (tarmoqlanuvchi sxemalar uchun) sohalarga bo'lish tavsiya etiladi.

Algoritmning har bir bloki qulaylik uchun raqamlanishi mumkin.

Barcha shakllar bir tekisda tasvirlanishi va bloklarning kattaligi bir o'lchamda chizilishi shart. Agar blokning balandligi  $v$  ga teng bo'lsa, uning uzunligi  $a=1,5*v$  ga teng bo'lishi kerak.

5. *Maxsus algoritmik tilda ifodalash.* Bu algoritmlarni bir xil va aniq yozish hamda ularni bajarish uchun ishlatiladigan belgilashlar va qoidalar tizimidir.

Birinchi dan algoritmik til oddiy tilga yaqin bo'lib, u oddiy matndek yozilishi va o'qilishi mumkin. Ikkinchi dan, algoritmik til matematik simvollar: sonlar, kattaliklar va funkstiyalarning belgilari, amallar ishoralari, qavslar va boshqalarni o'z ichiga oladi. Algoritmik til o'z lug'atiga

ega. Bu lug'atning asosini biror algoritm ijrochisining buyruqlari tizimiga kirgan buyruqlarni ifodalashda qo'llaniladigan so'zlar tashkil qiladi. Hozirgi kunda turli tuman algoritmik tillar mavjud. A. P. Ershov boshchiligida yaratilgan algoritmik tilda shaharlararo telefon-avtomatdan foydalanish ko'rsatmasini yozilishini ko'ramiz.

**alg** shaharlararo telefon-avtomatdan foydalanish

**boshl**

trubkani oling  
 telefon jetonini tashlang  
 uzluksiz signalni kuting  
 kerakli shahar kodini tering  
 abonent ovozi kuting  
 "gaplashish" tugmasini bosing  
 gaplashing

**tam**

6. *Dastur shaklida ifodalanishi.* Millionlab kompyuterlarning keng tarqalishi algoritmning dastur tarzidagi tasvirining keng ommalashib ketishiga katta turtki berdi. Chunki, kompyuterlar doimo dasturlar yordamida boshqariladi.

*Algoritmning turlari.* Har qanday murakkab algoritmlar amalda uch xil strukturaning kombinatsiyasidan iborat bo'ladi. Ular:

1. *Chiziqli algoritmlar.*
2. *Tarmoqlanuvchi algoritmlar.*
3. *Takrorlanuvchi algoritmlar.*

Algoritmning asosiy turlaridan tashqari shartli ravishda yana quyidagi turlarga ajratish mumkin:

- o *ichma-ich joylashgan takrorlanish algoritmlari;*
- o *rekurrent algoritmlar;*
- o *takrorlanishlar soni oldindan noma'lum algoritmlar;*
- o *ketma-ket yaqinlashuvchi algoritmlar.*

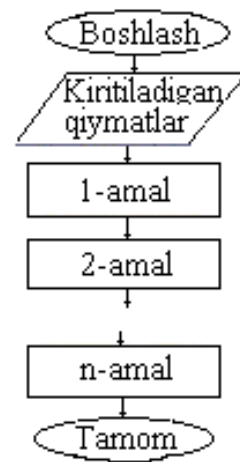
Algoritmning shartli ravishda qo'shimcha turlarga ajratgan bo'lsakda, aslida ular yuqorida keltirilgan asosiy turlarning kombinatsiyasidan tashkil topadi.

Ular eng soddasi chiziqli algoritmlar bo'lib hisoblanadi.

1. *Chiziqli algoritmlar.* Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-chiziqli algoritmlar deyiladi.

Chiziqli algoritmlar amallar ketma-ketligini tashkil qilib, hisoblash jarayonida bu amallar kelish tartibi bo'yicha faqat bir martagina bajariladi. Amalda chiziqli algoritmlar alohida juda kam uchraydi. Ular boshqa turdagi algoritmlar tarkibida qo'llaniladi. Bunday algoritmlardan, masalan, matematika, fizika, kimyo va boshqa formulalarni yuqori aniqlikda hisoblashda foydalanish mumkin. Ularda hisoblash amallari ketma-ket bajariladi.

Blok sxemadagi bloklarni turli xil holatlarda joylashtirish mumkin. Buning uchun eng asosiysi, ular ijrochiga tushunarli va qaysi qadamdan keyin qaerga borishi kerakligini aniq ko'rsatish kifoya qiladi.



$$y = (a + 2)^3$$

$$y = \sin x$$

$$i = i + 1$$

$$s = 0$$

$$z = y + 2x$$

Blok sxemada hisoblash amallari to'g'ri to'rt burchaklar ichida beriladi.

2. *Tarmoqlanuvchi algoritmlar.* Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo'yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi.

Tarmoqlanuvchi algoritmlar o'z tarkibida bir yoki bir nechta shartni tekshiruvchi blokning bo'lishi bilan xarakterlanadi va ularda ayri strukturasi ishlatiladi. Bu algoritmda shartning

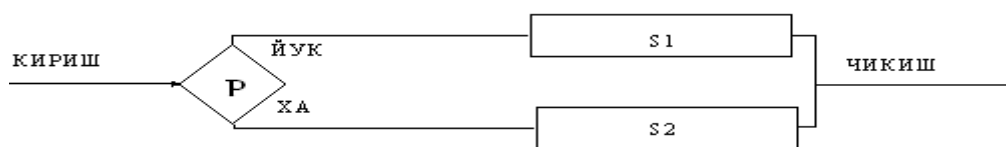
bajarilishi yoki bajarilmasligining natijasiga qarab u yoki bu amalning bajarishi ta'minlanadi. Shuning uchun shartning bajarilishiga qarab hisoblash jarayoni albatta tarmoqlanadi.

Berilgan shart romb orqali ifodalanadi, r-berilgan shart. Agar shart bajarilsa, "ha" tarmoq bo'yicha a amal, shart bajarilmasa "yo'q" tarmoq bo'yicha b amal bajariladi.

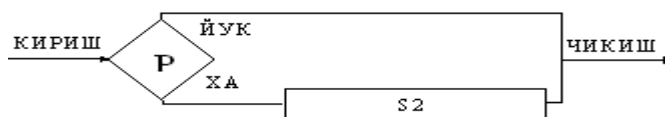
Tarmoqlanuvchi algoritmlarning blok-sxemasi turli xil holatda joylashtirilishi mumkin, eng asosiysi u tushunarli bo'lishi talab qilinadi. Ikkinchi holatdagi tasvirda romb ichidagi shartning qanoatlangan (ha) holdida A1, aks holda (yo'q) esa A2 bajariladi. Ikkala holatning biridan keyin A3 bajariladi.

Tarmoqlanuvchi algoritmlarning ikki xil konstruktsiyasi mavjud:

1. To'liq shartli konstruktsiya:



2. To'liq bo'lmagan shartli konstruktsiya:



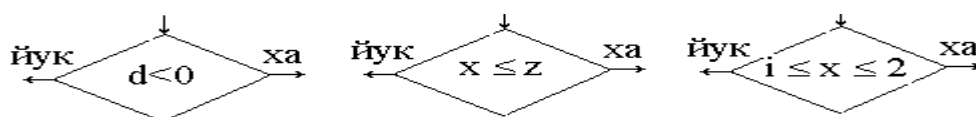
Ko'pgina masalalarni echishda, shart asosida tarmoqlanuvchi algoritmlarning ikkita tarmog'idan bittasining, ya'ni yoki "ha" yoki "yo'q" ning bajarilishi yetarli bo'ladi. Bu holat tarmoqlanuvchi algoritmning xususiy holi sifatida aylanish strukturasi deb atash mumkin. Aylanish strukturasi quyidagi ko'rinishlarga ega:

Demak, tarmoqlanuvchi algoritm to'la yoki qisqa bo'lishi mumkin.

Tarmoqlanishning to'la formasi: "Ha" va "Yo'q" shartlar qatnashadi.

Tarmoqlanishning qisqa formasi: "Ha" yoki "Yo'q" shartlar qatnashadi.

Shartlar romblar ichida yoziladi, masalan:

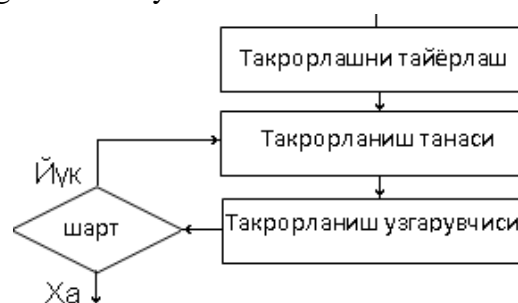


Shart sifatida ijrochiga tushunarli, rioya qilinishi yoki qilinmasligi mumkin bo'lgan har qanday ta'kid ishlatiladi. Ta'kid so'zlar yoki formula orqali ifodalanishi mumkin.

Ba'zi bir masalalarni echishda shartsiz o'tish ishlatiladi va u orqali boshqaruvni u yoki bu hisoblashga berish mumkin.

3. *Takrorlanuvchi algoritmlar.* Agar biror masalani echish uchun tuzilgan zarur bo'lgan amallar ketma-ketligining ma'lum bir qismi biror parametrga bog'liq holda ko'p marta qayta bajarilsa, bunday algoritm takrorlanuvchi yoki stiklik algoritmlar deyiladi.

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruqlar ketma-ketligini ma'lum shartlar asosida qayta-qayta bajarish zaruriyati tug'iladi. Ma'lum bir o'zgaruvchining turli qiymatlarida ma'lum buyruqlar tizimining biron bir qonuniyatga asosan takrorlanib bajarilishi takrorlanuvchi hisoblash jarayonini (stikl) hosil qiladi. Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini





takrorlanishning tanasi dab ataladi. Takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchini takrorlanish o'zgaruvchisi yoki takrorlanishning boshqaruvchi o'zgaruvchisi (stikl parametri) deb yuritiladi.

Takrorlanuvchi jarayonning algoritmi umumiy holda quyidagilarni o'z ichiga olishi kerak:

1. Takrorlanishni tayyorlash - takrorlanishni boshlashdan oldin, takrorlanishda qatnashadigan o'zgaruvchilarning boshlang'ich qiymatlari yoki takrorlanish o'zgaruvchisining boshlang'ich qiymati o'rnatiladi, takrorlanish o'zgaruvchisining o'zgarish qadami belgilanadi.

2. Takrorlanish tanasi - takrorlanish o'zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko'rsatiladi.

3. Takrorlanish o'zgaruvchisiga yangi qiymat berish - har bir takrorlanishdan avval o'zgaruvchiga o'zgarish qadamiga mos ravishda yangi qiymat beriladi.

4. Takrorlanishni boshqarish-takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o'tish ko'rsatiladi. Algoritmning takrorlash qismini umumiy holda quyidagi sxema orqali ko'rsatish mumkin:

Takrorlash algoritmi takrorlash tanasi va takrorlashni amalga oshiruvchi shartdan iborat.

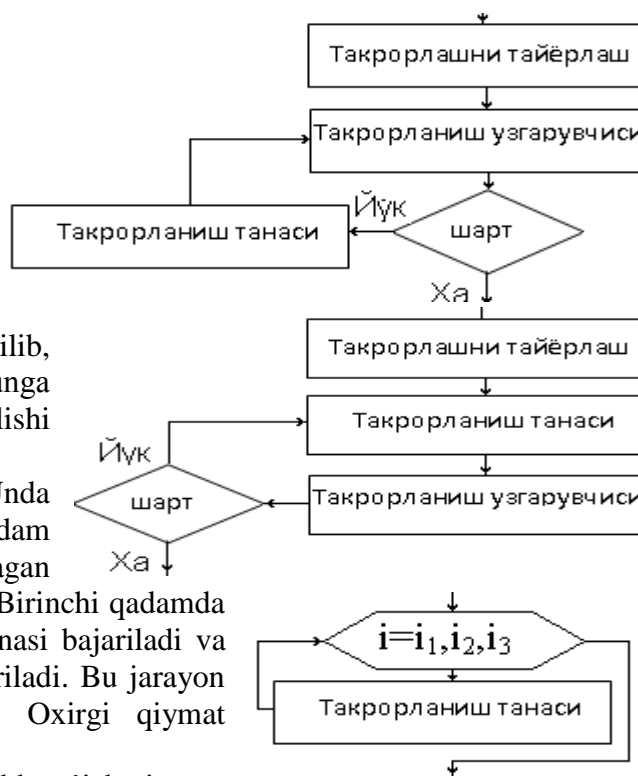
Takrorlash algoritmi ham uch turga bo'linadi:

1 - tur "toki" takrorlash deb atalib, bunda berilgan shart tekshiriladi, shu shart qanoatlangunga qadar stikl tanasi takrorlanaveradi. Shart bajarilmasa, stikldan chiqib ketiladi.

2 - tur algoritmi "gacha" takrorlashdir. "Gacha" stiklida dastlab stikl tanasi bajarilib, so'ngra shart tekshiriladi. Shart qanoatlangunga qadar stikl tanasi takrorlanaveradi. Shart bajarilishi bilanoq stikldan chiqiladi.

3 - tur parametr bo'yicha takrorlashdir. Unda boshlang'ich qiymat, oxirgi qiymat va qadam mavjud bo'ladi. Agar qadam ko'rsatilmagan bo'lsa, u holda qadam 1 ga teng deb olinadi. Birinchi qadamda boshlang'ich qiymat bo'yicha takrorlanish tanasi bajariladi va qadam oshirilib yana takrorlanish tanasi bajariladi. Bu jarayon oxirgi qiymat bajarilgancha davom etadi. Oxirgi qiymat bajarilgach takrorlanishdan chiqiladi.

Bu erda  $i$  - ixtiyoriy o'zgaruvchi,  $i_1$  - boshlang'ich qiymat,  $i_2$  - oxirgi qiymat va  $i_3$  - qadam.



#### 4. Chiziqli hisoblash jarayonlarini algoritmlash

Chiziqli jarayonlar algoritmlari amallar ketma-ketligini tashkil qilib, hisoblash jarayonida bu amallar kelish tartibi bo'yicha faqat bir martagina bajariladi. Amalda chiziqli algoritmlar juda kam uchraydi. Bunday algoritmlardan, asosan, katta formulalarni yuqori aniqlikda hisoblashda foydalanish mumkin. Chiziqli hisoblash jarayoni algoritmidan - bloklar yuqoridan quyiga tomon, ya'ni boshidan oxiriga tomon bir chiziq ostida (shuning uchun chiziqli deyiladi) ketma-ket bajariladi.

Chiziqli algoritmning ko'rgazmali sxemasida quyidagi ishlar bajariladi:

1. Algoritmni ishga tushirish.
2. Foydalanuvchidan ma'lumotlar olish.
3. Ma'lumotlarni qayta ishlash.
4. Natijalarni displeyga yoki printerga chiqarish.

Chiziqli algoritmga misol ko'raylik.



*Misol 1:* Kompyuter xotirasining A va B yacheykalardagi qiymatlarni almashtirish masalasi algoritmini tuzamiz:

Algoritmining sxemasi 5 blokdan iborat. “Boshlash” bloki algoritmni bajarishga sozlash, algoritmda ishlatiladigan o`zgaruvchilar uchun xotiradan joy ajratish ishini bajaradi.

2-blokda A va B o`zgaruvchilarning qiymatlari xotiraning shu nomli yacheykalariga uzatiladi.

3-blokda A yacheykadagi qiymat C o`zgaruvchi tomonidan o`zlashtiriladi, ya`ni C nomli yacheykaga uzatiladi. A yacheykadagi eski qiymat o`chib, uning o`rnida B o`zgaruvchining qiymati yoziladi. C yacheykadagi qiymat (A yacheykaning 1-qiymati) B yacheykaga uzatiladi. B o`zgaruvchining eski qiymati yangi C o`zgaruvchi qiymati bilan almashtiriladi.

4-blokda A va B yacheykalardagi yangi hosil bo`lgan qiymatlar talab qilingan tarzda chiqariladi.

5-blokda algoritm o`z ishini tugatadi.

*Misol 2:* Tomonlari A ga teng bo`lgan kvadratning perimetri (P) va yuzasini (S) topish algoritmini tuzamiz.

Algoritm blok-sxemasi olti blokdan iborat. Algoritmning bajarilishi 1-“Boshlash” blokdan boshlanadi. Bu blok avtomatni ulash, uni algoritmni bajarishga sozlash va algoritmda ishlatiladigan o`zgaruvchilarning hammasi uchun xotiradan joy ajratish kabi ishlarni bildiradi. Bu algoritmda bunday o`zgaruvchilarga A, P, S kiradi. Demak, ularning har biri uchun xotiradan bittadan yacheyka ajratiladi. Shu bilan 1-blok o`z ishini tugatadi.

Chizmadan ko`rinib turibdiki, 1-blok 2-blok bilan oqimning vertikal chizig`i orqali ulangan. Bu chiziq oqim yo`nalishini ko`rsatuvchi yo`nalishlarga ega emas. Shunday qilib, 1-blok bajarib bo`linganidan keyin boshqarish 2-blokka uzatiladi. 2-blok A-o`zgaruvchi qiymat o`zlashtirishini ko`rsatadi. Bu, o`zgaruvchiga avtomatik ravishda ajratilgan yacheykaga, o`zgarimas joylashtirish kerakligini bildiradi. Uzgarimasning kompyuterga kiritish usuli, kiritish bo`lagining qanday dasturlanganligiga bog`liq bo`ladi. Masalan, o`zgarimasni klaviaturadan kiritish yoki uni oldindan tayyorlangan fayldan olish mumkin. O`zgarimaslar ma`lumotlarni tashqi manbalardan, masalan, kompyuterga ulangan fizik qurilmalardan ham olishi mumkin.

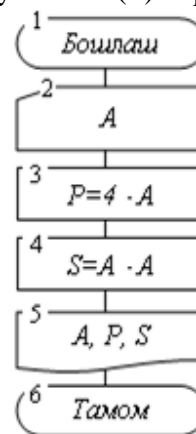
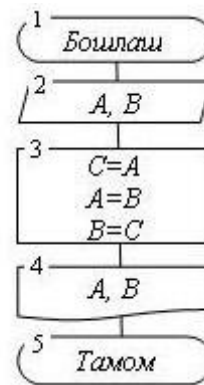
Mazkur misol uchun o`zgarimaslarni uzatish usuli emas, balki 2-blok bajarilganda A adresli yacheykaga aniq o`zgarimasning kiritilishi muhimdir. Masalan, 5 soni shunday o`zgarimas bo`lsin.

Shundan keyin, boshqarish oqim chizig`i bo`ylab, 3-“Jarayon” blokiga uzatiladi. Bu blokda, unda joylashtirilgan buyruqning bajarilishida 4 soni A yacheykada joylashtirilgan o`zgarimasga, ya`ni 5 ga ko`paytiriladi, natija, ya`ni P o`zgaruvchiga 20 o`zlashtiriladi (20 soni P adresli yacheykaga beriladi). Bu ishlar bajarilgach, boshqarish 4-blokka uzatiladi.

4-blokda xuddi shunday tarzda A o`zgaruvchining qiymatlari ko`paytiriladi va natija (25 soni) S o`zgaruvchi tomonidan o`zlashtiriladi (S adresli yacheykaga 25 soni kiritiladi). Shundan keyin 5-blokka o`tish sodir bo`ladi.

5-blok buyruqlarini bajarish vaqtida A, P, S yacheykalarda saqlangan o`zgaruvchilarning mos qiymatlari chiqariladi (masalan, ekranga, kog`ozga, tashqi faylga va h.k.). A=5 bo`lgan misol uchun 5, 20, 25 – o`zgarimaslar, ya`ni kvadrat tomon uzunligi, perimetri va yuzasi chiqariladi. Shundan keyin boshqarish 6-blokka uzatiladi.

6-“Tamom” blokida A, P, S o`zgaruvchilar band qilgan xotira yacheykalarini bo`shatish sodir bo`ladi va algoritm o`z ishini tugatadi. Bu algoritm qaytadan ishga tushirilganda (2-blokda A o`zgaruvchiga boshqa son berilganda), algoritmning 5-blokida boshqa sonlar paydo bo`ladi.



## 5. Algoritmlarqa qo'yiladigan talablar

Foydalanuvchi dastur bilan ishlaganda uni natija qiziqtiradi va dastur bu natijani qanday usullar bilan olganligi uni qiziqtirmaydi. Foydalanuvchi uchun dasturning tashqi ko'rinishi, ishlatishning soddaligi va qulayligi hamda ko'zlangan natijani tez, aniq va xatosiz berishi juda muhim hisoblanadi. Dastur tuzuvchilar esa unga ichki tomondan qarashadi. Ular dastur qanday ishlashini his qilishadi, nima uchun ish jarayonlari aynan shunday sodir etilayotganligini va xuddi shunday bo'lishini, aksincha bo'lishi mumkin emasligini tushunishadi. Ular asosan dastur kutilayotgan natijani har qanday holatda ham aniq, xatosiz va tez berishiga e'tibor qaratishadi. Bu tushunchalar nimani anglatishini ochib berishga harakat qilamiz.

Agar biror bir manzilga borish kerak bo'lsa va bu manzil qaerda ekanligini bilmasangiz, u erga qanday borish kerakligini biladigan kishidan yo'lni so'raysiz. Belgilangan manzilga turli xil yo'llar va vositalar yordamida borish mumkin. Agar e'tibor bergan bo'lsangiz kimdir boriladigan manzil yo'nalishini eng qisqasini aniq tartib bilan batafsil tushuntirsa, kimdir uzoqroq yo'lni qisqa va pala-partish tartibda tushuntiradi. Olingan ma'lumotlarni tahlil qilgan holda kerakli manzilga borishga oshiqamiz va kerak bo'lsa yo'l davomida yana qo'shimcha ma'lumot olamiz hamda ulardan xulosa chiqarib ko'zlangan manzilga etib boramiz. Masalan, Siz Buxoro shahriga sayohatga keldingiz va mashhur "Minorai Kalon"ni tomosha qilmoqdasiz. "Ismoil Somoniy" maqbarasi haqida eshitgansiz va u ham yaqin atrofda ekanligini bilasiz hamda uni ham ko'rish orzusidasiz. Shuning uchun unga qanday borish kerakligi haqida so'raysiz.

"Minorai Kalon" maydonidan "Ismoil Somoniy" maqbarasiga qanday borish kerakligi (piyoda sifatida) haqida quyidagicha ko'rsatma berishadi.

1. G'arb (quyosh botar tomon yoki qo'li bilan ko'rsatgan holda) tomonga to'g'ri ko'cha bo'ylab taxminan 800 m yurasiz.
2. Taxminan 700 m yurgach o'ng tomonda «Ark» ka duch kelasiz.
3. "Ark" ro'parasidagi ("Ark" ka kirish nazarda tutiladi) avtomobillirning kichik halqa yo'lini kesib o'tasiz va parkka kirasiz.
4. Parkka kirib to'g'ri taxminan 100 m yursangiz "Ismoil Somoniy" maqbarasini ko'rasiz.

Ushbu ko'rsatmani berganda kimdir, yo'lda taxminan 100 m yurgach o'ng tomonda bozorga duch kelishingizni, undan keyin yana taxminan 300 m yurgach "Ark" ka duch kelishingizni va u "qadimgi Buxoro davlati hukmdorlari qarorgohi-xohlasangiz kirib ko'rishingiz mumkin" degan izohni berishi ham mumkin. Kimdir esa "Ark" ro'parasidagi machitni ham ziyorat qilishni va parka qanday kirish mumkinligini batafsil tushuntirishi mumkin. Kimdir qo'shimcha ravishda "Ismoil Somoniy" maqbarasiga yaqin bo'lganligi uchun "Chashmai Ayub" hamda qadimiy Buxoro qal'asi darvozasi va devorlarini ko'rishni ham maslahat beradi. So'ralgan manzilga qanday borish kerakligi haqidagi tushuntirish ko'rsatmadir. Agar e'tibor bersangiz bir xil savolga har kim turlicha ko'rsatma beradi va izohlar bilan boyitadi. Ko'rsatma qisqa yoki to'liq yoki batafsil berilishidan qat'iy nazar uni tahlil qilamiz va kamchiliklarini mushohida orqali tuzatamiz.

Faraz qiling, sizning o'ringizda texnik "robot" va u "Minorai Kalon" maydonidan "Ismoil Somoniy" maqbarasiga borishi kerak. Unga, sizdan farqli ravishda, boriladigan yo'nalishni qadamba-qadam, tartib bo'yicha va batafsil tushuntirish talab qilinadi. To'g'riga necha metr yurishni, burilishlarda necha gradus burilish kerakligini aniq ko'rsatish talab qilinadi. Agar avtomobil yo'lini kesib o'tish tartibini tushuntirmasangiz, u yo'lni kesib o'tishda halokatga uchrashi ham mumkin. Agar unga bajaradigan ishlari tartibini o'zgartirgan holda taqdim qilsangiz, u noaniqlikka duch kelib nima qilish kerakligini bilmay qoladi. Avval parkka qanday kirish, keyin "Ark" yonidan o'tish va "Minorai Kalon" maydonidan qaysi tomonga borish tarzida gi tartibda ko'rsatma berilsa inson ularni o'z o'rniga qo'yib harakat qiladi. "Robot" esa qanday tartibda berilgan bo'lsa, shu tarzda, navbatma-navbat berilgan ko'rsatmalarni bajaradi. Shuning uchun "robot"ga bajaradigan ishlarini, tartibi bo'yicha va batafsil tushuntirish talab qilinadi. Agar ko'rsatma insonga berilsa u yozma yoki og'zaki tarzda bayon qilinadi. Ko'rsatma ogzaki tarzda bayon qilinsa, ko'rsatma beruvchi va uni bajaruvchi tushunadigan tildan foydalaniladi. Ko'rsatma yozma tarzda bayon qilinganda, ko'rsatma beruvchi va uni bajaruvchi tushunadigan til, alfavit, belgilar va imlo qoidalaridan foydalaniladi. Ko'rsatma "robot"ga faqat yozma tarzda, uning o'z

maxsus tili, alfaviti, belgilari va imlo qoidalaridan foydalangan holda bayon qilinadi.

Eng asosiysi, ishni bajarish ko`rsatmasi aniq, barcha imkoniyatlar hisobga olingan, qisqa, kam vaqt sarflanadigan va tez bajariladigan bo`lishi lozim. Bular algoritmlarqa qo`yiladigan eng muhim talablar hisoblanadi.

### **6. Algoritmlarni ta`riflash usullari**

Olimlarning izchil izlanishlariga qaramay, algoritm tushunchasiga bitta aniq ta`rif berishning imkoni bo`lmadi va natijada bir qancha ta`riflar ishlab chiqildi. Ilmiy tadqiqotlarning natijasiga ko`ra formal ta`riflarning o`zaro teng kuchli va ekvivalent ekanligi aniqlandi. Biz quyida ularning ba`zi birlarini keltirib o`tamiz.

Ta`rif 1. Algoritm – berilgan masalani echish uchun boshlang`ich ma`lumotlar ustida bajariladigan cheklangan ketma-ketlikdagi oddiy amallarning qaysidir tilda berilishidir.

Ta`rif 2. (Knut). Algoritm – bu chekli sondagi qoidalar to`plami bo`lib, u aniq masalalar to`plamini echish amallari ketma-ketligini aniqlaydi va beshta muhim xususiyatlarga ega: cheklanganlik, aniqlik, kiritish, chiqarish va unumdorlik.

Ta`rif 3. (Kolmogorov). Algoritm – bu qo`yilgan masala natijasiga qandaydir sondagi qadamlardan keyin olib keluvchi ma`lum qoidalar bo`yicha bajariluvchi har qanday hisoblash tizimidir.

Ta`rif 4. (Markov). Algoritm – bu boshlang`ich ma`lumotlardan izlangan natijaga olib keluvchi hisoblash jarayonini aniqlovchi aniq ko`rsatmalardir.

Ta`rif 5. Algoritm – bu qo`yilgan masalani echimini chekli qadamlardan so`ng olish uchun ijrochiga berilgan aniq va tushunarli ko`rsatmalardir.

Umuman olganda algoritm tushunchasi ta`riflari bir qator talablarga javob berishi kerak.

- Algoritm chekli sondagi elementlar bajaruvchi ko`rsatmalardan iborat bo`lishi kerak.
- Algoritm chekli sondagi qadamlardan iborat bo`lishi kerak.
- Algoritm barcha boshlang`ich ma`lumotlar uchun umumiy bo`lishi kerak.

### **7. Masalalarni algoritmlash va dasturlash asoslari**

Amaliyotda algoritm ijrochisi kompyuter hisoblanadi. Hozirgi zamon kompyuterlari inson qo`llaydigan rus, o`zbek yoki boshqa shu kabi biror-bir tilda tuzilgan algoritmni tushuna oladigan darajada takomillashgan emas. Shu bois, mashinaga mo`ljallangan buyruqlar kompyuter tushunadigan shaklda yozilishi zarur. Bu maqsadda algoritmik yoki dasturlash tillari deb ataluvchi sun`iy tillar qo`llaniladi. Algoritm ushbu sun`iy tilda yozilishi kerak. Ijrochi (kompyuter) berilgan buyruqlarni aniq va to`g`ri bajarishi uchun unga beriladigan buyruqlar aniq yozilishi va ijrochi tomonidan bir xil tushunilishi lozim. Shu sababli kompyuter tushunadigan til dasturlash tili, bu tilda yozilgan algoritm esa kompyuter dasturi deyiladi.

*Dastur* – kompyuterga algoritmni beruvchi ko`rsatmalar (buyruq yoki tasnif va operatorlar)ning ketma-ketligi.

Dastur kompyuter tomonidan qaysi tartibda, qaysi ma`lumotlar ustidan va qaysi amallar bajarilishi va natija qaysi shaklda taqdim etilishini ko`rsatadi. Kompyuterning boshqarish qurilmasi mashina buyruqlari ketma-ketligi shaklida berilgan dasturni qabul qiladi. Dasturni mashina tilida yaratish – noqulay va katta mehnat talab qiluvchi jarayon. Shuning uchun dastur inson tomonidan dasturlash tillarining birida yaratilib, keyin u mashina tiliga o`giriladi.

Dasturlash tillari paydo bo`lishidan oldin dasturlar mashina kodlarida (mashina kodi - proressor tomonidan bajariladigan xotiradagi instruksiyalar kema-ketligidir) tuzilar edi. Katta - katta dasturlarni tuzishda juda ko`p vaqt talab qilinadi, ularni xatolarini tuzatish juda qiyin, modifikastiyalash esa ko`p hollarda ilojisiz edi. Shularni hisobga olib, inson uchun tushunarli bo`lgan dasturlash tillarini tashkil qilish muammosi paydo bo`ldi va dasturlash tillari vujudga keldi.

*Dasturlash tili* – kompyuter uchun dasturlar (ko`rsatmalar yig`masi) yozilgan uni u yoki bu harakatlarni bajarishga majbur qiladigan rasmiy til.

Dasturlash tilining ta`rifi quyidagilarni o`z ichiga oladi:

1. Mumkin bo`lgan belgilar ro`yxati.
2. Zahiralangan so`zlar ro`yxati.

3. Sintaks (belgilarni va zahiralangan so'zlarni birikmalash usullari).

4. Semantika (dasturlash tilining birikmalari ma'nosi)

Dasturlash tilida yozilgan ko'rsatmalar dastlabki kod deb ataladi.

Dastlabki kod kompyuterda bajarilishidan avval, uni mashina kodiga o'girish lozim.

*Mashina kodi tili* kompyuter qurilmalari bilan bevosita bog'liq bo'lib, buyruqlar ularning kodlari yordamida yoziladi. Dastlabki kompyuterlar (Eniak, MESM va boshqalar) ana shu tilda ishlar edi.

Dasturlash tillari yaratilishi bo'yicha ikki guruhga bo'linadi:

○ Quyi darajadagi tillar yoki asseblerlar.

○ Yuqori darajadagi tillar.

*Quyi darajadagi* dasturlash tillarining buyruqlari faqat raqamlar emas, balki inson nutqiga yaqin so'zlar ham ishlatila boshlandi. Ular ko'pincha assebler tillari deb ham yuritiladi. Mazkur tilda tuzilgan dastur yuqori darajadagi tillarda tuzilgan dasturdan 10-15 marta tezroq ishlaydi. Unda boshqa darajadagi tillarda bo'lmagan amallar mavjud.

Assebler tillarida buyruqlar qisqartirilgan so'zlar yoki so'zlar majmuidan iborat bo'lib, gohida ularni mnemokodlar deb ham yuritiladi.

*Yuqori darajadagi* dasturlash tillari ancha rivojlangan tuzilishga ega bo'lib, unda ishlatiladigan xizmatchi so'zlar inson nutqiga juda yaqinlashtirilgan. Mazkur tilda foydalanuvchi amallarni ketma-ket yozadi, kompyuter esa uni kerakli ko'rinishga o'tkazib oladi. Dastlabki til Plankalkyul deb nomlanib, u 1946 yilda nemis olimi Konrad Stuze tomonidan yaratildi.

Mavjud dasturlash tillari bir necha guruhga bo'linadi:

○ Strukturali (tuzilmali)

○ Prostedurali

○ Mantiqiy

○ Ob'ektga yo'naltirilgan va h-zo

Hozirgi kunda dasturlash tillarini besh avlodga ajratishadi:

○ Birinchi avlod: mashina tili.

○ Ikkinchi avlod: Assebler tili.

○ Uchinchi avlod, yuqori bosqichli tillar: Masalan, C, C++, C#, Java, ...

○ Turtinchi avlod, ma'lumotlar bazalarini boshqarish tizimlari bilan ishlaydigan va ma'lumotlar oqimini boshqaradigan yuqori bosqichli tillar (4GL): Masalan, SQL, Focus, Metafont, PostScript, RPG-II, IDL-PV/WAVE, Gauss, Mathematica, AVS, Iris Explorer, ...

○ Beshinchi avlod, sun'iy intellekt va neyron tarmoqlarida masala echishda foydalanadigan tillar.

Dasturlash tillari inson uchun tushunarli bo'lgan dasturlarni tuzish imkonini beradi. Bunday dasturlarning matnini mashinalarlarda bajarish uchun, ularni mashina kodiga aylantirish zarur. Buning uchun translyator deb ataluvchi maxsus dasturlardan foydalaniladi.

*Translyator* (ingl. Translator — tarjimon) dasturlash tilida yozilgan matnni mashina kodiga aylantiradi (ya'ni, tarjima qiladi).

Translyator oldindan kompyuter xotirasiga yozilgan bo'ladi. Translyatorlar kompilyator va interpretator ko'rinishida bo'lishi mumkin. Kompilyator va interpretatorlar dasturni bajarishda tubdan farq qiladi.

*Kompilyator* (ingl. compiler — yig'uvchi) dasturni to'laligicha o'qiydi va uni tarjima qiladi (mashina tiliga o'giradi).

*Interpretator* (ingl. interpreter — izohlovchi, og'zaki tarjimon) dasturni tarjima qiladi va satrma-satr bajaradi.

Dastur kompilyastiya qilingandan so'ng dasturlash tilida yozilgan dastur matni va kompilyatorga ehtiyoj qolmaydi. Masalan, info.pas fayli kompilyastiya qilingandan so'ng info.exe fayliga o'zgaradi va bu fayl kompilyatorsiz ham ishlashi mumkin. Agar info.pas faylini interpretator qayta ishlaganda har doim dastur ishini bajarilishi uchun interpretator ishga tushishi shart. Shu sababli kompilyator va interpretatorlarning ishlash prinsipi turlichadir.

Kompilyatsiya qilingan dasturlar tezroq ishlaydi, lekin interpretatsiya qilinayotgan dasturlarga o'zgartirish kiritish osonroq.

Har bir dasturlash tili qaysi maqsadda yaratilganligiga qarab kompilyatsiya yoki interpretatsiyaga mo'ljallangan bo'lishi mumkin.

*Bayt-kod* (bytecode) Hisoblash texnikasining va web texnologiyalarning rivojlanishi, aniq bir mashinaga va operatsion tizimga bog'liq bo'lmagan yuqori darajadagi dasturlash tillarining paydo bo'lishi hamda ularning rivojlanishiga olib keldi. Bu dasturlash tillari dastur matnini bayt-kod deb ataluvchi oraliq shaklga translyatsiya qiladi. Bu bayt-kodli dasturni ixtiyoriy joyda va ixtiyoriy vaqtda ishga tushirish mumkin. Masalan, Java va C# singari dasturlash tillari dastur matnini interpretator orqali bayt-kodga o'giradi. Bayt-kod ixtiyoriy platformada tezda mashina kodiga o'girilib ishga tushiriladi.

Dasturlash tili inson va mashinaning muloqot vositasi hisoblanadi va u quyidagi afzalliklarga ega:

- u jonli tilimizga o'xshash va uni o'rganish oson;
- bu tilda yozilgan dastur mashina tilidagidan qisqaroq bo'ladi;
- dastur yozishga kamroq vaqt sarflanadi va kam xatolikka yo'l qo'yiladi;
- yozilgan dasturni ixtiyoriy dasturchi o'qiy oladi;
- dasturlash tili mashina turiga bog'liq emas.

Demak, dasturlash (yuqori darajadagi) tilida dastur tuzish qulayroq, osonroq va buning uchun aynan bir mashina (quyi darajadagi) tilini bilish shart emas ekan.

Bugungi kunda yuzlab dasturlash tillaridan foydalaniladi. Har bir dasturlash tili ma'lum bir soha yoki yo'nalishga oid masalalarni hal qilish uchun mo'ljallangan.

#### **Nazorat savollari:**

1. Masalalarni kompyuterda echish bosqichlari deganda nimani tushunasiz?
2. Algoritm deyilganda nima tushuniladi?
3. Algoritmning qanday xossalari bilasiz?
4. Algoritmni yozishning qanday usullarini bilasiz?
5. Chiziqli algoritmlar deyilganda nima tushuniladi?
6. Tarmoqlanuvchi algoritmlar deyilganda nima tushuniladi?
7. Tarmoqlanuvchi algoritmlarning to'liq shartli konstruktsiyani izohlang.
8. Tarmoqlanuvchi algoritmlarning to'liq bo'lmagan shartli konstruktsiyasi.
9. Takrorlanuvchi algoritmlar deyilganda nima tushuniladi?
10. Takrorlanuvchi algoritmlarning "toki" takrorlanish turini izohlang.
11. Takrorlanuvchi algoritmlarning "gacha" takrorlanish turini izohlang.

## 6 mavzu. C++ algoritmik tili. Kirish

### Reja:

1. Dasturlash.
2. Kompyuter anatomiyasi, mashina kodi va dasturlash tillari, dasturlash muhiti bilan tanishish, ilk dasturni tahlil qilish, xatolar, muammoni yechish, dizayn algoritmi.

**Tayanch iboralar.** Dasturlash, dasturning boshlang`ich va ishchi matnlari, Assembler tili, yuqori darajali dasturlash tili,

### 1. Dasturlash

Algoritmning kompyuter tushunadigan maxsus til alfaviti, belgilari va imlo qoidalari asosida yaratilgan matni dastur deyiladi.

Dastur ikki xil tavsiflanadi. Algoritmning kompyuter tushunadigan maxsus til yordamida yaratilgan matni boshlang`ich dastur, kompyuterga taqdim qilingan boshlang`ich dastur asosida yaratilgan maxsus belgili matn ishchi dastur deyiladi. Kompyuter faqat ishchi dasturni taniydi va u bilan ishlaydi. Biz dasturning boshlang`ich va ishchi matnlarini umumiy holda dastur deb ataymiz.

Dastur matnini yaratish uchun foydalanilgan kompyuter tushunadigan maxsus til dasturlash tili deyiladi.

Kompyuter boshlang`ich dasturi matnini yaratish va uni xatosiz kompyuterga taqdim qilib, ishchi dasturni hosil qilish jarayoni dasturlashtirish deyiladi.

Dasturning boshlang`ich matni yaratilgandan keyin uni ishchi matnini hosil qilish uchun kompyuterga taqdim qilamiz. Agar boshlang`ich matnda dasturlash tili qoidalariga rioya qilinmagan bo`lsa, bu haqda ogohlantirish yoki xatolik sodir bulganligi haqidagi xabarni olamiz. Xabarga asoslangan holda boshlang`ich matnga o`zgartirish kiritib uni yana kompyuterga taqdim qilamiz. Matnda xato bo`lmasa ishchi dastur hosil qilinadi. Ba`zan to`g`ri ishchi dastur hosil qilinadi, ammo u ko`zlangan natijani bermasligi mumkin. Bu, ko`zlangan manzilga borish uchun noto`g`ri yo`nalish ko`rsatilganligi va biz unga amal qilganligimiz tufayli butunlay boshqa manzilga borib qolganligimizga o`xshaydi. Bundan tashqari uzoq va aylanma yo`llar orqali ham belgilangan manzilga boramiz, bu ortiqcha kuch va vaqt sarflanishiga olib keladi. Bu holda ham noto`g`ri ko`rsatma berilgan deyishimiz mumkin. Dasturda ham xuddi shunday holatlar bo`ladi:

- o Kompyuter dastur matnida mavjud bo`lgan imlo xatolarni ko`rsatadi va ularni tuzatganingizdan keyin ishchi dastur hosil qilinadi.

- o Dastur matnida imlo xatolari mavjud emas va ishchi dastur hosil qilinadi. Ammo kerakli natija to`lik olinmaydi, chunki matnda ish jarayoniga tegishli bo`lmagan yoki noto`g`ri bo`lgan ko`rsatmalar berilgan. Ularni kompyuter aniqlay olmaydi va xatoni tuzatish ancha qiyin. Dasturchi xato mavjudligini gumon qilmasligi ham mumkin.

- o Dastur matnida imlo xatolari mavjud emas va ishchi dastur hosil qilinadi hamda kerakli natija to`liq olinadi. Ammo dasturda ortiqcha va ko`p vaqtni oladigan hamda o`z o`rniga mos bo`lmagan ko`rsatmalar berilgan. Ularni kompyuter ham, muallif ham aniqlay olmaydi. Chunki muallif o`z malakasi va bilimi doirasidan kelib chiqqan holda dastur tuzgan. Bu xuddi yaqin manzilga eng qulay yo`nalish orqali emas, balki uzoq va aylanma yo`llar orqali borganga o`xshaydi.

Kompyuter uchun mo`ljallangan dastur qanday tartibda tuzilishi lozimligi haqida bir oz to`xtalib o`tamiz:

- o Dasturning biror bir qadami ish jarayonini tushirib qoldirmang va ularni bajarilish tartibi bo`yicha aniq bayon qiling.

- o Har bir qadamda bajariladigan ishlarni batafsil ko`rsating va ba`zi bir holatlarda sodir bo`lishi mumkin bo`lgan ishlarga e`tiborsizlik qilmang.

- o Dastur matnini imkon darajasida ixchamlashtiring va turli xil holatlarda ko`p marta takrorlanadigan ish jarayonlarini qayta-qayta bayon qilmasdan, ularni umumlashtiring hamda ularga murojaat qilish orqali ishni tashkil qiling.

Kompyuterga qanday ko`rsatma bersangiz u aynan shu tartibda ularni bajaradi. Agar unga avval A va B jarayonni keyin esa D jarayonni bajar deb ko`rsatma bersangiz, u hech qachon

“to`xta-to`xta” men D jarayondan avval S jarayonni bajarishim kerak emas edimi, bu erda qandaydir xatolik bor deb o`ylamaydi.

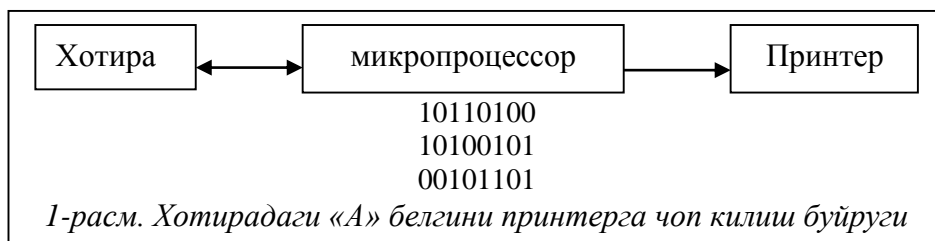
Kompyuterga ko`rsatma bajarilish tartibi bo`yicha aniq va to`g`ri hamda ish jarayonining biror bir qismini tushirib qoldirmasdan yoki ish jarayoniga ta`lukli bo`lmagan qo`shimchalarsiz berilishi shart.

## 2. Kompyuter anatomiyasi, mashina kodi va dasturlash tillari, dasturlash muhiti bilan tanishish, ilk dasturni tahlil qilish, xatolar, muammoni yechish, dizayn algoritmi.

*Kompyuter anatomiyasi, mashina kodi va dasturlash tillari, dasturlash muhiti bilan tanishish va xatolar.* Kompyuter uchun muljallangan dastur, u tushunadigan maxsus tilda yozilishi va taqdim qilinishi kerakligini bilamiz. endi kompyuter tushunadigan maxsus til - dasturlash tili nima ekanligi bilan tanishamiz.

Kompyuterning “ona plata”sida mikroprotsessor mavjud va u kompyuterda sodir bo`layotgan barcha jarayonlarni tashkil qiladi hamda boshqaradi. Agar kompyuterga xabarni ekran yoki printerga chiqarish buyrug`ini bersak mikroprotsessor mos elektr signallarini jo`natadi. U kompyuterga, kerakli xabarni xotiraning qaysi qismidan olish va uni qaerga jo`natish kerakligi haqidagi buyruqni beradi. Mikroprotsessor dasturni yuklaydi va undagi ko`rsatmalarni kelish tartibida ketma-ket bajarilishini ta`minlaydi. Uning ish jarayonini ko`rishning iloji yo`q, ammo shunday bo`lsada jarayonlar qanday bajarilishi haqida qisman bo`lsada tasavvurga ega bo`lish lozim bo`ladi.

Mikroprotsessor ish jarayonini oddiy tarzda tushuntirmoqchi bo`lsak, u bor yo`g`i 4 ta amalni bajarishini aytish kifoya qiladi. Ma`lumotni xotiraga joylashtiradi, xotiraning biror qismidagi ma`lumotni o`zgartiradi yoki uni olib boshqa qismga joylashtiradi, xotiraning aniq biror bir qismida kerakli ma`lumot mavjudligini tekshiradi va ko`rsatmani bajarilish tartibini ta`minlaydi hamda kerak bo`lganda uni o`zgartiradi. (Ko`rsatmada berilgan holatlarning birini tanlaydi.) Barcha jo`natish, ishlov berish, qabul qilish va kuzatish singari harakatlar berilgan elektr signallarining holatiga qarab bajariladi. Kompyuter elektr signallari ikki xil holatda bo`ladi. Yuqori darajali kuchlanish (“Yoqilgan” yoki “Ha” yoki 1) yoki quyi darajali kuchlanish (“O`chirilgan” yoki “Yo`q” yoki 0) signallari mikroprotsessor ish jarayonini tashkil qiladi. Kompyuterda topshiriqni signal shaklida ifodalash uchun 1-“yoqilgan” va 0-“o`chirilgan” sonlari ishlatiladi. Bu ikkilik sanoq tizimi sonlari hisoblanadi va kompyuter ish jarayoni ikkilik sanoq tizimiga asoslangandir. Kompyuterda ixtiyoriy ishni bajarish uchun unga 0 va 1 sonlari kombinatsiyalaridan tashkil topgan signallar to`plami beriladi.



Kompyuterga taqdim qilinayotgan har bir ma`lumot bit deb ataluvchi o`lchovda o`lchanadi: 1 bit = 0 yoki 1 (ikkilik kodlari ham deb ataladi), 1 bayt=8 bit va h-zo.

Boshlang`ich bosqichlarda yaratilgan elektron hisoblash mashinalarida (EHM) dastur 0 va 1 signallari kombinatsiyalari harakatlari bilan bajarilar va biror bir oddiy muammoni hal qilish uchun ham minglab signallarni mutanosib tarzda joylashtirish talab qilinardi. Buning uchun ko`p kuch va vaqt sarflanadi. Ularda mikroprotsessor bo`lmaganligi uchun dasturni ishga tushuruvchi texnik, EHM dastaklarni bir holatdan boshqasiga o`tkazuvchi nazoratchi sifatida faoliyat ko`rsatar edi.

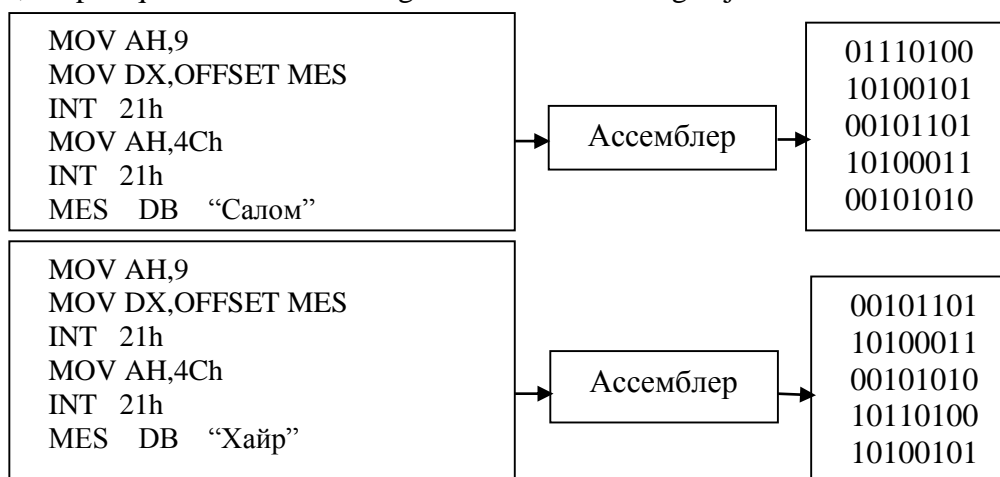
Elektron texnikalarining rivojlanishi natijasida dasturni avvaldan EHMga yuklab, keyin esa unda ko`rsatilgan buyruqlarni bajarishga majburlash vujudga keldi. Baribir eng avvalo minglab 0 va 1 signallarini o`z o`rniga qo`yib, kerakli dastur matnini hosil qilish, keyin esa uni yuklash talab qilinardi. Keyinchalik Assembler tili yaratildi.



Assembler tili yaratilgandan keyin 0 va 1 signallar o`rniga shartli kodlardan foydalanish holda dastur tuzish vujudga keldi. Shartli kodlar bevosita mikroprotsessorga mo`ljallangan bo`lib eslab qolish murakkab bo`lmagan mnemonik kodlardan tashkil topadi. Masalan MOV kodi ma`lumotni xotiraning bir qismidan boshqa qismiga o`tkazish, JMP kodi esa xotiraning boshqa qismiga o`tish singari buyruqlarni bajaradi.

Biror bir masalani echish algoritmi dasturlash tili shartli kodlari orqali ifodalanadi va u boshlangich matn hisoblanadi. Ushbu boshlangich matn maxsus dastur yordamida mashina kodiga o`giriladi. Chunki, kompyuter faqat 0 va 1 belgili signallardan tashkil topgan kodni tushunganligi uchun Assembler tilida yozilgan dastur kodini kompyuter kodiga tarjima qiluvchi maxsus oraliq buyruqlar to`plami ham yaratildi. U translyator deb ataladi. Dasturning boshlangich matnini yozish va uni mashina tiliga tarjima qilishni amalga oshiruvchi maxsus dasturlar majmuasi dasturlash tilining dasturlash muhiti deyiladi. Har bir dasturlash tilining dasturlash muhiti o`z matn muharriri va translyatoriga ega. Har bir dasturlash tilining translyatori aynan shu dasturlash tili kodlarini kompyuterning 0 va 1 bilan ifodalangan kodlariga o`tkazib beradi. Assembler translyatori aynan Assembler tili kodlarini kompyuterning 0 va 1 bilan ifodalangan kodlariga o`tkazib beradi.

Assembler tilida yozilgan dastur juda katta tezlikda ishlaydi, ammo bu tilda dasturlash ancha murakkab, ko`p vaqtni oladi va boshlang`ich dastur matnining hajmi ham katta bo`ladi.



2-расм. Ассемблер тили транслятори Ассемблер тилида ёзилган дастур матнини иккилик кодга таржима қилиб утказади.

Hozirgi kunda dasturchilar yuqori darajali dasturlash tillarida ishlashadi va unda ko`rsatmalar inson so`zlashuv tiliga yaqin bo`lgan so`zlar yordamida beriladi. Ularda mnemonik kod yoki 0 va 1 signallari ishlatilmaydi. Masalan, qandaydir so`zni kompyuter ekraniga chiqarish kerak bo`lsin.

```

С тилида:
puts("Салом");
Ассемблер тилида:
MOV AH,9
MOV DX,OFFSET NAME
INT 21h
MOV AH,4Ch
INT 21h
NAME DB "Салом"
Иккилик кодда:
1011010000001001101110100000000100001011
1100110100100001101101000100110011001101
0010000101010011010110010100001001000101
01011000

```

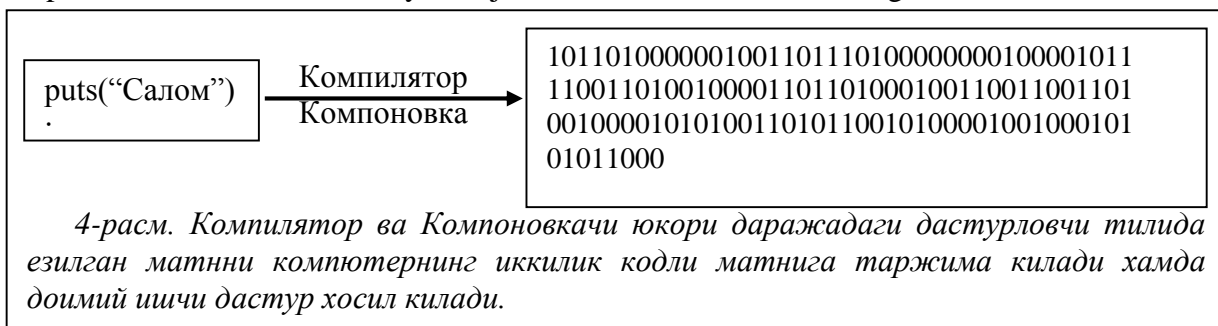
3-расм. Бир хил вазифани бажариши учун ёзилган С тили, Ассемблер тили ва иккилик код матнларини солиштириши.

Bir xil vazifani bajarish uchun yozilgan C++ tili, Assembler tili va ikkilik kod matnlarining murakkablik darajasi hamda hajmini solishtirishsak qaysi biri sodda ekanligini ko`rish mumkin. C++ tilida yozilgan dastur matnini kompyuter to`g`ridan to`g`ri tushunmaydi. Uni ham kompyuter kodiga tarjima qiluvchi maxsus oraliq buyruqlar to`plami ham yaratilgan. Ular aynan dastur matnini kompyuterning 0 va 1 bilan ifodalangan kodlariga o`tkazib beradi. Inson tiliga yaqin tilda yozilgan dastur matni ikkilik kodga translyator yodamida uch xil usuldan biri orqali o`tkaziladi. Ular kompilyator, interpretator va bayt kod deb ataladi.

*Kompilyatorlar.* Kompilyator boshlang`ich dastur matnini birdaniga ikkilik kodga tarjima qiladi va hosil bo`lgan ishchi dastur matnini diska yozib qo`yadi. Hosil bo`lgan dasturni ixtiyoriy vaqtda ishga tushirish mumkin.

Kompilyator ish jarayonini hayotiy misolda tushuntiramiz. Faraz qilaylik siz o`zbek tilida ma`ruza yozdingiz va uni biror bir xalqaro konferentsiyada ingliz tilida o`qishingizni so`rashdi. Siz tarjimonga murojaat qilasiz va u o`zbek tilidagi matnni ingliz tiliga tarjima qilib beradi. Agar matnda gramatik xato mavjud bo`lsa, tarjimon bu haqda sizga xabar beradi va xatoni tuzatib berasiz. Xatolar tuzatilgandan so`ng tarjimon uni to`liq tarjima qilib beradi va sizda ma`ruzaning ingliz tilidagi matni paydo bo`ladi. Tarjima qilingan matnni bir necha nusxada chop qilib konferentsiya hay`atiga taqdim qilasiz va konferentsiya ishtirokchilari undan bimalol foydalanishlari mumkin. Ushbu ma`ruzangizni bir yildan keyin boshqa bir xalqaro konferentsiyada yana ingliz tilida o`qishingizni so`rashdi. Sizda tayyor tarjima mavjud va uning nusxasini ko`paytirib taqdim qilasiz.

Kompilyator ham kompyuter dasturi hisoblanadi va u sizning boshlang`ich dasturingiz matnini kompyuter bajaruvchi ikkilik kodlarga (komponovkachi deb ataluvchi dastur yordamida) tarjima qilib beradi. Kompilyator eng avvalo dastur matni dasturlovchi tilning imlo qoidalariga rioya qilingan holda yozilganligini tekshiradi. Agar kompilyator, matnda dasturlovchi tilning imlo qoidalariga rioya qilinmaganlikka yoki notanish bo`lgan biror bir muammoga duch kelsa, bu haqda xabar beradi va ishni to`xtatadi. Xatolar tuzatilgandan keyin yana matn kompilyatorga taqdim qilinadi va biror bir xato bo`lmasa u ob`ekt fayl deb nomlanuvchi oraliq shakl tuzadi. Komponovkachi dastur ob`ekt faylni bajariluvchi ishchi dastur ko`rinishiga o`tkazadi.



muharriri yordamida dasturning boshlang`ich matni yoziladi. Dasturning boshlang`ich matnini o`qish, o`zgartirish va chop qilish mumkin. Dasturning boshlang`ich matni to`g`ri yozilganligiga ishonch hosil qilgandan so`ng kompilyatsiya yordamida ob`ekt fayl hosil qilinadi. Xatosiz ob`ekt fayldan komponovkachi dastur yordamida ishchi dastur yaratiladi va u mashina tiliga tarjima qilingan oxirgi natija hisoblandi. Boshlang`ich dastur matni muallifda saqlanadi, ishchi dastur esa foydalanuvchilarga taqdim qilinadi. Agar dasturga biror bir o`zgarish kiritish kerak bo`lsa, boshlang`ich dastur matni o`zgartiriladi va undan yana ishchi dastur hosil qilinadi. Bu endi dasturning boshqa versiyasi hisoblanadi.

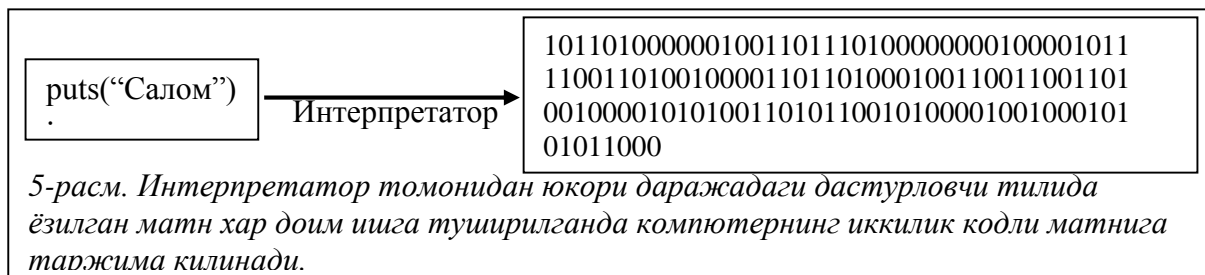
Delphi, C, C++ singari dasturlovchi tillar kompilyatsiya qilinuvchi tillar hisoblanadi. Har bir dasturlovchi tilning o`z kompilyatori mavjud va u faqat shu til imlo qoidalarini biladi hamda tushunadi.

*Interpretator.* Interpretator – boshlang`ich dastur matni ko`rsatmalarining barchasini kompyuter kodiga uni bajarish vaqtida tarjima qiladi.

Xalqaro konferentsiya uchun o`zbek tilida yozgan ma`ruza misoliga qaytamiz. Sizdan uni ingliz tilida o`qishingizni so`rashdi. Siz ma`ruzani konferentsiya ishtirokchilariga etkazish uchun

sinxron tarjimon yollaysiz. Tarjimon sizning o`zbek tilida o`qigan so`zingizni ingliz tiliga tarjima qilgandan keyin keyingi so`zni o`qiysiz va yana tarjima qilinishini kutasiz. Ma`ruza matni o`zbek tilida saqlanadi, ingliz tilidagi matni esa tarjimon xotirasida saqlanadi. Ushbu ma`ruzangizni bir yildan keyin boshqa bir xalqaro konferentsiyada yana ingliz tilida o`qishingizni so`rashdi. endi siz yana sinxron tarjimon yollaysiz.

Interpretator ham xuddi shu tarzda ishlaydi va u orqali ishlov beriluvchi dastur faqat boshlang`ich matn holatida saqlanadi.



Basic, FoxPro singari dasturlovchi tillar interpretatsiya qilinuvchi tillar hisoblanadi. Dasturlovchi tilning interpretatori faqat shu til imlo qoidalarini biladi hamda tushunadi.

Kompilyatorli tillar interpretatorli tillarga nisbatan ancha tez ishlaydi. Chunki interpretatorli tillarda eng avvalo interpretator dasturi kompyuter xotirasiga yuklanadi, keyin esa boshlang`ich dastur matni kompyuter kodiga o`girilgandan keyin ishga tushiriladi. Kompilyatorli tillarda esa avvaldan ikkili kodli ishchi dastur yaratiladi va u to`g`ridan to`g`ri ikkili kodli holatida ishga tushiriladi. Bundan tashqari kompilyatorli tillar ishchi dasturini ishga tushirish uchun barcha kompyuterda til kompilyatori bo`lishi shart emas. Ishchi dastur kompilyatorsiz o`zi mustaqil ishlay oladi. Interpretatorli tillar uchun esa dastur ishga tushiriluvchi kompyuterda til interpretatori bo`lishi talab qilinadi, aks holda dasturni ishlatib bo`lmaydi.

**C oilasi tillari tarixi.** C tili 1972 yilda Denis M. Ritchi tomonidan yaratildi. Uning imkoniyatlari yuqori bo`lganligi uchun turli kompaniyalar turli dasturlash muhitlarini ishlab chiqdilar va o`z qo`shimchalarini qo`shdilar. 1983 yilda ANSI (Amerika milliy standartlar instituti) tomonidan barcha dasturlash muhitlarini tahlil qilgan holda yagona standart ishlab chiqdi va barcha muhitlarning ushbu standartga rioya qilishi talab qilindi. Dasturlash muhitlarini yaratuvchilarga standart talablarining barchasini o`z tarkibida mujassamlashtirgan holda umumiy siyosatga zid bo`lmagan qo`shimchalar qo`shishiga ruxsat beriladi. Hozirgi kunda ancha ommabop kompilyator yaratuvchilar Borland, MicroSoft hisoblanishadi. Agar dasturning boshlang`ich matni standartga rioya qilingan holda yozilgan bo`lsa, u barcha muhitlar kompilyatorlarida kompilyatsiya qilinadi.

C++ tili 1980 yilda B`yarn Straustrup tomonidan yaratilgan va u C tilining ancha kengaytirilgan, takomillashtirilgan hamda ob`ektga yo`naltirilgan shaklidir. Straustrup C++ tilini yaratishdan avval o`z oldiga ikki maqsadni qo`ydi. Birinchidan C++ tilida C tili standartining barcha imkoniyatlarini saqlab qolgan holda qo`shimchalar qo`shish, ikkinchidan tilga ob`ektga yo`naltirilgan dasturlashning tuzilmalarini kiritish. U maqsadiga erishdi va C++ tili ommalashdi. C++ tilining ham yuqori tezlikka ega ekanligi, ko`chiruvchanligi va tuzilmaliligi sababli ko`pchilik operatsion tizimlar, kompilyatorlar va kommunikatsiya tizimi dasturlari ushbu tilda tuzilgan hamda tuzilmoqda.

C# tili 1998 yilda MicroSoft kompaniyasi tomonidan yaratilgan va u C++ tilining ancha kengaytirilgan, takomillashtirilgan hamda platformalarga bog`liq bo`lmagan shaklidir.

#### **C++ tilining alfaviti va xizmatchi so`zlari.**

C++ dasturlash tilining alifbosi deb, shu tilda ma`lumotlarni ifodalash va dastur ishlab chiqish jarayonida kompilyator tomonidan qabul qilishga ruxsat berilgan belgilar yoki maxsus belgilardan iborat zanjirlar to`plamiga aytiladi. Bu alifbo ASCII (xalqaro belgilar va ularning kodlari) jadvalining hamma belgilarini, ya`ni quyidagilarni o`z ichiga oladi:

- 1) lotin alifbosining katta va kichik harflari (A,B,...,Z,a,b,...,z);
- 2) 0 dan 9 gacha arab raqamlari;

3) tagiga chizish belgisi ( \_ );  
4) bo'sh joy belgisi;  
5) boshqaruvchi belgilar: ASCII jadvalidagi (1-ilovaga qarang) kodlari 0 dan 31 gacha bo'lgan belgilar. Bu belgilar satr va konstantalami ifodalashda qo'llanishi mumkin;

6) turli ko'rsatmalami yozish uchun ishlatiladigan maxsus belgilar:

+ - \* / = { } ( ) [ ] . , @ # ' ; ; \$ % A ! ? & \ < >

7) asosiy bo'lmagan belgilar (ASCII ni kengaytiruvchi, ya'ni kodi 128 dan 255 gacha bo'lgan belgilar; rus alifbosining katta va kichik harflari, psevdografika elementlari shu sinfga kiradi. Bu belgilar turli konstantalami hosil qilish, matnlami yozish, izohlami tashkil qilishda qo'llanishi mumkin);

8) murakkab belgilar: < = > = : = . . \* \* ;

9) xizmatchi so'zlar. Ular C++ dasturlash tilida ma'lum bir ma'no yoki ko'rsatmani anglatuvchi maxsus belgilar zanjiridan iborat bo'lib, bu zanjimi o'zgartirish yoki qisqartirib qo'llash mumkin emas. Shuningdek, ularni o'zgaruvchilar nomlari sifatida ishlatish mumkin emas.

C tilida quyidagi xizmatchi so'zlar mavjud:

- o Tiplar nomlari: char, short, int, unsigned, long, float, double.
- o Operatorlar nomlari: if, else, switch, case, while, do, for, default, break, continue, goto.
- o Xotira turlari: auto, register, static, extern.
- o Tiplar bilan ishlash: typedef, sizeof.
- o Struktura: struct, union.
- o Chiqish: return, entry.

C++ tilining xizmatchi so'zlari: asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed, sizeof, static, struct, swith, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.

10) ko'rinmaydigan belgilar ("umumlashgan bo'shliq belgilari"). Leksemalarni o'zaro ajratish uchun ishlatiladigan belgilar (misol uchun bo'shliq, tabulyatsiya, yangi qatorga o'tish belgilari).

Eslatma. Agar dastur tarkibida yuqorida sanab o'tilgan belgilardan boshqa belgi yoki xizmatchi bo'lmagan so'zlar uchrab qolsa, bu haqidagi maxsus axborot kompyuter ekraniga chiqariladi.

So'z - bo'sh belgiga ega bo'lmagan belgilar ketma-ketligidir. Xizmatchi so'z esa C++ tilida ma'lum bir buyruqni amalga oshirish ko'rsatmasini beruvchi so'zdir.

C++ tilida dasturni tavsiflash uchun izohlar ishlatiladi. Izohlarda, satrlarda va belgili konstantalarda boshqa literallar, masalan rus harflari ishlatilishi mumkin.

C++ tilida olti xil turdagi leksemalar ishlatiladi: erkin tanlanadigan va ishlatiladigan identifikatorlar, xizmatchi so'zlar, konstantalar (konstanta satrlar), amallar (amallar belgilari) va ajratuvchi belgilar.

*Identifikator.* Identifikatorlar lotin harflari, ostki chiziq belgisi va sonlar ketma-ketligidan iborat bo'ladi. Identifikator lotin harfidan yoki ostki chiziq belgisidan boshlanishi lozim.

Misol uchun:

A1, \_MAX, adress\_01, RIM, rim

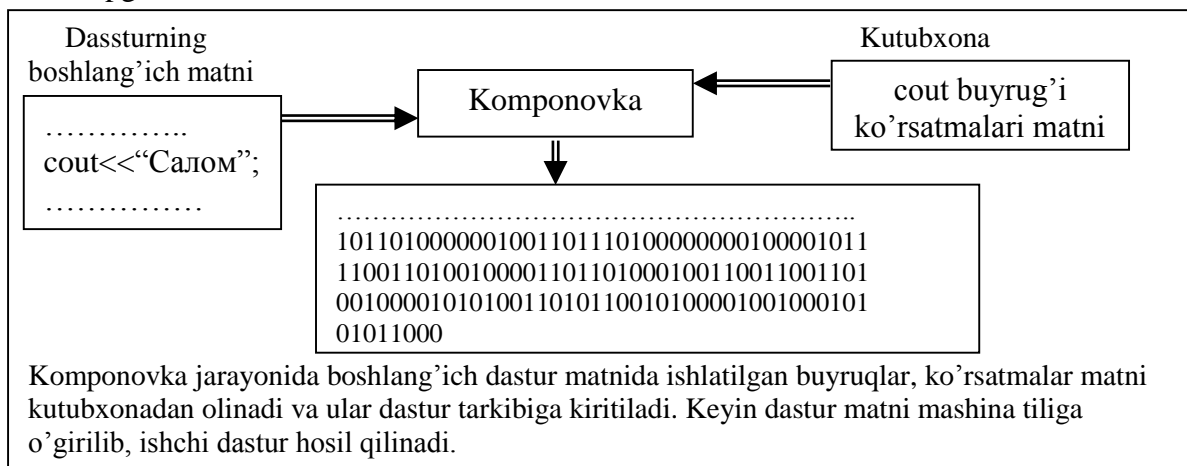
Katta va kichik harflar farqlanadi, shuning uchun oxirgi ikki identifikator bir-biridan farq qiladi.

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 harfi, ba'zi kompilyatorlarda 8 ta harfni inobatga oladi. Bu holda NUMBER\_OF\_TEST va NUMBER\_OF\_ROOM identifikatorlari bir biridan farq qilmaydi.

**Funksiyalar kutubxonasi.** C va C++ tillari kam sonli amallardan tashkil topgan. Bu tillarda, boshqa dasturlovchi tillardan farqli ravishda o'zining aniqlangan kiritish, chiqarish va satr (sotr-so'z yoki jumlani tashkil qilgan belgilar ketma-ketligi) bilan ishlash vositasi mavjud emas.

Masalan, C tilining boshlang'ich standarti bor-yo'g'i 27 kalit so'z yoki buyruqdan tashkil topgan edi.

C va C++ tillarining asosiy qudrati uning funksiyalaridadir va ular kompilyator bilan birga funksiyalar kutubxonasi tarzida taqdim qilinadi. Funktsiya maxsus masalalarni hal qilish uchun ishlatiladigan ketma-ket tartibda berilgan ko'rsatmalar to'plamidir. Kutubxona-kompilyatorga birlashtirilgan alohida fayl bo'lib, u turli xil muammolarni hal qiluvchi funksiyalar to'plamidan tashkil topgandir.

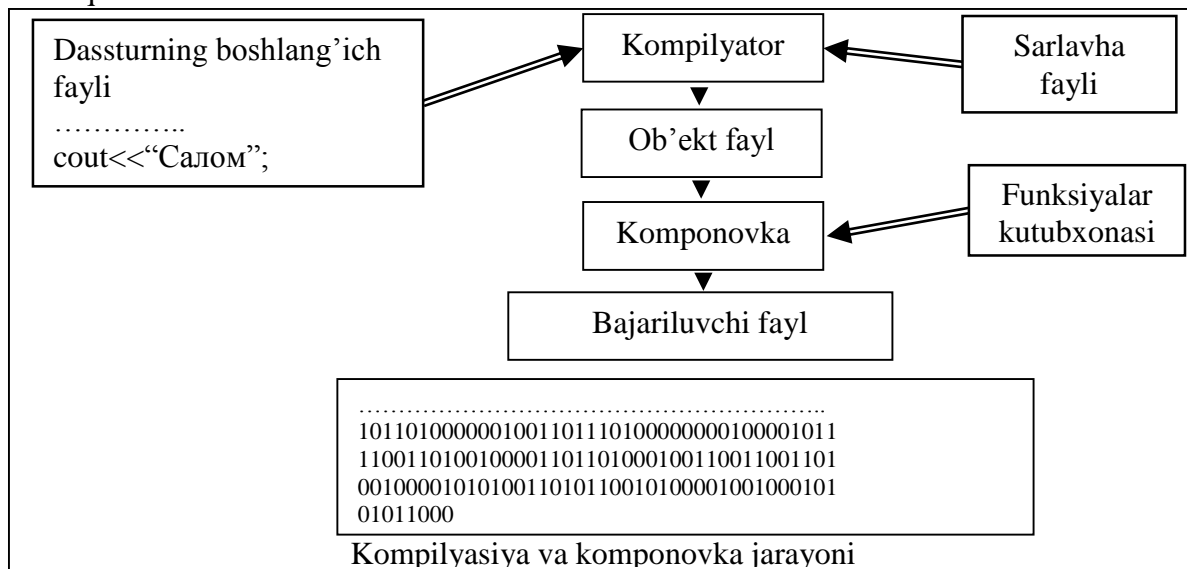


Masalan, ma'lumotni ekranga chiqarish uchun bir necha funktsiya mavjud bo'lib, ularning muvofiqlig'idan o'z o'rnida foydalanish orqali kerakli ish bajariladi. Har bir funktsiya o'z nomiga ega. Biror bir ishni bajarish uchun uni amalga oshiruvchi funktsiyaning nomini sintaksisiga muvofiq yozish kifoya qiladi.

Ba'zi bir funksiyalar avvaldan kompilyatsiya qilingan kodga ega. Agar kompilyator dastur matnida bunday funktsiyaga duch kelsa, uni ikkilik kodga o'girish bilan shug'ullanmaydi. Chunki, funktsiya matni ikkilik kodga o'girilgan bo'ladi. Komponovka jarayonida funktsiyaning ikkilik kodi kutubxonadan olinib dastur matniga birlashtiriladi.

Funktsiya avvaldan kompilyatsiya qilinganligi uchun juda ham unumli kodlashtirilgan hamda u to'liq optimallashtirilgan bo'ladi.

Kompilyator bilan birgalikda juda ko'p ishlatiladigan funksiyalarning boshlang'ich matni ham taqdim qilinadi. Ular *sarlavha fayllari* deb ataluvchi fayllarda saqlanadi va odatda \*.h kengaytmaga ega bo'lishadi. Sarlavha fayllari tarkibida kompilyator direktivalari va biror bir aniqlanishni ishlatish qo'llanmalari ham mavjud. Kompilyatsiya jarayonida ishlatilgan sarlavha fayli matni ham sizning xususiy matningiz tarkibiga kiritiladi va birgalikda umumiy ob'ekt fayl hosil qilinadi.



Sarlavha fayllari kutubxona fayllaridan farqli ravishda kompilyatsiya qilinmagan bo`ladi. Ularni ham xuddi boshlang`ich dasturingiz matni singari o`qish, o`zgartirish va chop qilish mumkin. Ammo, kompilyator bilan birga taqdim qilingan sarlavha fayllarini o`zgartirishdan saqlanishingizni maslahat beramiz, chunki ular barcha imkoniyatlarni hisobga olib tuzilgan va biror xatoga yo`l qo`ysangiz kompilyator hech qachon ular ob`ekt kodini generatsiya qila olmaydi.

Kompilyator bilan birga taqdim qilinayotgan kutubxonaga qo`shimcha, maxsus ma`lumotlar bazasini qurish, grafik, kommunikatsiya singari har xil kutubxonalarini ham olishingiz mumkin. Kutubxona qanchalik boy bo`lsa ishingiz unumli va tez bo`ladi. Bundan tashqari dasturlaringizda juda ko`p ishlatilayotgan ish jarayonlarini umumlashtirib uz xususiy kutubxonangizni ham tashkil qilishingiz mumkin. Masalan, ko`pchilik dasturlaringizda A va B jarayonni bajarishni keyin esa C jarayonni bajarish kerakligini ko`p marta takrorlaysiz. Ular ish jarayonini tavsiflovchi funksiyalarni yaratib ularni xususiy kutubxonangizga joylashtirasiz. Ular oldindan yozilgan va kompilyatsiya qilinganligi uchun har doim ehtiyoj tugilganda dasturlaringizda ulardan tez foydalana olasiz.

Til imlo qoidalariga rioya qilgan va kompilyator kutubxonasidagi funksiyalardan foydalangan holda yozilgan dasturlar oson ko`chiriluvchan bo`ladi. Chunki, har bir platforma kompilyatori standart talablarni qanoatlantiradi va yaratilgan ishchi dastur ushbu platformada xatosiz ishlaydi.

**Dasturning tuzilishi.** C++ tilida barcha dasturlar main() nomli funktsiya bilan boshlanishi lozim. U quyidagicha yoziladi. main() funktsiya bo`lgani uchun qiymat qaytarishi lozim. Shu sababli ko`pchilik C++ dasturlash muhitlarida *int main()* tipli sifatida (ya`ni butun sonli qiymat qaytaruvchi) boshlanadi. Bu holda dastur oxirida return 0 (yoki ixtiyoriy son) buyrug`i bo`lishi lozim. Ba`zi hollarda *void main()* tipsiz sifatida (ya`ni qiymat qaytarmaydigan) boshlanadi. Bu holda dastur oxirida return 0 (yoki ixtiyoriy son) buyrug`i bo`lmaydi.

Qavslar funktsiya nomi qismi bo`lib, aynan u kompilyatorga ingliz tilidagi oddiy main emas, balki funktsiya ishlatilayotganligini bildiradi. Qavslar har bir funktsiyaning ajralmas qismi bo`lib, ko`pchilik hollarda qavs ichida funktsiya parametrlari beriladi.

main() dan keyin ko`rsatmalar beriladi. Ko`rsatmalar standart buyruq va kutubxona tarkibidagi standart funktsiya yoki o`zimiz yozgan xususiy funktsiya shaklida beriladi. Birinchi ko`rsatmadan oldin ochuvchi figurali qavs {}, oxirgi ko`rsatmadan keyin esa yopuvchi figurali qavs {} qo`yiladi. Ochuvchi va yopuvchi figurali qavslar orasidagi funktsiyani tashkil qiluvchi ko`rsatmalar ketma-ketligi, ko`pincha funktsiya tanasi deyiladi. Shunday qilib, C++ tilida yozilgan eng oddiy dasturning tuzilishi quyidagicha (Biz tipli, int main() konstruksiyasini qo`llaymiz):

```
int main() //dasturning boshlanishini bildiruvchi funktsiya. Kirish nuqtasi.
{
    //funktsiya shu yerdan boshlanadi.
    .....;
    .....; //kompyuter bajarishi kerak bo`lgan ko`rsatmalar
    .....; //shu erda joylashtiriladi.
    return 0; //funktsiya qiymat qaytaradi.
} //funktsiya shu erda tugaydi.
```

Ochuvchi va yopuvchi figurali qavslar cheklovchi deb ataladi hamda ular kodning ma`lum bir qismini yagona blok sifatida ajratish uchun xizmat qiladi. Funktsiya ochuvchi figurali qavs bilan boshlanib, yopuvchi figurali qavs bilan yakunlanishi lozim. Bundan tashqari funktsiya ichida alohida bloklar xususiy figurali qavslar bilan ajratilishi va ular bir nechta bo`lishi mumkin. Har bir ochuvchi figurali qavsning o`z yopuvchi figurali qavsi mavjud.

```
int main()
{
    .....;
    {
        .....;
        {
            .....;
        }
    }
}
```



Bu holda kompilyator sarlavha faylini faqat joriy katalogdan izlaydi, agar uni topa olmasa xatolik mavjudligi xabarini beradi. Shuning uchun kompilyator hujjati bilan tanishib, qaysi holat o`rinli ekanligini aniqlang.

```
using namespace std;
```

ko`rsatmasi Dev C++ muhitida qo`yilishi lozim, Borland C++ muhitida bu ko`rsatma berilmaydi.

Dasturning asosiy kodi main() funksiyasini chaqirish bilan boshlanadi. C++ dasturida albatta main() funksiyasi bo`lishi shart va dastur shu funksiyani bajarish bilan o`z ishini boshlaydi. Funksiya bu bir yoki bir necha amalni bajaruvchi dastur blokidir. Odatda funksiyalar boshqa funksiyalar orqali chaqiriladi, lekin main() funksiyasi alohida xususiyatga ega bo`lib u dastur ishga tushirilishi bilan avtomatik tarzda chaqiriladi.

main() funksiyasini boshqa funksiyalar kabi qaytaradigan qiymati tipini e`lon qilish lozim. SALOM.cpp dasturida main() funksiyasi int (integer – butun so`zidan olingan) tipli qiymat qaytaradi, ya`ni bu funksiya ishini tugatgandan so`ng operatsion sistemaga butun sonli qiymat qaytaradi. Operatsion sistemaga qiymat qaytarish unchalik muhim emas, umuman sistema bu qiymatdan foydalanmaydi, lekin C++ tili standarti main() funksiyasi barcha qoidalarga muvofiq e`lon qilinishini talab qiladi.

Eslatma: Ayrim kompilyatorlar main() funksiyasini void tipidagi qiymat qaytaradigan qilib e`lon qilish imkonini beradi. C++ da bundan foydalanmaslik kerak, chunki hozirda bunday uslub eskirgan. main() funksiyasini int tipini qaytaradigan qilib aniqlash lozim va buning hisobiga funksiyaning oxiriga return 0 ifodasi yoziladi.

Barcha funksiyalar ochiluvchi figurali qavs ({} bilan boshlanadi va {}) yopiluvchi qavs bilan tugaydi. main() funksiyasi figurali qavsida 3 – satrdan 6 – satrgacha joylashtirilgan. Figurali qavslarni ichida joylashgan barcha satrlar funksiya tanasi deb aytiladi.

Bizning oddiy dasturimizning barcha funktsionalligi 4–satrdan keltirilgan. cout ob`ekti ekranga ma`lumotni chiqarish uchun qo`llaniladi. cin va cout ob`ektlari mos ravishda ma`lumotlarni kiritish (masalan, klaviatura orqali) va ularni chiqarish (ekranga chiqarish) uchun qo`llaniladi. main() funksiyasi 6 – satr bilan tugallanadi.

Dastur tanasida konsol rejimi (Consol – rejimi bu MS DOS oynasi ko`rinishiga o`xshash oyna bo`lib, unda foydalanuvchi dastur tuzuishda faqat dastur kodlari bilan ishlaydi. Graphic interface – rejimida esa faqat tilning kodlari bilangina emas muhitning menyulari, komponentalari bilan ham ishlashi mumkin bo`ladi) da belgilar ketma-ketligini oqimga chiqarish amali qo`llanilgan.

Ekranga ma`lumotni chiqarish uchun cout so`zini, undan so`ng chiqarish operatorini (oqimini) (<<) kiritish lozim. C++ kompilyatori (<<) belgisini bitta operator deb qaraydi.

2 – misol. cout ob`ektini qo`llanilishi

```
//cout ob`ektini qo`llanilishi
```

```
# include <iostream>
```

```
using namespace std; //Dev C++ muhitida qo`yilishi lozim
```

```
int main()
```

```
{ cout << "Bu son 5 ga teng:" << 5 << "\n";  
  cout << "endl operatori ekranda yangi  
  cout << " satrga o`tish amalini bajaradi";  
  cout << endl;  
  cout << "Bu katta son:\t" << 70000 << endl;  
  cout << "Bu 5 va 8 sonlarining yig`indisi:  
  << \t" << 8+5 << endl;  
  cout << "Bu kasr son:\t\t" << (float) 5\8  
  << endl;  
  cout << "Bu esa juda katta son: \t";  
  cout << (double) 7000*7000 << endl;  
  return 0; }
```



NATIJA:

Bu son 5 ga teng: 5  
endl operatori ekranda yangi satrga o`tish amalini bajaradi  
Bu katta son: 70000  
Bu 5 va 8 sonlarining yig`indisi: 13  
Bu kasr son: 0.625  
Bu esa juda katta son: 4.9e+07

Eslatma: endl operatori end line (catr oxiri) degan so`zdan olingan bo`lib «end-el» deb o`qiladi.

**Izohlar.** Siz dastur yozayotgan vaqtingizda nima ish qilmoqchi ekanligingiz doimo aniq bo`ladi. Lekin bir oydan so`ng bu dasturga qaytish lozim bo`lsa dasturga tegishli detallar va ularning vazifalari nimadan iborat ekanligini bilmasligingiz mumkin.

Dasturni butunlay xotirangizdan o`chirib yubormaslik va boshqalarga ham tushunarli bo`lishi uchun izohlardan foydalanish lozim. Izohlar kompilyator tomonidan tushirib qoldiriladigan dasturning alohida satrida yoki butun bir blokida qo`llaniladi.

```
/* va */ belgilar orasidagi har qanday ifoda izoh hisoblanadi.  
// belgidan keyin qatordagi har qanday ifoda izoh hisoblanadi.  
// C++ tilida yozilgan eng oddiy ko`rinishdagi dastur tuzilmasi  
# include <iostream> /* direktiva orqali chaqirilgan sarlavha fayli*/  
using namespace std; //Dev C++ muhitida qo`yilishi lozim  
int main() /* dasturga kirish nuqtasini ko`rsatuvchi funktsiya*/  
{ /* blokning boshlanishini qayd qiluvchi figurali qavs*/  
    cout << "Salom"; /* natijani ekranga chiqarish operatori*/  
    /*va ular oxirini ko`rsatuvchi nuqta vergul belgisi*/  
    return 0; //funktsiya qiymat qaytaradi.  
} // blokning tugashini qayd qilish
```

Ushbu dastur C++ tili qoidalariga rioya qilingan holda yozilgan.

Dastur matnidagi ishlatilgan nuqta vergul ko`rsatma oxirini qayd qiladi va u ajratgich deb ataladi. Ajratgich kompilyatorga ko`rsatma tugaganligini va undan keyin boshqa ko`rsatma berilishi yoki dastur tugashi kerakligini bildiradi. Har bir alohida ko`rsatma oxiriga albatta nuqta vergul qo`yilishi lozim. Har bir qator oxiriga emas, balki alohida tugallangan ko`rsatma oxiriga nuqta vergul qo`yiladi. Ba`zan ko`rsatma bir qatorga sig`maydi va ikkinchi qatordan davom ettiriladi. Bu holda ko`rsatma tugagandan keyin bir marta nuqta vergul qo`yiladi.

3– misol.

```
# include <iostream>  
using namespace std;  
int main()  
{ cout << "Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik  
nohiyada istiqomat qilaman.");  
    cout << "Dastur tuzishni o`rganyapman va bu mening birinch dasturim.");  
    return 0;}
```

Dastur ishga tushirilsa ekranga quyidagi ikki qatordan iborat bo`lgan ma`lumot chiqadi.

Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik nohiyada istiqomat qilaman.

Dastur tuzishni urganyapman va bu mening birinch dasturim.

C++ tili erkin formatli til hisoblanadi. Shuning uchun dastur qatori qaerdan boshlanishi va cheklagichlar qaerga qo`yilishi ahamiyatga ega emas. Yuqoridagi misolni bir necha xil usulda yozsak ham natija bir xilda bo`laveradi.

4– misol.

```
# include <iostream>  
using namespace std; //Dev C++ muhitida qo`yilishi lozim
```

```
int main() { cout <<"Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik nohiyada istiqomat qilaman."); cout << "Dactur tuzishni urganyapman va bu mening birinch dasturim."); return 0;}
```

yoki

```
#include <iostream>
```

```
using namespace std; //Dev C++ muhitida qo'yilishi lozim
```

```
int main()
```

```
{ cout <<"Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik nohiyada istiqomat qilaman."); cout << "Dactur tuzishni urganyapman va bu mening birinch dasturim.");
```

```
return 0;}
```

yoki

```
#include <iostream>
```

```
using namespace std; //Dev C++ muhitida qo'yilishi lozim
```

```
int main()
```

```
{
```

```
cout <<"Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik nohiyada istiqomat qilaman.");
```

```
cout << "Dactur tuzishni o'rganyapman va bu mening birinch dasturim.");
```

```
return 0;
```

```
}
```

Dastur yozishda har bir dasturchining o'z uslubi mavjud. Ammo shunday bo'lsada har bir dasturchi ma'lum bir tartibga rioya qilgan holda, dasturning o'qilishini osonlashtirish maqsadida, probellardan unumli foydalanishi lozim. Chunki, kompilyatsiya jarayonida dastur matnidagi probellar olib tashlanadi.

Quyidagi matn yuqorida keltirilgan matnlarga nisbatan ancha o'qimishli hisoblanadi.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
cout <<"Salom, bu men Sohibov Boburman. Buxoro shahar 5 kichik nohiyada istiqomat qilaman.");
```

```
cout << "Dactur tuzishni urganyapman va bu mening birinch dasturim.");
```

```
return 0;
```

```
}
```

Dastur matni o'qimishli bo'lishi uchun bir qancha tavsiyalarni berib o'tamiz va ularga rioya qilsangiz maqsadga muvofiq bo'lar edi.

- o Doimo main() funksiyasini alohida qatorga yozing.

- o Doimo ochuvchi va yopuvchi figurali qavslarni alohida qatorga yozing.

- o Dastur matnini tabulyatsiyadan foydalangan holda yozing. Agar matn hajmi katta bo'lsa chekinishlar yordamida dastur tuzilmasini tushunarli tarzda yozish va mantiqiy birliklarini ajratish natijasida uni ancha o'qimishli qilishingiz mumkin.

Bu talablarga rioya qilishingiz natijasida sizning dasturingiz matni boshqa dasturchilar uchun ham o'qimishli bo'ladi. Dasturda ortiqcha yoki kam sondagi probellarning bo'lishi kompilyator uchun ahamiyatga ega emas. Ammo qavslar va nuqta vergullarning tushirib qoldirilishi xatolik hisoblanadi. Bu ko'rinishdagi xatoliklar sintaksis xatoliklari deb ataladi. Sintaksis xatoliklar tuzatilgandan so'ng kompilyator ishchi dastur kodini yarata oladi.

**Direktivalar** – funksiyalar kutubxonasini chaqirish. Ular maxsus include katalogida joylashgan va .h kengaytmali fayllar bo'ladi. C++ tilida masalaning qo'yilishiga qarab kerakli kutubxonalar chaqiriladi. Bus esa dasturning xotirada egallaydigan joyini minimallashtiradi.

Masalan, ma'lumotlarni kiritish-chiqarish proseduralari uchun:

```
#include <iostream.h> tizimdan chaqirish.
```

```
#include "iostream.h" joriy katalogdan chaqirish.
```

#include direktivasi kompilyatorga iostream.h sarlavha faylidagi ma`lumotdan foydalanish zarurligini ko`rsatadi. Qisqartirilgan stdio standart kiritish\chiqarish (standart input/output) so`zini anglatadi. iostream.h fayl, komilyator disk fayllari va printer bilan ishlashi uchun kerakli bo`lgan ko`rsatmalarni saqlaydi.

Sarlavha faylida saqlanadigan ko`rsatmalarni main() funtsiyadan oldin joylashtirish lozim.

Sarlavha fayli (< >) belgilari orasida berilsa, bu fayl INCLUDE katalogida mavjud bo`lishi mumkinligini kompilyatorga bildiradi. Instalyatsiya jarayonida sarlavha fayllari shu katlogga joylashtiriladi. Agar komilyatsiya jarayonida sarlavha fayllari joriy katalogdan topilmasa, kompilyator uni INCLUDE katalogidan izlaydi.

Siz sarlavha faylini qo`shirnoq ichiga olib ham yozishingiz mumkin.

```
#include "iostream.h"
```

Bu holda kompilyator sarlavha faylini faqat joriy katalogdan izlaydi, agar uni topa olmasa xatolik mavjudligi xabarini beradi. Shuning uchun kompilyator hujjati bilan tanishib, qaysi holat o`rinli ekanligini aniqlang.

C++ dasturlash tili bilan ishlovchi eng sodda dasturlash muhiti Dev C++ va CodeBlocks dasturlari hisoblanadi. Ularning tarkibida 300 dan ortiq kutubxonalar mavjud. Eng ko`p ishlatiladigan kutubxonalar quyidagilar:

```
#include<iostream.h>,  
#include <math.h>  
#include <conio.h>  
#include <graphics.h>  
#include <memory.h> va boshqalar
```

Makrolar (#define) – dastur bajarilishi davomida o`zgaruvchi ko`rsatilgan qiymatni qabul qilishi uchun (const). Unda makroning nomi va qiymati ko`rsatiladi. Masalan:

```
#define pi 3.1415  
#define x 556  
#define s[100]  
#define M x*x*x
```

#### **Nazorat savollari:**

1. Dasturlash deganda nimani tushunasiz?.
2. Kompyuter anatomiyasi haqida tushuncha bering.
3. Mashina kodi nima?
4. Dasturlash tillari haqida tushuncha bering.
5. Dasturlash muhiti deganda nimani tushunasiz.
6. C++ tilida dastur tuzilmasini tavsiflang.
7. Dasturdagi xatolar haqida tushuncha bering.
8. Kompilyator va interpretator orasidagi farq nimadan iborat?
9. C++ tilining kompilyatorlari orasida tafovut mavjudmi?
10. Assembler tilining yuqori darajali tillardan farqi nimada?
11. Dastur matni boshlang`ich fayli nima?
12. Kompilyatsiya xatoligining bajarilish xatoligidan farqi nimada?

## 7 mavzu. Asosiy ma'lumot turlari

### Reja:

1. O'zgaruvchilar, o'zgarmaslar, arifmetik amallar, kiritish va natija, muammoni yechimi va boshqalar.

**Tayanch iboralar:** Operatorlar, toifalar, protseduralar, ma'lumotla toifalari, Identifikator, o'zgaruvchilar, tarmoqlanuvchi dastur tuzilmasi, shartli, shartsiz va tanlash operatorlari, prepotssessor, konsol rejimi, o'zgaruvchilar, o'zgarmaslar, sanab o'tiluvchi tur, ifoda.

### 1. O'zgaruvchilar, o'zgarmaslar, arifmetik amallar, kiritish va natija, muammoni yechimi va boshqalar.

**Identifikatorlar.** Dasturlash tillarida identifikator tushunchasi mavjud bo'lib, u dasturda obyektlarni nomlash uchun ishlatiladi. Masalan: `axc`, `alfa`.

*Identifikator* deb harf, raqam va maxsus belgilar ketma-ketligidan iborat bo'lib, birinchi belgisi harf yoki maxsus belgi bilan yozilgan ixtiyoriy uzunlikdagi nomga aytiladi.

O'zgarmaslarni, o'zgaruvchilarni, belgi (metka), protsedura va funksiyalarni belgilashda ishlatiladigan nom identifikatorlar deyiladi. Identifikatorlar lotin alfaviti harflaridan boshlanib, qolgan belgilari harf yoki raqamlar ketma-ketligidan tashkil topgan bo'lishi mumkin. Maxsus belgi sifatida ostiga chizig'i belgisi (`_`) ishlatiladi. Atalishiga ko'ra mos keluvchi bosma va yozma harflar ishlatilgan ikkita identifikator turli xil hisoblanadi. Masalan, `abc`, `ABC`, `A125B`, `a128b`. Identifikatorni yozishda muhim xususiyatlarni qayd etish kerak.

Birinchidan, identifikator asosiy kalit so'zlar, rezervlangan so'zlar va SI tili kompilyatori kutubxona xizmati nomlari bilan mos kelmasligi kerak.

Ikkinchidan, (`_`) ostiga chizish belgisining identifikatorning birinchi belgisi sifatida ishlatilishiga alohida e'tibor berish kerak. Chunki `C++` tilining ichki o'zgaruvchi nomlari bilan to'g'ri kelishi mumkin. Shuning uchun identifikatorni nomlashda bu belgini birinchi belgi sifatida ishlatishni tavsiya etmaymiz.

Uchinchidan, o'zgaruvchilarni aniqlash uchun ishlatiladigan identifikatorlar tashqi aloqalar muharririda ishlatiluvchi chegaralarda shakllantirib qo'yilishi shart.

**O'zgaruvchilar va o'zgarmaslar.** Dastur o'zi ishlatadigan ma'lumotlarni saqlash imkoniyatiga ega bo'lishi lozim. Buning uchun o'zgaruvchilar va o'zgarmaslardan foydalaniladi.

**O'zgarmaslar.** Dasturlash tillarida dastur bajarilishi vaqtida qiymati o'zgaraydigan identifikatorlar o'zgarmaslar deyiladi. O'zgarmaslar beshta guruhga bo'linadi – butun, haqiqiy (suzuvchi nuqtali), sanab o'tiluvchi, belgi (literli) va satr ("string", literli satr).

`C++` tilida o'zgarmas (`const`) – bu fiksirlangan sonni, satrni va belgini ifodalovchi leksema hisoblanadi.

Kompilyator o'zgarmasni leksema sifatida aniqlaydi, unga xotiradan joy ajratadi, ko'rinishi va qiymatiga (tipiga) qarab mos guruhlariga bo'ladi.

**Butun o'zgarmaslar:** ular quyidagi formatlarda bo'ladi

- o'nlik son;
- sakkizlik son;
- o'n oltilik son.

O'nlik o'zgarmas 0 raqamidan farqli raqamdan boshlanuvchi raqamlar ketma-ketligi va 0 hisoblanadi: `0`; `123`; `7987`; `11`.

Manfiy o'zgarmas – bu ishorasiz o'zgarmas bo'lib, unga faqat ishorani o'zgartirish amali qo'llanilgan deb hisoblanadi.

Sakkizlik o' 0 raqamidan boshlanuvchi sakkizlik sanoq sistemasi (0,1,...,7) raqamlaridan tashkil topgan raqamlar ketma-ketligi:

`023`; `0777`; `0`.

O'n oltilik o'zgarmas `0x` yoki `0X` belgilaridan boshlanadigan o'n oltilik sanoq sistemasi raqamlaridan iborat ketma-ketlik hisoblanadi:

`0x1A`; `0X9F2D`; `0x23`.

Harf belgilar ixtiyoriy registrlarda berilishi mumkin.

Kompilyator sonning qiymatiga qarab unga mos tipni belgilaydi. Agar tilda belgilangan tiplar dastur tuzuvchini qanoatlantirmasa, u oshkor ravishda tipni ko'rsatishi mumkin. Buning uchun butun o'zgarma raqamlari oxiriga, probelsiz l yoki L (long), u yoki U (unsigned) yoziladi. Zarur hollarda bitta o'zgarma uchun bu belgilarning ikkitasini ham ishlatish mumkin: 451u, 012Ul, 0xA2L.

**Haqiqiy o'zgarmlar:** Haqiqiy o'zgarmlar – suzuvchi nuqtali son bo'lib, u ikki xil formatda berilishi mumkin:

- O'nlik fiksirlangan nuqtali formatda. Bu ko'rinishda son nuqta orqali ajratilgan butun va kasr qismlar ko'rinishida bo'ladi. Sonning butun yoki kasr qismi bo'lmasligi mumkin, lekin nuqta albatta bo'lishi kerak. Fiksirlangan nuqtali o'zgarmlarga misollar: 24.56; 13.0; 66.; .87;
- Eksponensial shaklda haqiqiy o'zgarma 6 qismdan iborat bo'ladi:
  - 1) butun qismi (o'nli butun son);
  - 2) o'nli kasr nuqta belgisi;
  - 3) kasr qismi (o'nlik ishorasiz o'zgarma);
  - 4) eksponenta belgisi 'e' yoki 'E';
  - 5) o'n darajasi ko'rsatkichi (musbat yoki manfiy ishorali o'nli butun son);
  - 6) qo'shimcha belgisi ('F' yoki f, 'L' yoki 'l').
- Eksponensial shakldagi o'zgarma sonlarga misollar: 1E2; 5E+3; .25E4; 31.4E-1.

**Belgi o'zgarmlar:** Belgi o'zgarmlar qo'shtirnoq (' , ' -apostroflar) ichiga olingan alohida belgilardan tashkil topadi va u char kalit so'zi bilan aniqlanadi. Bitta belgi o'zgarma uchun xotirada bir bayt joy ajratiladi va unda butun son ko'rinishidagi belgining ASCII kodi joylashadi. Quyidagilar belgi o'zgarmlarga misol bo'ladi:

'e', '@', '7', 'z', 'w', '+', 'sh', '\*', 'a', 's'.

C++ tilida escape – belgilar jadvali

Escape belgilari	Ichki kodi (16 son)	Nomi	Belgining nomlanishi va unga mos amal
\\	0x5C	\	Teskari yon chiziqni chop etish
\'	0x27	'	Apostrofni chop etish
\"	0x22	“	Qo'shtirnoqni chop etish
\\?	0x3 G'	?	So'roq belgisi
\\a	0x07	bel	Tovush signalini berish
\\b	0x08	Bs	Kursorni 1 belgi o'rniga orqaga qaytarish
\\f	0x0C	ff	Sahifani o'tkazish
\\n	0x0A	lf	Qatorni o'tkazish
\\r	0x0D	cr	Kursorni ayni qatorning boshiga qaytarish
\\t	0x09	ht	Kursorni navbatdagi tabulyatsiya joyiga o'tkazish
\\v	0x0D	vt	Vertikal tabulyatsiya (pastga)
\\000	000		Sakkizlik kodi
\\xNN	0xNN		Belgi o'n oltilik kodi bilan berilgan

Ayrim belgi o'zgarmlar '\\' belgisidan boshlanadi, bu belgi birinchidan, grafik ko'rinishga ega bo'lmagan o'zgarmlarni belgilaydi, ikkinchidan, maxsus vazifalar yuklangan belgilar – apostrof belgisi, savol belgisini (?), teskari yon chiziq belgisini (\) va ikkita qo'shtirnoq belgisini (“) chop qilish uchun ishlatiladi. Undan tashqari, bu belgi orqali belgini ko'rinishini emas, balki oshkor ravishda uning ASCII kodini sakkizlik yoki o'n oltilik shaklda yozish mumkin. Bunday belgidan boshlangan belgilar escape ketma-ketliklar deyiladi.

C++ tilida belgili o'zgarmani aniqlashning ikki usuli mavjud:

1. # define direktivasi yordamida o'zgarmani aniqlash.
2. const kalitli so'zi orqali o'zgarmani aniqlash.

An'anaviy usul hisoblangan #define direktivasi orqali o'zgarmani aniqlashni quyidagi misolda ko'rishimiz mumkin.

```
#define StudentsPerClass 15
```

Bu holda StudentsPerClass o'zgarmas hech qanday tipga tegishli bo'lmaydi.

Preprotessor StudentsPerClass so'ziga duch kelganida uni 15 literaliga almashtiradi.

C++ tilida #define direktivasidan tashqari o'zgarmasni aniqlashning nisbatan qulayroq bo'lgan yangi usuli ham mavjud:

```
const unsigned short int StudentsPerClass=15
```

Bu misolda ham belgili konstanta StudentsPerClass nomi bilan aniqlanayapti va unga unsigned short int tipi berilyapti. Bu usul bir qancha imkoniyatlarga ega bo'lib u sizning dasturingizni keyingi himoyasini engillashtiradi. Bu o'zgarmasni oldingisidan eng muhim afzalligi uning tipga egaligidir.

Belgili o'zgaraslarni literal o'zgaraslarga nisbatan ishlatish qulayroqdir. Chunki agarda bir xil nomli literal o'zgaruvchini qiymatini o'zgartirmoqchi bo'lsangiz butun dastur bo'yicha uni o'zgartirishga to'g'ri keladi, belgili o'zgaraslarni esa faqatgina birining qiymatini o'zgartirish etarli.

**Turlangan o'zgaraslar:** Turlangan o'zgaraslar xuddi o'zgaruvchilardек ishlatiladi va initsializatsiya qilingandan (boshlang'ich qiymat berilgandan) keyin ularning qiymatini o'zgartirib bo'lmaydi.

Turlangan o'zgaraslar const kalit so'zi bilan e'lon qilinadi, undan keyin o'zgaras tipi va albatta initsializatsiya qismi bo'lishi kerak.

1-misol. Turlangan va literli o'zgaraslardan foydalangan holda radius berilganda aylana yuzasini hisoblaydigan dastur.

```
#include <iosream>
using namespace std;
int main (){
    const double pi=3.1415;
    const int radius=3;
    double square=0;
    square=pi*radius*radius;
    cout<<square<< '\n';
    return 0; }
```

Dastur bosh funksiyasining boshlanishida ikkita – pi va radius o'zgaraslari e'lon qilingan. Aylana yuzasini aniqlovchi square o'zgaras deb e'lon qilinmagan, chunki u dastur bajarilishida o'zgaradi. Aylana radiusini dastur ishlashida o'zgartirish mo'ljallanmagan, shu sababli u o'zgaras sifatida e'lon qilingan.

**Sanab o'tiluvchi tur:** Ko'p miqdordagi, mantiqan bog'langan o'zgaraslardan foydalanganda sanab o'tiluvchi turdan foydalanish ma'qul. Sanab o'tiluvchi o'zgaraslar enum kalit so'zi bilan aniqlanadi. Mazmuni bo'yicha bu o'zgaraslar oddiy butun sonlardir. Sanab o'tiluvchi o'zgaraslar C++ standarti bo'yicha butun tipdagi o'zgaraslar hisoblanadi. Har bir o'zgarasga (songa) mazmunli nom beriladi va bu identifikatorni dasturning boshqa joylarida nomlash uchun ishlatilishi mumkin emas. Sanab o'tiluvchi tur quyidagi ko'rinishga ega:

```
enum <Sanab o'tiladigan tur nomi> { <nom1> =<qiymat1>, <nom2> = <qiymat2>,...
<nom3>=<qiymat3> } ;
```

bu yerda, enum – kalit so'z (inglizcha enumerate – sanamoq);

<Sanab o'tiladigan tur nomi> – o'zgaraslar ro'yxatining nomi;

<nom> – butun qiymatli konstantalarning nomlari;

<qiymati> – shart bo'lmagan initsializatsiya qiymati (ifoda).

**O'zgaruvchilar.** Dastur ishlashi mobaynida qiymatlari o'zgarishi mumkin bo'lgan identifikatorga o'zgaruvchilar deyiladi.

Dasturlash tillarida dastur bajarilishi paytida qandaydir berilganlarni saqlab turish uchun o'zgaruvchilar va o'zgaraslardan foydalaniladi. O'zgaruvchi-dastur obyekti bo'lib, xotiradagi bir nechta yacheykalarni egallaydi va berilganlarni saqlash uchun xizmat qiladi. O'zgaruvchi nomga, o'lchamga va boshqa atributlarga – ko'rinish sohasi, amal qilish vaqti va boshqa

xususiyatlarga ega bo'ladi. O'zgaruvchilarni ishlatish uchun ular albatta e'lon qilinishi kerak. E'lon natijasida o'zgaruvchi uchun xotiradan qandaydir soha zahiralanadi, soha o'lchami esa o'zgaruvchining aniq turiga bog'liq bo'ladi. Shuni qayd etish zarurki, bitta tipga turli apparat platformalarda turlicha joy ajratilishi mumkin.

C++ tilida o'zgaruvchi e'loni uning tipini aniqlovchi kalit so'zi bilan boshlanadi va '=' belgisi orqali boshlang'ich qiymat beriladi (shart emas). Bitta kalit so'z bilan bir nechta o'zgaruvchilarni e'lon qilish mumkin. Buning uchun o'zgaruvchilar bir-biridan ',' belgisi bilan ajratiladi. E'lonlar ';' belgisi bilan tugaydi. O'zgaruvchi nomi 255 belgidan oshmasligi kerak.

O'zgaruvchilarni e'lon qilish dastur matnining istalgan joyida amalga oshirilishi mumkin.

**Ma'lumotlar tiplari.** C++ tilida o'zgaruvchini aniqlash uchun komp'yuterga uning tipi (masalan, int, char yoki boshqa) haqida ma'lumot beriladi. Bu axborot asosida kompilyatorga o'zgaruvchi uchun qancha joy ajratish lozim va bu o'zgaruvchida qanaqa tipdagi qiymat saqlanishi mumkinligi haqida ma'lumot aniq bo'ladi.

C++ tilining tayanch tiplari.

Tip nomi	Baytlardagi o'lchami	Qiymat chegarasi
bool	1	True yoki false
unsigned short int	2	0..65535
short int	2	-32768..32767
unsigned long int	4	0..42949667295
long int	4	-2147483648..2147483647
int(16 razryadli)	2	-32768.. 32767
int (32 razryadli)	4	-2147483648..2147483647
unsigned int (16 razryadli)	2	0..65535
unsigned int (32 razryadli)	4	0..42949667295
unsigned char	1	0..255
char	1	-128.. 127
float;	4	1.2E-38..3.4E38
double	8	2.2E-308..1.8E308
long double(32 razryadli)	10	3.4e-4932..-3.4e4932
void	2 ёки 4	-

2 – misol. Tayanch tiplar uchun kompyuter xotirasidan ajratiladigan baytlarni aniqlash.

```
#include <iostream>
using namespace std;
int main()      {
    cout << "int tipining o'lchami: \t" << sizeof(int) << " bayt.";
    cout << "short int tipining o'lchami: \t" << sizeof(short) << " bayt.";
    cout << "long int tipining o'lchami: \t" << sizeof(long) << " bayt.";
    cout << "char tipining o'lchami: \t" << sizeof(char) << " bayt.";
    return 0;    };
```

NATIJA:

```
int tipining o'lchami: 4 bayt.
short int tipining o'lchami: 2 bayt.
long int tipining o'lchami: 4 bayt.;
char tipining o'lchami: 1 bayt;
```

C++ tilida ham o'zgaruvchilarning turlari bir necha guruhlarga ajraladi. Ularni quyida qarab chiqamiz.

**Butun son tiplari.** Butun son qiymatlarni qabul qiladigan o'zgaruvchilar int(butun), short(qisqa) va long(uzun) kalit so'zlar bilan aniqlanadi. O'zgaruvchi qiymatlari ishorali bo'lishi yoki unsigned kalit so'zi bilan ishorasiz son sifatida qaralishi mumkin.

**Belgi tipi.** Belgi tipidagi o‘zgaruvchilar char kalit so‘zi bilan beriladi va ular o‘zida belgining ASCII kodini saqlaydi. Belgi tipidagi qiymatlar nisbatan murakkab bo‘lgan tuzilmalar – satrlar, belgilar massivlari va hokazolarni hosil qilishda ishlatiladi.

**Haqiqiy son tipi.** Haqiqiy sonlar float kalit so‘zi bilan e‘lon qilinadi. Bu tipdagi o‘zgaruvchi uchun xotiradan 4 bayt joy ajratiladi va <ishora><tartib><mantissa> qolipida sonni saqlaydi. Agar kasrli son juda katta (kichik) qiymatlarni qabul qiladigan bo‘lsa, u xotirada 8 yoki 10 baytli ikkilangan aniqlik ko‘rinishida saqlanadi va mos double va long double kalit so‘zlari bilan e‘lon qilinadi. Oxirgi holat 32-razryadli platformalar uchun o‘rinli.

**Mantiqiy tip.** Bu tipdagi o‘zgaruvchi bool kalit so‘zi bilan e‘lon qilinib, xotiradan 1 bayt joy egallaydi va 0 (false, yolg‘on) yoki (true, rost) qiymat qabul qiladi. Mantiqiy tip o‘zgaruvchilar qiymatlar o‘rtasidagi munosabatlarni ifodalaydigan mulohazalarni rost (true) yoki yolg‘on (false) ekanligi tavsifida qo‘llaniladi va ular qabul qiladigan qiymatlar matematik mantiq qonuniyatlariga asoslanadi.

Mantiqiy mulohazalar ustida uchta amal aniqlangan:

1) **inkor** – A mulohazani inkori deganda A rost bo‘lganda yolg‘on yoki yolg‘on bo‘lganda rost qiymat qabul qiluvchi mulohazaga aytiladi. C++ tilida inkor – “!” belgisi bilan beriladi. Masalan, A mulohaza inkori “!A” ko‘rinishida yoziladi;

2) **konyunksiya**- ikkita A va B mulohazalar konyunksiyasi yoki mantiqiy ko‘paytmasi “A&&B” ko‘rinishga ega. Bu mulohaza faqat A va B mulohazalar rost bo‘lgandagina rost bo‘ladi, aks holda yolg‘on bo‘ladi (odatda “&&” amali “va” deb o‘qiladi). Masalan «bugun oyning 5- kuni va bugun chorshanba» mulohazasi oyning 5- kuni chorshanba bo‘lgan kunlar uchungina rost bo‘ladi;

3) **dizyunksiya** – ikkita A va B mulohazalar dizyunksiyasi yoki mantiqiy yig‘indisi “A || B” ko‘rinishida yoziladi. Bu mulohaza rost bo‘lishi uchun A yoki B mulohazalardan biri rost bo‘lishi yetarli. Odatda “||” amali “yoki” deb o‘qiladi.

Yuqorida keltirilgan fikrlar asosida mantiqiy amallar uchun rostlik jadvali aniqlangan.

Mantiqiy amallar uchun rostlik jadvali

Mulohazalar		Mulohazalar ustida amallar		
A	B	Not A	A and B	A or B
!A		A&&B		A  B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

Mantiqiy tip qiymatlari ustida mantiqiy ko‘paytirish, qo‘shish va inkor amallarini qo‘llash orqali murakkab mantiqiy ifodalarni qurish mumkin. Misol uchun, “x – musbat va uning qiymati [1..3] sonlar oralig‘iga tegishli emas” mulohazasini mantiqiy ifoda ko‘rinishi quyidagicha bo‘ladi:  $(x>0)\&\&(y<1||y>3)$ .

**Void tipi.** C++ tilida void tipi aniqlangan bo‘lib bu tipdagi dastur obyektini hech qanday qiymatga ega bo‘lmaydi va bu tipdan qurilmaning til sintaksisiga mos kelishini ta‘minlash uchun ishlatiladi. Masalan, C++ tili sintaksisi funksiya qiymat qaytarishini talab qiladi. Agar funksiya qiymat qaytarmaydigan bo‘lsa, u void kalit so‘zi bilan e‘lon qilinadi.

Misollar.

`int a=0 A=1; float abc=17.5;`

`double Ildiz;`

`bool ok=true;`

`char LETTER='z';`

`void mening_funksiyam(); /*funksiya qaytaradigan qiymat inobatga olinmaydi */`

**Tipni boshqa tipga keltirish:** C++ tilida bir tipni boshqa tipga keltirishning oshkor va oshkormas yo‘llari mavjud. Umuman olganda, tipni boshqa tipga oshkormas keltirish ifodada har xil tipdagi o‘zgaruvchilar qatnashgan hollarda amal qiladi (aralash tiplar arifmetikasi). Ayrim hollarda, xususan tayanch tiplar bilan bog‘liq tipga keltirish amallarida xatoliklar yuzaga kelishi



mumkin. Masalan, hisoblash natijasining xotiradan vaqtincha egallagan joyi uzunligi, uni oʻzlashtiradigan oʻzgaruvchi uchun ajratilgan joy uzunligidan katta boʻlsa, qiymatga ega razryadlarni yoʻqotish holati yuz beradi.

Oshkor ravishda tipga keltirishda, oʻzgaruvchi oldiga qavs ichida boshqa tip nomi yoziladi: 3 – misol.

```
#include <iostream>
using namespace std;
int main() {
    int integer_1=54;
    int integer_2;
    float floating=15.854;
    integer_1=(int) floating; // oshkor keltirish;
    integer_2=(int) floating // oshkormas keltirish;
    cout<< "yangi integer (oshkor): "<<integer_1<<"\n";
    cout<< "yangi integer (oshkormas): "<<integer_2<<"\n";
    return 0; }
```

Dastur natijasi quyidagi koʻrinishda boʻladi:

Yangi integer (oshkor):15

Yangi integer (oshkormas):15

4 – misol. Berilgan belgining ASCII kodi chop etilsin. Masala belgi tipidagi qiymatni oshkor ravishda butun son tipiga keltirib chop qilish orqali yechiladi.

```
#include <iostream>
using namespace std;
int main() {
    unsigned char A;
    cout<< "belgini kiriting: ";
    cin>>A;
    cout<< "\n"<<A<<" "-belgi ASCII kodi="(int)A<<"\n";
    return 0; }
```

Dasturning belgini kiriting soʻroviga

A <enter> amali bajarilsa, ekranga "A"-belgi ASCII kodi=65 satri chop etiladi.

**Ishorali va ishorasiz tiplar.** Dasturda qoʻllaniladigan butun sonli tiplar ishorali va ishorasiz boʻlishi mumkin. Baʼzan oʻzgaruvchi uchun faqatgina musbat sonni qoʻllash foydali boʻladi. Unsigned kalitli soʻziciz keltirilgan butun sonli tiplar (short va long) ishorali hisoblanadi. Ishorali butun sonlar manfiy va musbat boʻlishi mumkin. Ishorasiz sonlar esa doimo musbat boʻladi.

**Oʻzgaruvchiga qiymat berish.** Oʻzgaruvchilarga qiymat berish uchun oʻzlashtirish operatori qoʻllaniladi. Masalan, Width oʻzgaruvchisiga 5 qiymatni berish uchun quyidagilarni yozish lozim:

```
unsigned short Width;
Width = 5;
```

Bu ikkala satrni Width oʻzgaruvchisini aniqlash jarayonida birgalikda yozish mumkin.

```
unsigned short Width = 5;
```

Bir necha oʻzgaruvchilarni aniqlash vaqtida ham ularga qiymat berish mumkin:

```
Long width = 5, length = 7;
```

Bu misolda long tipidagi width oʻzgaruvchisi 5 qiymatni, shu tipdagi length oʻzgaruvchisi esa 7 qiymatni qabul qildi. Quyidagi misolda oʻzgaruvchilarni aniqlashga oid misolni qaraymiz.

5 – misol. Oʻzgaruvchilarning qoʻllanishi.

```
# include <iostream>
using namespace std;
int main() {
    int Buyi=5, Eni=10, Yuzasi;
    cout << "Boʻyi:" << Buyi << "\n";
    cout << "Eni:" << Eni << endl;
```

```

Yuzasi= Buyi*Eni;
cout << "Yuzasi:" << Yuza << endl;
return 0;}

```

NATIJA:

Bo`yi: 5

Eni: 10

Yuzasi: 50

**typedef kalitli so`zi.** unsigned short int kabi kalit so`zlarni ko`p martalab dasturda yozilishi zerikarli va diqqatvozklik talab qilganligi uchun C++ tilida bunday tiplarni typedef kalitli so`zi yordamida psevdonimini (taxallusini) tuzish imkoniyati berilgan. typedef so`zi tipni aniqlash ma`nosini bildiradi.

Psevdonim tuzishda tipning nomi yangi tuziladigan tip nomidan farqli bo`lishi lozim. Bunda birinchi typedef kalitli so`zi, keyin mavjud tip nomi, undan so`ng esa yangi nom yoziladi. Masalan: typedef unsigned short int ushort

Bu satrdan so`ng ushort nomli yangi tip hosil bo`ladi va u qaerda unsigned short int tipidagi o`zgaruvchini aniqlash lozim bulsa, shu joyda ishlatiladi.

6 – misol . typedef operatori orqali tiplarning aniqlanishi.

```

#include <iostream>
typedef unsigned short int ushort;
using namespace std;
int main()
{ ushort Buyi = 5;
  ushort Eni = 10;
  ushort Yuzasi = Buyi* Eni;
  cout << "Yuzasi:" << Yuzasi << endl;
  return 0; }

```

NATIJA:

Area: 50

**Arifmetik amallar.** Berilganlarni qayta ishlash uchun dasturlash tillarida amallarning juda keng majmuasi aniqlangan. Amal - bu qandaydir harakat bo`lib, u bitta (unar) yoki ikkita (binar) operandlar ustida bajariladi, hisob natijasi uning qaytaruvchi qiymati hisoblanadi.

Tayanch arifmetik amallarga qo`shish (+), ayirish (-), ko`paytirish (\*), bo`lish (/) va bo`lishdagi qoldiqni olish (%) amallarini keltirish mumkin.

Amallar qaytaradigan qiymatlarni o`zlashtirish uchun C++ tilida "=" va uning turli modifikatsiyalari ishlatilib, quyidagilar hisoblanadi: qo`shish, qiymat berish bilan (+=); ayirish, qiymat berish bilan (-=); ko`paytirish, qiymat berish bilan (\*=); bo`lish, qiymat berish bilan (/=); bo`lish qoldig`ini olish, qiymat berish bilan (%=) va boshqalar. Bu holatlarning umumiy ko`rinishi:

<o`zgaruvchi><amal>=<ifoda>;

**Ifoda.** C++ tilida ifodalar biror bir hisoblash natijasini qaytaruvchi boshqa ifodalar ketma-ketligini boshqaradi yoki hech nima qilmaydi (nol ifodalar). C++ tilida barcha ifodalar nuqtali vergul bilan yakunlanadi. Ifodaga misol qilib o`zlashtirish amalini olish mumkin.

$x=a+b;$

Algebradan farqli ravishda bu ifoda  $x = a+b$  ga teng ekanligini anglatmaydi. Bu ifodani quyidagicha tushinish kerak:

$a$  va  $b$  o`zgaruvchilarni qiymatlarini yig`ib natijasini  $x$  o`zgaruvchiga beramiz yoki  $x$  o`zgaruvchiga  $a+b$  qiymatni o`zlashtiramiz. Bu ifoda birdaniga ikkita amalni bajaradi, yig`indini hisoblaydi va natijani o`zlashtiradi. Ifodadan so`ng nuqtali vergul qo`yiladi. (=) operatori o`zidan chap tomondagi operandga o`ng tomondagi operandlar ustida bajarilgan amallar natijasini o`zlashtiradi.

**Bo`sh joy (probel) belgisi.** Bo`sh joy belgilariga nafaqat probel, balki yangi satrga o`tish va tabulyatsiya belgilari ham kiradi. Yuqorida keltirilgan ifodani quyidagicha ham yozish mumkin:

$x = a; \quad + \quad b ;$

Bu variantda keltirilgan ifoda ko`rimsiz va tushunarsiz bo`lsa ham to`g`ridir.

Bo`sh joy belgilari dasturning o`qilishligini ta`minlaydi.

**Bloklar va kompleks ifodalar.** Ba`zan dastur tushunarli bo`lishi uchun o`zaro mantiqiy bog`langan ifodalarni blok deb ataluvchi komplekslarga birlashtirish qulaydir. Blok ochiluvchi figurali qavs ({} bilan boshlanadi va yopiluvchi figurali qavs (}) bilan tugaydi. Blok ochilganda va yopilganda nuqtali vergul qo`yilmaydi.

```
{ temp= a;
  a = b;
  b = temp; }
```

Bu blok xuddi bir ifodadek bajariladi, u a va b o`zgaruvchilar qiymatlarini almashtiradi.

**Amallar.** Bajarilishi natijasida biror bir qiymat qaytaradigan barcha ifodalar C++ tilida amallar deyiladi. Amallar albatta biror bir qiymat qaytaradi. Masalan, 3+2 amali 5 qiymatni qaytaradi.

7 – misol . Quyidagi dastur matnida ayrim amallarga misollar keltirilgan.

```
#include <iostream>
using namespace std;
int main()
{ int a=0 , b=4, c=90; char z='t';
  a=b; cout<<a<<z; //a=4
  a=b+c+c+b; cout<<a<<z; //a=4+90+90+4=188
  a=b-2; cout<<a<<z; //a=2
  a=b*3 cout<<a<<z; //a=4*3=12
  a=c/(b+6); cout<<a<<z; //a=90/(4+6)=9
  cout<<a%2<<z; //9%2=1
  a+=b; cout<<a<<z; //a=a+b=9+4=13
  a*=c-50; cout<<a<<z; //a=a*(c-50)=13*(90-50)=520
  a-=38; cout<<a<<z; //a=a-38=520-38=482
  a%=8; cout<<a<<z; //a=a%8=482%8=2
  return 0; }
```

Dastur bajarilishi natijasida ekranda quyidagi sonlar satri paydo bo`ladi:

4 188 2 12 9 1 482 2

**Operatorlar.** Operator - bu qandaydir amalni bajarish tug`risida kompilyatorga uzatiladigan literaldir. Operatorlar operandlarga ta`sir qiladi. C++ da operandlar deb alohida literallar va butun ifodalar tushuniladi.

C++ tilida ikki ko`rinishdagi operatorlar bor:

- o`zlashtirish operatorlari;
- matematik operatorlar.

**O`zlashtirish operatori.** O`zlashtirish operatori (=) o`zidan chap tomonda turgan operand qiymatini tenglik belgisidan o`ng tomondagilarni hisoblangan qiymatiga almashtiradi. Masalan,

$x = a+b;$

ifodasi x operandga a va b o`zgaruvchilarni qiymatlarini qo`shishdan hosil bo`lgan natijani o`zlashtiradi.

O`zlashtirish operatoridan chapda joylashgan operand adresli operand yoki l–qiymat deyiladi. O`zlashtirish operatoridan o`ngda joylashgan operand operatsion operand yoki r–qiymat deyiladi.

O`zgarmlar faqatgina r–qiymat bo`lishi mumkin va hech qachon adresli operand bo`la olmaydi, chunki dasturning bajarilishi jarayonida o`zgarmlar qiymatini o`zgartirib bo`lmaydi.

$35 = x // notugri!$

l–qiymat esa r–qiymat bo`lishi mumkin.

**Matematik operatorlar.** C++ tilida 5 ta asosiy matematik operatorlar qo'llaniladi: qo'shish (+), ayirish (-), ko'paytirish (\*), butun songa bo'lish (/) va modul bo'yicha bo'lish (%) (qoldiqni olish).

Ishorasiz butun sonlarni ayirishda, agarda natija manfiy son bo'lsa g'ayrioddiy natija beradi. Buni 2. misoldan ko'rishimiz mumkin.

8 – misol. Ayirish natijasida butun sonni to'lib qolishiga misol

```
#include <iostream>
using namespace std;
int main()
{ unsigned int ayirma;
  unsigned int kattaSon = 100;
  unsigned int kichikSon = 50;
  ayirma = kattaSon - kichikSon;
  cout << "Ayirma":<< ayirma<< " ga teng\n";
  ayirma = kichikSon - kattaSon ;
  cout << "Ayirma":<< ayirma<< " ga teng\n"<<endl;
  return 0;}
```

HATIJA:

Ayirma: 50 ga teng

Ayirma: 4294967246 ga teng

**Butun songa bo'lish va qoldiqni olish operatorlari.** Butun songa bo'lish odatdagi bo'lishdan farq qiladi. Butun songa bo'lishdan hosil bo'lgan bo'linmaning faqatgina butun qismi olinadi. Masalan, 21 sonini 4 ga bo'lsak 5 soni va 1 qoldiq hosil bo'ladi. 5 butun songa bo'lishni qiymati, 1 esa qoldiqni olish qiymati hisoblanadi.

**Razryadli mantiqiy amallar.** Dastur tuzish tajribasi shuni ko'rsatadiki, odatda qo'yilgan masalani yechishda biror holat ro'y bergan yoki yo'qligini ifodalash uchun 0 va 1 qiymat qabul qiluvchi bayroqlardan foydalaniladi. Bu maqsadda bir yoki undan ortiq baytli o'zgaruvchilardan foydalanish mumkin. Masalan, mantiqiy turdagi o'zgaruvchini shu maqsadda ishlatsa bo'ladi. Boshqa tomondan, bayroq sifatida baytning razryadlaridan foydalanish ham mumkin. Chunki razryadlar faqat ikkita qiymatni – 0 va 1 sonlarini qabul qiladi. Bir baytda 8 razryad bo'lgani uchun unda 8 ta bayroqni kodlash imkoniyati mavjud.

Faraz qilaylik, qo'riqlash tizimiga 5 ta xona ulangan va tizim taxtasidagi 5 ta chiroqcha (indikator) xonalar holatini bildiradi: xona qo'riqlash tizimi nazoratida ekanligini mos indikatorning yonib turishi (razryadning 1 qiymati) va xonani tizimga ulanmaganligini indikator o'chganligi (razryadning 0 qiymati) bildiradi. Tizim holatini ifodalash uchun bir bayt yetarli bo'ladi va uning

kichik razryadidan boshlab beshtasini shu maqsadda ishlatish mumkin:

7	6	5	4	3	2	1
			Ind5	ind4	ind3	ind2

Masalan, baytning quyidagi holati 1, 4 va 5 xonalar qo'riqlash tizimiga ulanganligini bildiradi:

7	6	5	4	3	2	1	0
x	x	x	1	1	0	0	1

Quyidagi jadvalda C++ tilida bayt razryadlari ustida mantiqiy amallar majmuasi keltirilgan.

Amallar		Mazmuni
and	&	Mantiqiy VA (ko'paytirish)
or		Mantiqiy yoki (qo'shish))
xor	^	Istisno qiluvchi YOKI
not	~	Mantiqiy INKOR (inversiya)

C++ tilida razryadli mantiqiy amallarni qiymat berish operatori birgalikda bajarilishining quyidagi ko'rinishlari mavjud:

&= – razryadli VA qiymat berish bilan;

| = – razryadli YOKI qiymat berish bilan;

^ = – razryadli istisno qiluvchi YOKI qiymat berish bilan.

**Chapga va o'ngga surish amallari.** Baytdagi bitlar qiymatini chapga yoki o'ngga surish uchun, C++ tilida “<<” va “>>” amallari qo'llanilib, amaldan keyingi son bitlarni nechta o'rin chapga yoki o'nga surish kerakligini bildiradi.

Masalan:

unsigned char A=12; //A=000011002=0x0C16

A=A<<2; // A=001100002=0x3016=48

A=A>>3; //A=000001102=0x0616=610

Razryadlarni n ta chapga (o'nga) surish sonni 2 soniga ko'paytirish (bo'lish) amali bilan ekvivalent bo'lib va nisbatan tez bajariladi. Shuni e'tiborga olish kerakki, operand ishorali son bo'lsa, u holda chapga surishda eng chapdagi ishora razryadi takrorlanadi (ishora saqlanib qoladi) va manfiy sonlar ustida bu amal bajarilganda matematika nuqtai-nazardan xato natijalar yuzaga keladi:

unsigned char B=-120; // B=100010002=0x8816

B=B<<2; // B=001000002=0x2016=3210

B=-120; // B=100010002=0x88

B=B>>3; // B=111100012=0xG'1=-1510

Shu sababli, bu razryadli surish amallari ishorasiz (unsigned) tipdagi qiymatlar ustida bajarilgani ma'qul.

**Taqqoslash amallari.** Dasturlash tillarida qiymatlarni solishtirish uchun taqqoslash amallari aniqlangan. Taqqoslash amali binar amal bo'lib, quyidagi ko'rinishga ega:

<operand1> <taqqoslash amali> <operand2>

Taqqoslash amallarining natijasi – taqqoslash o'rinli bo'lsa, true (rost), aks holda false (yolg'on) qiymat bo'ladi. Agar taqqoslashda arifmetik ifoda qatnashsa, uning qiymati 0 qiymatidan farqli holatlar uchun 1 deb hisoblanadi.

Taqqoslash amallari va ularning qo'llanishi.

Amallar		Qo'llanishi		Mazmuni (o'qilishi)
<	<	a<b	a<b	“a kichik b”
<=	<=	a<=b	a<=b	“a kichik yoki teng b”
>	>	a>b	a>b	“a katta b”
>=	>=	a>=b	a>=b	“a katta yoki teng”
=	==	a=b	a==b	“a teng b”
<>	!=	a<>b	a!=b	“a teng emas b”

**Amallarining ustunliklari va bajarilish yo'nalishlari.** An'anaviy arifmetikadagidek dasturlash tillarida ham amallar ma'lum bir tartib va yo'nalishda bajariladi. Ma'lumki, matematik ifodalarda bir xil ustunlikdagi (“приоритет”dagi) amallar uchrasa (masalan, qo'shish va ayirish), ular chapdan o'ngga bajariladi. Bu tartib dasturlash tillarida ham o'rinli, biroq ayrim hollarda amal o'ngdan chapga bajarilishi mumkin (qiymat berish amalida).

Ifodalar qiymatini hisoblashda amallar ustunligi hisobga olinadi, birinchi navbatda eng yuqori ustunlikka ega bo'lgan amal bajariladi.

Operatorlarning tavsifi.

Operator	Tavsifi	Ustunligi	Yo'nalishi
::	Ko'rinish sohasiga ruxsat berish	16	=>
[]	Massiv indeksi	16	=>
()	Funksiyani chaqirish	16	=>
->	Tuzilma yoki sinf elementini tanlash	16	=>
++	Postfiks inkrement	15	<=
--	Postfiks dekrement	15	<=
++	Prefiks inkrement	14	<=
--	Prefiks dekrement	14	<=

sizeof	O'lchamni olish	14	<=
(<type>)	Tipga akslantirish	14	
~	Bitli mantiqiy INKOR	14	<=
!	Mantiqiy inkor	14	<=
-	Unar minus	14	
+	Unar plyus	14	<=
&	Adresni olish	14	<=
*	Vositali murojaat	14	<=
new	Dinamik obyektни yaratish	14	<=
delete	Dinamik obyektни yo'q qilish	14	<=
casting	Tipga keltirish	14	
*	Ko'paytirish	13	=>
/	Bo'lish	13	=>
%	Bo'lish qoldig'i	13	=>
+	Qo'shish	12	=>
-	Ayirish	12	=>
»	Razryad bo'yicha o'ngga surish	11	
«	Razryad bo'yicha chapga surish	11	
<	Kichik	10	=>
<=	Kichik yoki teng	10	=>
>	Katta	10	=>
>=	Katta yoki teng	10	
==	Teng	9	=>
!=	Teng emas	9	=>
&	Razryadli VA	8	=>
^	Razryadli istisno qiluvchi YOKI	7	=>
	Razryadli YOKI	6	=>
&&	Mantiqiy VA	5	=>
	Mantiqiy yoki	4	=>
?:	Shart amali	3	<=
=	Qiymat berish	2	<=
*=	Ko'paytirish qiymat berish amali bilan	2	<=
/=	Bo'lish qiymat berish amali bilan	2	<=
%=	Modulli bo'lish qiymat berish amali bilan	2	<=
+=	Qo'shish qiymat berish amali bilan	2	
-=	Ayirish qiymat berish amali bilan	2	<=
<<=	Chapga surish qiymat berish amali bilan	2	<=
>>=	o'ngga surish qiymat berish amali bilan	2	<=
&=	Razryadli va qiymat berish bilan	2	<=
^=	Razryadli istisno qiluvchi yoki qiymat berish bilan	2	<=
=	Razryadli yoki qiymat berish bilan	2	<=
Throw	Istisno holatni yuzaga keltirish	2	<=
,	Vergul	1	<=

Agarda ikkita matematik ifodaning prioriteti teng bo'lsa, ular chapdan o'ngga qarab ketma-ket bajariladi.

Demak

$$x=5+3+8*9+6*4$$

ifodada birinchi ko'paytirish amallari chapdan o'ngga qarab bajariladi  $8*9=72$  va  $6*4=24$ .

Keyin bu ifoda soddaroq ko'rinish hosil qiladi.

$$x=5+3+72+24$$

Endi qo`shishni ham xuddi shunday chapdan unga qarab bajaramiz:

$5+3=8$ ;  $8+72=80$ ;  $80+24=104$ ;

Dastur tuzuvchisi amallarni bajarilish tartibini o`zgartirishi ham mumkin. Xuddi matematikadagidek, amallarni qavslar yordamida guruhlarga jamlash mumkin. Qavs ishlatishga cheklov yo`q.

9 – misol. Qavs yordamida amallarni bajarilish tartibini o`zgartirish.

```
#include <iostream>
using namespace std;
int main(){
int x=0,y=0;
int a=3, b=34, c=82;
x=a*b+c;
y=(a*(b+c));
cout<<"x="<<x<<"\n";
cout<<"y="<<y<<"\n"; return 0;}
```

Dasturda amallar ustunligiga ko`ra x qiymatini hisoblashda oldin a o`zgaruvchi b o`zgaruvchiga ko`paytiriladi va unga c o`zgaruvchining qiymati qo`shiladi. Navbatdagi ko`rsatmani bajarishda esa birinchi navbatda ichki qavs ichidagi ifoda  $-(b+c)$  ning qiymati hisoblanadi, keyin bu qiymat a ga ko`paytirilib, y o`zgaruvchisiga o`zlashtiriladi.

Dastur bajarilishi natijasida ekranga

x=184

y=348

satrlari chop etiladi.

**Inkrement va dekrement.** Dasturlarda o`zgaruvchiga 1 ni qo`shish va ayirish amallari juda ko`p hollarda uchraydi. C++ tilida qiymatni 1 ga oshirish inkrement, 1 ga kamaytirish esa dekrement deyiladi. Bu amallar uchun maxsus operatorlar mavjuddir.

Inkrement operatori (++) o`zgaruvchi qiymatini 1 ga oshiradi, dekrement operatori (--) esa o`zgaruvchi qiymatini 1 ga kamaytiradi. Masalan, s o`zgaruvchisiga 1 qiymatni qo`shmoqchi bo`lsak quyidagi ifodani yozishimiz lozim.

Bu ifodani quyidagicha yozishimiz mumkin edi.

$s=s+1$ ;

Bu ifoda o`z navbatida quyidagi ifodaga teng kuchli:

$s+=1$ ;

**Prefiks va postfix.** Inkrement operatori ham, dekrement operatori ham ikki variantda ishlaydi: prefiksli va postfixli. Prefiksli variantda ular o`zgaruvchidan oldin (++)Age), postfixli variantda esa o`zgaruvchidan keyin (Age++) yoziladi.

Oddiy ifodalarda bu variantlarni qo`llanishida farq katta emas, lekin bir o`zgaruvchiga boshqa o`zgaruvchining qiymatini o`zlashtirishda ularning qo`llanilishi boshqacha xarakterga ega. Prefiksli operator qiymat o`zlashtirilguncha, postfixli operator esa qiymat o`zlashtirilgandan keyin bajariladi. Buni quyidagi misoldan ko`rishimiz mumkin:

10 – misol. Prefiksli va postfixli operatorlarni qo`llanishi.

```
#include <iostream>
using namespace std;
int main() {
int myAge = 39; int yourAge = 39;
cout << "Men" << MyAge << "yoshdaman \n";
cout << "Siz" << yourAge << "yoshdasiz \n";
myAge++; // postfixli inkrement
++yourAge; // prefiksli inkrement
cout << "Bir yil o`tdi ... \n";
cout << "Men" << MyAge << "yoshdaman \n";
cout << "Siz" << yourAge << "yoshdasiz \n";
```

```

cout << "Yana bir yil o`tdi \n";
cout << "Men" << myAge++ << "yoshdaman \n";
cout << "Siz" << ++yourAge << "yoshdasiz \n";
cout << "Ma`lumotlarni qaytadan"
cout << "chiqaraylik \n";
cout << "Men" << myAge << "yoshdaman \n";
cout << "Siz" << yourAge << "yoshdasiz \n";
return 0;}

```

HATIJA:

```

Men 39 yoshdaman
Siz 39 yoshdasiz
Bir yil o`tdi ...
Men 40 yoshdaman
Siz 40 yoshdasiz
Yana bir yil o`tdi ...
Men 40 yoshdaman
Siz 41 yoshdasiz
Ma`lumotlarni qaytadan chiqaraylik
Men 41 yoshdaman
Siz 41 yoshdasiz

```

Biz dasturlashga doir keltirgan misollarimiz asosan chiziqli jarayonlar algortmlariga tegishli edi. Chiziqli jarayonlar algortmlari amallar ketma-ketligini tashkil qilib, hisoblash jarayonida bu amallar kelish tartibi bo`yicha faqat bir martagina bajariladi.

#### Nazorat savollari:

1. O`zgaruvchilar deganda nimani tushunasiz?
  2. Butun sonlar tiplarini tavsiflang.
  3. O`zgaruvchiga qiymat berish haqida tushuncha bering.
  4. typedef kalitli so`zi tavsifi.
  5. Maxsus belgilar haqida tushuncha bering.
  6. O`zgarmaslar deganda nimani tushunasiz?
  7. Belgili o`zgarmaslar deganda nimani tushunasiz?
  8. C++ tilida arifmetik amallarning bajarilish tartibi qanday?
  9. Ifoda nima?
  10. Bloklar va kompleks ifodalar deganda nimani tushunasiz?
  11. O`zlashtirish operatori haqida tushuncha bering.
  12. Matematik operatorlar haqida tushuncha bering.
  13. Inkrement va dekrement haqida tushuncha bering.
  14. Prefiks va postfiks haqida tushuncha bering.
  15. Chiqarish operatorini tavsiflang.
  16. Kiritish va natija.
  17. Ifoda nima?
  18.  $x=5+7$  yozuvi ifoda bo`la oladimi? Uning qiymati nechaga teng?
  19.  $201/4$  ifodaning qiymati nechaga teng?
  20.  $201\%4$  ifoda qiymati nechaga teng?
  21. Agarda MyAge, a va b o`zgaruvchilarining tiplari int bo`lsa ularning qiymatlari quyidagi ifodalar bajarilgandan so`ng nechaga teng bo`ladi.
- ```

MyAge = 39;
a=MyAge++;
b=++MyAge;

```



## 8 mavzu. Operatorlar va standart funksiyalar

### Reja:

1. if operatori, sonlarni va qatorlarni solishtirish, birdan ortiq tanlovlar, muammo yechimi: diogrammalar, birlashtirilgan o'tishlar, muammo yechimi: kiruvchi ma'lumotni tasdiqlash, operatorlar va o'zgaruvchilar.

**Tayanch iboralar.** Operator, if operatori,

**1. if operatori, sonlarni va qatorlarni solishtirish, birdan ortiq tanlovlar, muammo yechimi: diogrammalar, birlashtirilgan o'tishlar, muammo yechimi: kiruvchi ma'lumotni tasdiqlash, operatorlar va o'zgaruvchilar.**

**Tarmoqlanuvchi dastur tuzilmasi.**

**if operatori.** Odatda dastur satrma–satr tartib bilan bajariladi. if operatori shartni tekshirish (masalan, ikki o'zgaruvchi tengmi) va uning natijasiga bog'liq ravishda dasturni bajarilish tartibini o'zgartirish imkonini beradi. if operatorining oddiy shakli quyidagi ko'rinishdadir:

*if (shart)*

*ifoda;*

Qavs ichidagi shart ixtiyoriy ifoda bo'lishi mumkin. Agarda bu ifoda false qiymatini qaytarsa undan keyingi ifoda yoki blok tushirib qoldiriladi. Agarda shart true qiymat qaytarsa navbatdagi ifoda bajariladi. Quyidagi misolni qaraymiz:

```
if (kattaSon>kichikSon)
```

```
    KattaSon=kichikSon;
```

Bu erda kattaSon va kichikSon o'zgaruvchilari taqqoslanayapti. Agarda kattaSon o'zgaruvchisi qiymati katta bo'lsa, bu navbatdagi qatorda unga qiymat sifatida kichikSon o'zgaruvchisining qiymati o'zlashtiriladi.

if operatorida figurali qavs ichiga olingan ifodalar blokini ham ishlatish mumkin.

```
if (shart)
```

```
{
```

```
    1 - ifoda
```

```
    2 - ifoda
```

```
    3 - ifoda
```

```
}
```

Quyida ifodalar blokining qo'llanilishiga oid misol keltirilgan

```
if(kattaSon>kichikSon)
```

```
{ kattaSon=kichikSon
```

```
    cout<<"kattaSon:"<<kattaSon << "\n";
```

```
    cout<<"kichikSon:"<<kichikSon<< "\n"; }
```

Bu holda kattaSon o'zgaruvchisiga nafaqat kichikSon o'zgaruvchisi o'zlashtirilayapti, balki ekranga bu haqida axborot ham chiqarilayapti.

1 – misol. Munosabat operatorining qo'llanilishi orqali tarmoqlanishga misol.

```
// 1. – misolda munosabat operatori bilan birgalikda if instruksiyasining
```

```
// qo'llanilishi namoyish etilgan
```

```
#include <iosrteam>
```

```
using namespace std;
```

```
int main( ) {
```

```
    int BuxoroGol, PaxtakorGol;
```

```
    cout<<"Buxoro komandasi kiritgan to'plar"
```

```
    << "sonini yozing:";
```

```
    cin >> BuxoroGol;
```

```
    cout<<"Paxtakor komandasi kiritgan"
```

```
    << "to'plar sonini yozing:";
```

```
    cin >> PaxtakorGol;
```

```
    cout << "\n";
```

```
    if ( BuxoroGol>PaxtakorGol)
```

```

cout << "Yashasin Buxoro!\n"
if (BuxoroGol < PaxtakorGol)
{
    cout << "Yashasin PaxtakorGol\n"
    cout << "Bugun Toshkentda bayram!\n";
}
if (BuxoroGol==PaxtakorGol)
{
    cout << "Durrangmi? Yo-oq? Bo`lishi"<<
    " mumkin emas\n";
    cout << "Paxtakorning kiritgan to`plari"
    << "haqida ma`lumotni qaytadan yozing\n"
    cin >> PaxtakorGol;
    if (BuxoroGol>PaxtakorGol)
    {
        cout<< "Buxoro yutishini oldindan bilgan"
        << " edim! Shuning uchun qayta so`radim\n";
        cout<< "Yashasin Buxoro!";
        if (BuxoroGol<PaxtakorGol)
        {
            cout<< "Paxtakor yutishini oldindan bilgan"
            << " edim! Shuning uchun qayta so`radim\n";
            cout<< "Yashasin Paxtakor!";
            cout << "Bugun Toshkentda bayram!\n";
            if (BuxoroGol==PaxtakorGol)
                cout<< "Qoyil! Haqiqatan ham during ekan\n";
        }
        cout<< "\n Ma`lumotingiz uchun rahmat\n";
    }
}
return 0; }

```

NATIJA:

```

Buxoro komandasi kiritgan to`plar sonini yozing:3
Paxtakor komandasi kiritgan to`plar sonini yozing:3
Durrangmi? Yo-oq? Bo`lishi mumkin emas
Paxtakorning kiritgan to`plari haqida ma`lumotni qaytadan yozing: 2
Buxoro yutishini oldindan bilgan edim! Shuning uchun qayta so`radim
Yashasin Buxoro!

```

**else kalit so`zi.** Dasturlarda ko`p hollarda biror bir shartning bajarilishiga (ya`ni bu shart true qiymat qaytarsa) bog`liq ravishda bir blok, uning bajarilmasligiga asosan esa (ya`ni bu shart false qiymat qaytarsa) boshqa bir blokning bajarilishi talab qilinadi. 1. – misolda birinchi tekshirish (BuxoroGol>PaxtakorGol) true qiymat qaytarsa ekranda bir xabar, false qiymatida esa boshqa bir xabar chiqariladi.

Bunday masalalarni yuqorida ko`rsatilgan usul, ya`ni qator shartlarni tekshirish uchun bir nechta if operatorini qo`llash orqali hal qilish mumkin, lekin bu tushunish uchun biroz murakkabroqdir. Dasturning soddaligini ta`minlash uchun else kalitli so`zidan foydalanish mumkin.

```
if (shart)
```

```
    Ifoda;
```

```
else
```

```
    Ifoda;
```

2 – misol. else kalitli so`zining ishlatilishi.

```

// 2 misol. If va else kalit so`zlarini ishlatilishiga oid misol
# include <iostream>
using namespace std;
int main()    {
    int BirinchiSon, IkkinchiSon;
    cout << "Katta sonni kiriting:
    cin >> BirinchiSon;
    cout<< "\n Kichik sonni kiriting:";
    cin >> IkkinchiSon;
    if (BirinchiSon > IkkinchiSon)
        cout << "\n Rahmat! \n";
    else
        cout << "\n Ikkinchisi katta son-ku!";
    return 0;    }

```

NATIJA:

Katta sonni kiriting: 10

Kichik sonni kiriting: 12

Ikkinchisi katta son - ku!

**if operatori orqali murakkab konstruktsiyalarni hosil qilish.** if–else konstruktsiyasida ifodalar blokida ixtiyoriy operatorlarni ishlatishda hech qanday chegara yo`q. SHu jumladan, ifodalar bloki ichida yana if–else operatorlarini ishlatish mumkin. Bu holda bir nechta if operatoridan iborat ichma – ich konstruktsiya hosil bo`ladi.

```

if (1–shart)
{
    if (2–shart)
        1–ifoda
    else
    {
        if (3–shart)
            2–ifoda
        else
            3–ifoda
    } }
else
    4–ifoda;

```

Ushbu bir nechta if operatoridan tashkil topgan konstruktsiya quyidagi tartibda ishlaydi: agarda 1–shart va 2–shart rost bo`lsa 1–ifoda bajariladi. Agarda 1–shart rost va 2–shart yolg`on natija qaytarsa, u holda 3–shart tekshiriladi va agarda bu shart rost bo`lsa 2–ifoda, yolg`on bo`lsa esa 3–ifoda bajariladi. Va eng oxiri, agarda 1–shart yolg`on bo`lsa 4–ifoda bajariladi. Bunday murakkab konstruktsiyaga misol 3 – misolda keltirilgan.

3 – misol. if operatori ichki bo`lgan murakkab konstruktsiya.

```

// 3 – misol. If operatori ichki bo`lgan murakkab konstruktsiyaga misol
# include <iostream.h>
using namespace std;
int main()    {
    // Ikkita son kiritamiz.Ularni BirinchiSon va IkkinchiSon o`zgaruvchilariga beramiz
    //Agarda KattaSon qiymati KichikSon qiymatidan katta bo`lsa, katta son kichigiga
    //qoldiqsiz bo`linishini tekshiramiz. Agarda u koldiksiz bo`linsa ular teng yoki
    //teng emasligini tekshiramiz.
    int BirinchiSon, IkkinchiSon;
    cout<< "Ikkita son kiriting.\n Birinchisi: ";

```

```

cin >> BirinchiSon;
cout << "\n Ikkinchisi:
cin >> IkkinchiSon;;
cout << "\n\n;
if (BirinchiSon>=IkkinchiSon;)
    if ( (BirinchiSon%IkkinchiSon)==0)
    {
        if (BirinchiSon==IkkinchiSon;)
            cout<< "Ular bir – biriga teng!\n";
    }
else
    cout<< "Birinchi son ikkinchisiga"<<
    "karrali!\n";
}
else
    cout<< "Ikkinchi son katta!\n"
return 0;    }

```

#### NATIJA

Ikkita son kiriting  
 Birinchisi:            9  
 Ikkinchisi:    3  
 Birinchi son ikkinchisiga karrali!

**Mantiqiy operatorlar.** Dasturlashda bir emas balki bir nechta shartli ifodalarni tekshirish zaruriyati juda ko'p uchraydi. Masalan, x o'zgaruvchisi y o'zgaruvchisidan, y esa o'z navbatida z o'zgaruvchisidan kattami sharti bunga misol bo'la oladi. Bizning dasturimiz mos amalni bajarishdan oldin bu ikkala shart rost yoki yolg'onligini tekshirishi lozim.

Quyidagi mantiq asosida yuqori darajada tashkil qilingan signalizatsiya sistemasini tasavvur qiling. Agarda eshikda signalizatsiya o'rnatilgan bo'lsa VA kun vaqti kech soat olti VA bugun bayram YOKI dam olish kuni BO'LMASA politsiya chaqirilsin. Barcha shartlarni tekshirish uchun C++ tilining uchta mantiqiy operatori ishlatiladi.

Mantiqiy operatorlar

| Operator | Belgi | Misol            |
|----------|-------|------------------|
| VA       | &&    | 1ifoda && 2ifoda |
| YOKI     |       | 1ifoda  2ifoda   |
| INKOR    | !     | !ifoda           |

**Mantiqiy ko'paytirish operatori.** Mantiqiy ko'paytirish operatori ikkita ifodani hisoblaydi, agar ikkala ifoda true qiymat qaytarsa VA operatori ham true qiymat qaytardi. Agarda sizning qorningiz ochligi rost bo'lsa VA sizda pul borligi ham rost bo'lsa siz supermarketga borishingiz va u erdan o'zingizga tushlik qilish uchun biror bir narsa xarid qilishingiz mumkin. YOki yana bir misol, masalan,

```
if(x==5)&&(y==5)
```

mantiqiy ifodasi agarda x va u o'zgaruvchilarini ikkalasining ham qiymatlari 5 ga teng bo'lsagina true qiymat qaytaradi. Bu ifoda agarda o'zgaruvchilardan birortasi 5 ga teng bo'lmagan qiymat qabul qilsa false qiymatini qaytaradi. Mantiqiy ko'paytirish operatori faqatgina o'zining ikkala ifodasi ham rost bo'lsagina true qiymat qaytaradi.

Mantiqiy ko'paytirish operatori && belgi orqali belgilanadi.

**Mantiqiy qo'shish operatori.** Mantiqiy qo'shish operatori ham ikkita ifoda orqali hisoblanadi. Agarda ulardan birortasi rost bo'lsa mantiqiy qo'shish operatori true qiymat qaytaradi. Agarda sizda pul YOKI kredit kartochkasi bo'lsa, siz schyotni to'lay olasiz. Bu holda ikkita shartning birdaniga bajarilishi: pulga ham va kredit kartochkasiga ham ega bo'lishingiz shart emas. Sizga ulardan birini bajarilishi etarli. Bu operatorga oid yana bir misolni qaraymiz. Masalan,

```
if(x==5)||(y==5)
```

ifodasi yoki x o`zgaruvchi qiymati, yoki u o`zgaruvchi qiymati, yoki ikkala o`zgaruvchining qiymati ham 5 ga teng bo`lsa rost qiymat qaytaradi.

**Mantiqiy inkor operatori.** Mantiqiy inkor operatori tekshirilayotgan ifoda yolg`on bo`lsa true qiymat qaytaradi. Agarda tekshirilayotgan ifoda rost bo`lsa inkor operatori false qiymat qaytaradi. Masalan,

```
(if!(x==5))
```

ifodasining qiymati, agarda x o`zgaruvchisi 5 ga teng bo`lmasa true qiymat qaytaradi. Bu ifodani boshqacha ham yozish mumkin:

```
if(x!=5)
```

**switch operatori.** Ayrim masalalarni echishda if operatori ichida ko`p sondagi if operatorlarini qo`llashga to`g`ri keladi. Bu esa dasturni yozishni ham, uni tushinishni ham murakkablashtirib yuboradi. Bunday muammoni echish uchun C++ tilida switch operatori qo`llaniladi. Bu operatorning if operatoridan asosiy farqi shuki, unda bir yo`la bir nechta shart tekshiriladi. Natijada dasturni tarmoqlanishi nisbatan samaraliroq bo`ladi. switch operatorining sintaksisi quyidagichadir.

```
switch(ifoda)
{
  case Birinchi qiymat:  operator;
                        break;

  case Ikkinchi qiymat: operator;
                        break;

  .....

  case N nchi qiymat:   operator;
                        break;

  default:              operator;
}

```

switch operatoridan keyingi qavs ichida tilning konstruktsiyasi nuqtai–nazaridan to`g`ri bo`lgan ixtiyoriy ifodani ishlatish mumkin. Operator identifikatori o`rnida ham ixtiyoriy operator yoki ifoda, hamda operator va ifodalarning ketma-ketligini ishlatish mumkin. Lekin bu erda mantiqiy operatsiyalar yoki taqqoslash ifodalarni ishlatish mumkin emas.

**switch operatorining qo`llanilishi.** switch operatorining qo`llanilish sintaksisi quyidagicha:

```
switch(ifoda)
{
  case 1-nchi qiymat: ifoda;
  case 2-nchi qiymat: ifoda;
  ...
  case n-nchi qiymat: ifoda;
  default : ifoda;
}

```

switch operatori orqali dasturning tarmoqlanishi bir necha mumkin bo`lgan qiymatlarni qaytaruvchi ifodaning natijasi asosida tashkil etiladi. switch operatoridagi qavs ichida berilgan ifodaning qaytargan qiymati case operatoridan keyinda ko`rsatilgan qiymat bilan solishtiriladi. Ifodaning qiymati bilan case operatoridan keyingi qiymat mos kelsa tanlangan case operatoridan keyingi barcha satrlar bajariladi. Bunda amallarni bajarilishi break operatorigacha davom etadi.

Agarda case operatorlari qiymatidan birortasi ham qaytarilgan qiymatga mos kelmasa default operatoridan keyingi dastur satrlari bajariladi. Agarda bu operator mavjud bo`lmasa boshqaruv switch bloki tanasidan chiqadi va keyingi dastur satrlariga beriladi.

4-misol.

```
switch(choice)
{
  case 0:

```

```

        cout<< "zero!"<< endl;
        break;
case 1:
        cout<< "one!"<< endl;
        break;
case 2:
        cout<< "two!"<<endl;
        break;
default:
        cout<< "default!"<<endl;
}
5-misol
switch (choice)
{
case 0:
case 1:
case 2:
        cout< "Less than 3!"<< endl;
        break;
case 3:
        cout<< "Equals 3!" << endl;
        break;
default:
        cout<< "Greater than 3 !" << endl;}

```

Operator yoki ifodalardan keyin break operatori qo'llanilmasa joriy case operatoridan keyingi case blokidagi barcha ifodalar bajariladi. Ko'p hollarda bunday situatsiyada xatolik ro'y beradi. Shuning uchun, break operatorini tushirib qoldirsangiz bu amalni tavsiflovchi mos izohni yozishni unutmang.

switch operatorining qo'llanilishiga misol 6.-misolda keltirilgan.

6 – misol. switch operatorining qo'llanilishi.

```

include <iostream.h>
using namespace std;
int main()
{ unsigned short int number;
  cout<< "1 dan 5 gacha bo'lgan sonlarning birini kiriting va Enter klavishini bosing: "
  cin>> number;
  switch (number)
  {
    case 0: cout << "Salom!";
            break;
    case 5: cout<< "Amirxon\n";
    case 4: cout << "Rayhon\n" ;
    case 3: cout<< "Madina\n";
    case 2: cout << "Gulruh\n"
    case 1: cout << Bobur\n";
            break;
    default: cout << "Hisobdan tashqari.\n";
            break;
  }
  cout<< "\n\n";
  return 0;}
HATIJA:

```

1 dan 5 gacha bo'lgan sonlarning birini kiriting va Enter klavishini bosib: 3  
Madina  
Gulruh  
Bobur

1 dan 5 gacha bo'lgan sonlarning birini kiriting va Enter klavishini bosib: 8  
Hisobdan tashqari.

TAHLIL. Dastur oldin son kiritishni so'raydi. Keyin esa kiritilgan son switch operatori orqali tekshiriladi. Agarda 0 kiritilgan bo'lsa unga muvofiq ravishda ekranga 9 – satrda yozilgan kichik son xabari chiqariladi. Va undan keyin yozilgan break operatori switch konstruktsiyasini bajarilishini yakunlaydi. Agarda 5 soni kiritilsa boshqaruv 11 – satrga beriladi va unga muvofiq xabar chiqariladi. Undan keyingi toki break so'ziga barcha satrlar ketma – ket bajariladi.

Agarda dasturga 5 dan katta son kiritilsa 17 – satrdagi default bajariladi, ya'ni ekranga Big xabari chiqariladi.

#### Nazorat savollari:

1. Ifoda nima?
2.  $x=5+7$  yozuvi ifoda bo'la oladimi? Uning qiymati nechaga teng?
3.  $201/4$  ifodaning qiymati nechaga teng?
4.  $201\%4$  ifoda qiymati nechaga teng?
5. Agarda MyAge, a va b o'zgaruvchilarining tiplari int bo'lsa ularning qiymatlari quyidagi ifodalar bajarilgandan so'ng nechaga teng bo'ladi.

MyAge = 39;

a=MyAge++

b=++MyAge

6. If(x = 3) va if(x ==3) ifodalarning farqi nimadan iborat?

## 9 mavzu. C++ dasturlash tizimida takrorlanuvchi jarayonlarni dasturlash

### Reja:

1. Takrorlash (tsikl) operatorlari. Ularning turli shakllari (parametrlil, shartni oldin va shartni keyin tekshiruvchi operatorlar).
2. Takrorlanuvchi strukturali dasturlar.
3. Funktsiyalarni joriy qilish, parametrni o'tkazish, qaytish qiymatlari, qaytish qiymatlarsiz funksiyalar, muammo yechimi: qayta ishlatiladigan funksiyalar, muammo yechimi: bosqichma bosqich detallashtirish, o'zgaruvchini aniqlash maydoni va global o'zgaruvchilar, izohlar parametri, funksiyalar.

**Tayanch iboralar.** Takrorlash (tsikl) operatorlar, parametrli, shartni oldin va shartni keyin tekshiruvchi operatorlar, funktsiya, parametr, argument, standart matematik funksiyalar, lokal o'zgaruvchi, global o'zgaruvchi, statik, dinamik o'zgaruvchi, funktsiya prototiplari.

### 1. Takrorlash (tsikl) operatorlari

Har qanday dasturning strukturasi tarmoqlanish va takrorlanish to'plamining kombinatsiyasidan iborat bo'ladi. Oldingi mavzuda dasturning tarmoqlanishi if operatori orqali tashkil etilishini ko'rib chiqqan edik. Ushbu ma'ruzada biz quyidagilarni bilib olamiz:

1. Takrorlanish nima va ular qanday ishlatiladi?
2. Takrorlanishni tashkil etishning qanday usullari bor?
3. Ko'p tarmoqlanuvchi holda if/else konstruktsiyasi o'rniga boshqa konstruktsiyani ishlatilishi.

**Takrorlanishni tashkil etish.** Qator masalalarni echish uchun ko'pincha bitta amalni bir necha marotaba bajarish talab qilinadi. Amaliyotda bu rekursiyalar va iterativ algoritmlar yordamida amalga oshiriladi. Iterativ jarayonlar – bu operatsiyalar ketma-ketligini zaruriy sonda takrorlanishidir.

Takrorlash operatori «Takrorlash sharti» deb nomlanuvchi ifodaning rost qiymatida dasturning ma'lum bir qismidagi operatorlarni (takrorlash tanasini) ko'p marta takror ravishda bajaradi (itarativ jarayon).

Takrorlash o'zining kirish va chiqish nuqtalariga ega, lekin chiqish nuqtasining bo'lmashligi mumkin. Bu holda takrorlashga cheksiz takrorlash deyiladi. Cheksiz takrorlash uchun takrorlashni davom ettirish sharti doimo rost bo'ladi.

Takrorlash shartini tekshirish takrorlash tanasidagi operatorlarni bajarishdan oldin tekshirilishi mumkin (for, while takrorlashlari) yoki takrorlash tanasidagi operatorlari bir marta bajarilgandan keyin tekshirilishi mumkin (repeat-until, do-while).

#### **while operatori orqali takrorlanishni tashkil etish.**

while operatori yordamida takrorlanishni tashkil etishda operatsiyalar ketma-ketligi takrorlashning davom etish sharti "to'g'ri" bo'lsagina uning navbatdagi operatsiyalari amalga oshiriladi.

1 – misol. while operatori yordamida takrorlashni tashkil etish.

```
include <iostream>
using namespace std;
int main() {
int counter=0; //Birlamchi kiymatni o`zlashtirish
while(counter<5)//Takrorlash shartini tekshirish
{counter ++;
cout << "counter : " << counter << ".\n";}
cout<<"Tsikl tugadi.Counter:"<<counter<<".\n";
return 0; }
```

NATIJA:

```
counter : 1
counter : 2
counter : 3
counter : 4
```



counter : 5

Tsikl tugadi.Counter: 5.

*while operatori orqali murakkab konstruktsiyalarni tuzish.*

while operatori shartida murakkab mantiqiy ifodalarni ham qo'llash mumkin. Bunday ifodalarni qo'llashda && (mantiqiy ko'paytirish), || (mantiqiy qo'shish), hamda !(mantiqiy INKOR) kabi operatsiyalardan foydalaniladi. 2. - misolda while operatori konstruktsiyasida murakkabroq shartlarni qo'yilishiga misol keltirilgan .

2 – misol. while konstruktsiyasidagi murakkab shartlar.

```
include <iostream>
using namespace std;
int main() {
    unsigned short kichik;
    unsigned long katta;
    const unsigned short MaxKichik=65535;
    cout << "Kichik sonni kiriting:";
    cin >> kichik;
    cout << "Katta sonni kiriting:";
    cin >> katta;
    cout << "kichik son:" << kichik << "...";
    //Har bir iteratsiyada uchta shart tekshiriladi.
    while (kichik<katta && katta>0 &&
           kichik< MaxKichik )
    {
        if(kichik%5000==0) //Xar 5000 satrdan
            //keyin nuqta chikariladi
            cout<< ".";
        kichik++;
        katta-=2;
    }
    cout<< "\n kichik son:" << kichik << " katta son:"
    << katta << endl;
    return 0; }
```

NATIJA:

Kichik sonni kirit : 2

Katta sonni kirit : 100000

Kichik son : 2 .....

Kichik son :33335 katta son : 33334

TAHLIL

Misolda keltirilgan dastur quyidagi mantiqiy o'yinni ifodalaydi. Oldin ikkita son – kichik va katta kiritiladi. Undan so'ng toki ular bir biriga teng bo'lmaguncha, ya'ni "uchrashmaguncha" kichik son birga oshiriladi, kattasi esa ikkiga kamaytiriladi. O'yinni maqsadi qiymatlar "uchrashadigan" sonni topishdir.

9 – 12 – satrlarda qiymatlar kiritiladi. 15 – satrda esa takrorlashni davom ettirishning quyidagi uchta sharti tekshiriladi:

1. kichik o'zgaruvchisi qiymati katta o'zgaruvchisi qiymatidan oshmasligi.
2. katta o'zgaruvchisi qiymati manfiy va nolga teng emasligi.
3. kichik o'zgaruvchisi qiymati MaxKichik qiymatidan oshib ketmasligi.

17 – satrda esa kichik soni 5000 ga bo'lingandagi qoldiq hisoblanadi. Agarda kichik 5000 ga qoldiqsiz bo'linsa bu operatsiyaning bajarilishi natijasi 0 ga teng bo'ladi. Bu holatda hisoblash jarayonini vizual ifodasi sifatida ekranga nuqta chiqariladi. Keyin esa kichik qiymati bittaga oshiriladi, katta qiymati esa 2 taga kamaytiriladi. Takrorlash agarda tekshirish sharti tarkibidagi birorta shart bajarilmasa to'xtatiladi va boshqaruv 24 – satrga o'tadi.

**break va continue operatorlari.** Ko`pincha takrorlashning navbatdagi iteratsiyasiga takrorlash tanasidagi boshqa operatorlar (navbatdagi operatorlar) bajarilmasdan turib o`tish zaruriyati tug`iladi. Bunday holatlarda continue operatori qo`llaniladi. Bundan tashqari, takrorlashni bajarilishi sharti qanoatlanirilganda ham, qator hollarda undan chiqib ketish zaruriyati paydo bo`ladi. Bu holda esa break operatori ishlatiladi. Bunday operatorlarni qo`llanilishiga misol 3. – misolda keltirilgan. Bu misol bizga oldingi misoldan tanish bo`lgan o`yinning biroz murakkablashtirilgan variantidir. Bu erda kichik va katta qiymatlardan tashqari qadam va maqsadli kattalikni kiritish talab etiladi. Agarda kichik son qiymati qadam o`zgaruvchisiga (qadam) karrali bo`lmasa katta qiymati ikkiga kamaytiriladi. Qachonki, kichik o`zgaruvchisi qiymati katta o`zgaruvchisi qiymatidan katta bo`lsa o`yin tugatiladi. Agarda katta o`zgaruvchisi qiymati maqsadli kattalik(maqsad)ning qiymati bilan ustma – ust tushsa o`yin bekor qilinganligi haqida xabar ekranga chiqariladi.

3 – misol. break va continue operatorlarining qo`llanilishi.

```
include <iostream>
using namespace std;
int main()      {
    unsigned short kichik ;
    unsigned long  katta;
    unsigned long  qadam;
    unsigned long  maqsad ;
    const unsigned short MaxKichik = 65535;
    cout<< "Kichik nomerni kiriting:";
    cin >>kichik ;
    cout<< "Katta nomerni kiriting :";
    cin >>katta ;
    cout<< "Qadam qiymatini kiriting:";
    cin >>qadam ;
    cout<< "Maqsadli kattalik qiymatini kiriting:;
    cin >> maqsad ;
    cout << "\n";
    while(kichik<katta && katta>0 &&
    kichik<MaxKichik)
    {
        kichik++;
        if(kichik%qadam==0)
        {      cout << "qadam:" << kichik << endl ;
                continue ;          }
        if(katta==maqsad) //maqsadli nuqtaga
        // tengligini tekshirish
        {
            cout << "Maqsadga erishildi !;
            break;
        }
        katta -= 2;
    }
    cout<< "\n Kichik son:" << kichik <<
    << " katta son:" << katta << endl ;
    return 0;}

```

NATIJA:

Kichik sonni kiriting: 2

Katta sonni kiriting: 20

Qadam qiymatini kiriting: 4

Maqsadli kattalik qiymatini kiriting: 6

Qadam :4

Qadam: 8

Kichik son : 10 Katta son:8

*while(true) konstruktsiyasini qo'llanilishi.*

Takrorlashning navbatdagi iteratsiyasiga o'tishda shart sifatida C++ tilida sintaksisi bo'yicha to'g'ri bo'lgan ixtiyoriy ifoda qatnashishi mumkin. Bunda shart "to'g'ri" bo'lsa takrorlash bajarilaveradi. Cheksiz takrorlanishni tashkil etish uchun shart sifatida true mantiqiy o'zgarma qo'llaniladi. Bu 4. - misolda ko'rsatilgan.

4 – misol. while operatorini qo'llashga oid yana bir misol.

```
#include <iostream>
using namespace std;
int main() {
int counter = 0 ;
while(true)
{
counter++;
if(counter>10)
break ;
}
cout<<"counter:"<<counter << "\n " ;
return 0 ; }
```

HATIJA:

Counter: 11

***do...while konstruktsiyasi yordamida takrorlash tashkil etish.***

Ayrim hollarda while operatori yordamida takrorlanishni tashkil etishda uning tanasidagi amallar umuman bajarilmasligi mumkin. Chunki takrorlashni davom etish sharti har bir iteratsiyadan oldin tekshiriladi. Agarda boshlang'ich berilgan shart to'g'ri bo'lmasa takrorlash tanasining birorta operatori ham bajarilmaydi. Bu holga 5. – misolda misol keltirilgan.

5–misol. while takrorlashi tanasidagi amallar bajarilmay qolishi

```
#include <iostream>
using namespace std;
int main() {
int counter ;
cout << "How manu hellos ?:";
cin >> counter ;
while (counter>0 )
{ cout << "Hello !\n";
counter -- ; }
cout<<"Counter is OutPut ;" << counter ;
return 0; }
```

HATIJA:

How manu hellos ? : 2

Hello !

Hello !

counter is OutPut : 0

How manu hellos ? : 0

counter is OutPut : 0

*do...while konstruktsiyasining qo'llanilishi.*

do...while konstruktsiyasida takrorlash sharti uning tanasidagi operatsiyalar bir marta bajarilgandan so'ng tekshiriladi. Bu takrorlash operatorlarini hech bo'lmaganda bir marta bajarilishini kafolatlaydi.

Quyidagi dasturda oldingi dasturda keltirilgan variantning bir oz o`zgartirilgan shakli, ya`ni while operatori o`rniga do...while konstruktsiyasi qo`llangan shakli keltirilgan.

6–misol. do...while konstruktsiyasining qo`llanilishi.

```
#include <iostream>
using namespace std;
int main()      {
int counter;
cout<<"How manu hellos ?";
cin >>counter;
do
{
cout << "hello \h" ;
counter --;
}
while(counter>0)
cout << "Counter is : " << counter <<endl ;
return 0 ; }
```

HATIJA :

```
how manu hellos ? 2
hello
hello
Sounter is : 0
How manu hellos ? 0
Hello
Counter is: - 1
```

### **for operatori.**

while operatori yordamida takrorlanishni tashkil etishda 3 ta zaruriy amallar: takrorlash o`zgaruvchisiga boshlang`ich qiymat berish, har bir iteratsiyada takrorlashni davom etish sharti bajarilishini tekshirish va takrorlash o`zgaruvchisi qiymatini o`zgartirishni bajarishimiz kerak.

7 – misol. while operatorining ishlatilishiga yana bir misol

```
#include <iostream>
using namespace std;
int main()      {
int counter=0;
while (counter <5)
{
counter++ ;
cout << "Looping!"
}
cout << "\n Counter:" << Counter << "\n";
return 0;      }
```

NATIJA:

```
Looping! Looping! Looping! Looping! Looping!
Counter: 5
```

for operatori takrorlashni ishlashi uchun zarur bo`ladigan uchta operatsiyani o`zida birlashtiradi. Bu operatsiyalarni qisqacha quyidagicha xarakterlash mumkin: boshlang`ich qiymatni o`zlashtirish, shartni tekshirish, takrorlash schyotchigini qiymatini oshirish. for operatori ifodasidagi qavsning ichida shu uchchala operatsiyani amalga oshiruvchi ifodalar yoziladi. Qavs ichidagi ifodalar nuqtali vergul orqali ajratiladi.

for takrorlashining birinchi ifodasi takrorlash schyotchigiga boshlang`ich qiymatni o`zlashtiradi. Schyotchik – to`g`ridan-to`g`ri for takrorlashida e`lon qilinadigan va qiymat

o`zlashtiriladigan butun sonli o`zgaruvchidir. C++ da bu o`rinda schyotchikka qiymat beradigan ixtiyoriy ifoda yozilishiga imkon berilgan. for takrorlashining ikkinchi parametrda takrorlashni davom etish sharti aniqlanadi. Bu shart while konstruktsiyasining sharti bajaradigan vazifani amalga oshiradi. Uchinchi parametrda esa takrorlash schyotchigi qiymatini o`zgartiruvchi (oshiruvchi yoki kamaytiruvchi) ifoda yoziladi. for takrorlashidan foydalanishga misol 8. – misolda keltirilgan.

8 – misol. for takrorlashining qo`llanilishiga misol.

```
#include< iostream>
using namespace std;
int main()      {
int counter;
for (counter=0 ; counter<5; counter++ )
cout<< "Looping!";
cout<< "\n Counter:" << counter<< ".\n";
return 0;      }
```

NATIJA:

Looping! Looping! Looping! Looping! Looping!

Counter: 5

*for operatori uchun murakkab ifodalarni berilishi.* for takrorlashi dasturlashning kuchli va qulay instrumentidir. for operatorida takrorlashni o`zaro bog`liq bo`lmagan parametrlar (boshlang`ich qiymat o`zlashtirish, bajarilish sharti va qadam) ni qo`llanilishi takrorlash ishini boshqarishda juda yaxshi imkoniyatlarni ochib beradi.

for takrorlashi quyidagi ketma–ketlikda ishlaydi.

1. Takrorlash schetchigiga boshlang`ich qiymat o`zlashtiriladi.
2. Takrorlashni davom etish shartidagi ifoda qiymati hisoblanadi.
3. Agarda shart ifodasi true qiymat qaytarsa oldin takrorlash tanasi bajariladi, keyin esa takrorlash schyotchigi ustida berilgan amallar bajariladi.

Har bir iteratsiyada 2 – va 3 – qadamlar takrorlanadi.

*Takrorlashda bir nechta schyotchikni qo`llanilishi.* for takrorlashining sintaksisi unda bir nechta o`zgaruvchi - schetchikni qo`llanilishiga, takrorlashni davom etishini murakkab shartlarini tekshirishga va takrorlash schyotchiklari ustida ketma-ket bir nechta operatsiyani bajarilishiga imkon beradi.

Agarda bir nechta schyotchikka qiymat o`zlashtirilsa yoki ular o`rtasida bir nechta operatsiya bajarilsa, bu ifodalar vergul bilan ajratilgan holda ketma – ket yoziladi. 9. – misolda ikkita schyotchikli for operatorining qo`llanilishini ko`rib chiqamiz.

9–misol. for takrorlashida bir nechta schyotchikni qo`llanilishi.

```
#include< iostream>
using namespace std;
int main() {
for (int i=0, j=0; i<3; i++, j++)
cout<< "i:" <<i<< "j:" <<j<< endl;
return 0;}
```

HATIJA:

i: 0                    j: 0

i: 1                    j: 1

i: 2                    j: 2

*for takrorlashida nol parametrlarni ishlatilishi.* for takrorlashining ixtiyoriy parametri tushirib qoldirilishi mumkin. Nol parametrlar for takrorlashini boshqa parametrlaridan nuqtali vergul (;) bilan ajratiladi. Agarda for takrorlashini 1 – va 3 – parametrlarini tushirib qoldirsak, u xuddi while operatoridek qo`llaniladi.

10 – misol. for takrorlashining nol parametrlari.

```

#include < iostream>
using namespace std;
int main() {
int counter=0;
for ( ; counter<5 ; )
{
counter++;
cout << "Looping!";
}
cout<< "\n Counter:" << counter<< ".\n";
return 0;}

```

NATIJA:

Looping! Looping! Looping! Looping! Looping!

Counter: 5

TAHLIL

Bu takrorlashni bajarilishi 7. – misolda keltirilgan while takrorlashini bajarilishiga o`xshash tarzda amalga oshiriladi. 4-satrdan sounter o`zgaruvchisiga qiymat o`zlashtirilayapti. for takrorlashida esa parametr sifatida faqatgina takrorlashni davom etish shartini tekshirish ifodasi ishlatilgan. Takrorlash o`zgaruvchisi ustida operatsiya ham tushirib qoldirilgan. Bu holda ushbu takrorlashni quyidagicha ifodalash mumkin:

```
while(counter<5)
```

Qaralgan misolimiz yana bir marta C++ tilida bir xil masalani bir necha usul bilan hal qilish imkoniyati borligini ko`rsatadi. Bundan tashqari, for takrorlashining 3 ta parametrini ham tushirib qoldirish va takrorlashni break va continue operatorlarini qo`llash orqali boshqarish mumkin. for konstruktsiyasini parametrlarsiz qo`llanilishiga misol 11 – misolda ko`rsatilgan.

11 – misol. for operatorlarini parametrlarsiz qo`llanilishi.

```

#include< iostream>
using namespace std;
int main() {
int counter=0;
int max;
cout<< "How many hellos?";
cin>> max;
for( ; ; )
{
if (counter <max)
{
cout << "Hello! \n";
counter++;
}
else
break;
}
return 0;}

```

NATIJA:

How many hellos? 3

Hello!

Hello!

Hello!

for takrorlashining tanasi bo`sh bo`lgan holda qo`llanilishi. Takrorlashda for operatori orqali uning tanasiga hech qanday operator yozmasdan turib ham biror bir amalni bemalol bajarish

mumkin. Bunda takrorlash tanasi bo`sh satrdan iborat bo`ladi. Tanasi bo`sh bo`lgan for takrorlashiga misol 12. – misolda keltirilgan.

12 – misol. For takrorlashining tanasi bo`sh bo`lgan holda qo`llanilishi.

```
#include< iostream>
using namespace std;
int main() {
for (int i=0; i<5; cout<< "i" <<i++ <<endl)
;
return 0; }
```

HATIJA:

i:0  
i:1  
i:2  
i:3  
i:4

*Ichki takrorlanish.* Boshqa takrorlashning ichida tashkil etilgan takrorlash ichki takrorlash deb aytiladi. Bu holda ichki takrorlash tashqi takrorlashni har bir iteratsiyasida to`liq bajariladi.

13. – misolda matritsa elementlarini ichki takrorlash orqali to`ldirilishi namoyish qilingan.

13 – misol. Ichki takrorlanish.

```
#include< iostream>
using namespace std;
int main()      {
int rows, columns;
char theChar;
cout << "How many rovs?";
cin >> rows;
cout << "How many columns?";
cin >> columns;
cout << "What character?";
cin>>theChar;
for ( int i=0; i<rows; i++)
{
for (int j=0; j<columns; j++)
cout << the char;
cout<< "\n";
}
return 0; }
```

HATIJA:

How many rows? 4  
How many columns? 12  
What character? x  
x x x x x x x x x x x  
x x x x x x x x x x x  
x x x x x x x x x x x  
x x x x x x x x x x x

*for takrorlashi schyotchigining ko`rinish sohasi.* ANSI ning yangi standarti bo`yicha takrorlashda e`lon qilingan o`zgaruvchining ko`rinish sohasi faqat takrorlash ichidangina iborat. Lekin ko`pgina kompilyatorlar eski standartlarni ham qo`llab – kuvvatlaydilar. Quyida keltirilgan dastur kodini kiritib o`zingizning kompilyatoringiz yangi standartga mos kelish- kelmasligini tekshirishingiz mumkin.

14 – misol.

```
#include <iostream>
```

```
using namespace std;
int main() {
    for ( int i = 0; i<5; i++ )
    {
        cout << " i: " << i << endl ;
    }
    i=7; // i ko`rinish sohasi chegarasidan tashqarida
    return 0;}

```

Agarda bu dastur xatoliksiz kompilyatsiya qilinsa demak u ANSI ning yangi standartini qo'llab - quvvatlamaydi. Yangi standartga muvofiq kompilyatorlar i=7 ifoda uchun xatolik xaqida xabar berishi kerak. Dastur kodiga ozgina o`zgartirish kiritilganda so`ng dastur barcha kompilyatorlar uchun xatoliksiz ishlaydi.

15 – misol.

```
#include <iostream>
using namespace std;
int main() {
    int i;
    for ( int i = 0; i<5; i++ )
    {
        cout << " i: " << i << endl ;
    }
    i=7; //Endi i barcha kompilyatorlar tomonidan //xatoliksiz qabul qilinishi mumkin.
    return 0; }

```

## 2. Funktsiyalar

Ko'p hollarda dasturning biror–bir qismini turli xil qiymatlarda ko'p marta takroran bajarishga to'g'ri keladi. Bunday hollarda dasturning ushbu qismi alohida bir qism sifatida ajratib qo'yiladi. Qachonki, ushbu qismga tegishli hisoblashlarni bajarishga to'g'ri kelsa, bu qismga kerakli qiymatlarni bergan holda murojaat tashkil qilinadi. Agar bu qism natija qaytarsa, u funktsiy, aks holda prosedura deyiladi.

Funktsiyaga bir necha qiymat uzatishi mumkin, biroq u bitta qiymat qaytariladi. Funktsiya ichida dasturlashning barcha qoidalari bajariladi.

Funktsiyalar dasturchi ishini juda yengillashtiradi. Funktsiyalar yordamida dastur modullashadi, qismlarga bo'linadi. Bu esa keyinchalik dasturni rivojlantirishni osonlashtiradi. Bunda dasturchi yozgan funktsiyalar C++ ning standart kutubhonasi va boshqa kutubhonalar ichidagi funktsiyalar bilan birlashtiriladi. Bu esa ishni osonlashtiradi. Ko'p holda dasturda takroran bajariladigan amalni funktsiya sifatida yozish va kerakli joyda ushbu funktsiyani chaqirish mumkin. Dastur yozilish davrida hatolarni topishni yengillashtiradi.

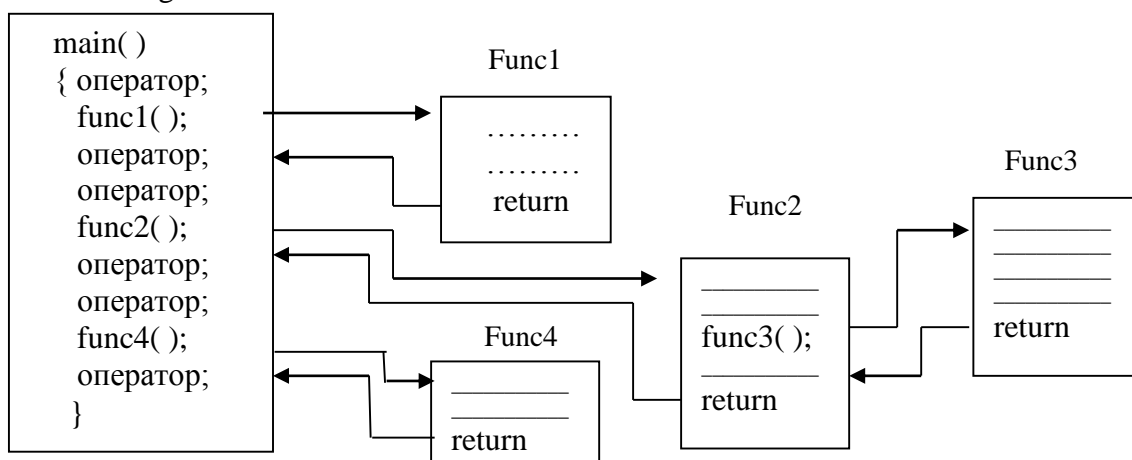
C++ tilida asosiy funktsiya main() hisoblanadi. Dasturda ushbu funktsiya albatta bo'lishi shart, dasturning bajarilishi shu funktsiyabilan boshlanadi, boshqa barcha funktsiyalar ana shu funktsiya orqali chaqiriladi. Dasturning qismida qaysi funktsiya nomi yozilgan bo'lsa, boshqaruv shu funktsiya tanasiga o'tadi. Bu jarayon funktsiyani chaqirilishi (yoki funktsiyaga murojaat qilish) deb aytiladi. Funktsiya ishini tugatgandan so'ng dastur o'z ishini funktsiya chaqirilgan qatorning keyingisidan boshlab davom ettiradi, ya'ni boshqarish murojaat qilingan dastur qismiga qaytariladi.

Funktsiya dasturining nom berilgan shunday qismiki, unga dasturning boshqa funktsiyalarning tuzilish usuliga to'xtalib o'tishdan oldin funktsiyaga har bir bo'limdan qancha murojat qilish talab qilinsa, shuncha murojat qilish mumkin.

Funktsiya bu ma'nosiga ko'ra dastur osti bo'lib, u ma'lumotlarni o'zgartirishi va biror bir qiymat qaytarishi mumkin. C++ da har bir dastur hech bo'lmaganda bitta main() funktsiyasiga ega bo'ladi. main() funktsiyasi dastur ishga tushirilishi bilan operatsion sistema tomonidan avtomatik chaqiriladi. Boshqa funktsiyalar esa u tomonidan chaqirilishi mumkin.



Har bir funktsiya o`zining nomiga egadir. Qachonki, dasturda bu nom uchrasa boshqaruv shu funktsiya tanasiga o`tadi. Bu jarayon funktsiyani chaqirilishi (yoki funktsiyaga murojaat qilish) deb aytiladi. Funktsiya ishini tugatgandan so`ng dastur o`z ishini funktsiya chaqirilgan qatorning keyingisidan boshlab davom ettiradi. Dastur bajarilishining bunday sxemasi yuqoridagi rasmda ko`rsatilgan.



Funktsiyaning chaqirilishi tartibi

**Standart va foydalanuvchi funktsiyalari.** C++ tili foydalanuvchilarga juda ko'p funktsiyalarni taqdim qiladi. Ular kodi avvaldan yozib qo'yilgan. Foydalanuvchi kerakli funktsiya nomini va parametrlari qiymatini berishi kifoya qiladi. Ushbu funktsiyalar standart funktsiyalar deyiladi. Ular ma'lum bir guruhlar shaklida jamlangan.

**Standart matematik funktsiyalar.** Standart matematik funktsiyalari ko'pgina amallarni bajarishga imkon beradi. Masalan bizning dasturimizda quyidagi satr bor bo'lsin:

```
double = k;
int m = 123;
k = sin(m);
```

Kompilyator ushbu satrni ko'rganida, standart kutubhonadan sin funktsiyasini chaqiradi. Kirish qiymati sifatida m ni berdik. Javob, ya'ni funktsiyadan qaytgan qiymat k ga berildi. Funktsiya agumentlari o'zgarmas sonlar (const), o'zgaruvchilar, ifodalar va boshqa mos keluvchi qiymat qaytaradigan funktsiyalar bo'lishi mumkin. Masalan:

```
int g = 49, k = 100;
cout << "4900 ning ildizi -> " << sqrt( g * k );
```

Ekkranda:

```
4900 ning ildizi -> 70;
```

Matematik funktsiyalar aksariyat hollarda double tipidagi qiymat qaytarishadi.

Kiruvchi argumentning tipi sifatida esa double ga keltirilishi mumkin bo'lgan tip beriladi. Bu funktsiyalarni ishlatish uchun math.h (yangi ko'rinishda cmath) e'lon faylini include bilan asosiy dastur tanasiga kiritish kerak.

Quyida matematik funktsiyalar kutubhonasining ba'zi bir a'zolarini keltiramiz. x va y o'zgaruvchilari double tipiga ega.

Matematik funktsiyalar

| Funktsiya      | Aniqlanishi                                                                    | Misol                                                          |
|----------------|--------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>ceil(x)</b> | x ni x dan katta yoki unga teng bo'lgan eng kichik butun songacha yaxlitlaydi. | ceil(12.6) = 13.0<br>ceil(-2.4) = -2.0                         |
| <b>cos(x)</b>  | x ning trigonometrik kosinusi (x radianda).                                    | cos(0.0) = 1.0                                                 |
| <b>exp(x)</b>  | e ning x chi darajasi (eskponetsial funktsiya).                                | exp(1.0) = 2.71828<br>exp(2.0) = 7.38906                       |
| <b>fabs(x)</b> | x ning absolut qiymati.                                                        | x>0 => abs(x) = x<br>x=0 => abs(x) = 0.0<br>x<0 => abs(x) = -x |

|                  |                                                                 |                                          |
|------------------|-----------------------------------------------------------------|------------------------------------------|
| <b>floor(x)</b>  | x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi. | floor(4.8) = 4.0<br>floor(-15.9) = -16.0 |
| <b>fmod(x,y)</b> | x/y ning qoldig'ini kasr son tipida beradi.                     | fmod(7.3,1.7) = 0.5                      |
| <b>log(x)</b>    | x ning natural lagorifmi (e asosiga ko'ra).                     | log(2.718282)= 1.0                       |
| <b>log10(x)</b>  | x ning 10 asosiga ko'ra lagorifmi.                              | log10(1000.0)=3.0                        |
| <b>pow(x,y)</b>  | x ning y chi darajasini beradi.                                 | pow(3,4)= 81.0<br>pow(16,0.25) = 2       |
| <b>sin(x)</b>    | x ning trigonometrik sinusi (x radianda).                       | sin(0.0)= 0.0                            |
| <b>sqrt(x)</b>   | x ning kvadrat ildizi.                                          | sqrt(625.0)= 25.0                        |
| <b>tan(x)</b>    | x ning trigonometrik tangensi (x radianda).                     | tan(0.0) = 0                             |

Har bir funksiya – funksiyalar shabloni yoki funksiyalar shabloni to'plami yordamida ifodalanadi. Shunday qilib, funksiya har xil tipdagi qiymatlarga ega bo'lgan har xil konteynerlar bilan ishlay oladi. Barcha funksiyalarni argumentlari (begin, end) yarim oraliqlar bo'ladi.

### Parametrlarni funktsiyalarga uzatish.

**Qaytariladigan qiymatlar, parametrlar va argumentlar.** Funktsiya biror bir qiymat qaytarishi mumkin. Funktsiyaga murojaat qilingandan so'ng u qandaydir amallarni bajaradi, keyin esa u o'z ishining natijasi sifatida biror bir qiymat qaytaradi. Bu qaytariladigan qiymat deb ataladi va bu qiymatning tipi oldindan e'lon qilinishi lozim. Quyidagi yozuvda myFunction funktsiyasi butun sonli qiymat qaytaradi.

```
int myFunction()
```

Funktsiyaga ham o'z navbatida biror bir qiymat uzatish mumkin. Uzatiladigan qiymatlar funktsiyaning parametrlari deb aytiladi.

```
int myFunction (int Par, float ParFloat);
```

Bu funksiya nafaqat butun son qaytaradi, balki parametr sifatida butun va haqiqiy sonli qiymatlarni qabul qiladi.

Parametrdagi funksiya chaqirilganda unga uzatiladigan qiymat tipi aniqlanishi lozim. Funktsiyaga uzatiladigan haqiqiy qiymatlar argumentlar deb aytiladi.

```
int theValueReturned=myFunction(5,6,7);
```

Bu erda theValueReturned nomli butun sonli o'zgaruvchiga argument sifatida 5, 6 va 7 qiymatlar berilgan myFunction funktsiyasining qaytaradigan qiymati o'zlashtirilayapti. Argument tiplari e'lon qilingan parametr tiplari bilan mos kelishi lozim.

**Funktsiyani e'lon qilish va aniqlash.** Dasturda funktsiyani qo'llash uchun, oldin uni e'lon qilish, keyin esa aniqlash lozim. Funktsiyani e'lon qilishda kompilyatorga uning nomi, qaytaradigan qiymatlari va parametrlari haqida xabar beriladi. Funktsiyani aniqlanishidan kompilyator uning qanday ishlashi haqida ma'lumot oladi. Dasturdagi biror funktsiyani oldindan e'lon qilmasdan turib chaqirish mumkin emas. Funktsiyani e'lon qilinishi uning prototipini (timsolini) hosil qilish deb ataladi.

**Funktsiyani e'lon qilish.** Funktsiyani e'lon qilishning uch xil usuli mavjud:

- o Funktsiya prototipi faylga yoziladi, keyin esa u #include ifodasi qo'llanilib kerakli dasturga qo'shib qo'yiladi.

- o Funktsiya ishlatiladigan faylga uning prototiplari yoziladi.

- o Funktsiya uni chaqiruvchi ixtiyoriy funktsiyadan oldin yoziladi va bu holda funksiya e'lon qilinishi bilan bir vaqtda aniqlanadi.

Funktsiyani prototipini tuzmasdan turib ham uni ishlatishdan oldin e'lon qilish mumkin. Lekin, dasturlashning bunday uslubi quyidagi uchta sababga ko'ra yaxshi hisoblanmaydi.

Birinchi, funktsiyani faylda ko'rsatilgan tartibda yozish, uni dastur ishlatilishida o'zgartirish jarayonini murakkablashtiradi.

Ikkinchi, quyidagi ko'p uchraydigan holatni amalga oshirish imkoniyati mavjud emas.

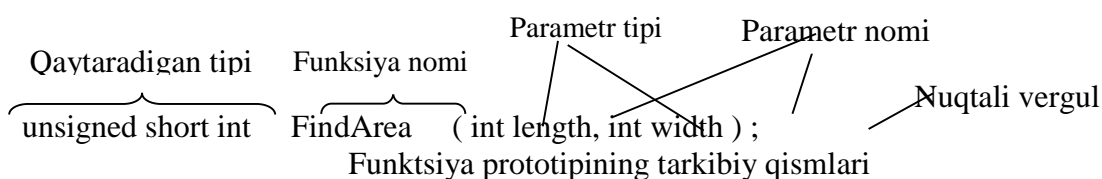
A() funksiya B() funktsiyani chaqirsin. Xuddi shuningdek, dasturning biror bir qismida B() funksiya A() funktsiyani chaqirsin. U holda biz A() funktsiyani B() funksiya aniqlanmasdan turib ishlata olmaymiz.

Bu holda hech bo'lmaganda bitta funksiya oldindan e'lon qilinishi lozim.

Uchinchidan, funktsiyaning prototiplari dasturni tekshirish jarayonida juda yaxshi ishlatiladi. Agarda funktsiya prototipi aniqlangan bo`lsa unga muvofiq funktsiya aniqlangan parametrini qabul qiladi yoki aniqlangan biror bir qiymat qaytaradi. Dasturda e`lon qilingan prototipga muvofiq bo`lmagan funktsiyani ishlatishga urinsak kompilyator bu xatolikni kompilyatsiya jarayonini o`zidayoq aniqlaydi va dastur ishlashida turli noxush xatoliklarni ro`y berishining oldini oladi.

**Funktsiya prototiplari.** Ko`pgina ichki qurilgan funktsiyalarning prototiplari dasturga #include kalit so`zi yordamida qo`shiladigan fayl-sarlavhasida yoziladi. Foydalanuvchi tomonidan tuziladigan funktsiyalar uchun esa ularning mos prototiplarini dasturga qo`shish dasturchi tomonidan bajarilishi lozim.

Funktsiyaning prototipi nuqtali vergul orqali tugaydigan funktsiyani qaytaradigan qiymati va signaturasidan iboratdir. Funktsiyani signaturasi deb uning nomi va parametrlar ro`yxati tushiniladi.



Formal parametrlar ro`yxati barcha parametrlar va ularning tiplarini ifodalaydi. Funktsiya prototipining tarkibiy qismlari 2. - rasmda ko`rsatilgan.

Funktsiyaning prototipi hamda aniqlanishidagi uning qaytaradigan qiymati tipi va signaturasi mos bo`lishi lozim. Agarda bunday mutanosiblik bo`lmasa kompilyator xatolik haqida xabar beradi. Funktsiya prototipida parametr nomlarisiz tiplarni ko`rsatilishi etarlidir. Masalan, quyida keltirilgan misol tug`ridir:

```
long Area(int, int)
```

Bu prototip ikkita butun sonli parametrni qabul qilib, long tipidagi qiymat qaytaradigan Area() nomli funktsiyani e`lon qiladi. Prototipning bunday yozilishi unchalik yaxshi variant emas. Prototipga parametrlarning nomlarini qo`shilishi uni tushunarliroq bo`lishini ta`minlaydi.

Har bir funktsiyaning qaytaradigan qiymati tipi aniqlangan bo`ladi. Agarda u ochiq aniqlanmagan bo`lsa avtomatik ravishda int tipini qabul qiladi.

16–misol. Funktsiyani e`lon qilinishi, aniqlanishi va ishlatilishi.

```
// 16. – Misol. Funktsiya prototipini qo`llanishi
#include <iostream>
using namespace std;
//funktsiya prototipi
int Yuza(int uzunlik, int kenglik);
int main()    {
    int YerUzunligi, int YerKengligi, int YerMaydoni;
    cout<< "\n Yerning uzunligi necha metr?\n";
    cin >> YerUzunligi;
    cout<< "\n Yerning kengligi necha metr?";
    cin >> YerKengligi;
    YerMaydoni=Yuza(YerUzunligi,YerKengligi);
    cout << "\n Yer maydoni yuzasi ";
    cout >>YerMaydoni;
    cout << "kvadrat metr\ n \ n";
    return 0; }
int Yuza(int YerUzunligi,int YerKengligi)
{ return YerUzunligi* YerKengligi }
```

NATIJA

```
Yerning uzunligi necha metr? 200
Yerning kengligi necha metr? 100
```

Yer maydoni yuzasi 20000 kvadrat metr

**Funktsiyaning aniqlanishi.** Funktsiyaning aniqlanishi ikki qismdan – uning sarlavhasi va tanasidan iboratdir. Funktsiyaning sarlavhasi uning prototipiga o`xshash aniqlanadi, faqatgina bu holda parametrlar nomlangan bo`lishi shart va sarlavha oxirida nuqtali vergul qo`yilmaydi. Funktsiya tanasi figurali qavsga olingan ifodalar to`plamidan iborat. Funktsiyaning sarlavhasi va tanasi quyida ko`rsatilgan.

| Qaytaradigan tipi             | Nomi                      | Parametrlari                                           |
|-------------------------------|---------------------------|--------------------------------------------------------|
| $\overbrace{\text{int}}$      | $\overbrace{\text{Yuza}}$ | $\overbrace{(\text{int uzunlik}, \text{int kenglik})}$ |
| { - ochiluvchi figurali qavs. |                           |                                                        |
| // funktsiya tanasi           |                           |                                                        |
| return (uzunlik*kenglik);     |                           |                                                        |
| } - yopiluvchi figurali kavs. |                           |                                                        |

**Funktsiyaning bajarilishi.** Funktsiya chaqirilganda unda ko`rsatilgan amallar ochiluvchi figurali qavsdan ({} keyingi birinchi ifodadan boshlab bajariladi. Funktsiya tanasida if shartli operatoridan foydalanib tarmoqlanishni ham amalga oshirish mumkin.

Funktsiya o`z tanasida boshqa funktsiyalarni va hatto o`z – o`zini ham chaqirishi mumkin.

**Lokal, statik, dinamik o`zgaruvchilar.**

**Lokal o`zgaruvchilar.** Funktsiyaga qiymatlar uzatish bilan birga uning tanasida o`zgaruvchilarni e`lon qilish ham mumkin. Bu lokal o`zgaruvchilar orqali amalga oshiriladi. Qachonki dasturni bajarilishi funktsiyadan asosiy qismga qaytsa, bu funktsiyadagi lokal o`zgaruvchilar xotiradan o`chiriladi.

Lokal o`zgaruvchilar xuddi boshqa o`zgaruvchilar kabi aniqlanadi. Funktsiyaga beriladigan parametrlarni ham lokal o`zgaruvchilar deb atash mumkin va ularni funktsiya tanasida aniqlangan o`zgaruvchilar kabi ishlatish mumkin. 2. – misolda funktsiya parametrlari va funktsiya ichida aniqlangan lokal o`zgaruvchilarni qo`llashga oid misol keltirilgan.

17. – misol. Funktsiya lokal o`zgaruvchilari va parametrlarining qo`llanilishi.

```
#include <iostream>
using namespace std;
float Almashtirish(float);
int main()    {
    float TempFer;
    float TempCel;
    cout << "Feringait bo`yicha temperaturani
    << "kiriting:";
    cin >> TempFer;
    TempCel = Almashtirish(TempFer);
    cout << "\n Bu temperatura selziy shkalasi
    << "bo`yicha: ";
    cout << TempCel << endl;
    return 0;    }
float Almashtirish(float TempFer)
{
    float TempCel;
    tempCel=((TempFer-32)*5)/9;
    return TempCel;
}
```

NATIJA:

Feringait bo`yicha temperaturani kiriting: 50

Bu temperatura selziy shkalasi bo`yicha: 10

**Global o`zgaruvchilar.** main() funksiyasida aniqlangan o`zgaruvchilar dasturdagi barcha funksiyalar uchun murojaat qilishga imkonli va ko`rinish sohasiga ega hisoblanadi. Bunday o`zgaruvchilar dasturdagi funksiyalar uchun global o`zgaruvchilar deyiladi.

Global o`zgaruvchi nomi bilan funksiya ichida nomlari ustma-ust tushadigan lokal o`zgaruvchilar faqatgina joriy funksiyaning ichidagina global o`zgaruvchining qiymatini o`zgartiradi. Lekin global o`zgaruvchi funksiya o`z ishini tugatgach u chaqirilishidan oldingi qiymatini saqlab qoladi, ya`ni funksiya tanasida e`lon qilingan lokal o`zgaruvchi funksiyaning ichida global o`zgaruvchini yashiradi xolos. Bunda lokal o`zgaruvchi alohida hosil qilinadi va funksiya ishlash vaqtida global va lokal o`zgaruvchilarning nomlari bir xil bo`lsa faqatgina lokal o`zgaruvchi ustida amallar bajariladi. Global o`zgaruvchi esa funksiyaning bajarilishi davomida oldingi qiymatini saqlab turadi. Bu g`oya 3.- misolda namoyish qilingan.

18.–misol. Global va lokal o`zgaruvchilarning qo`llanishi.

```
# include <iostream>
using namespace std;
void MeningFunktsiyam() ; // prototip
int x = 5, y = 7; // global o`zgaruvchilar
int main()      {
    cout << "main()dagi x ning qiymati:"
    <<x<< "\n";
    cout<< "main()dagi y ning qiymati y:"
    <<y<< "\n";
    MeningFunktsiyam();
    cout << "MeningFunktsiyam()funksiyasi"<<
    "ishini tugatdi!\n \n";
    cout<< "main()dagi x ning qiymati:"
    <<x<< "\n";
    cout<< "main()dagi y ning qiymati:"
    <<y<< "\n";
    return 0;      }
void MeningFunktsiyam();
{ int y = 10;
    cout<< "MeningFunktsiyam()dagi"<<
    << "x:" << x << "\n";
    cout<< "MeningFunktsiyam()dagi"<<
    << "y:" << y << "\n";    }
```

HATIJA:

```
main()dagi x ning qiymati: 5
main()dagi y ning qiymati: 7
MeningFunktsiyam()dagi x: 5
MeningFunktsiyam()dagi y: 10
MeningFunktsiyam()funksiyasiishini tugatdi!
main()dagi x ning qiymati: 5
main()dagi y ning qiymati: 7
```

Funksiyani ichida aniqlangan o`zgaruvchilar lokal ko`rinish sohasiga ega deyiladi. Yuqorida aytib o`tilganidek, bu o`zgaruvchilarni faqatgina funksiyaning ichidagina qo`llash mumkinligini anglatadi. C++da o`zgaruvchilarni nafaqat dasturning boshida balki ixtiyoriy joyda aniqlash mumkin. Agarda o`zgaruvchi funksiya tanasidagi biror bir blok ichida aniqlangan bo`lsa, bu o`zgaruvchi faqatgina shu blok ichidagina ta`sirga ega bo`lib butun funksiyaning ichida ko`rinish sohasiga ega bo`lmaydi.

19. – misol. Lokal o`zgaruvchini ko`rinish sohasi.

```
# include <iostream.h>
using namespace std;
```

```

void MeningFunktsiyam();
int main() {
    int x=5;
    cout<<"\n\n main()dagi x ning qiymati:"
    <<x;
    MeningFunktsiyam();
    cout<<"\n main()dagi x ning qiymati:"<<x;
    return 0; }
void MeningFunktsiyam();
{int x = 8;
    cout <<"\n\n MeningFunktsiyam()dagi"
    <<"local x ning qiymati: "<<x << endl;
    {
        cout << "\ n\ MeningFunktsiyam() "
        << "funktsiyasi blokidagi x ning qiymati"
        << " x:"<<x;
        int x = 9;
        cout<<"\n Blok ichida aniqlangan"
        << x ning qiymati:"<<x;
    }
    cout<<"\n MeningFunktsiyam()dagi" << "blockdan tashqarisida x ning qiymati:"
    <<x<<endl; }

```

HATIJA:

main()dagi x ning qiymati: 5

MeningFunktsiyam()dagi local x ning qiymati: 8

MeningFunktsiyam() funktsiyasi blokidagi x ning qiymati: 8

Blok ichida aniqlangan x ning qiymati: 9

MeningFunktsiyam()dagi blockdan tashqarisida x ning qiymati: 8

main()dagi x ning qiymati: 5

**Funktsiyada ishlatiladigan operatorlar.** Funktsiyada ishlatiladigan operatorlar soni va tipiga hech qanday chegara yo`q. Funktsiya ichida istalgancha boshqa funktsiyalarni chaqirish mumkin bo`lsada, lekin funktsiyalarni aniqlash mumkin emas.

C++ tilida funktsiyaning hajmiga hech qanday talab qo`yilmasa ham uni kichikroq tarzda tuzgan ma`quldir. Har bir funktsiya tushunish oson bo`ladigan bitta masalani bajarishi lozim. Agarda funktsiya hajmi kattalashayotganligini sezsangiz, bu funktsiyani bir nechta kichikroq funktsiyalarga ajratish haqida o`ylashingiz kerak.

**Funktsiya argumentlari.** Funktsiyaning argumentlari turli tipda bo`lishi mumkin. Shuningdek, argument sifatida C++ tilidagi biror bir qiymat qaytaradigan o`zgarmlar, matematik va mantiqiy ifodalar va boshqa ixtiyoriy funktsiyalarni berish mumkin.

Misol sifatida bizda biror bir qiymat qaytaruvchi double(), triple(), square() va cube() funktsiyalari berilgan bo`lsin. Biz quyidagi ifodani yozishimiz mumkin:

```
Answer=(double(triple(square(my Value))))
```

Bu ifodada myValue o`zgaruvchisini qabul qilinadi va u cube() funktsiyasiga argument sifatida uzatiladi.

cube() funktsiyasi qaytargan qiymat esa square() funktsiyasiga argument sifatida uzatiladi. square() funktsiyasi qiymat qaytargandan keyin, buning qiymati o`z navbatida triple() funktsiyasiga argument sifatida beriladi. triple() funktsiyasining qiymati esa double() funktsiyasiga argument qilib beriladi va u Answer o`zgaruvchisiga o`zlashtiriladi.

**Yangi stildagi e'lon fayllari va ismlar sohasi tushunchasi.** C++ tilining standarti .h bilan tugaydigan (stdio.h ...) standart kutubxona e'lon fayllarini yangittan nomlab chiqdi. Bunda .h qo'shimchasi olib tashlandi. C tilidan qolgan fayllar ismiga esa c harfi qo'shildi.

Misol uchun:

```
iostream.h -> iostream
string.h -> cstring
stdlib.h -> cstdlib
time.h -> ctime
```

C tilidan meros qolgan kutubhona 18 ta e'lon fayli orqali berilgan. C++ tiliga tegishli standart kutubhonada esa 32 ta e'lon fayl bor. Fayllarni yangittan belgilashdan maqsad kutubhodadagi funksiya va ob'ektlarni

std deb ataluvchi ismlar sohasiga (namespace) kiritishdir.

Ismlar sohasining o'zi ham nisbatan yangi tushuncha. Ismlar sohasini alohida dastur qismlari deb faraz qilsak boladi. Boshqa-boshqa sohalarda ayni ismli funksiya, o'zgaruvchi nomlari va ob'ektlar berilishi mumkin. Va bunda hech qanday ismlar to'qnashuvi sodir bo'lmaydi. Misol uchun bizda global, std va fun::obj degan ism sohalari bo'lsin. Ularning har birining ichida esa cout nomli ob'ekt aniqlangan bo'lsin. C++ tilida to'liq aniqlangan ism (fully qualified name) degan tushuncha bor. Shunga ko'ra har bir cout ob'ektinig to'liq ismi quyidagicha bo'ladi:

```
Ismlar sohasi ob'ekt
global      ::cout
std         std::cout
fun::obj    fun::obj::cout
```

:: operatori sohalarni bog'lash uchun qo'llaniladi. fun::obj nomli ismlar sohasida obj fun ichida joylashgan ism sohasidir. Global ismlar sohasida aniqlangan funksiya va boshqa turdagi dastur birliklariga dasturning istalgan yeridan yetishsa bo'ladi. Masalan global ismlar sohasida e'lon qilingan int tipidagi k ismli o'zgaruvchimiz bol'sa, uning ustidan dasturning xohlagan blokida amal bajarsak bo'ladi. Ismlar sohasi mehanizmi dasturchilarga yangi kutubhonalarni yozish ishini ancha osonlashtiradi. Chunki, yangi kutubhonada ayni ismlar qo'llanishiga qaramay, ismlar konflikti yuz bermaydi. Dastur yoki kutubxona yozganda yangi ismlar sohasini belgilash uchun

```
namespace istalgan_ism {
...
foo();
int k;
String str;
...
}
```

deb yozamiz. Dasturimizda ushbu ismlar sohasida aniqlangan o'zgaruvchilarni ishlatish uchun ularning to'liq ismini yozishimiz kerak. Masalan:

```
istalgan_ism::foo();
```

Ammo bu usul ancha mashaqqatli bo'ladi. Har bir funksiya yoki o'zgaruvchi oldiga uning to'liq aniqlangan ismini yozish ko'p vaqt oladi. Buning o'rniga biz **using namespace** istalgan\_ism; deb yozib o'tsak kifoya. using (ishlatish, qo'llash) buyrug'i bizning ismlar sohamizni dasturimiz ichiga tanishtiradi. Eng asosiysi biz bu amalni sohada aniqlangan va biz qo'llamoqchi bo'lgan ismlarning ilk chaqirig'idan oldin yozishimiz kerak. C++ ning standart kutubhonasida aniqlangan ifodalarni qo'llash uchun biz using namespace std; deymiz. Va albatta qo'llanilayotgan e'lon fayllari yangi tipda bo'lishi kerak. Endi bu tushunchalarni ishlatadigan bir dasturni keltiraylik.

20 – misol.

```
//Yangi tipdagi e'lon fayllari va ismlar sohasini qo'llash.
```

```
# include <iostream>
using namespace std;
int main()      {
    std::cout << "Hello!\n";
    cout << "Qale!";
    return (0); }
}
```

Ekranda:

```
Hello!
```

Qale!

`std::cout << "Hello\n";` satrida biz chiqish oqimi ob'ekti `cout` ning to'liq ismini qo'lladik. Keyingi satrda esa biz yana ayni ob'ektni ishlatdik, lekin endi uni to'liq atab o'tirmadik, chunki biz `std` ismlar sohasini using bilan e'lon qilib bo'ldik. Ismlarni global ismlar sohasida e'lon qilish uchun ularni blok va funktsiyalar tashqarisida aniqlash kerak. Masalan:

```
#include <stdio>
int i;
int main()    {
...
int k;
...
return (0);  }
```

Bu yerda `i` global ismlar sohasida joylashgan, `k` esa `main()` funksiyasiga tegishli. O'zgaruvchilarni global ism sohasida aniqlashning boshqa yo'li, ularni nomsiz ismlar sohasida belgilashdir. Yani:

```
namespace { int j; }
j o'zgaruvchisi global boldi. Uni ishlatish uchun:
::j = ::j + 7;
:: operatorini qo'llashimiz mumkin, yoki oddiygina qalib faqat o'zini:
j = j * 9;
```

kabi yozishimiz mumkin. Ammo agar biz ishlayotgan dastur blokida ayni ismli o'zgaruvchi bo'lsa, masalan `j`, unda ushbu lokal aniqlangan `j` bizning global `j` imizni berkitib qo'yadi. Biz `j` ni o'zini qo'llasak, lokal `j` ga murojat qilgan qilgan bo'lamiz. Global `j` ni ishlatish uchun endi `::` operatorini qo'llashga majburlanmiz. Bu mulohazalar umuman olganda boshqa ismlar sohaslarini ishlatganimizda ham o'rinnidir.

### Nazorat savollari:

1. `for` takrorlashida bir nechta schyotchik ishlatish mumkinmi?
2. Tanasida biror amal yozilmagan `for` operatori yordamida takrorlash tashkil etish mumkinmi?
3. `for` takrorlashi ichida `while` takrorlashini tashkil etish mumkinmi?
4. Hech qachon tugallanmaydigan takrorlash tashkil etish mumkinmi?
5. `while` va `do .. while` operatorlarini qanday farqlari bor?
6. Nima uchun funktsiyaga argument sifatida uzatilgan o'zgaruvchilar qiymati funktsiya tanasida o'zgartirilsa dasturning asosiy kodidagi shu o'zgaruvchi qiymatiga akslantirilmaydi.
7. Funktsiya prototipini e'lon qilish va funktsiyani aniqlash o'rtasida qanday farq bor?
8. Funktsiya prototipini ko'rsatishda, aniqlashda va chaqirishda uning parameterlari nomlari ustma – ust tushishi kerakmi?
9. Agarda funktsiya hech qanday qiymat qaytarmasa uni qanday e'lon qilish kerak?
10. Lokal o'zgaruvchi nima?
11. Nima uchun barcha o'zgaruvchilarni global deb e'lon qilish maqsadga muvofiq emas.
12. Lokal o'zgaruvchilar o'zlari e'lon qilingan funktsiya yoki blok chegarasida ko'rinish sohasiga ega buladimi?
13. Funktsiyalar qanday turlarga bo'linadi?
14. C++ tilidagi har qanday dasturda qaysi funktsiya bosh funktsiya bolishi kerak?
15. Lokal o'zgaruvchi yashash vaqti qanday aniqlanadi?



## O'QUV-USLUBIY VA AXBOROT TA'MINOTI

### ASOSIY ADABIYOTLAR

1. Introduction to Software for Chemical engineers. Mariano Martin. CRC Press is an imprint of the Taylor&Francis Group an information business. London, New York. 2010.
2. An Intraduction to spreadsheet optimization using MS Excel Solver. Lancaster university.
3. CAY HORSTMAN. C++ dasturlash tili. (TATU tarjiması).
4. MatCad 15\ MatCad Prime 1.0 Санле-Петербург. 2012.
5. Axborot-kommunikatsiya texnologiyalari. Toshkent. 2012.
6. Informatika va axborot texnologiyalari. Toshkent. 2012.
7. MATLAB help files. Mathworks Inc, 2011.
8. Introduction to MatLab for engineering students. David Houcque Northwestern University, 2005.
9. David Watson, Helen Williams. Computer science, IGCSE is the registered trademark of Cambridge International Examinations. 2014 y.
10. John Walkenbach and other. Microsoft Office 2010. Bible. Canada Wiley Publishengm 2010.
11. M.Aripov, V.Begalov, U Begimqulov, M.Mamarajabov. Axborot texnologiyalari. O'quv qo'llanma. –T.: “Noshir”, 2009.
12. Э.Таненбаум. Архитектура компьютера 5-Е издание. Питер, Москва 2010.
13. В.В.Дригалкин. Веб – сайт на 100 %”. Диалектика, Москва, Санк-Петербург, Киев 2011.
14. Т.А.Павловская. С/С + + Программирование на языке высокого уровня. Питер, Москва 2013.
15. Лешек Рутковский. Методы и технологии искусственного интеллекта. Горячая линия телеком, Москва 2010.
16. В.Олифер Н.Олифер. Компьютерные сети. Питер. Москва 2013.
17. А.П.Зайцев, Р.В.Мещеряков, А.А.Шелупанов. Технические средства и методы защиты информации. Горячая линия телеком. Москва 2012.
18. Т.С.Тимофеева, О.А.Тихонова, И.Г.Ларионова. Задачи по программированию. Вузовская книга. Москва 2011.
19. Джесс Либерти. Освой самостоятельно С++ за 21 день. “Вильямс”. Москва, Санкт-Петербург. Киев. 2003.
20. Том Сван. Программирование для в Windows Borland С++. –М.: Бином. 1995. -480 с.
21. R.V.Qobulov va boshqalar. С va С++. О'quv qo'llanma. –Toshkent: Aloqachi. 2013.
22. Alex Allain. Jamping into С++ Ebook. 2014
23. В.В.Мо'minov. Informatika. 2014
24. Ирэ Пол. Объектно-ориентирование программирование с использованием С++. – К.: НИПФ “ДиаСофтЛтд.”, 1995.-480 с.

### QO'SHIMCHA ADABIYOTLAR

25. Mirziyoev Sh.M. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiyoti va xalq farovonligining garovi. 48 b. T.O'zbekiston 2017 yil.
26. Mirziyoev Sh.M. Erkin va farovon, demokratik O'zbekiston davlatini birgalikda barpo etamiz. 56 b. T.O'zbekiston 2016 yil.
27. Mirziyoev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. 488 b. T.O'zbekiston 2017 yil.

28. O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida. - T.:2017 yil 7 fevral', PF-4947-sonli Farmoni.
29. MatCad User's Guide MathSoft, Inc. 101 Main Street Cambridge.
30. MATLAB help files. Mathworks Inc, 2011.
31. S.S.G'ulomov, B.A.Begaliev. Informatika va axborot texnologiyalari. Toshkent., FAN., 2010.
32. M.A.Aripov, A.R.Maraximov. Informatika, informatik texnologiyalar. Darslik, T:TDYUL., 2004.
33. M. Aripov. Internet va elektron pochta asoslari. O'quv qo'llanma. –Toshkent: TDTU. 2000
34. G'ulomov S. S. va boshqalar. Axborot tizimlari va texnologiyalari. Toshkent., FAN., 2000.
35. M.Aripov, T.Imomov va boshqalar. Informatika. Axborot texnologiyalari. O'quv qo'llanma. 1-2 qism. –Toshkent: TDTU. 2005.
36. M.Mamarajabov, S.Tursunov. Kompyuter grafikasi va Web dizayn. Darslik. –T.: "Cho'lpon", 2013.
37. Nazirov Sh.A., Qobulov R.V. Ob'ektga mo'ljallangan dasturlash. O'quv qo'llanma. – Toshkent: Aloqachi. 2007.
38. Joseph Valacich, Information Systems Today (Looseleaf) - 7th edition, 2016, ISBN13: 978-0133940473
39. W., Introduction to Computational Angela B. & George Science, 2014. 856 pages. 3 halftones. 204 line illus. 80 tables. CI: 978-0-691-16071-9
40. David Evans, Introduction to Computing, Explorations in Language, Logic, and Machines, University of Virginia, Version: August 19, 2011
41. Justin Zobel, Writing for Computer Science, Third Edition, Department of Computing and Information Systems The University of Melbourne, Parkville Australia, 2014.
42. Christos Kalloniatis, Modern Information Systems, 2012.
43. Sh.I.Razzakov, Sh.S.Yo'ldoshev, U.M.Ibragimov "Kompyuter grafikasi". -T.: "Noshir". -2014 y. -326 b.
44. Джон Бентли. Жемчужины программирования. СПб: Питер.,2002.–272 с.
45. Конноли Т., Брегг К. Базы данных, проектирование, реализация и сопровождения, теория и практика, Университет Пейсли, Шотландия, изд. М.- СПб.- Киев, 2003.
46. A.Sattorov. "Ma'lumotlar bazasini boshqarish sistemasi". Toshkent, Fan va texnologiya. 2006 y. 302 b.
47. Стефан Р. Дэвис. С++ для "чайников". ДИАЛЕКТИКА. Москва, Санкт-Петербург. Киев. 2003
48. S.Eshtemirov. "Elektron jadvallar" O'quv qo'llanma. Toshkent. O'zbekiston. 2006 y. 174 b.

#### **Internet saytlar**

1. [www.gov.uz](http://www.gov.uz) – O'zbekiston Respublikasi hukumat portali.
2. [www.lex.uz](http://www.lex.uz) – O'zbekiston Respublikasi Qonun hujjatlari ma'lumotlari milliy bazasi.
3. [www.ZiyoNET.uz](http://www.ZiyoNET.uz) - O'zbekiston Respublikasi axborot-ta'lim portali.
4. [www.ru.wikipedia.org](http://www.ru.wikipedia.org) - Vikipediya. Erkin entsiklopediya:
5. <http://virtual-university-eurasia.org/> - Virtual universitet Evroosiyo
6. <http://www.searchengines.ru/> - Qidiruv tizimlari entsiklopediyasi
7. [http://linux.manas.kg/books/how\\_intranets\\_work/ch32.htm](http://linux.manas.kg/books/how_intranets_work/ch32.htm) - How Intranet Search Tools and Spiders Work
8. <http://info.webcrawler.com/mak/projects/robots/threat-or-treat.html> - Martijn Koster "Robots in the Web: threat or treat?"
9. <http://www.opennet.ru>
10. [www.intuit.ru](http://www.intuit.ru)