

ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
САМАРҚАНД ФИЛИАЛИ

Ахборот ва педагогик технологиялар факультети

**«Информатика ва ахборот технологиялари»
кафедраси**

Я.Ёрбеков, М.Джумаева, Г. Сафарова

Дастурлаш технологиялари

фанидан

*амалий ва лаборатория маишулотларини бажариш учун услубий
кўрсатмалар*

Самарқанд-2010

Ушбу услубий кўрсатма 5521900 – «Информатика ва ахборот технологиялари» ҳамда 5149900 «Касб таълими (Информатика ва ахборот технологиялари)» бакалавиятура йўналишларининг II курсида ўқитиладиган «Дастурлаш технологиялари» фани дастури асосида яратилган.

Дастурлаш технологиялари фанидан амалий ва лаборатория машғулотларини бажариш учун услубий кўрсатмалар, ТАТУ Самарқанд филиали ўқув-услубий кенгашининг 2010 йил __ - _____ №__ сон қарори билан нашрга тавсия қилинган.

Я.Ёрбеков М.Джумаева, Г. Сафарова

ТАТУ Самарқанд филиали, 2010 й., 65 бет.

Дастурлаш технологиялари фанидан амалий ва лаборатория машғулотларини бажариш учун услубий кўрсатмалар 65 бетдан иборат. Услубий кўрсатмаларда замонавий дастурлаш технологиялари асосида дастурий воситаларни яратиш усуллари, дастурий воситаларни тестлаш, созлаш, оптималлаш, хужжатлаштириш ва дастурий воситаларни яратишда компьютерлардан фойдаланиш, дастурий восита сифати ва асосий характеристикалари мисолларда баён қилинади.

Бу услубий кўрсатмалар ўзбек тилидаги ягона ўқув адабиёти бўлгани учун унга қизиқиш катта бўлиши табиий ҳол.

*Тақризчилар: Самарқанд иқтисодиёт ва сервис
институтини доценти,
ф.м.ф.н., В.Ф.Бурнашев;
ТАТУ Самарқанд филиали
доценти, ф.м.ф.н.,
З.М.Махмудов*

М у н д а р и ж а

Амалий машғулотлар учун услубий кўрсатмалар

1. Дастурий таъминотни яратиш учун техник топшириқни яратиш.....	5
2. Дастурий таъминотнинг ҳаётий циклини аниқлаш.....	9
3. Дастурий воситанинг функционал схемасини ишлаб чиқиш...	12
4. Дастурий восита архитектурасини яратиш.....	15
5. Ишончли дастур яратиш. Содда дастурлар яратиш ва уларда таҳрирлаш ишларини амалга ошириш, дастурдаги хатоларни аниқлаш.....	17
6. Дастурий воситаларни яратиш ва созлаш.....	20
7. Модулли дастурлашга мисоллар.....	25
8. Объектга йўналтирилган дастурлар яратиш.....	30

Лаборатория машғулотларини бажариш учун услубий кўрсатмалар

1. Дастурни лойиҳалаш. Қўйилган масалани тизимли таҳлил қилиш. Дастурий таъминот яратиш босқичларини аниқлаш	36
2. Дастурлашни лойиҳалаш. Содда масалалар учун техник топшириқ ва дастур яратиш	41
3. Дастурни яратиш. Дастур хатоларини баҳолаш ва тузатиш. Дастур асосида олинган натижаларни таҳлил қилиш. Процедуралар ва функциялар.	46
4. Модулли дастурлар яратиш.....	52
5. Объектга йўналтирилган дастурлаш технологияси.....	58
6. Фойдаланилган адабиётлар.....	69

АМАЛИЙ МАШҒУЛОТЛАР

1-амалий машғулот.

Мавзу: Дастурий таъминотни яратиш учун техник топшириқни яратиш

Дастурий восита (ДВ) яратувчи ташкилотларда тизимли таҳлил натижасида «Талаблар бўйича келишув» хужжати ёки «Техник топшириқ» хужжати шакллантирилади, бу хужжатни баъзан масаланинг қўйилиши деб ҳам юритишади.

«Техник топшириқ» хужжати қуйидаги бўлимларга эга:

- кириш;
- ДВ ни яратишни асослаш;
- ДВ нинг вазифаси;
- ДВ га талаблар;
- дастур хужжатларига талаблар;
- техник-иқтисодий кўрсаткичлар;
- ДВ яратишнинг босқичлари;
- назорат қилиш тартиби.

Бу бўлимларнинг ҳар бирини мукамал кўриб чиқамиз.

Кириш қисми дастурнинг номи ва унинг қўлланиш соҳаси тўғрисида қисқача характеристика бериши лозим. Кириш қисмининг асосий мақсади – ушбу яратилаётган ДВ нинг актуаллигини кўрсатиши ва шунга ўхшаш дастурлар орасида қандай ўрин эгаллашини кўрсатиши лозим.

ДВ ни яратишни асослаш қисмида ДВ ни яратишга киришишга асос бўлувчи хужжат номи ва уни тасдиқловчи ташкилот номи келтирилади. Бундай хужжат шартнома, буйруқ ва ҳ.к. бўлиши мумкин.

ДВ нинг вазифасида унинг функционал ва эксплуатацион вазифалари тавсифланиши лозим.

ДВ га талаблар қуйидагилардан ташкил топади:

- функционал характеристикаларига талаблар;
- барқарорлигига талаблар;
- эксплуатация қилиниш шартлари;
- техник таъминоти таркиби ва характеристикаларига талаблар;

Бу талаблар ичида энг муҳими бу функционал характеристикаларига талаблар ҳисобланади. Бу бўлимда ДВ нинг асосий бажариши лозим бўлган функциялари келтирилади ва тавсифланади.

Системали таҳлил итератив характерли жараёнга эга ва унинг асосий вазифаси талаблар ва мақсадларнинг тўлиқ комплексини шакллантиришдир.

ДВ ни лойиҳалашнинг биринчи босқичида фойдаланувчининг нимани хошлаётганини жуда аниқ билиб олиш лозим. Фойдаланувчи талабларини тўғри тушуниб олиш ДВ ни тўғри лойиҳалашга ва дастурлашга олиб келади.

Техник топшириқни ишлаб чиқиш қуйидаги тартибда амалга оширилади. Аввало, бажарилиши лозим бўлган функциялар терилмаси ва берилмалар характеристикалари аниқланади. Сўнгра, натижалар ва уларнинг характеристикалари, уларни тасвирлаш усуллари аниқланади.

Шундан кейин ДВ нинг функциялашиш муҳити, шу жумладан фойдаланилаётган ОС, техник таъминотлар ва бошқалар аниқланади.

Техник топшириқни махсус стандартлар орқали баҳолаш мумкин.

1- Мисол. $Y=f(x)$ функциянинг қийматлари ва графигини кўринарли тасвирловчи ДВ яратиш талаб қилинади. Яратилаётган дастур функция қийматларини ва функция графигини берилган кесмада ясаши лозим.

1. Кириш

Ушбу техник топшириқ бир ўзгарувчили функцияларнинг жадвал қийматларини ҳисоблаш ва унинг графигини яратишга мўлжалланган бўлиб, ундан умумтаълим мактабларининг катта синфлари ўқувчилари фойдаланишлари мумкин.

Функцияларни таҳлил қилиш мактабларда ўтиладиган алгебра фанининг энг мураккаб мавзуларидан бири ҳисобланади. Бу мавзуни ўқувчилар ўрганишларида улар функциянинг характеристик нуқталаридан фойдаланиб бир ўзгарувчили функцияларни текшириш ва уларнинг графикларини кўришни ўрганишади. Мавжуд шундай хилдаги масалаларни ечишга мўлжалланган универсал дастурий таъминот, масалан, Eucisa ёки MathCad дастурлари ҳисобланади. Аммо бу дастурлар институтларда олий математика курсини ўтган талабаларга мўлжалланган ва нисбатан мураккаб фойдаланувчи интерфейсига эга. Шу сабабли бу дастурлардан фойдаланиш ўқувчилар учун анча қийинчилик туғдиради. Биз яратаётган ДВ эса ўқувчиларга шу мавзу бўйича билимларини текшириб кўришда ва мустақил тадқиқот ишларини олиб боришда ёрдам беради ва ундан фойдаланиш жуда ҳам соддалаштирилган.

2. ДВ яратишга асослар

ДВ «Информатика ва АТ» кафедрасининг ўқув режаси асосида ва кафедранинг Самарқанд шаҳридаги №8 мактаби билан 4.09.09 да тузилган шартнома асосида яратилмоқда.

3. Вазифаси

Дастурнинг асосий вазифаси ўқувчиларнинг алгебра курсининг «Бир аргументли функцияларни текшириш» мавзусини ўрганишларида ёрдамлашиш.

4. ДВ га қўйилаётган талаблар

4.1. Функционал характеристикаларига талаблар.

4.1.1. Дастур қуйидаги функцияларни бажаришга мўлжалланган:

- бир аргументли функциянинг аналитик кўринишини киритиш ва уни сақлаш;
- функция аниқланиш соҳаси интервалини киритиш ва уни ўзгартириш;
- аргумент қадамини киритиш ва уни коррективка қилиш;

- функциянинг аниқланиш интервали ичидаги қийматлари жадвалини ҳосил қилиш ва графигини чизиш.

4.1.2. *Берилмалар:*

- функциянинг аналитик берилиши;
- функциянинг аниқланиш интервали;
- аргументнинг ўзгариш қадами.

4.2. Ишончлилигига қўйилган талаблар

4.2.1. ДВ нинг киритилаётган ахборотларни назорат қилиши назарда тутилиши лозим.

4.2.2. Фойдаланувчи томонидан қилинадиган но коррект ҳаракатларни блокировка қилиш назарда тутилиши лозим.

4.3. *Техник воситалар параметрларига талаблар.*

5. *Дастур ҳужжатларига талаблар.*

2-Мисол.

«Талабалар ўзлаштиришини ҳисобга олувчи» тизимни ишлаб чиқиш учун техник топшириқ яратиш талаб қилинади. Бу тизим талабаларнинг назорат вақтларидаги олган баҳоларини оператив тарзда деканат ва деканат ходимлари томонидан ҳисобга олиб бориш учун хизмат қилади. Ўзлаштириш маълумотлари талабаларининг бутун ўқиш муддати давомида, яъни тўрт йил сақланиши лозим. Бу маълумотлардан талабаларнинг дипломлари иловаларини тўлдиришда фойдаланилади.

Қуйида техник топшириқ матни келтирилади.

1. Кириш

Ушбу техник топшириқ талабалар ўзлаштиришини ҳисобга олиб бориш тизимини яратишга мўлжалланган бўлиб, ундан талабалар томонидан назоратларни топшириш ахборотларини йиғиш ва сақлаш учун хизмат қилади. Бу тизим талабаларнинг назорат вақтларидаги олган баҳоларини оператив тарзда деканат ва деканат ходимлари томонидан ҳисобга олиб бориш учун хизмат қилади.

Ўзлаштиришни автоматлаштирилган тарзда ҳисобга олиб бориш имтихонларни топширишни назорат қилиш сифатини оширади ва бу жараёни бошқаришни енгиллаштиради.

2. *ДВ яратишга асослар*

Бу тизим факультет деканининг ўқув жараёнини такомиллаштириш тўғрисидаги тадбирлар режасини амалга ошириш мақсадида берган буйруғига асосан яратилади.

3. *Вазифаси*

ДВ асосий вазифаси талабалар ўзлаштиришини ҳисобга олиб бориш тизимини яратиш, талабалар томонидан назоратларни топшириш ахборотларини автоматлаштирилган тарзда йиғиш ва сақлашдан иборат.

4. *ДВ га талаблар*

4.1. Функционал характеристикаларига талаблар.

4.1.1. Тизим қуйидаги функцияларни бажаришга мўлжалланган:

- ўқув гуруҳлари рўйхатларини киритиш, ўқув режаси асосида ўтилаётган фанлар рўйхатини киритиш ва уларни инициализация қилиш ва ҳ.к.;
- талабаларнинг назоратни топширишлари ҳақидаги жорий ахборотларни киритиш ва уларни таҳрирлаш;
- талабаларнинг назоратни топширишлари ҳақидаги жорий ахборотларини ўқув муддати давомида сақлаб бориш;
- талабаларнинг назоратни тоширишлари ҳақидаги жорий ахборотларни олиш.

4.1.2. *Берилмалар:*

- ўқув гуруҳлари рўйхатлари;
- кафедра ўқув режалари – фанларнинг номлари рўйхати, ҳар бир фан бўйича назорат тадбирлари ҳужжатлари;
- назорат графиги;
- ҳар бир талаба томонидан назоратни топширилиши ҳақидаги жорий ахборотлар.

4.1.3. *Натижалар:*

- ҳар бир талаба томонидан назоратни топширилиши ҳақидаги натижавий ахборотлар;
- ўқув гуруҳлари томонидан назоратни топширилиши ҳақидаги якуний натижалар;
- ўқув гуруҳларидаги конкрет фанлар бўйича ўзлаштириш фоизлари;
- курслар бўйича ўзлаштириш фоизлари;
- факультет бўйича ўзлаштириш фоизлари;
- жорий вақтдаги ўқув гуруҳлари бўйича қарздорлар рўйхати.

4.2. *Ишончлигига талаблар*

4.2.1. ДВ нинг киритилаётган ахборотларни назорат қилиши назарда тутилиши лозим.

4.2.2. Фойдаланувчи томонидан қилинадиган ноқоррект ҳаракатларни блокировка қилиш назарда тутилиши лозим.

4.2.3. *Сақланаётган ахборотларнинг яхлитлиги сақланиши лозим.*

4.3. *Техник воситалар параметрларига талаблар.*

5. *Дастур ҳужжатларига талаблар.*

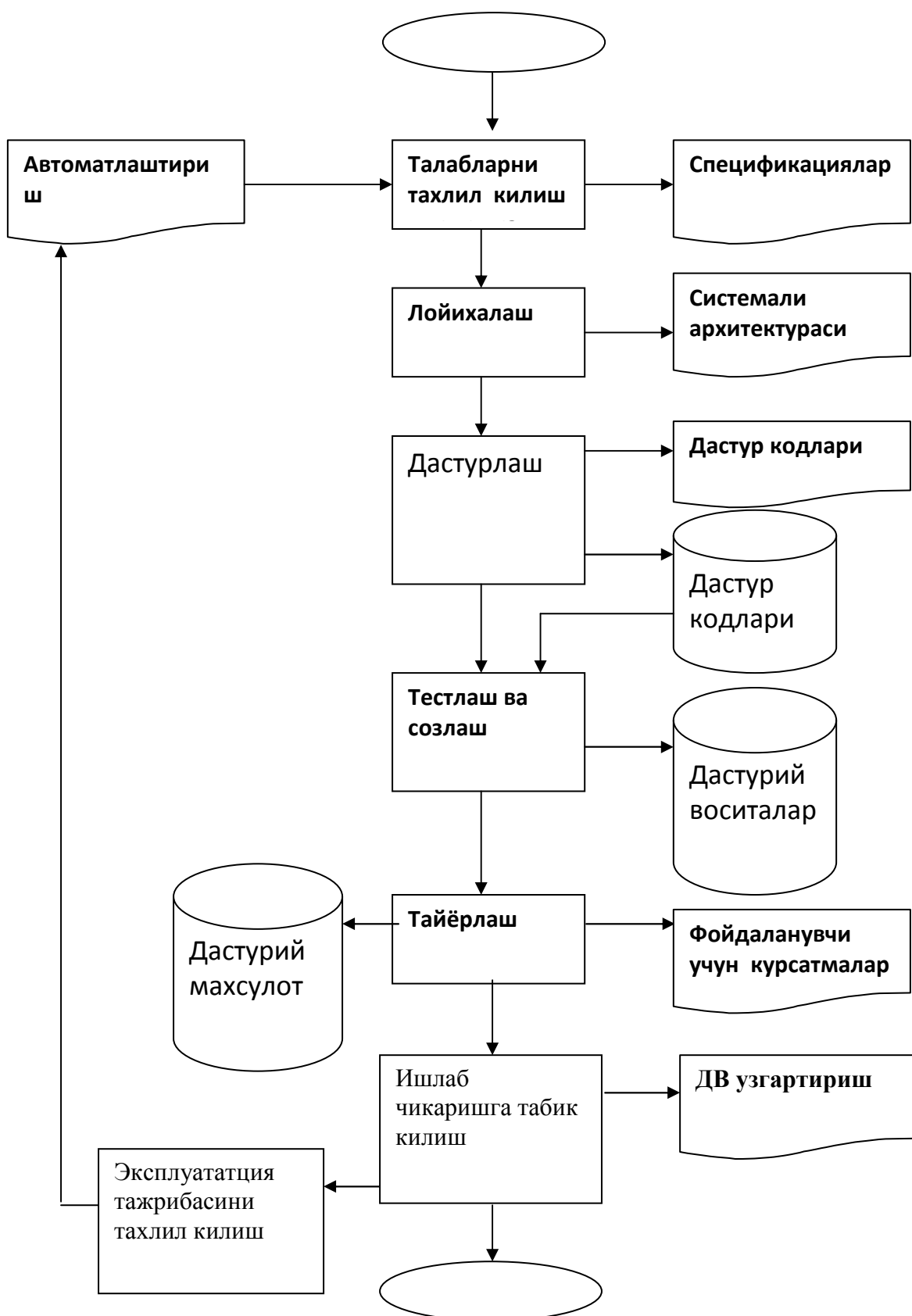
6. Фойдаланувчи интерфейсига талаблар.

2-амалий машғулот

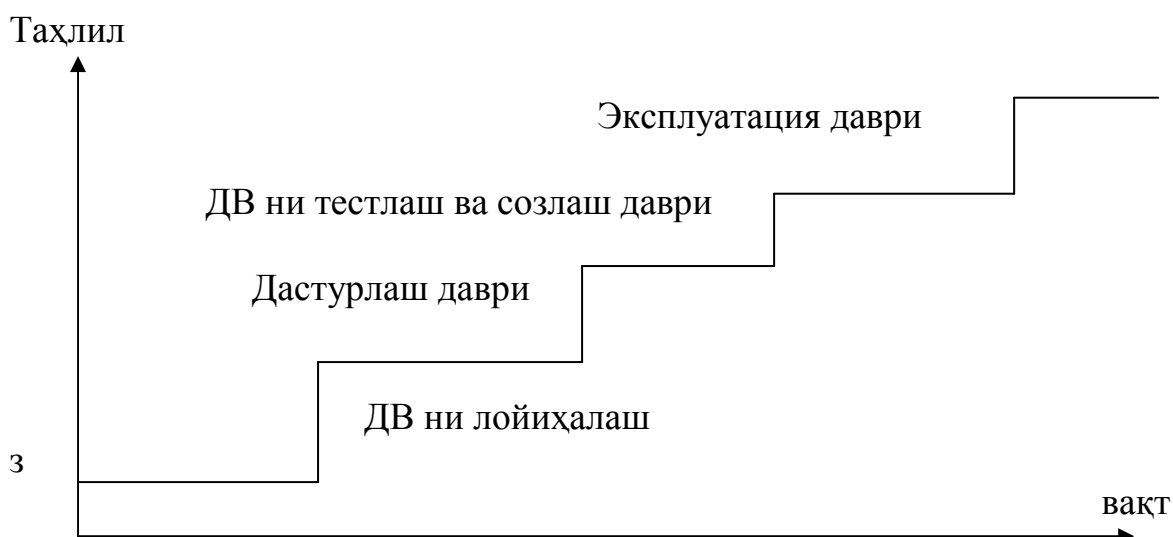
Мавзу: *Дастурий таъминотнинг ҳаётий циклини аниқлаш*

ДВ нинг ҳаётий цикли деб, уни лойиҳалашдан бошлаб бутун эксплуатация даврига айтилади. Лекин ДВ нинг ҳаётий цикли унинг яратилишига ва сифатига жуда ҳам боғлиқ. Шу сабабли ДВ ҳаётий циклини аниқлаш учун ДВ нинг яратилишининг бутун технологияси ўрганилиб чиқирилиши лозим. Аслида ДВ спецификациясининг ДВ хусусиятларига мос келиши ўрганирилиши лозим.

ДВ ҳаётий циклини қуйидагича босқичларга бўлиб схемалаштириш мумкин:



2.1-расм. ДВ ҳаётий циклининг схематик ифодаланиши.



ДВ ҳаётий циклини мукамал тавсифлаш

ДВ ларнинг ҳаётий циклини мукамал тавсифлашни қуйидаги жадвалда келтирамиз:

ДВ нинг анъанавий босқичлари	Яратиш босқичларининг ташкил этувчилари
Таҳлил	Талабларни шакллантириш ва мақсадларни ўрнатиш
Лойиҳалаш	Мукамал ташқи лойиҳалаштириш Система таркибини лойиҳалаш ДВ модуллари таркибларини лойиҳалаш МБ ни лойиҳалаш
Кодлаш	ДВ модулларининг модулларини лойиҳалаш
Тестлаш	Комплекс ДВ ни тестлаш Комплекс дастурий воситаларни созлаш
Жўнатиш	Тажрибавий эксплуатацияга қўллаш Саноатга тадбиқ қилиш ДВ ни жўнатиш

Мақсадларни шакллантириш

Бу босқичда лойиҳалаш учун техник топшириқлар ишлаб чиқилади. Ахборотларни ўзгартириш жараёнига алоҳида аҳамият берилади. Ҳар бир масала алоҳида –алоҳида қаралади ҳамда киритиш ва чиқариш формалари шакллантирилади. Таҳлил натижасида масалалар тақсимооти, функциялари бўйича таклифлар ишлаб чиқилади.

Ташқи дастлабки лойиҳалаш

Бу босқичда ДВ мақсадларини қуйидаги схема асосида таркиблаш киради: асосий мақсадлар → 1-босқичли кичик мақсадлар. . . →. . . i-босқичли кичик мақсадлар. →. . . . → n-босқичли кичик мақсадлар.

Бу босқич натижасида мақсадлар таркиби шаклланади ва сўзларда тасвифланади.

Мукаммал ташқи лойиҳалаштириш

Бу босқичда ДВ спецификацияси яратилади. Аслида спецификация бу алгоритмнинг тасвифланишидир.

Жараёнларнинг спецификациясини лойиҳалашни ечимлар жадваллари орқали ифодалаш мумкин. Бу жадвал кириш шартлари тўпламидан ва ечимлар тўпламидан ташкил топади. Ечимлар жадвали икки қисмдан иборат бўлади. Жадвалнинг юқори қисми шартларни (тармоқланишларни аниқлаш) аниқлаш учун ишлатилади. Жадвалнинг қуйи қисмига шу шартлар бажарилишидан кейин бажариладиган амаллар киради.

Юқоридаги фикримизни кириш оқимидан белгиларни қуйидаги тартибда танлаш масаласида мисолда кўрайлик:

а) агар навбатдаги белги бошқарувчи бўлса, у ҳолда товуш сигнали ва хато коди чиқарилсин;

б) шаклланаётган қатор буфери тўлган бўлса, у ҳолда товуш сигнали ва хато коди чиқарилсин;

в) агар навбатдаги белги берилган диапазондаги бўлмаса, у ҳолда товуш сигнали ва хато коди чиқарилсин;

г) акс ҳолда белги буферга жойлаштирилсин, танланган белгилар счетчиги биттага оширилсин.

ечимлар жадвали:

Шартлар	1	2	3	4	5	6	7	8
C1 белги бошқарувчими?	Ҳа	Ҳа	Ҳа	Ҳа	Йўқ	Йўқ	Йўқ	Йўқ
C2 буфер тўлганми?	Ҳа	Ҳа	Йўқ	Йўқ	Ҳа	Ҳа	Йўқ	Йўқ
C3 белги 'а' дан то 'я' оралиғидами?	Ҳа	Йўқ	Ҳа	Йўқ	Ҳа	Йўқ	Ҳа	Йўқ
Амаллар								
D1 -товуш сигнали берилсин	1	1	1	1	1	1		1
D2 хато коди чиқарилсин (>0)	2	2	2	2	2	2		2
D3 счетчик қийматини ошириш							2	
D4 белгини буферга жойлаштириш							1	

3-амалий машгулот

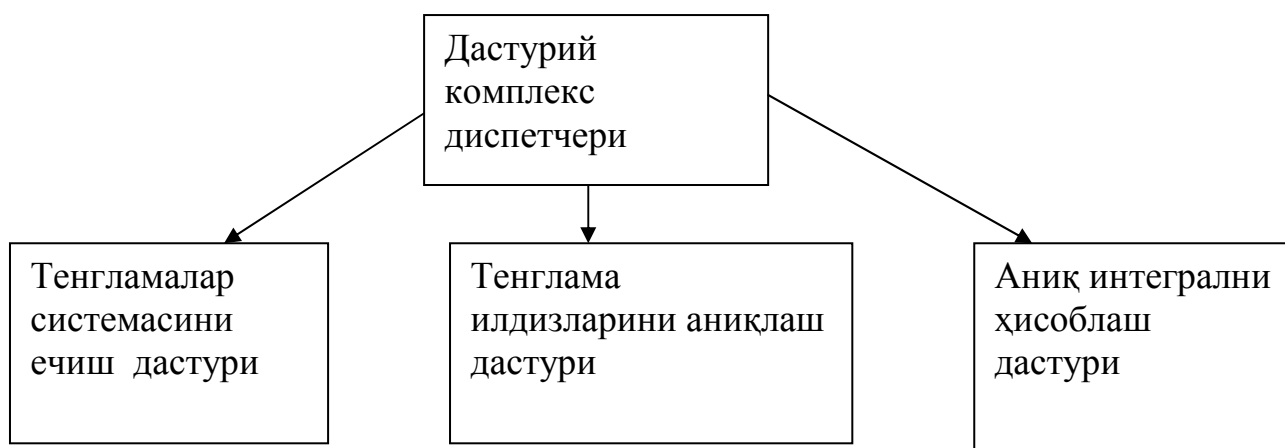
Мавзу: Дастурий воситанинг функционал схемасини ишлаб чиқиш

ДВ яратишдаги функционал ёндошув деганда дастурни ёки дастурий тизимни функционал принципга асосан декомпозиция қилиш тушунилади. Декомпозициялашнинг ҳамма усулларида энг содда интерфейслар ишлатилади: примитив интерфейслар ва анъанавий менюлар бўлиб, улар берилмалар структурасини ва уларни ишлаш дастурларини таҳлил қилишга ва лойиҳалашга қаратилган бўлади. Бу ҳолда одатда ишланувчи компоненталар ва таркибларни лойиҳалаш амалга оширилади.

1. Структурали ва функционал схемани ишлаб чиқиш

Мураккаб дастурий таъминотни лойиҳалаш жараёни унинг таркибини аниқлашдан бошланади, яъни таркибий компоненталарни ва улар орасидаги боғланишларни аниқлашдан бошланади. Бу аниқланишлар натижасида эса компоненталарнинг структурали ёки функционал схемаси яратилади.

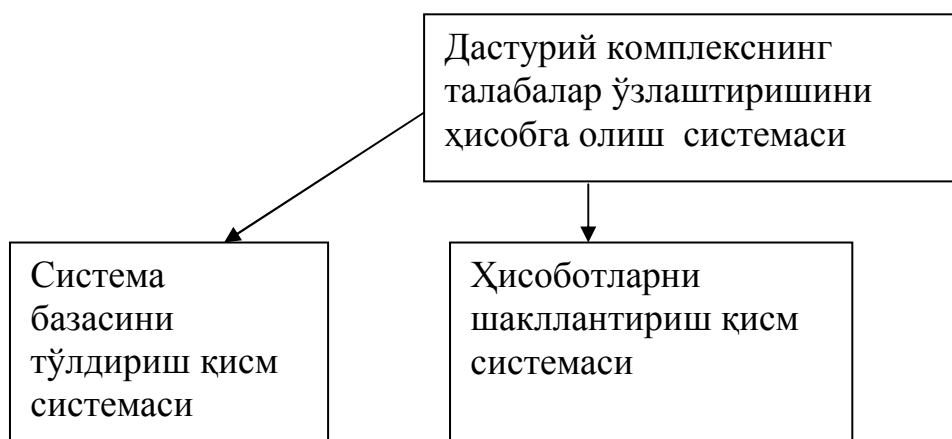
Яратилаётган дастурий таъминотнинг структурали схемаси. Структурали схема деб яратилаётган дастурий таъминотнинг таркиби ва ўзаро боғланишларини акс эттирувчи схемага айтамыз.



3.1-расм. Дастурий комплекс структурали схемасига мисол.

Энг содда дастурий таъминотнинг таркибий схемаси компоненталари сифатида қисм дастурлар ва кутубхоналар ресурсларини ифодалаш мумкин. Дастурий таъминотнинг структурали схемасини яратишни одатда қадамма-қадам деталлаштириш усули билан амалга оширилади. Дастурий система компоненталари сифатида, қисм системалар, маълумотлар баъзалари, ресурслар библиотекаси ва ҳоказолар қаралиши мумкин. Дастурий комплекснинг структурали схемаси диспетчер-дастурдан бошқаришни мос дастурга узатишни намоиш қилади (3.1-расм).

Дастурий система структурали схемаси, структурада қисм системаларнинг мавжудлигини ёки бошқа структура компоненталарининг мавжудлигини аниқлайди. Дастурий комплексдан фарқли равишда дастурий системанинг баъзи қисмлари интенсив ахборот алмашувчи бўлади.



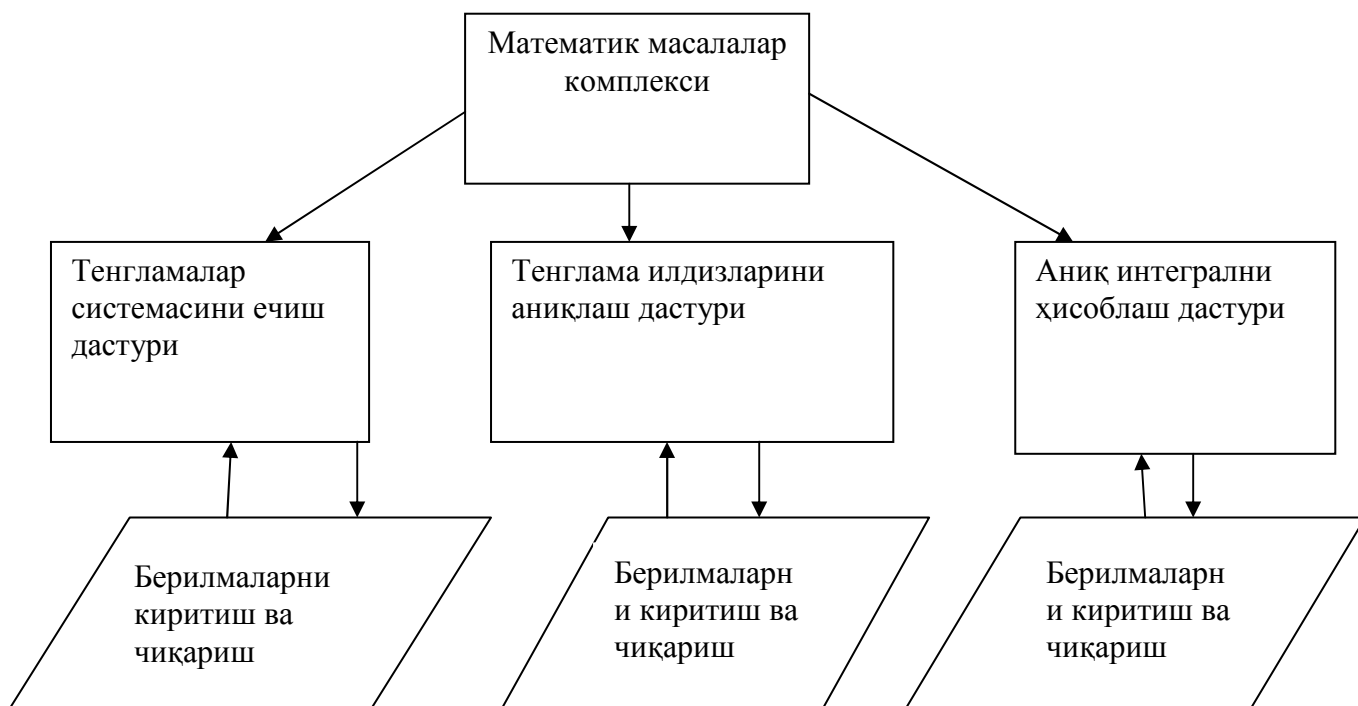
3.2-расм. Дастурий система структурали схемасига мисол.

Лойиҳаланаётган дастурий таъминот ҳақида тўлиқроқ тасаввурни уларнинг компоненталарининг боғланиши нуқтаи назардан функционал схемалар беради.

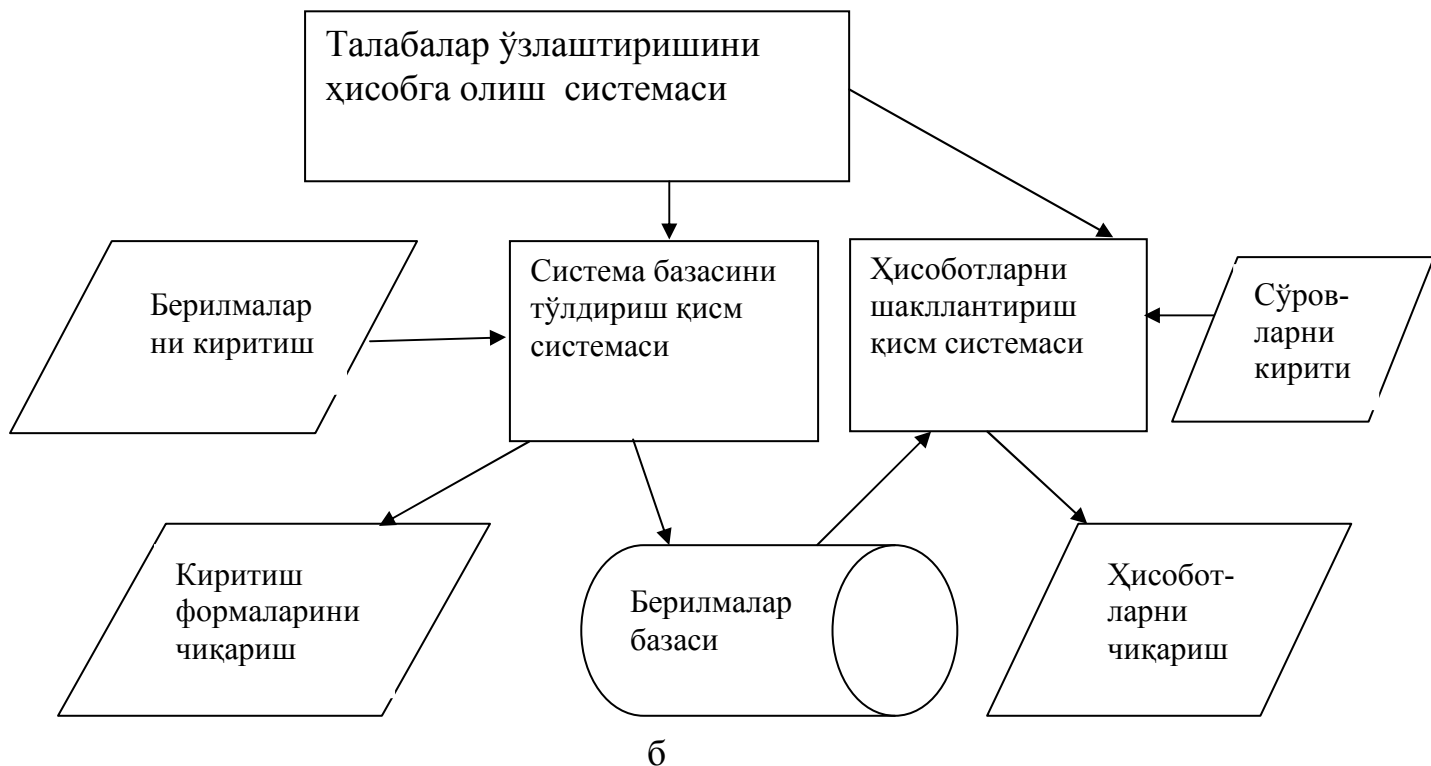
Функционал схема. Функционал схема ёки берилмалар схемаси – бу дастурий таъминот компоненталарининг информацион оқимлар тавсифи билан ўзаро боғланишини акс эттирилишидир. Функционал схемани яратишда махсус стандартлар орқали ўрнатилган белгилашлар ишлатилади.

Функционал схемалар структурали схемаларга қараганда анча информатив ҳисобланади.

Структурали ва функционал схемаларнинг ҳамма компоненталари албатта тавсифланиши лозим. Структурали ёндошувда айниқса дастурлараро интерфейслар спецификациясига катта эътибор берилади.



а



3.3-расм. Дастурий система функционал схемасига мисол: а-дастурий комплекс, б- дастурий система.

4-амалий машгулот

Мавзу: Дастурий восита архитектурасини яратиш

ДВ архитектурасини яратишдан мақсад:

1. Вариант масалаларида архитектура тушунчаси ва уни тавсифлаш масалаларини ўрганиш.
2. ДВ архитектуралари синфларини аниқлашни ўрганиш.
3. Архитектура блоклари орасидаги боғланишларни аниқлашни ўрганиш.
4. Архитектура функцияларини аниқлашни ўрганиш.

1. ДВ архитектураси тушунчаси.

ДВ архитектураси – бу унинг қандай кўринишдалигини ифодаловчи қурилма, яъни ДВ ни ўзаро боғланган қисм системалар тўплами тарзида ифодаланишидир. Бу қисм системалар сифатида одатда алоҳида дастурлар ишлатилади. ДВ архитектурасини яратиш ДВ қийинчиликлари билан курашнинг биринчи босқичи ҳисобланади.

ДВ архитектурасини яратишдаги асосий масалалар:

- Дастурнинг қисм системаларини ажратиш ва уларда ташқи функцияларини ифодалаш (улар ташқи тавсифлашда келтирилади).
- Дастурий қисм системалар боғланишларини аниқлаш.

Шу боскичда қабул қилинган ечимлар асосида кейинчалик ДВ нинг функционал спецификацияси конкретлаштирилади.

2. ДВ архитектураларининг асосий синфлари.

ДВ лар архитектураларининг қуйидаги асосий синфлари мавжуд:

- * Мақсадли дастур.
- * Автоном бажарилувчи дастурлар комплекси.
- * Қатламли дастурий система.
- * Параллел бажарилувчи дастурлар коллективи.

Мақсадли дастур. Бундай архитектурали ДВ га фақат битта дастур киради. Бундай архитектура ҳеч қанақа тавсифланишга эга эмас, чунки ташқи функцияларни акс эттириш шарт эмас.

Автоном бажарилувчи дастурлар комплекси бир нечта дастурлардан иборат бўлиб:

- Улардан исталгани фойдаланувчи томонидан ишга туширилиши мумкин.
- Фаоллашган дастур иши тугамагунча бошқа дастурлар ишга тушурилиши мумкин эмас.

Қатламли дастурий система - бир нечта қатлам деб аталувчи қисм системалардан ташкил топади:

- Ҳар бир қатламда хусусият тўғрисида ҳеч нарса маълум бўлмайди.
- Ҳар бир қатлам бошқарилиш бўйича боғланади.
- Ҳар бир қатлам маълум бир заҳирага эга бўлади.

Параллел бажарилувчи дастурлар коллективи – бажарилиши стадиясида бўлувчи бир бири билан боғлана олувчи дастурлар терилмаси шаклида таркибланади.

3. Архитектура функциялари.

Қисм системалар орасида ўзаро боғланишларни амалга ошириш учун баъзан қўшимча дастур компоненталари талаб қилинмайди – бунинг учун олдиндан аниқланган келишувлик стандарт база дастурий таъминотига киритилган бўлиши лозим. Масалан, автоном бажарилувчи дастурлар комплекси учун ташқи спецификациялар етарли бўлади. Ана шу спецификацияларни амалга оширувчи функциялар архитектура функцияларига киради.

4. ДВ архитектурасини назорат қилиш.

ДВ архитектурасини назорат қилиш учун аралаш назорат ва қўлда имитация қилиш ишлатилади.

ДВ архитектурасини аралаш назорат қилиш – бу ДВ яратувчиларининг ДВ ташқи тавсифини назорат қилишдан иборат бўлади: спецификация сифати ва функционал спецификацияларни назорат қилишдир.

Кўлда имитация қилишда ДВ қисм системаларнинг ўзаро боғланишлари назоратга олинади.

5-амалий машғулот

Мавзу: Ишончли дастур яратиш. Содда дастурлар яратиш ва уларда тахрирлаш ишларини амалга ошириш, дастурдаги хатоларни аниқлаш.

ДВ ларни яратишда инсон тизимлар билан ишлайди. Тизим деганда бир - бири билан ўзаро таъсирда бўлган элементлар тўпламини тушунамиз. Алоҳида олинган ихтиёрий дастур ҳам тизим бўла олади. Тизимни тушуниш – бу унинг барча элементлари ўртасидаги таъсир қилувчи йўллارни ҳаёлан танлашдир. Бундай тизимларга шундай дастурлар мисол бўла оладик, улар учун дастурчиларнинг «ҳар қандай созланган дастурда камида битта хато мавжуд» деган ибораси ўринли. ДВ ларни яратаётганда мумкин бўлган хатоликлар туфайли унинг барча элементлари орасидаги алоқаларни тўлиқ била олмаймиз. Шунинг учун тизимнинг мураккаблигини унинг элементлар сонига қараб баҳолаш фойдалидир: унинг элементлари орасидаги ўзаро таъсир йўллариининг сони $n!$ га тенг, бу ерда n -тизимнинг элементлар сони. Тизимни кичик деб атаемиз, агар $n < 7$ ($6! = 720 < 1000$) бўлса, тизимни катта деб атаемиз, агар $n > 7$ да оралиқ тизимлар синфига эга бўламиз. Кичик тизимлар ҳар доим оддий бўлади, катта тизимлар ҳам оддий, ҳам мураккаб бўлиши мумкин. Дастурлаш технологиясининг асосий вазифаси - катта тизимларни кичик тизимларга олиб келишни ўрганишдан иборат.

Оддий тизимларни унинг элементлар сонига қараб баҳолаш амалиётда кенг қўлланилади. Масалан, бирор коллективни раҳбарига ўзаро бир-бири билан боғлиқ ишчилар сони олтидан кўп бўлмаслиги афзалдир. Қуйидаги қоидага амал қилиш жуда муҳимдир: «ҳар бир айтиладиган гап, олти ва ундан кам пунктларда айтилиши зарур». Биз ҳам ушбу қоидага амал қиламиз: ҳар бир ўзаро боғлиқ бўлган тасдиқлар (тавсиянома талабнома ва ҳ.к) мос равишда гуруҳлаштирилади ва умумлаштирилади. ДВ ларни яратишда ҳам ушбу қоидага риоя қилиш фойдалидир.

1. Нотўғри таржима қилиш ДВ ларда хатоликларнинг асосий сабабидир.

ДВ ларни яратиш ва ишлатишда биз кўп маротиба маълумотларни бир кўринишдан бошқа кўринишга ўтказишга (таржима) қилишга дуч келамиз. Буюртмачи ўзининг ДВ ларга бўлган заруратини қандайдир талаблар асосида ифода қилади. Бу талаблардан келиб чиққан ҳолда дастурчи ДВ нинг ташқи тасвирини яратади.

Бунда берилган аппаратуранинг специфик тасвиридан, эҳтимолки, таянч дастурий таъминот спецификацисидан фойдаланади.

Дастурлаш тилининг ташқи тасвири ва унинг ўзига хос спецификациясига асосан бу алгоритмик тилда ДВ нинг ташқи тасвирига кўра фойдаланувчи учун керакли дастурий хужжатлар тайёрланилади. Ҳар бир дастурнинг матни уни қайта ишлашда ва хатоликларни тузатишда бошланғич маълумот ҳисобланади. Фойдаланувчи дастурий хужжатлар асосида ДВ дан фойдаланиш учун керакли амалларни бажаради ва олинадиган натижаларни интерпретация қилади. Бу босқичларнинг ҳар бирида маълумотларни бир кўринишдан бошқасига ўтказиш хато амалга оширилиши мумкин. Бунга маълумотларни бошланғич тасвирлашнинг нотўғри тушунилиши сабаб бўлиши мумкин.

2. Хатоликлар билан курашнинг асосий йўллари.

Инсоннинг маълумотларни бир кўринишдан бошқа кўринишга ўтказишдаги юқорида кўриб ўтилган фаолиятнинг хусусиятларини ҳисобга олиб, хатоликлар билан курашнинг қуйидаги йўллари кўрсатиб ўтиш мумкин:

- Саралаш фазосини кичрайтириш (яратиладиган тизимларни соддалаштириш)
- Дастур тузувчилар тайёргарлигининг зарур савиясини таъминлаш (бу дастур тузувчилар коллективи менежерининг вазифасидир)
- Маълумотларни тасвирлашнинг бир қийматли интерпретациясини таъминлаш.
- Таржиманинг тўғрилигини назорат қилиш (интерпретациянинг бир қийматлилигини назорат қилиш ҳам киради).
- Дастурни ихчамлаштириш натижасида хатоларнинг пайдо бўлиш эҳтимолини камайтириш. Мисол.

$C1:=10+B1*D^{**}2;$

$C2:=B+\sin(R);$

$C3:=5+A(1)*A(2);$

$C:=(C1+C2)*C3;$

Бу операторлар кетма-кетлигини битта оператор билан алмаштириш мумкин:

$C:=(10+B1*D^{**}2+ B+\sin(R))*(5+A(1)*A(2));$

Бу операторга юқоридагига қараганда оз машина буйруқлари ишлатилади. Масалан, учта қийматлаш конструкцияси биттага, учта ҳисоблаш модули биттага қисқартирилади.

3. Хатоларни тестлаш орқали аниқлаш

Чегаравий шартларни тестлаш

Биз дастур ёзилишида ҳамма ёзган кодларимизни жиддий назорат қилиб борсак, масалан цикл охирги қиймати массив чегарасидан чиқиб кетишининг олдини олиш ёки бошқаришнинг тўғри узатилишини назорат қилиш натижасида кодимиз хатосиз ёзилиши мумкин. Тестлаш усуллари

бири чегаравий шартларни тестлашдан иборатдир: ҳар сафар кичик код ёзишимиз билан масалан, цикл ёки шартли ифодани ёзиш биланок биз циклнинг керакли марта такрорланишини текшириб чиқишимиз, шартли ифода эса тармоқланишни тўғри амалга ошираётганини текширишимиз лозим. Бу жараён чегаравий шартларни тестлаш деб аталади. Бу текширишларда экстремал қийматлар назорат қилинади. Агар коднинг қайси бир блокида хато бўлса (одатда у чегарада бўлади) ва аксинча экстремал қийматларда код яхши ишласа кодда деярли хато бўлмайди.

1-Мисол.

```
Var  
N,i: integer;  
S:array [1..n] of integer;  
begin  
for i:=1 To n do  
S[i+1]:=0;  
end;
```

Бу циклнинг тўғри ишлашини текширамыз. Текширишимиз лозим бўлган биринчи чегаравий шарт – бу циклнинг экстремал қийматидир. Бу ҳолда цикл $i=n$ қийматда тўхтайди, чунки S массивнинг четки қийматидан чиқиб кетади. $S[n+1]$ -аниқланмаган. Бу чегаравий хатодир, яъни массив чегарасидан ташқарига чиқади. Демак биринчи чегаравий шартни текшириш хатони аниқлади. Худди шу усулда энди бошқа чегаравий шартларни ҳам текшириб чиқиш лозим, агар улар бўлса.

Саволлар ва машқлар

1-мисол. Қуйидаги дастур фрагментда компилятор аниқлай олмайдиган жиддий мантикий хато бор, уни аниқланг ва дастурни таҳрирланг:

```
x:=0;  
While x<=10 do;  
Begin  
X:=x+1;  
End;  
Writeln(x);
```

2-мисол. Қуйидаги дастур фрагмент бажарилишида компилятор қанақа хатони қайд қилади:

```
Var  
X:1..5;  
Begin  
x:=17;  
Writeln(x); Writeln(pred(x)); Readln; End.
```

3-мисол. Факториални ҳисоблашдаги хатони аниқланг.

```
Int factorial(int n);
Var
N,fac:integer;
Begin
fac:=1;
While n>=0 do
Begin
fac:=fac*n; n:=n-1; end; end.
```

4-мисол. S массиви элементлари қийматларини инициализация қилиш дастуридаги хатони аниқланг.

```
Uses
SysUtils;
Const border:=10;
Var
s: array[1..10] of integer; I:integer;
begin
I:=10;
While i<=border do
begin
s[i+1]:=1; end; end.
```

5-мисол. Дастурни ихчамлаштириш натижасида хатоларнинг пайдо бўлиш эҳтимолини камайтириш учун қуйидаги мисолни қандай ёзиш мумкин?

```
C1:=10+B1*D**2;
C2:=B+sin( R);
C3:=5+A(1)*A(2);
C:=(C1+C2)*C3;
```

6-мисол. Дастурни ихчамлаштириш натижасида хатоларнинг пайдо бўлиш эҳтимолини камайтириш учун қуйидаги операторларни қандай ёзиш мумкин?

```
C1:=10+cos(0.25*D);      C2:=D+exp(D);      C3:=15+M(1)*M(2);
C:=(C1+C2)*C3;
```

7-мисол. Бу дастур фрагментда компилятор қанақа хатони аниқлайди:

```
Var
X:1..25;
Begin
X:=0;Writeln(x);Writeln(Pred(x));
Readln; End.
```

6-амалий машғулот

Мавзу: Дастурий воситаларни яратиш ва созлаш

Созлаш турлари

Синтактистик хатолар ҳақида хабарлар тўхташи биланоқ созлаш бошланади. Созлаш жараёни бошида оддий тест маълумотларидан фойдаланиш керак. Агар бунда тўғри натижалар ҳосил бўлса унда жуда кўп мураккаб маълумотлар билан ишлаб кўриш керак. Агар натижалар нотўғри бўлса унда қуйидаги ҳоллар бўлиши мумкин:

1. Синтаксис хато йўқ, лекин дастур компиляция бўлмаган.
2. Дастур ишляпти, компиляция бўлган, лекин натижа чиқармаяпти.
3. Дастур компиляция бўлган ишляпти, лекин кутилмаган узилишлар бўляпти.
4. Дастур компиляция бўлган, ишляпти, лекин нотўғри натижа чиқаряпти.
5. Дастур цикл ичида ишлаб тўхтаб қолмоқда (циклланиб қолмоқда)
Қуйида ҳар бир ҳолатни кўриб чиқамиз:
 - 1 - ҳолат. Бу ҳолатларни тузатишда системани яхши биладиган ходимлардан маслаҳат сўраш керак.
 - 2 - ҳолат. Бу ҳолатлар қандайдир логик ва системали ҳолатлар орқали бўлган бўлиши мумкин. Масалан, алгоритм бошланиши билан қийматига қараб кетиши мумкин.
 - 3 - ҳолат. Программа кўзда тутилгандан олдинроқ тўхтаб қолиши оғир ҳолатлардан биридир. Бу ҳолатларни топиш учун хатоларни топиш усулларидан фойдаланиш маъқул.
 - 4 - ҳолат. Бу ҳолат программа тўғри тузилганлигини, лекин унда хатолик борлигини кўрсатади.
 - 5- ҳолат. Бу ҳолда қайси циклдан тўхташ бўлмаганлигини топиш керак. Бунинг учун гумон қилинган циклдан олдин ва кейин чиқариш операторларини қўйиб текшириш керак.

Юқоридагиларни ҳисобга олиб қуйидаги умумий таклифларни бериш мумкин:

- дастур ҳажми қанча кўпайса, шунча кўп хато бўлиши мумкин.
- тузатиш жараёнини дастур яратилиши босқичида ўйлаш керак.
- тузатиш жараёнида ишлатилаётган ўзгарувчи константаларнинг рўйхати бўлиши керак.
- топилган хатоларни албатта кетма - кет тузатиш керак.
- ҳар доим чиқарилган натижаларни яхшилаб кўриб таҳлил қилиш зарур.
- программа вариантларини санаси билан алоҳида сақланишини таъминлаш керак.

Созлаш воситалари.

Дастурни ёзиш вақтида унинг ичига киритиладиган созлаш воситалари энг самарали ҳисобланади. Бу ҳолатда дастурчи томонидан хатоларни аниқлаш тез ва аниқ бўлади. Дастурлашда қуйидаги созлаш воситалари қўлланилади:

1. Хотира таркибини таҳлил қилиш.
2. Алгоритм бажарилишини назорат қилиш.
3. Ўзгарувчилардан фойдаланишни назорат қилиш.
4. Индексларни текшириш.
5. Кичик дастурлардан фойдаланишни назорат қилиш.
6. Ўзгарувчиларни белгиларини тасвирлаш.
7. Хотира таркибидагиларни ёзувга чиқариш.

Дастурни ҳозирги ҳолатини уни бажарилишига қадар ҳисобга олишга ёки белгилашга ёрдам беради.

Ёзувдаги маълумотлар амалиётда фойдаланиладиган ҳолатига умуман мос келмаслиги сабабли, созлаш мосламасига интилиш ҳосил бўлади ва бу мослама маълумотни қулай ҳолатга (ўқиш учун) келтириб боради.

Алгоритмнинг бажарилиши йўлларини назорат қилиш.

Дастурни бажарилишини мантиқий йўлини ҳисобга олади. Ундан дастурчилар томонидан берилган жараёни бажарилиш кетма-кетлигини ва ўзгарувчиларни ҳозирги вақтдаги белгисини бажарилишини ва ҳисобга олишини текширишда фойдаланилади. Бундай кузатишларнинг учта тури мавжуд:

1. Ўзгарувчиларнинг белгиларини назорат қилиш.
2. Кичик дастурларни чиқаришни ҳисобга олишни назорат қилиш.
3. Дастурдаги буйруқларни етказишни назорат қилиш.

Иккинчи турдаги кузатиш асосан кичик дастурлардан фойдаланилган пайтда созлашда қўлланилади. Ҳар бир муомалада ёзувга кичик дастурнинг номи чиқади, ундан чиқилганда асосий дастурга қайтилганлиги ҳақида маълумот чиқади.

Бу турдаги текшириш дастурчига дастурий хатоларни аниқлашдаги барча керакли маълумот билан таъминлаб туради. Унинг камчилиги машина вақтини кўп сарф бўлишидир, ҳамда берилаётган маълумотлар бир неча минг катордан иборат бўлишидадир. Шунинг учун юқоридаги камчиликлардан воқиф бўлиш учун дастурни қисмларга ажратиб белгиланган жойларда ўчириб текширилади.

Ўзгарувчилардан фойдаланишни назорат қилиш мосламаси.

Назорат мосламаси шундай тузиладикки, унда ўзгарувчилар бирданига эмас, балки аниқ рўйхат бўйича кўрсатилгандек тартиб билан ёзилади.

Индексларни текшириш. Массивларни эълон қилинган чегараларни ва уларнинг индексларни солиштириш орқали номланган массивларни индексацияланганлигини ва тўғрилигини текшириш.

Ўзгарувчиларнинг мазмунини тақрибий чиқариш Display номли созловчи буйруқ орқали бажарилади. Чиқариладиган маълумотни тўлиқ

чиқишини таъминлаш учун ўзгарувчининг номини ва унинг ҳозирги ҳолат мазмуни кўрсатилган буйруқ орқали берилади.

Интерактив режимида созлаш.

Жуда кўп ҳолатларда, дастурлар дастурчини машина билан ўзаро таъсири остида бажарилади, шунинг учун бу ҳолатларда қулайликлардан кенгроқ фойдаланиш маъқулроқ, яъни созлаш жараёнининг самарадорлигини ошириш учун терминалдан кенгроқ фойдаланиш керак. Пакетли созлаш системасида қўлланиладиган усулларнинг кўп қисмидан вақти тақсимланган системада ҳам фойдаланиш мумкин, лекин интерактив жараён ҳаракатида қўшимча факторларни ва янги созловчи усулларга ёндашишга ҳам аҳамият бериш керак.

Дастурнинг стил бирлиги

Дастур қанчалик мантиқли ва кетма кетликда жиҳозланган бўлса, у шунчалик яхши ёзилган дейилади. Агар кодни форматлаш олдиндан айтиб бўлмайдиган даражада ўзгариб борса: масалан, цикл массивни гоҳ қийматларининг ўсиб бориши гоҳо камайиб бориши тартибида кўриб чиқиш назарда тутилса, у ҳолда дастур яхши ўқилмайди ва дастурнинг ишлаш ишончилиги ҳам камайади. Дастур жиҳозланишида отступлар албатта дастурни ўқишни осонлаштиради. Масалан, қавсларни қандай ишлатиш лозим? Қавсларни if конструкцияси ёзилган қаторда ишлатиш керакми ёки навбатдаги қаторда ёзиш лозимми? Дастурчилар доимо код матнини мумкин бўлгунча яхши жойлаштиришга ҳаракат қилишади. Дастурчилар аллақачон дастур матнини жойлаштириш стилини танлашган. Қавслар фақат шартли операторларда ва цикл танасини аниқлашда ишлатилади.

Дастур стилини ёзиш албатта катта кўникма ва усталикни, ижодий ишни талаб қилади, буни қуйидаги мисолда келтирамиз. Битта ўзгарувчи қийматини ҳар хил қийматларда таққослаш талаб қилинганда баъзи дастурчилар қуйидаги дастурни ёзишади:

```
IF A=1 THEN ...;
```

```
IF A=2 THEN ...;
```

```
IF A=3 THEN...;
```

Бу дастурда агар A=1 бўлса ҳам барибир ҳамма операторлар бажарилади. Аммо биз дастурда ELSE операторини ишлатсак, бошқа таққослашлар тўхтатилади:

```
IF A=1 THEN ...
```

```
ELSE IF A=2 THEN...
```

```
ELSE IF A=3 THEN ...;
```

Қуйидаги мисолларда дастурлашнинг ташкил этилишини ва айниқса қуйидаги параметрларнинг бажарилишини ўрганиб чиқинг:

1. Дастурнинг тўғри ёзилишини.
2. Дастурлаш стилининг ишлатилиши.
3. Дастурда изоҳларнинг ишлатилиши.

4. Дастурда чегаравий шартларнинг бажарилишини.
5. Дастурда ўзгарувчилар ва константаларнинг тўғри ишлатилиши.
6. Дастурлашда ишлатилган технология тури.
7. Дастурдаги хатолар ва уларнинг манбалари (агар улар бўлса).
8. Дастурнинг ишлашини тушунтириш.
9. Дастурдаги ишлатилган асосий дастурлаш техникаси элементларини.

1- Мисол. Квадрат тенгламани ечиш дастурини ёзиш талаб қилинсин.

Дастур коди:

```
Program kvadrat;
```

```
Var
```

```
X1,x2,a,b,c,d,z: real;
```

```
Begin
```

```
Read (a,b,c); d:=b*b-4*a*c; z:=2*a;
```

```
if d>=0 then
```

```
begin
```

```
x1:=( b+sqr(d))/z; x2:=( b-sqr(d))/z; end
```

```
else begin
```

```
x1:=- b/z; x2:=sqrt(Abs(d))/z; end; Writeln('x1=',x1,',x2=',x2,',d=',d );end.
```

1 4 2

x1=34,x2=-30,d=8.

Дастурда изоҳларнинг ишлатилиши

Изоҳлар дастурни ўқишни осонлаштириши лозим. Улар шундай ҳам тушунарли нарсаларни тушунтириши мумкин эмас. Яхши изоҳлар дастурни тушунишга ёрдам беради, у қисқача энг керакли деталларни очишга ёрдам беради. Масалан, изоҳлар операторларни тушунтиришга ёзилиши мумкин эмас, чунки бу операторнинг ишлашини шундай ҳам биламиз. Изоҳларда коднинг катта фрагменти ҳақида ахборот берилмайди. Кодда жуда қийин тушуниладиган жараёнларга ёзилиши мақсадга мувофиқдир. Изоҳлар функциялар, глобал берилмаларга, константаларни аниқлашга, таркиблардаги ва синфлардаги майдонларга, ҳамда объектларга ёзилиши лозим. Глобал ўзгарувчилар дастурнинг ҳар хил қисмларида пайдо бўлади ва албатта уларни изоҳлар билан изоҳлаб бориш фойдалидир. Баъзан дастур коди ҳақиқатан ҳам мураккаб ёки берилмалар таркиби жуда мураккаб шакллантирилган бўлса, у ҳолда бу жараёнлар изоҳлар орқали тушунтирилиб борилади, баъзан эса бу ишлатиладиган алгоритмларнинг қаерда келтирилганлиги ҳақида ҳам хабарларни сақлаши мумкин. Қуйида изоҳларнинг жуда ўринли ишлатилишига мисол келтирамиз:

2-мисол.

```
Program CelsiusFahrenheit;
```

```
{Бу дастур Цельсия градусларини Френгейт градусларига ўтказди }
```

```
{Fahrenheit=NCelsius*(212-32)/100+32}
```

```
Var
```

```
NFactor, NCelsius: integer; NFahrenheit:real;
```

```

Begin
{Температура қийматини Цельсий градусларида киритинг }
Write ('Celsius: ');
Read(NCelsius);
{келтирилган ўтазиш формуласи учун }
{ўтказиш қийматини ҳисоблаймиз}
NFactor:=212-32;
{ бу натижани Fahrenheit қийматини ҳисоблашда ишлатамиз}
NFahrenheit:= NFactor* NCelsius/100+32;
Write ('NFahrenheit=',NFahrenheit);
end.

```

Celsius: 45 gr.
NFahrenheit=113 gr.

7—амалий машғулот

Мавзу: Модулли дастурлашга мисоллар

Модуллар

Модулларни яратиш дастурни бажариш тартибини, уни созлаш, тестдан ўтказиш ва такомиллаштиришни анча осонлаштиради. Бу эса дастур ёзишни бошиданок дастурлаш вақтидан ва машина вақтидан ютишга олиб келади. Дастурни лойиҳалаш, созлаш, тестдан ўтказишни бир - бирига боғлиқ бўлмаган ҳолда мустақил бажариш мумкин. Натижада дастурчи машина вақтидан ютади. Дастурдан фойдаланишдаги камчиликлардан бири унинг ҳажми катта бўлишига қаратилади. Дастурни кетма-кетлик билан бажаришда алоқа буйруғи узиб қўйилади ва бунинг оқибатида хотира ва вақт меъёрланади.

Модулли дастурлаш - бу модел деб аталувчи мантиқий қисмларга ажратиш жараёнидир. Бу қисмлар алоҳида кетма - кет лойиҳаланади, чунки катта масала бир неча кичик масалаларга бўлинса уни ҳал этиш ёки ечиш анча енгиллашади. Агар масала юқоридан қўйишга лойиҳаланса, унда у албатта тахминий модуллар учун бир неча кичик вазифаларга бўлинади. Модулли дастурлаш қўйидаги имкониятларни яратади:

1) Модул қўлланилган контекстга боғлиқ бўлмаган ҳолда, тўғри ва аниқ дастурий модулга эришиш керак.

2) Олдиндан модулнинг ички иши тўғрисида ҳеч андай маълумотга эга бўлмаган ҳолда модул ёрдамида катта дастурни шакллантиришга интилади. Тадбиқий дастурлаш ва стандарт жараёнлар қулай модуларга асосланади.

Одатда лойиҳа "таркибини" бир киши тузиши мумкин. Бу эса лойиҳани бутунлигини таъминлайди ва дастурни тушуниш осон бўлади.

Модулли принцип қўлланилаётганлиги сабабли асосий дастур қисқа бўлиши, модуллараро боғланишлар вужудга келиши лозим. Кўмаклашувчи дастур ҳақиқий дастур яралгунга қадар ўрнини босувчи қисқа буйруқлар кетма - кетлигидан иборат. Кўмаклашувчи дастур 2 га бўлинади. Сохта ва алмашинувчан модуллар. Сохта модул ҳеч қандай иш бажармайди, у фақат чақирувчи модулларга бошқарувни қайтариб беради. Вақтинча алмашинувчи модулни мураккаб модул яратулгунга қадар оддий қайта ишлашни бажаради. Алмашинувчи модул натижани бермагунга қадар чиқарувчи модул ишини давом эттирмаслиги керак. Кўмаклашувчи дастурлар дастурнинг бошқа бўғинларини (сегментга) тестдан ўтказиш учун ҳам хизмат қилади.

Асосий дастурни 2 та муваффақиятли лойиҳалари ишлаб чиқарилгандан сўнг таркибий дастурларнинг ҳамма усуллари бирлаштирилиб асосий дастур шакллантирала бошланади.

Кўп масалаларда тез - тез қўлланилиб турадиган модуллар, стандарт программалар бир хил қоида асосида тузилади (расмийлаштирилади), яъни уларга мурожат қилиш, фойдаланиш ва улардан натижа олиш бир хил қоида асосида ташкил этилади. Бу эса программа тузишда улардан осонлик билан фойдаланиш имконини беради.

Модуллардан икки хил усулда фойдаланиш мумкин:

Биринчи усул. Асосий программанинг зарур бўлган қисмига модулларни жойлаштириш йўли билан. Иккинчи усул - ҳар бир машинанинг ўзига хос буйруқлари мавжуд бўлиб, бу буйруқлар ёрдамида модулларга мурожат қилиш йўли билан фойдаланиш мумкин. Иккинчи усул кўпроқ қўлланилади. Чунки бу ҳолда оператив хотирада модуллар кутубхона шаклида жойлашади ва бу модуллардан исталган вақтда фойдаланиш мумкин. Шу сабабли модулларни ишдан озод қилишда уч хил адрес қўлланилади:

1. Абсолют адреслар - уларнинг қийматлари модулларнинг тутган ўрнига қараб ўзгармайди;
2. Ички адреслар - модулларнинг жойлашган жойига боғлиқ ҳолда ҳисобланади;
3. Ташқи адреслар - бошқа модулларнинг тутган ўрнига қараб ўзгаради.

Модулларнинг дастурий таркиби:

UNIT <номи>;

INTERFACE

<интерфейс қисми>

IMPLEMENTATION

<бажарилувчи қисм>

BEGIN

<инициализация қилинувчи қисм> **END**

Бу ерда UNIT – калит сўз, у модул сарлавҳасини бошлаб беради; <ном> - модул номи, INTERFACE – калит сўз (интерфейс), модулнинг интерфейс қисми. IMPLEMENTATION – калит сўз (бажарилиши), бажарилувчи қисмни бошлайди; BEGIN–калит сўз, модул инициализациясини бошлайди.

Шундай қилиб, сарлавҳа ва учта қисмдан иборат экан, улардан исталгани бўш бўлиши мумкин.

Модул сарлавҳаси ва модулларнинг бир- бири билан боғланиши

Модулнинг сарлавҳаси UNIT калит сўз ва ундан кейин келувчи модул номидан иборат бўлади, бу ном диск файли номи билан устма-уст тушиши лозим, унда модул матни жойлаштирилади. Агар сарлавҳа Unit Global кўринишида бўлса, у ҳолда модул GLOBAL.PAS номли диск файлига жойлаштирилиши лозим. Модул номи унинг бошқа модуллар билан боғланишини таъминлайди. Бу боғланишлар махсус илова билан ўрнатилади:

USES <модуллар рўйхати>

Бу ерда USES – калит сўз;

<модуллар рўйхати> - модуллар рўйхати ва улар билан боғланишларни амалга оширилади; рўйхатнинг элементлари модуллар номларидан ташкил топади, масалан:

Uses CRT, Graph, Global;

Агар USES... эълон қилиниши фойдаланилаётган бўлса, у асосий дастур тавсифларини очиши лозим.

Интерфейс қисми

Интерфейс қисми INTERFACE билан очилади. Бу ерда модулнинг ҳамма глобал объектлари эълон қилинади (тур, константалар, ўзгарувчилар ва қисм дастурлар), улар асосий дастурдан ёки модулларда фойдаланилиши лозим. Эълон қилиниши:

Unit Cmplx;

Interface

type

complex = record

re, im : real end;

Procedure AddC (x, y : complex; var z : complex);

Procedure MulC (x, y : complex; var z : complex);

Агар асосий дастурда қуйидагича ёзилса

Uses Cmplx;

Бу эълон қилинишдан кейин дастурда COMPLEX турлар ва CMPLX модулидаги иккита процедурлар - ADDC ва MULC дастурда ишлатилиши мумкин бўлади.

Бажарилувчи қисм

Бажарилувчи қисм IMPLEMENTATION калит сўз билан бошланади, у дастурнинг интерфейс қисмида эълон қилинади. Унда модул учун локал

объектлар, ёрдамчи турлар, константалар, меткалар ва блоклар эълон қилинади. Масалан:

```
Unit Cmplx;  
Interface  
type  
complex = record  
re, im : real  
end;  
Procedure AddC (x, y : complex; var z : complex);  
Implementation  
Procedure AddC;  
begin  
z.re := x.re + Y.re;  
z.im := x.im + y.im  
end; end.
```

Инициализация қилинувчи қисм

Инициализация қилинувчи қисм модулни тугаллайди. Унда бажарилувчи операторлар жойлашади. Бу операторлар асосий дастурга бошқарилиши узатилишидан олдин бажарилиши лозим. Масалан:

```
Unit FileText;  
Interface  
Procedure Print(s : string);  
Implementation  
var  
f: text; const  
name = 'output.txt'; Procedure Print;  
begin  
WriteLn(f, s)  
end;  
{ Инициализация қилиниш қисми: }  
begin  
assign(f, name);  
rewrite(f);  
{ Инициализация қилиниш қисмининг тугалланиши }  
end.
```

1-мисол. Комплекс сонлар арифметикасини амалга оширувчи модуль яратиш талаб қилинади:

```
UNIT Cmplx;  
{-----}  
INTERFACE  
{-----}  
type complex = record  
re, im:real
```

```

end;
Procedure AddC (x, y : complex; var z : complex) ;
Procedure SubC (x, y : complex; var z : complex) ;
Procedure MulC (x, y : complex; var z : complex) ;
Procedure DivC (x, y : complex; var z : complex) ;
const
c : complex =    (re : 0.1; im :  -1);
{-----}
IMPLEMENTATION
{-----}

```

```

Procedure AddC; begin
z.re := x.re + y.re; z . im := x.im + y. im
end {AddC};
Procedure SubC;
begin
z.re := x.re - y. re ;
z.im := x.im - y.im
end {SubC};
Procedure MulC;
begin
z.re := x.re*y.re - x.im*y. im;
z.im := x.re*y.im + x.im*y.re
end {MulC};
Procedure DivC;
var
zz : real;
begin
zz := sqr(y.re) + sqr(y.im);
z. re := (x.re * y.re + x.im * y.im) / zz;
z.im := (x.re * y.im - x.im * y.re) / zz
end {DivC};
end.

```

Бу модуль матнини CMPLX.PAS файлига жойлаштирамиз. Биз уни TPU-файл яратиб компиляция қилишимиз ҳам мумкин. Шундан кейин бу модульга муружаат қилиш мумкин бўлади (2-мисол).

2-мисол.

```

Uses Cmplx;
var
a, b, c : complex;

```

```

begin
a.re := 1; a.im := 1;
b.re := 1; b.im := 2;
AddC(a, b, c);
WriteLn('Қўшиш: 'c.re:5:1, c.im:5:1,'i') ;
SubC(a, b, c) ;
WriteLn('Айириш: 'c.re:5:1, c.im:5:1,'i');
MulC(a, b, c);
WriteLn('кўпайтириш: 'c.re:5:1, c.im:5:1,'i') ;
DivC(a, b, c);
WriteLn('Бўлиш: 'c.re:5:1, c.im:5:1,'i');
end.

```

Дастурда Uses Cmplx эълон қилиниши билан унинг ҳамма объектлари доступга эга бўлиб қолди.

8-амалий машгулот

Мавзу: Объектга йўналтирилган дастурлар яратиш

1. C++ Class лари ва уларнинг таърифи, асосий тушунчалари

Жуда кўп ҳолларда дастурлашда берилмалар тўпламлари билан ишлашга тўғри келади. Масалан, ёзувлар, ёзувлар бинар дарахти билан, тармоқ берилмалари билан ва ҳ.к. лар билан ишлашга тўғри келади. Ёзувлар ҳар турдаги берилмаларни бирлаштиради. Ҳар бир тавсифланаётган ахборот алоҳида ҳеч нарсани билдирмайди. Улар биргаликда эса бирон бир объектни тавсифлаши мумкин. Энг содда таркиб, масалан, массив алоҳида бир хил турдаги қийматларни яхши сақлайди, аммо ҳар хил турдаги катталикларни сақлашга мутлақо ярамайди. Бундай ахборотлар комплексларини объектлар деб атаيمиз. Шу вақтгача биз дастурлаш тилида аниқланган турлар билан ишлаш дастурларини яратдик.

Синфлар — бу маълум бир дастурда аниқланган турдир. Синфни аниқлаш уни ташкил этувчи атрибутлар ва улар устида қандай операцияларни бажарилишини аниқлашдир. Бир нечта ҳар хил турдаги катталикларни битта объектга бирлаштирувчи конструкция синф деб аталади.

Синф объектлар турини ва унинг модули бирлигини ҳам тавсифлайди. ОЙД да бош дастур тушунчаси йўқолади, шунингдек қисм дастурлар ҳам йўқолади. Улар фақат синфлар қисмлари шаклида қатнашишлари мумкин. Пакетлар тушунчаси ҳам йўқ. Ҳатто Integer ва Real каби базавий турлар ҳам базавий синфларга кирувчи синф сифатида қаралади. ОЙД да фақат битта базавий ҳисоблаш механизми бор. Маълум бир объект борки у ҳамиша бирон бир синф экземплярни ҳисобланади, ҳисоблаш эса ушбу объект чақирувчи синф компоненти амали ҳисобланади. Масалан, ойнани экранда кўрсатиш

учун ойнани ифодаловчи объектнинг Window объекти экземпляри бўлган display компоненти чақирилади.

Инкапсуляция

Синфларни яратишда кўпинча фақат синфни ўз ичида ишлатилувчи компоненталарини ишлатишга тўғри келади. Бундай компоненталардан фойдаланувчилар фойдаланишмайди ва биз уларни инкапсуляция қилишимиз, уларни ёпишимиз лозим бўлади. Синфлар мутлақо компоненталарини чақириш орқали ахборот алмашади.

Ворислик

ДВ ларни яратишда кўплаб синфлар яратилади. Уларнинг кўплари олдин яратилган синфларнинг вариантлари бўлиши мумкин. Бундай мураккаб тизимларни соддароқ бошқариш учун синфлаштириш – ворислик механизми ишлатилади. А синф В синфнинг вориси ҳисобланади, агарда у В синфнинг баъзи компоненталарини меросласа.

Полиморфизм

Полиморфизм (polymorphism) – бу мумкин бўлган ҳар хил турдаги объектларни синф моҳиятига бирлаштира олиш имкониятидир. Полиморфизм ворислик усули билан назорат қилинади. Полиморфизм - энг универсал ва радикал воситадир. У инкапсуляциялаш ва ворислик билан чамбарчас боғлиқ. Полиморфизм шундай ҳолатки, бунда қандайдир битта нарсга кўп шаклларга эга бўлади. Дастурлаш тилида «кўп шакллар» дейилганда битта ном автоматик механизм томонидан танлаб олинган турли кодларнинг номидан иш кўриш тушунилади. Шундай қилиб, полиморфизм ёрдамида битта ном турли хулқ - атворни билдириши мумкин

2. Class формати

Class ни умумий ҳолда қуйидагича аниқлаш мумкин:

```
Class className
{ private:           //қисмий
<privat маълумотлар аъзолари, конструкторлар, усуллар>
protected:         //himoyalangan
<ҳимояланган маълумот аъзолари, конструкторлар ва усуллар>
public:             //умумий
<оммавий хусусиятлар> <оммавий маълумот аъзолари > <оммавий
конструкторлар ва деструкторлар > <оммавий усуллар> }
```

Мисол. Кубикларни ташлаш ўйини class ни яратиш.

//Класни аниқлаш (кубикларни ташлаш):

```
class Tdice
{ public:
int LastValue;
intGetTDiceValue(); };
```

Metodini aniqlash:

```
Int Tdice::GetDiceValue()  
{   LastValue=random(n)+1;  
    return LastValue; }
```

Class аниқланишидаги очилган фигурали кавсдан кейин public калит сўзи ёзилади ва бу калит сўздан кейин синфнинг объектлари.

3. Class лардан фойдаланиб дастурлаш

Масаланинг қўйилиши.

Қуйидаги функциялар қийматини ҳисоблаш дастурини яратишда қуйидаги ишларни бажариш талаб қилинади:

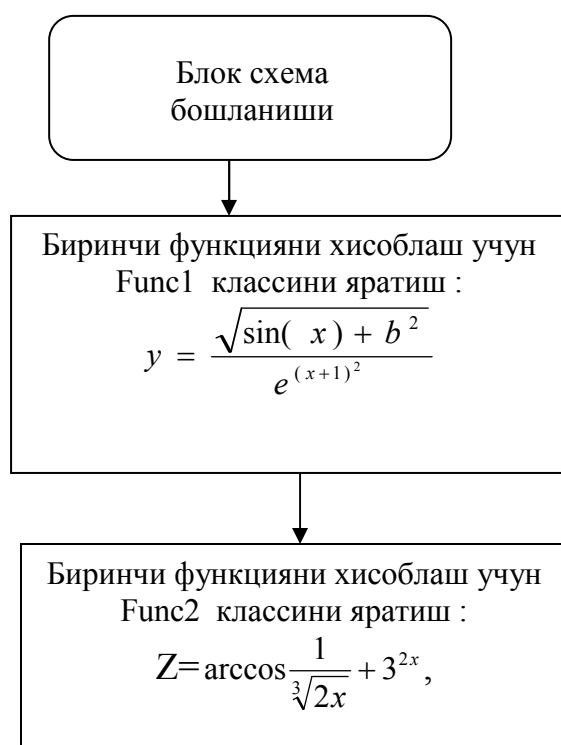
1. Вариант масаласини ўрганиш.
2. ДВ архитектурасини яратиш.
3. Масаланинг дастурини (ҳар бир функция учун синф яратинг) синфларни яратиш билан тузиш.
4. Дастурни отладка қилиш.
5. Мустақил иш ҳисоботини яратиш.

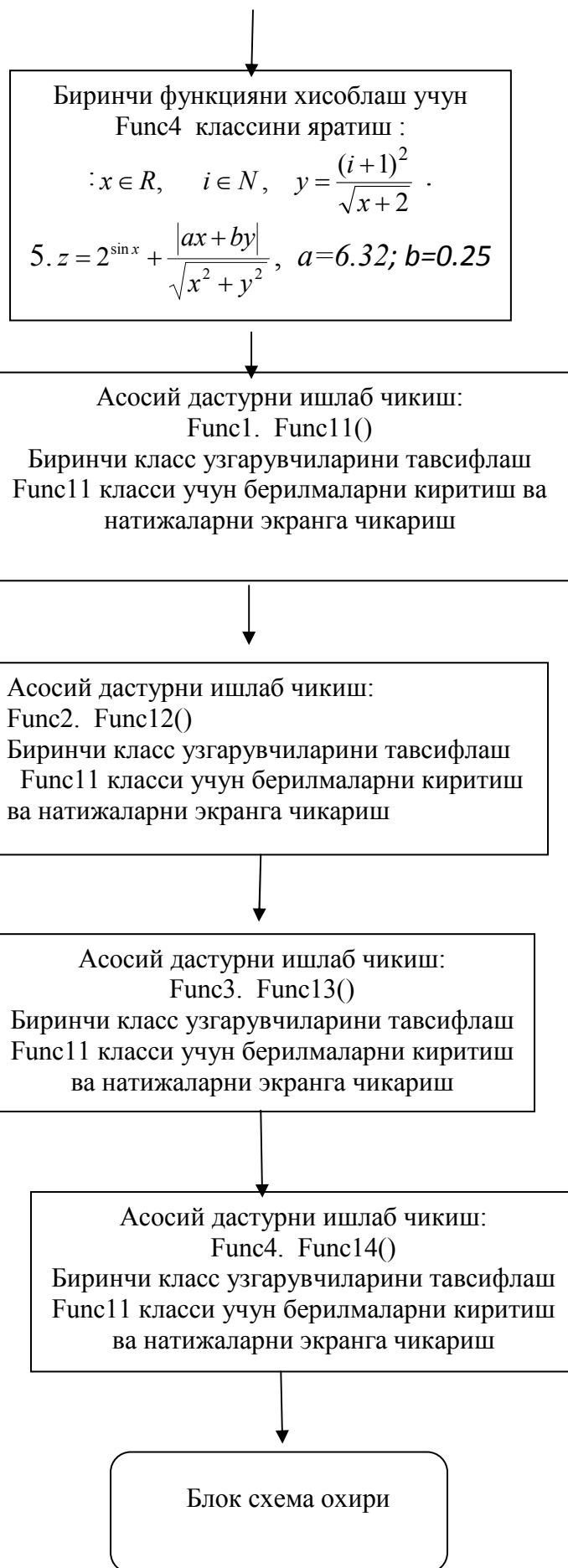
$$y = \frac{\sqrt{\sin(x) + b^2}}{e^{(x+1)^2}}, \quad b = 3.6 : \quad 2. \quad Z = \arccos \frac{1}{\sqrt[3]{2x}} + 3^{2x},$$

$$3. \quad Z1 = ax^2 + bx + c, \quad a = 3.6, \quad b = 6.3 \quad c = 6.32$$

$$4. \quad x \in R, \quad i \in N, \quad y = \frac{(i+1)^2}{\sqrt{x+2}}. \quad 5. \quad z = 2^{\sin x} + \frac{|ax+by|}{\sqrt{x^2+y^2}}, \quad a=6.32; \quad b=0.25 \quad y=5.326$$

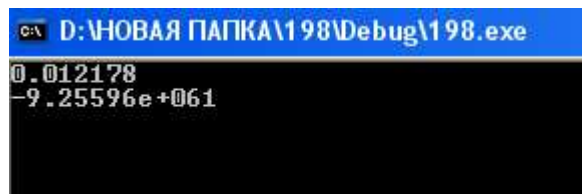
Алгоритмнинг блок схемаси





Дастур коди:

```
#include "stdafx.h"
#include <iostream.h>
#include <Math.h>
class Func
{public:
double Value;
FuncValue(double x,double b);};
Func::FuncValue(double x,double b)
{
    Value=sqrt(sin(x)+b*b)/exp((x+1)*(x+1));}
int main()
{Func Fun;
    Func Fun1;
    double x,b,z;
    x=1.2;b=1.2; z=1.1;
    Fun.FuncValue(b,x);
    cout<<Fun.Value<<endl;
    x=1.2; b=1.2;
    cout<<Fun1.Value<<endl;
    cin >>z;}
```



// 198.cpp : Defines the entry point for the console application.

```
#include "stdafx.h"

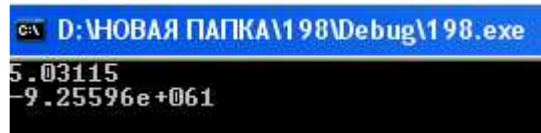
#include <iostream.h>
#include <Math.h>
class Func
{public:
double Value;
FuncValue(int i, double x);};
Func::FuncValue(int i, double x)
{
    Value=((i+1)*(i+1)/sqrt(x+2));}
int main()
{Func Fun;
```



```

Func Fun1;
double x,z; int i;
x=1.2;z=1.1; i=2;
Fun.FuncValue(i,x);
cout<<Fun.Value<<endl;
x=1.2; i=2;
cout<<Fun1.Value<<endl;
cin >>z;}

```



Адабиётлар

1. Фридман А.Л.
Язык программирования Си++
Интернет-университет информационных технологий - ИНТУИТ.ру, 2004
2. Анисимов А.Е., Пупышев В.В.
Сборник заданий по основам программирования
БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2006
3. Непейвода Н.Н.
Стили и методы программирования
Интернет-университет информационных технологий - ИНТУИТ.ру, 2005
4. Бертран Мейер
Объектно-ориентированное конструирование программных систем + CD
Интернет-университет информационных технологий - ИНТУИТ.ру, Русская Редакция, 2005
5. Борисенко В.В.
Основы программирования
Интернет-университет информационных технологий - ИНТУИТ.ру, 2005
6. Терехов А.Н.
Технология программирования
БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2007

ЛАБОРАТОРИЯ МАШҒУЛОТЛАРИ

1-лаборатория машғулоти

Мавзу: Дастурни лойиҳалаш. Қўйилган масалани тизимли таҳлил қилиш. Дастурий таъминот яратиш босқичларини аниқлаш.

Ишнинг мақсади: Талабаларни дастурий таъминотни яратиш босқичлари билан таништириш.

Қурилмалар: мос дастурий таъминотлар ўрнатилган ўқув компьютери.

Режа:

1. Дастурни лойиҳалаш. Дастурий таъминотни яратиш босқичлари.
2. Лаборатория ишини бажаришга мисол.
3. Лаборатория ишлари учун топшириқ вариантлари.
4. Назорат саволлари.

1. Дастурни лойиҳалаш. Дастурий таъминотни яратиш босқичлари.

Масалани компьютерда ечиш қуйидаги босқичларда амалга оширилади.

1. Масаланинг қўйилиши:

- Масала ҳақида ахборотлар йиғиш;
- Масаланинг шартини шакллантириш;
- Масалани ечишнинг охирги мақсадларини аниқлаш;
- Натижаларни ҳосил қилиш формаларини аниқлаш;
- Берилмаларни тавсифлаш (уларнинг турлари, катталиклар ҳаракат диапазони, структуралар ва ҳоказолар).

2. Масалани таҳлил қилиш ва моделини яратиш:

- Мавжуд аналогларни таҳлил қилиш;
- Техник ва дастурий таъминотларни таҳлил қилиш;
- Математик моделини ишлаб чиқиш;
- Берилмалар таркибини ишлаб чиқиш.

3. Алгоритмни ишлаб чиқиш:

- Алгоритмни лойиҳалаш усулини танлаш;
- Алгоритмни ёзиш формасини танлаш (блок-схемалар, псевдокод ва бошқалар);
- Тест ва тестлаш усуллари танлаш;
- Алгоритмни лойиҳалаш.

4. Дастурлаш:

- Дастурлаш тилини танлаш;
- Берилмаларни ташкил қилиш усуллари аниқлаш;
- Алгоритмни танланган алгоритмик тилда ёзиш.

- Тестлаш ва тузатиш;
 - Синтаксис тузатиш;
 - Дастур семантикасини ва мантикий таркибини тузатиш;
 - Тестли ҳисоблашлар ва тестлаш натижаларини таҳлил қилиш;
 - Дастурни такомиллаштириш.
6. Масалани ечиш натижаларини таҳлил қилиш ва керак бўлганда математик моделни аниқлаш.
7. Дастурни кузатиш:
- Маълум бир конкрет масалани ечишга дастурни мослаштириш;
 - Масалани ечишга, математик моделга, тестлашга ва дастурдан фойдаланишга ҳужжат тайёрлаш.

2. Лаборатория ишини бажаришга мисол

Топшириқ. Қўйилган масалани ечиш учун дастурий восита яратишнинг ҳамма босқичларини бажаринг.

Тўғри тўртбурчакли майдоннинг бир томони иккинчи томонидан 5 м. узунроқ, уларнинг узунликлари йиғиндиси эса 17 м. Шу майдоннинг томонлари узунликларини аниқланг.

1- босқич. Масала шартини шакллантирамиз: Тўғри тўртбурчакнинг бир томони иккинчисидан $Marta$ узунроқ, уларнинг йиғиндиси эса $summa$ га тенг.

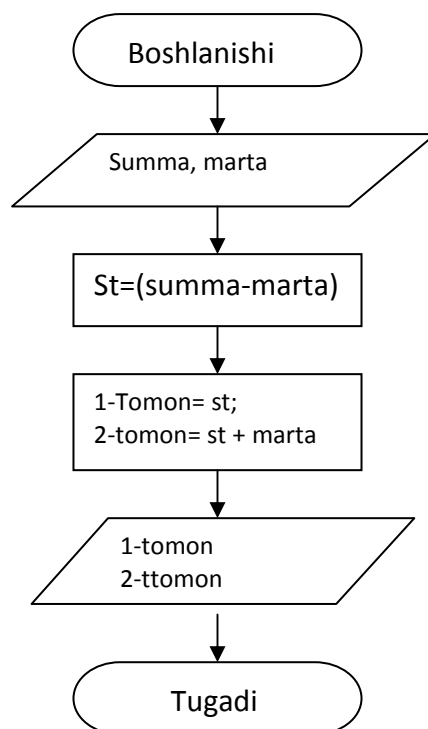
Масаланинг мақсади: Шу майдончанинг томонлари узунликларини топиш. Натижа экранга чиқарилади. Берилмалар $real$ турига эга.

2-босқич. Бу масаланинг математик модели тўғри тўртбурчакнинг томонларини қуйидаги тенгламадан топиш бўлади:

$$a-b=5, a+b=17.$$

Бу формулалардан $st=(summa - raz) / 2$ ни ҳосил қиламиз, бу ерда биз дастурлаш учун Паскаль алгоритмик тилини танлаймиз.

3-босқич. Чизикли таркибли алгоритмдан фойдаланамиз.



4 -босқич. Қўйилган масаланинг Паскал тилидаги дастури:

```

uses crt;
var summa,marta,st: real;
tomon1, tomon2: real;
begin
ClrScr;
writeln('tomonlar yig'indisini kiriting');
readln(summa);
writeln('bir tomon ikkinchisidan qanchaga uzunligi kiritilsin');
readln(marta);
st:= (summa -marta) / 2;
tomon1:= st;
tomon2:= st + raz;
write ('eni-', tomon1:7:2,'sm. ');
write ('bo'yi-', tomon2:7:2,' sm. ');
end.
  
```

5 - босқич. Дастурни тушунтиришлар - изоҳлар билан таъминлаш мумкин.

6 - босқич. Дастур компьютерда ишлатилгач қуйидаги тест берилмалари киритилди ва натижалар олинди:

Берилмалар		Натижалар	
Summa	marta	tomon1	tomon2
15	3	6	9

52	8	22	30
24	0	12	12
30	5	12,5	17,5

Бу тест натижалари дастурнинг тўғри ишлаётганлигини билдиради.

7-босқич. Тўғри тўртбурчакнинг икки томони йиғиндиси ва уларнинг айирмасини қуйидагича оламиз: 10 ва 4. У ҳолда $st=3$ бўлади ва у томонлардан бирининг учунлигидир ($tomon1$), иккинчи томон эса 7 га тенг бўлади.

Лаборатория ишлари учун вариант топшириқлари.

1. Нефтебаза икки кун ичида 2560 л. Бензин таркатди. Иккинчи куни 280 л. Кўпроқ таркатди. Нефтебаза ҳар куни қанча литр бензин таркатган?
2. Дўкондан харид қилинган товарларнинг қийматини тўлаш учун алгоритм ва дастур яратинг: 'a' сўмга қўлқоп сотиб олинган, 'b' сўмга портфел олинган, 'c' галстук сотиб олинган. Ҳамма харид моллари учун дастлаб ажратилган пул 'd' сўмга тенг. Пул етмай қолганда қайтим манфий бўлади.
3. Сотувчи бир ой мабойнида ҳар куни 4 литрдан сут уйларга етказиб берди. Март ойида сутнинг нархи ҳар литрга X сум эди, 1 апрелдан эса ҳар литри $[x+a]$ сўмга қимматлашди. Сотувчига ҳамма етказиб берилган сутлар учун апрел охирида қанча тўлаш лозим?
4. Уй эгаси ўз уйини обойлар билан деворларини клейлаб чиқмоқчи. Уй девори баландлиги 'a' м ва узунлиги b м. Обой рулоннинг узунлиги 12 м ва эни 1 м. Агар обойнинг 1 метри d сўм турса бутун деворга қанча сўмлик обой кетади?
5. Бирон бир автомат иккита сонни сўрайди ва 3 та буйруқни бажаради. Улардан a буйруғи берилган бир жуфт сонни (x, y) бошқа бир жуфтга айлантиради $(x-y, y)$, улардан b буйруғи эса (x, y) ни $(x+y, y)$ ўзгартиради, c буйруғи эса (x, y) ни (y, x) га ўзгартиради. Шу автоматнинг ишлашининг алгоритми ва дастурини ёзинг.
6. Шундай учта сонни топингки, бошқа иккита соннинг орасида жойлашган бўлсин. (Составьте алгоритм и программу выбирающую из трех чисел то которое лежит между двумя другими)
7. Иккита нуқталардан қайси бири қуйидагилардан узоқроқ туради: координата бошидан; маркази координата бошида бўлган берилган радиусли айлана.
8. Иккита сондан каттасини аниқланг.
9. Шундай алгоритм ёзингки, у "N" ўқисин ва экранга 1 дан N гача бўлган сонлар квадратларини чиқарсин.
10. A нуқтидан энг кўпроқ узоқлашган B нуқти координаталари номерини экранга чиқарувчи алгоритм ва дастур ёзинг.
11. Жадвалга математика ва информатика фанлари бўйича бир йиллик баҳоларни киритиш алгоритми ва дастурини ёзинг.
12. Экрanga чемпион фамилияси ва натижасини чиқариш алгоритми ва дастурини ёзинг.

13. Аъло баҳога топширишган талабалар рўйхатини чиқариш алгоритми ва дастурини ёзинг.
14. Жадвалдан энг баланд бўйли талабани танланг ва унинг фамилиясини ва бўйини аниқловчи рақамни экранга чиқариш алгоритми ва дастурини ёзинг.
15. 8 литр бензин массаси 5,68 кг ташкил этади. Цистерна эса 500 м^3 ҳажмга эга. Унга қанча тонна бензин жойлаштириш мумкинлигини ҳисобловчи ва натижани экранга чиқариш алгоритми ва дастурини ёзинг.
16. Узунлиги 5 м бўлган мис симнинг массаси 430 г чиқади. Квртирага симни ўтказиш учун С метр керак бўлади. Ҳаммаси бўлиб қанча грамм мис ишлатилган мис сим керак бўлади?
17. Ракета экватордаги нуқтадан учирилади ва у v км/с тезликни олади. Агар $v \leq 7.8$ км/с бўлса, у ҳолда ракета ерга тушади, агар $7.8 < v < 11.2$ бўлса, у ҳолда ракета ернинг сунъий йўлдоши бўлиб қолади, агар $11.2 \leq v \leq 16.4$ бўлса, у ҳолда ракета куёшнинг сунъий йўлдошига айланади, агар $v > 16.4$ бўлса, у ҳолда ракета куёш системасидан чиқиб кетади. Шу масалани ечиш алгоритми ва дастурини ёзинг.
18. «Электрон офис мутахассиси» мутахассислигига ўтказилган конкурснинг финал қисмига уч киши киритилди: Аҳмедов, Салимов ва Тошбеков. Мусобақа уч турда ўтказилди. Аҳмедов биринчи турда m_1 балл, иккинчи турда n_1 , учинчида p_1 балл олди. Салимов биринчи турда m_2 балл, иккинчи турда n_2 , учинчида p_2 балл олди. Тошбеков эса биринчи турда m_3 балл, иккинчи турда n_3 , учинчида p_3 балл олди. Ҳолиб неча балл олганини аниқланг ва унинг фамилиясини экранга чиқариш алгоритми ва дастурини ёзинг.
19. Берилган иккита соннинг энг кичик умумий бўлувчисини аниқлаш алгоритми ва дастурини ёзинг.
20. Берилган иккита бутун сонларнинг энг кичик умумий бўлувчисини аниқлаш алгоритми ва дастурини ёзинг.
21. Берилган сон икки хонали тоқ сон эканлигини аниқловчи алгоритм ва дастур ёзинг.
22. Квадрат ва доира юзалари берилган. Квадратнинг доирага жойлашишини аниқловчи дастур ёзинг.
23. Икки текис фигуралар умумий нуқтага эга эканлигини аниқловчи алгоритм ва дастур ёзинг: учининг координаталари билан берилган учбурчак ва берилган радиусли маркази координата бошида бўлган доира.
24. Бутун $A > 1$ сони берилган. Энг кичик бутун шундай k ни топиш лозимки унда $2k > A$ шарт бажарилсин.
25. Рим санок системасида берилган сонларни ўнли санок системаси сонларига ўтказиш алгоритми ва дастурини ёзинг.

4. Назорат саволлари.

1. Масалани компьютерда ечишнинг қандай асосий босқичларини биласиз?
2. Масалани компьютерда ечиш босқичларидан қайси бирлари компьютер иштирокисиз амалга оширилади?

3. Масаланинг математик модели нимани ифодалайди?
4. Нима учун объектларнинг хулқ атворини аниқ таҳлил қилиш мумкин эмас?
5. Компьютер ёрдамида моделлаштиришнинг қандай усуллари амалга оширилади?
6. Дастурни яратиш жараёни қандай кетма-кетликда амалга оширилади.

2-лаборатория иши

Мавзу: Дастурлашни лойихалаш. Содда масалалар учун техник топшириқ ва дастур яратиш.

Ишнинг мақсади: Талабаларни дастур яратишни лойихалашга ўргатиш.

Қурилмалар: мос дастурий таъминотлар ўрнатилган ўқув компютери.

Режа :

1. Масаланинг қўйилиши.
2. Дастурий восита яратиш учун техник топшириқ яратиш.
3. Алгоритмнинг блок схемасини яратиш.
4. Дастурни яратиш .
5. Лаборатория иши натижаларини таҳлил қилиш.
6. Лаборатория иши бўйича ҳисоботни ҳимояга тайёрлаш.

1. Масаланинг қўйилиши.

Функция қийматини ҳисоблаш ва унинг графигини яратиш учун қуйидаги ишларни амалга ошириш:

1. Вариант масаласини ўрганиш.
2. Техник топшириқни яратиш.
3. ДВ архитектурасини яратиш.
4. Масаланинг дастурини (ҳар бир функция учун синф яратинг) синфларни яратиш билан тузиш.
5. Дастурни тестлаш.
6. Дастурни отладка қилиш.
7. ДВ ни ҳужжатлаштириш.
8. ДВ нинг сифатини аниқлаш.
9. Мустақил иш ҳисоботини яратинг.

$$y = \frac{\sqrt{\sin(x) + b^2}}{e^{(x+1)^2}}, \quad b = 3.6 :$$

$$2. Z = \arccos \frac{1}{\sqrt[3]{2x}} + 3^{2x},$$

$$3. Z1 = ax^2 + bx + c, \quad a = 3.6, \quad b = 6.3 \quad c = 6.32$$

1. Техник топшириқни яратиш

ДВ яратувчи ташкилотларда тизимли таҳлил натижасида «Талаблар бўйича келишув» ҳужжати ёки «Техник топшириқ» ҳужжати шакллантирилади, бу ҳужжатни баъзан масаланинг қўйилиши деб ҳам юритишади.

«Техник топшириқ» ҳужжати қуйидаги бўлимларга эга:

- кириш;
- ДВ ни яратишни асослаш;
- ДВ нинг вазифаси;
- ДВ га талаблар;
- Дастур ҳужжатларига талаблар;
- техник-иқтисодий кўрсаткичлар;
- ДВ яратишнинг босқичлари;
- назорат қилиш тартиби.

Бу бўлимларнинг ҳар бирини мукаммал кўриб чиқамиз.

Кириш қисми дастурнинг номи ва унинг қўлланиш соҳаси тўғрисида қисқача характеристика бериши лозим. Кириш қисмининг асосий мақсади – ушбу яратилаётган ДВ нинг актуаллигини кўрсатиши ва шунга ўхшаш дастурлар орасида қандай ўрин эгаллашини кўрсатиши лозим.

ДВ ни яратишни асослаш қисмида ДВ ни яратишга киришишга асос бўлувчи ҳужжат номи ва уни тасдиқловчи ташкилот номи келтирилади. Бундай ҳужжат шартнома, буйруқ ва ҳ.к. бўлиши мумкин.

ДВ нинг вазифасида унинг функционал ва эксплуатацион вазифалари тавсифланиши лозим.

ДВ га талаблар қуйидагилардан ташкил топади:

- функционал характеристикаларига талаблар;
- барқарорлигига талаблар;
- эксплуатация қилиниш шартлари;
- техник таъминоти таркиби ва характеристикаларига талаблар;

Бу талаблар ичида энг муҳими бу функционал характеристикаларига талаблар ҳисобланади. Бу бўлимда ДВ нинг асосий бажариши лозим бўлган функциялари келтирилади ва тавсифланади.

Системали таҳлил итератив характерли жараёнга эга ва унинг асосий вазифаси талаблар ва мақсадларнинг тўлиқ комплексини шакллантиришдир.

ДВ ни лойиҳалашнинг биринчи босқичида фойдаланувчининг нимани хохлаётганини жуда аниқ билиб олиш лозим. Фойдаланувчи талабларини тўғри тушуниб олиш ДВ ни тўғри лойиҳалашга ва дастурлашга олиб келади. Техник топшириқни ишлаб чиқиш қуйидаги тартибда амалга оширилади. Аввало, бажарилиши лозим бўлган функциялар терилмаси ва берилмалар характеристикалари аниқланади. Сўнггра натижалар ва уларнинг характеристикалари, уларни тасвирлаш усуллари аниқланади.

Шундан кейин ДВ нинг функциялашиш муҳити аниқланади, шу жумладан фойдаланилаётган ОС, техник таъминотлар ва бошқалар аниқланади.

Техник топшириқни махсус стандартлар орқали баҳолаш мумкин.

2- Мисол. $Y=f(x)$ функциянинг қийматлари ва графигини кўринарли тасвирловчи ДВ яратиш талаб қилинади. Яратилаётган дастур функция қийматларини ва функция графигини берилган кесмада ясаши лозим.

1. Кириш

Ушбу техник топшириқ бир ўзгарувчили функцияларнинг жадвал қийматларини ҳисоблаш ва унинг графигини яратишга мўлжалланган бўлиб, ундан умумтаълим мактабларининг катта синфлари ўқувчилари фойдаланишлари мумкин.

Функцияларни таҳлил қилиш мактабларда ўтиладиган алгебра фанининг энг мураккаб мавзуларидан бири ҳисобланади. Бу мавзунинг ўқувчилар ўрганишларида улар функциянинг характеристик нуқталаридан фойдаланиб бир ўзгарувчили функцияларни текшириш ва уларнинг графикларини кўришни ўрганишади. Мавжуд шундай хилдаги масалаларни ечишга мўлжалланган универсал дастурий таъминот, масалан, Eureka ёки MathCad дастурлари ҳисобланади. Аммо бу дастурлар институтларда олий математика курсини ўтган талабаларга мўлжалланган ва нисбатан мураккаб фойдаланувчи интерфейсига эга. Шу сабабли бу дастурлардан фойдаланиш ўқувчилар учун анча қийинчилик туғдиради. Биз яратаётган ДВ эса ўқувчиларга шу мавзу бўйича билимларини текшириб кўришда ва мустақил тадқиқот ишларини олиб боришда ёрдам беради ва ундан фойдаланиш жуда ҳам соддалаштирилган.

2. ДВ яратишга асослар

ДВ «Информатика ва АТ» кафедрасининг ўқув режаси асосида ва кафедранинг Самарқанд шаҳридаги № мактаби билан 4.09.09 да тузилган шартнома асосида яратилмоқда.

3. Вазифаси

Дастурнинг асосий вазифаси ўқувчиларнинг алгебра курсининг «Бир аргументли функцияларни текшириш» мавзусини ўрганишларида ёрдамлашиш.

4. ДВ га қўйилаётган талаблар

4.1. Функционал характеристикаларига талаблар.

4.1.1. Дастур қуйидаги функцияларни бажаришга мўлжалланган:

- бир аргументли функциянинг аналитик кўринишини киритиш ва уни сақлаш;
- функция аниқланиш соҳаси интервалини киритиш ва уни ўзгартириш;
- аргумент қадамини киритиш ва уни коррективка қилиш;
- функциянинг аниқланиш интервали ичидаги қийматлари жадвалини ҳосил қилиш ва графигини чизиш.

4.1.2. Берилмалар:

- функциянинг аналитик берилиши;
- функциянинг аниқланиш интервали;
- аргументнинг ўзгариш қадами.

4.2. Ишончилигига қўйилган талаблар

4.2.1. ДВ нинг киритилаётган ахборотларни назорат қилиши назарда тутилиши лозим.

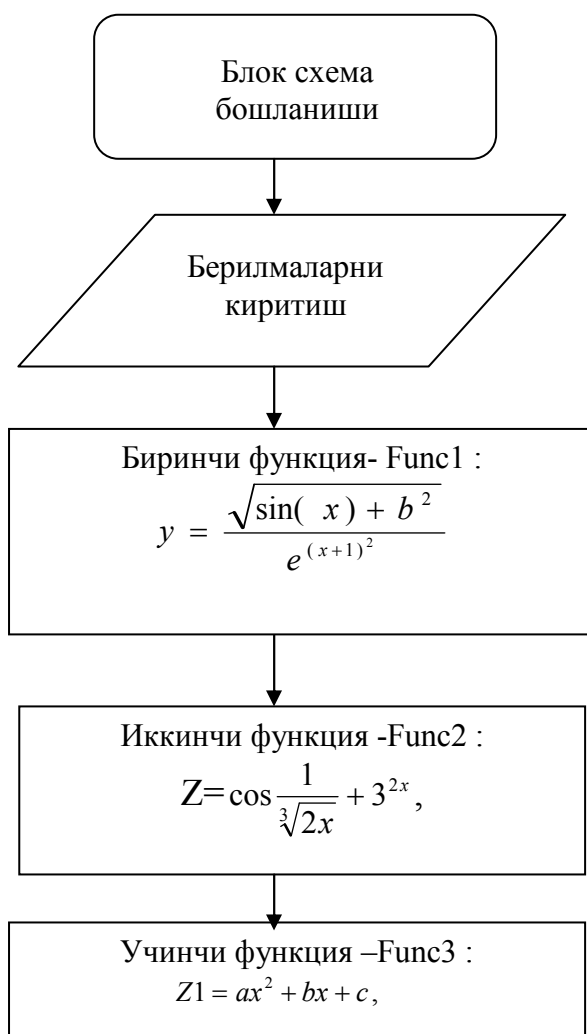
4.2.2. Фойдаланувчи томонидан қилинадиган но коррект ҳаракатларни блокировка қилиш назарда тутилиши лозим.

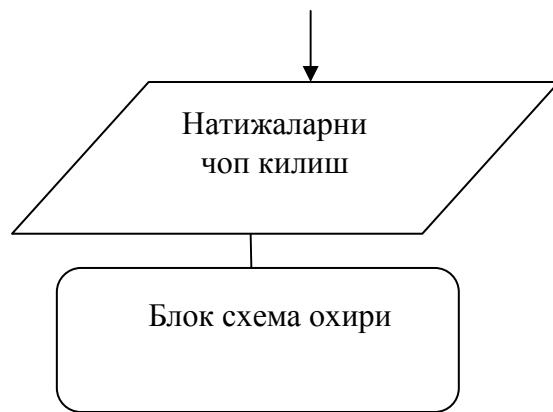
4.3. Техник воситалар параметрларига талаблар.

5. Дастур ҳужжатларига талаблар.

2. Лаборатория ишини бажаришга мисол

Алгоритм блок схемаси





Масалани ечишнинг Паскал алгоритмик тилидаги дастури:

```

program fun;
var
a,b,c,y,z,z1:real;
x,i:integer;
begin
read(a,b,c,x);
for i:=1 to x do
begin
y:=sqrt(sin(x)+b*b)/exp(i/x+1);
z:= cos(1/(exp(1/3*ln(2*i))));
z1:=a*i*i+b*i+c;
write('y=',y);
write('z=',z);
write('z1=',z1);writeln(' i=',i);
end;
end.
  
```

Натижалар:

```

2 3 4 6
y=0.919593518935123, z=0.701211821723673, z1=9, i=1
y=0.778419108106433, z=0.808050764202867, z1=18, i=2
y=0.658917549317746, z=0.852356586685558, z1=31, i=3
y=0.557761663707182, z=0.877582561890373, z1=48, i=4
y=0.472135055172714, z=0.894198425262554, z1=69, i=5
y=0.399653695883924, z=0.906114166933524, z1=94, i=6
  
```

3-лаборатория иши

Мавзу: *Дастурни яратиш. Дастур хатоларини баҳолаш ва тузатиш. Дастур асосида олинган натижаларни таҳлил қилиш. Процедуралар ва функциялар.*

Ишнинг мақсади: *Талабаларни дастурий таъминотни яратишда процедура ва функцияларни ташиқ қилиб дастурлар яратишга ўргатиш.*

Қурилмалар: *мос дастурий таъминотлар ўрнатилган ўқув компютери.*

Режа:

1. Дастурларни дастурлаш тиллари ёрдамида яратиш.
2. Дастур хатоларини баҳолаш ва тузатиш.
3. Дастур асосида олинган натижаларни таҳлил қилиш.
4. Процедуралар ва функциялар.
5. Лаборатория иши учун вариант масалалар.
6. Назорат саволлари.

1. Дастурларни дастурлаш тиллари ёрдамида яратиш.

Дастурий модулни яратишда қуйидаги тартибга амал қилиш тавсия этилади:

- модул спецификациясини назорат қилиш ва ўрганиш, дастурлаш тилини танлаш;
- алгоритмни танлаш ва берилмалар структурасини аниқлаш;
- модулни дастурлаш (кодлаш);
- модулни текшириш;
- модулни компиляция қилиш.

Биринчи босқич. Дастурий модулни яратишнинг бу босқичи асосан дастур структурасини пастдан юқорига назорат қилиб чиқишни тақозо қилади. Модул спецификацияларини ўрганиб чиқиш эса шу модулни лойиҳалашда керак бўлади. Шундан кейин дастурлаш тили танланади.

Иккинчи босқичда қўйилган масалани ечишнинг алгоритми мавжудлиги аниқланади. Агар бизнинг масалага мос алгоритм топилса, ундан фойдаланиш лозим бўлади.

Учинчи босқичда модул танланилган алгоритмик тилда яратилади.

Туртинчи босқичда дастурий таъминот сифати спецификацияси асосида модул матни такомиллаштирилади.

Бешинчи босқичда дастурий таъминотнинг мантиқи қўлда назорат қилинади.

2. Дастур хатоларини баҳолаш ва тузатиш.

Дастурий таъминотни тузатиш – бу дастурдаги хатоларни аниқлаш ва уларни тузатиш жараёнидир.

Дастурий таъминотни тестлаш – бу маълум бир олдиндан режа асосида дастурий таъминотни назорат қилиш жараёнидир.

Шундай қилиб, тузатиш жараёни учта операцияни кўплаб марта такрорлашдир:

- тестлаш, унинг ёрдамида дастурий таъминотдаги хатолар аниқланади;
- дастурлардаги ва дастурий таъминот ҳужжатларидаги хатоларни локализациялаш;
- дастурни таҳрирлаш.

3. Олинган натижаларни таҳлил қилиш.

Тест берилмалари хатоларни келтириб чиқарадиган шартларнинг ҳаммасини текширишни таъминлаши лозим:

- алгоритмнинг ҳар бир тармоқланиши текширилиши лозим;
- ҳар бир навбатдаги тест олдингиларидан фарқли шартларни текшириши лозим;
- биринчи тест жуда ҳам содда бўлиб, у фақат дастурнинг ишлашини текшириши лозим;
- Тестдаги арифметик операциялар сони максимал камайтирилиши лозим;

Мисол. Квадрат $ax^2 + bx + c = 0$ тенглама ечимларини топишга тестлар системаси:

Тест номери	Текширилувчи хол	Кoeffициентлар			Натижалар
		a	b	c	
1	$d > 0$	1	1	-2	$x_1 = 1, x_2 = -2$
2	$d = 0$	1	2	1	Тенглама ечимлари: $x_1 = -1, x_2 = -1$
3	$d < 0$	2	1	2	Ҳақиқий ечимлари йўқ
4	$a=0, b=0, c=0$	0	0	0	Ҳамма коoeffициентлар нулга тенг.
5	$a=0, b=0, c \neq 0$	0	0	2	Нотўғри тенглама
6	$a=0, b \neq 0$	0	2	1	Чизиқли тенглама. Битта ечим бор: $x = -0,5$
7	$a \neq 0, b \neq 0,$	2	1	0	$x_1 = 0, x_2 = -0,5$

	c = 0				
--	-------	--	--	--	--

4. Процедуралар ва функциялар. Модуллар.

Масаланинг ечилиш алгоритми бутун масалани қисм масалаларга лойиҳалаштиради. Қисм масалалар одатда қисм дастурлар шаклида реализация қилинади.

Қисм дастур – бу дастурнинг фақат бир жойида аниқланган ва ёзилган операторлар кетма-кетлигидан иборат бўлиб, уни дастурнинг исталган нуктасидан чақириш мумкин бўлади. Ҳар бир қисм дастур ўзининг уникал номига эга бўлади. Паскал тилида икки хил қисм дастурлар ишлатилади-процедуралар ва функциялар.

Процедура ёки функция - бу номланган операторларнинг тавсифланиш кетма - кетлигидир. Процедура ёки функциялардан фойдаланишда дастур процедура ёки функция матнини ва унга мурожаат қилишни сақлаши лозим. Процедура ёки функциялар матни Паскал дастурининг процедура ва функциялар тавсифланиши бўлимига жойлаштирилади. Процедуралар ва функционал турлар дастурнинг турлар қисмида тавсифланади:

type

FuncType = Function(z: Real): Real;

ProcType = Procedure (a,b: Real; var x,y: Real);

Процедуралар ва функционал турлар процедуралар ва функциялар сарлавҳалари сингари тавсифланади.

Мисол. Симпсон усули билан аниқ интегрални ҳисоблаш дастурини яратиш талаб қилинади. Интеграл тагидаги функцияни дастурда функцияларни ташкил этиш орқали амалга ошириш талаб қилинади. Аниқ интегрални Симпсон усули билан ечиш алгоритми куйидаги формула билан аниқланади:

$$ISimps = 2 \cdot h / 3 \cdot (0.5 \cdot F(A) + 2 \cdot F(A+h) + F(A+2 \cdot h) + 2 \cdot F(A+3 \cdot h) + \dots + 2 \cdot F(B-h) + 0.5 \cdot F(B))$$

Бу ерда А ва В – интеграллашнинг куйи ва юқори чегаралари, N – интеграллаш интервалини бўлишлар сони, $h = (B-A)/N$, бу ерда N фақат жуфт бўлиши лозим.

Program INTEGRAL;

type

Func= function(x: Real): Real;

var

I,TN,TK:Real; N:Integer;

{SF+}

Function Q(t: Real): Real;

Begin Q:=2*t/Sqrt(1-Sin(2*t)); end;

```

{$F-}
Procedure Simps(F:Func; a,b:Real; N:Integer; var INT:Real);
var
  sum, h: Real;    j:Integer;
begin
  if Odd(N) then N:=N+1;
  h:=(b-a)/N;    sum:=0.5*(F(a)+F(b));
  for j:=1 to N-1 do
    sum:=sum+(j mod 2+1)*F(a+j*h);
    INT:=2*h*sum/3;  end;
begin
  WriteLn(' BBEDI TN,TK,N');
  Read(TN,TK,N);  Simps(Q,TN,TK,N,I);
  WriteLn('I=',I:8:3); end.

```

Ҳар бир дастурни яратишда биз унга жуда катта система сифатида қарашимиз лозим ва биз уни соддалаштириш учун ҳамма мумкин бўлган ишларни қилишимиз лозим.

5. Лаборатория ишини бажаришга вариант топшириқлари.

1. Компонентлари $A(4, 4)$, $B(3, 3)$, $C(2, 2)$ квадрат матрицалар элементларининг йиғиндиларидан ташкил топувчи $x = \{x_1, x_2, x_3\}$ векторнинг узунлигини ҳисобловчи алгоритм ва дастур ёзиш талаб қилинади. Квадрат матрицалар элементлари йиғиндисини ҳисоблашни функция шаклида расмийлаштириш талаб қилинади.
2. Қуйида берилган учта $A(8, 10)$, $B(4, 8)$, $C(3, 7)$ матрицалардаги мусбат, манфий ва нул элементлар сонини ҳисобловчи алгоритм ва дастур ёзиш талаб қилинади. Кўрсатилган сонларни ҳисоблашда функция ташкил қилиш талаб қилинади.
3. Қуйидаги учта бир ўлчамли массивларнинг элементлари йиғиндисини функциялар ташкил қилиш йўли билан ҳисобловчи алгоритми ва дастурини ёзиш талаб қилинади.
4. Қуйидаги ҳақиқий сонлар берилган $a_0, \dots, a_6$. $x = 1, 2, 3, 4$ лар учун $P(x + 1) - P(x)$ ларни ҳисоблаш талаб қилинади, бу ерда $P(y) = a_6y^6 + a_5y^5 + \dots + a_0$. $P(y)$ ни ҳисоблаш функция ташкил қилиш билан амалга оширилсин.
5. Қуйидаги векторларнинг узунликлари йиғиндисининг ярмини ҳисобловчи дастур ёзинг: $A(1,5; 2,5; -0,3)$ ва $B(-11,7; 9,3; 2,5; 3,7; -1,2)$. Векторлар узунлигини аниқлаш функция ташкил қилиш билан амалга оширилсин.
6. Қуйидагини ҳисоблаш дастури ёзилсин: $v = a*b*c*\sin(a-c)*\exp(-b)$, бу ерда a, b, c – лар қуйидаги массивларнинг энг кичик элементлари $A(10, 10)$, $B(6, 8)$, $C(4, 10)$. Энг кичик элементни аниқлаш функция ташкил қилиш билан амалга оширилсин.
7. Қуйидаги матрицалар кўпайтмасини функция шаклида дастурлаш талаб қилинади $A(3, 4)$, $B(6, 8)$.

8. Қуйидаги ҳақиқий сонлар берилган $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Ўнбурчакнинг учларининг координаталари қуйидагича аниқланади $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Икки нукта орасидаги масофани ҳисоблашга функция ташкил қилинг.
9. Ҳақиқий сонлар берилган s, t . Ҳисобланг: $f(t, -2s, 1, 07) + f(2, 2, t, s-t)$, бу ерда дастурда $f(a, b, c) = (2a - b - \sin(c)) / (5 + |c|)$. $f(a, b, c)$ ни функция сифатида дастурланг.
10. $ax^2 + bx - 4 = 0$; $y^2 - cy + d = 0$; $2z^2 + dz - 1 = 0$. Тенгламаларнинг катта ечимларини ҳисобланг ва уларни $U(3)$ массивга ёзинг. Агар ечим комплекс бўлса, уни нулга тенг деб олинг. Илдизларни ҳисоблашни функция ёрдамида амалга оширинг.
11. Ҳисобланг: $S = N! + K! + (I + K)!$, бу ерда $N = 3, K = 5, I = 8$. Факториални ҳисоблашга функция яратинг.
12. Иккита массив $A(10)$ ва $B(20)$ ни битта массивга шундай жойлаштирингки, бу массивда мусбат элементлар массив бошида жойлашсин, манфийлари эса охирига жойлашсин. Нул элементлар эса чиқариб ташлансин. Массив элементларини тартиблаштиришни процедурада амалга оширинг.
13. Бутун турдаги $X(40, 35)$ массивида ҳар бир қатордаги жуфт сонли элементлар сонини ва уларнинг энг каттасини аниқланг. Жуфт элементларни аниқлашда функциядан фойдаланинг.
14. Назорат натижалари бўйича гуруҳлардаги ўртача балл ҳисоблансин. Гуруҳларнинг баллари қуйидаги матрицага киритилган $A(25, 5), B(23, 5), C(22, 5), D(24, 5)$. Ўртача баллни ҳисоблаш функциялар орқали амалга оширилсин.
15. Қуйидаги массив учун $T(33, 19)$ қаторлар кўпайтмасини ҳисоблаш дастури ёзилсин: биринчи қаторни охириги қатор билан, иккинчини охиригидан битта олдинги қатор билан. Қаторларни кўпайтириш учун функция ташкил қилиш талаб қилинади.
16. Бутун турдаги массив $Z(10, 15)$ да устунлар бўйича максимал тоқ элементларнинг энг каттасини топиш талаб қилинади. Тоқ элементларни ҳисоблашда функция ташкил қилиш талаб қилинади.
17. Қуйидаги натурал сонлар берилган: n, m ($n < m$). Бу сонлардан қуйидагилар ташкил этилади $n, n + 1, \dots, m$. Бу сонларнинг қайси бири катта йилларга мос келишини аниқланг. Текширишни функция сифатида расмийлаштиринг.
18. Қуйидаги $A(12, 15)$ массивида устунлар кўпайтмаларини ҳисоблаш дастурини яратиш талаб қилинади: иккинчисини унинчисига, учинчисини тўққизинчисига. Устунларни кўпайтиришга функциялар ташкил қилинг.
19. Иккита a ва b ҳақиқий сонлари берилган. Қуйидагиларни ҳосил қилиш дастурини яратиш талаб қилинади: $U = \min(a, b), V = \min(ab, a+b), \min(U+V, 3.14)$. Иккита сондан кичигини ҳисоблашда функция ташкил қилинг.

20. Қуйидаги учта бутун турдаги массивларда $A(40)$, $B(30)$, $C(60)$ 3 га каррали ҳамма элементларини топиш дастурини яратиш талаб қилинади. Излашга функция аниқланг.

21. Қуйидагини ҳисоблашга дастур яратинг: $Z = (x + y) / (kn)$, бу ерда x ва k $A(40)$ массивининг мусбат элементлари йиғиндиси ва сони, y ва n эса $B(50)$ массивининг манфий элементлари йиғиндиси ва сони. Массивларнинг мусбат ва манфий элементлари йиғиндиси ва сонини аниқлашга функциялар ташкил килинг.

22. $A(5,7)$ матрицанинг нулдан фарқли элементларининг ҳар бир устун бўйича кўпайтмасини ҳисоблаш дастурини яратинг. Бунда устун элементларини кўпайтиришга функция ташкил килинг.

23. Иккита квадрат $A(3, 3)$ ва $B(4, 4)$ матрицалар берилган. Уларни транспонировка қилиш дастурини яратинг. Функция шаклида расмийлаштиринг.

24. Қуйидаги $X(60)$, $Y(75)$, $Z(80)$ массивларнинг мусбат элементлари ўрта арифметик қийматини ҳисоблаш дастурини ёзинг. Бунда ўрта арифметикларни ҳисоблаш учун функция ташкил килинг.

6. Назорат саволлари

1. Тестлаш ва тузатишни қандай тушунасиш?
2. Хатоларнинг қанақа турларини биласиз?
3. Процедура ва функцияларнинг вазифалари?

4-лаборатория иши

Мавзу: Модулли дастурлар яратиш

Ишнинг мақсади: Талабаларни модулли дастур яратишга ўргатиш.

Қурилмалар: мос дастурий таъминотлар ўрнатилган ўқув компютери.

Режа :

1. Асосий тушунчалар.
2. *Pascal* нинг стандарт модуллари.
3. Модуллерни дастурга улаш.
4. Модулли дастурни яратиш.
5. Модуллер яратишга мисоллар.
6. Лаборатория иши натижаларини таҳлил қилиш.
7. Лаборатория иши бўйича ҳисоботни ҳимояга тайёрлаш.

1. Асосий тушунчалар.

Модул – дастур қисми бўлиб, у дастурнинг бошқа қисмларидан алоҳида компиляция қилинади. Модулларнинг алоҳида компиляция қилиниши модулларнинг асосий афзалликларидан бири ҳисобланади.

Энг содда дастур модулига процедура ва функциялардан фойдаланиб эришилиши мумкин аммо бу ҳамиша ҳам мумкин бўлавермайди.

2. Pascal нинг стандарт модуллари.

Pascal тилининг библиотекаси таркибига киритилган модуллар жуда кўп ва уларни бирма бир кўриб ўта олмаймиз. Уларнинг ичида асосий модул бу **System** модули ҳисобланади.

System модули

System модули асосий модул ҳисобланади ва унинг таркибига Pascal тилининг ҳамма стандарт процедуралари ва функциялари киритилган. Бу модул махсус уланишни талаб қилмайди: ундан бевосита мурожаат қилиб фойдаланиш мумкин.

Бу модул қуйидагиларни ўз таркибида сақлайди:

1. **Даражали катталиклар турларларини ишлаш қисм дастурлари** (**dec, inc, odd, pred, succ**).
2. **Арифметик функциялар.**
3. **Берилмалар турларини ўзгартириш функциялари** (**chr, ord, round, trunc**).
4. **Дастур бажарилиши жараёнини бошқариш процедуралари** (**break, continue, exit, halt**).
5. **Қаторларни ишлаш қисм дастурлари** (**concat, copy, delete, insert, length, pos, str, val**).
6. **Файлли киритиш ва чиқариш қисм дастурлари.**

3. Модулларни улаш

Бирон бир модулни дастурга улаш учун дастур сарлавҳасидан кейин қуйидаги қаторни ёзиш лозим: `uses <модул номи>;`

Агар уланувчи модуллар бир нечта бўлса, у ҳолда бу қатор қуйидагича ёзилади: `uses <модул номи _1>,...,<модул номи _N>;`

Модулда тавсифланган ҳамма константалар, берилмалар турлари, ўзгарувчилар, процедуралар ва функциялар бу модул асосий дастурга улангандан кейин қўшимча тавсифланмасдан шу дастурга доступли

ҳисобланади. Масалан, `abs()` функциясидан уни эълон қилмасдан ва тавсифламасдан фойдаланиш мумкин, чунки у System стандарт модули таркибига киритилган.

4. Модулли дастурни яратиш

Модул таркиби

Стандарт модулар тўртта секциядан ташкил топади (улардан исталгани бўш бўлиши мумкин, аммо секция номи барибир ёзилади):

```
unit <модул номи>;
interface           {ташқи боғланиш секцияси}
implementation      {амалга ошириш секцияси}
begin               {инициализация қилиш секцияси}
end.
```

Бу секцияларнинг вазифаларини кўриб чикайлик.

Модул номи

Дастур сарлавҳасини ёзмаслик мумкин (**program** <дастур номи>;),
модул номи эса албатта ёзилиши лозим (**unit** <модул номи>;).

Одатда модул номи ва уни сақловчи файл номи бир хил бўлиши лозим.
Яъни, `modul_1` номли модул `modul_1.pas` номли файлга жойлаштирилиши лозим.

Ташқи боғланишлар секцияси

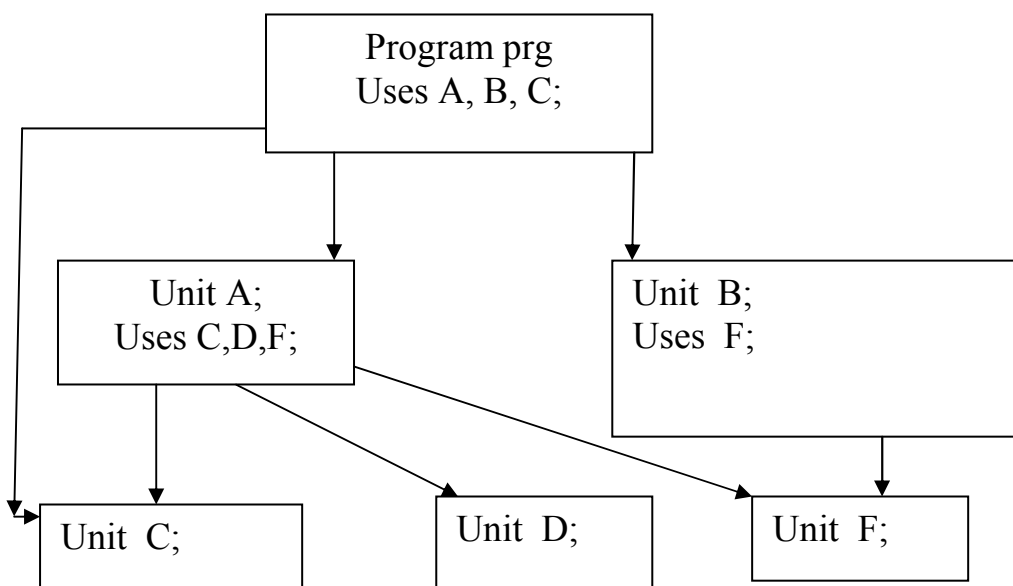
Ташқи боғланишлар секцияси модулдан ташқарида кўринарли бўлиб турилиши лозим бўлган константалар, турлар, ўзгарувчилар, қисм дастурлар ва ҳоказоларни тавсифлаш учун хизмат қилади. Агар бирон бир объектда тасвифланган объект тўғрисида маълумотлар керак бўлса, у ҳолда шу модулнинг номи шу секцияда келтирилиши лозим:

```
interface
[ uses <ёрдамчи модуларнинг рўйхати> ; ]
[ const <ташқи константалар рўйхати> ; ]
[ type <ташқи берилмалар турлари рўйхати> ; ]
[ var <ташқи ўзгарувчилар рўйхати> ; ]
[ procedure <ташқи процедурани эълон қилиш> ; ]
[ function <ташқи функцияларни эълон қилиш> ; ]
```

Масалан, бизга иккита модул берилган бўлсин: базавий константаларни ва берилмалар турларини тавсифловчи `mod_const` модули ва шу тавсифларни

ишлатувчи mod1 модули, у ҳолда ташқи алоқалар секцияси қуйидагича ёзилади:

```
unit mod_const;  
  interface  
    const sto = 100;  
    type one_to_sto = 1..sto;  
  ...  
unit mod1;  
  interface  
    uses mod_const;  
    const dvesti = 2*sto;  
    type massiv = array[1..dvesti] of byte;  
    var a: massiv;  
    b: one_to_sto;  
    function min(x,y:one_to_sto):one_to_sto;  
  ...
```



1-расм. Модулли дастур таркиби

Тадбиқ қилиш секцияси

Модулнинг бу секцияси ташқи боғланиш секциясида аниқланган ҳамма қисм дастурларни тадбиқ қилишни амалга оширади. Бундан ташқари, шу секцияда ички белгилар (нишонлар), константалар берилмалар турлари, ўзгарувчилар ва қисм дастурлар эълон қилинади.

```
implementation  
[uses <ёрдамчи модуллар рўйхати>;]  
[const <ички константалар рўйхати >;]
```

```
[type <ички берилмалар турлари рўйхати >]
[var <ички ўзгарувчилар рўйхати >]
[procedure <ташқи процедураларни тавсифлаш >]
[function <ташқи функцияларни тавсифлаш >]
[procedure <ички процедураларни эълон қилиш ва тавсифлаш >]
[function <ички функцияларни эълон қилиш ва тавсифлаш >]
```

Модулнинг бу секциясида ҳар хил модуллар бир-бирларидан фойдаланишлари мумкин, яъни қуйидаги ёзув ўринли:

```
unit mod_1;
  interface
    ...
  implementation
    uses mod_2;
  ...
unit mod_2;
  interface
    ...
  implementation
    uses mod_1;
  ...
```

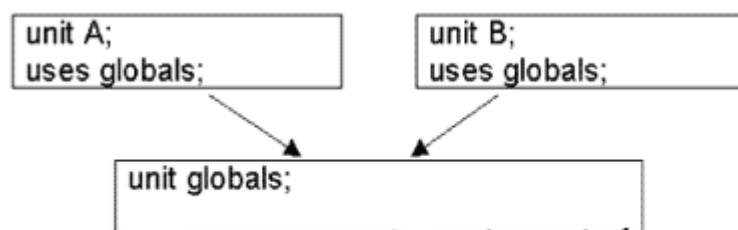
Инициализация қилиш секцияси

Инициализация қилиш секцияси ҳамма дастурга уланган модуллар учун асосий дастур бажарилишидан олдин бир марта бажарилади:

```
begin
  <операторлар кетма-кетлиги >
end.
```

Модуллarning боғланиши

Агар бир нечта тенг кучли модуллар битта константалар, ўзгарувчилар ва ҳоказоларга доступга эга бўлишлигини хоҳласак, у ҳолда қўшимча модул киритиш лозим бўлади ва унда ҳамма глобал ўзгарувчилар ва константалар ва ҳоказолар глобал шаклида тавсифланади. Бу янги модулнинг тадбиқ қилиш секцияси бўш бўлади:



2-расм. Аниқловчи модуллардан фойдаланиш.

Модулларни компиляция қилиш.

Модулларнинг матнлари **.pas** кенгайтмали файлларда сақланади, компиляция натижалари эса **.tpu** кенгайтмали файлларда сақланади.

Модулга мисол

Мисол сифатида бутун сонлар учун **min** ва **max** арифметик функцияларини ва даражага кўтариш функциясини сақловчи модулни оламиз. Бу функциялар **System** стандарт модулида мавжуд эмас.

```

unit modul1;
interface
  function min(a,b: integer): integer;
  function max(a,b: integer): integer;
  function deg(a,b: integer): integer;

implementation
  function min;
  begin if a>b then min:= b
        else min:= a;
  end;

  function max;
  begin if a<b then max:= b
        else max:= a;
  end;

  function deg;
  begin deg:= exp(b*ln(a));
  end.
  
```

Энди бу модулни асосий дастурга улаб бу учала функциядан ҳам фойдаланиш мумкин.

Энди асосий дастурни қуйидагича ёзиш мумкин:

```

program modulPR;
uses modul1;
var
k, i,j:integer;
begin
readln(i,j);
writeln(max(i,j));
writeln(min(i,j));
writeln(deg(i,j));
k:= max(i,j)+ min(i,j)+ deg(i,j);
writeln(k);
end.

```

Худди шунингдек, динамик таркибли берилмаларни (рўйхатлар, дарахтлар стеклар ва бошқаларни) бир марта қисм дастур шаклида ёзиб олиб уни таҳрирлаш, отладка қилиш ва компиляция қилиб олиб бирон бир махсус модулда сақлаш лозим ва улардан ҳохлаган вақтда фойдаланиш мумкин. Бундай ёндошув дастурлаш технологияси ёндошувларига киради ва улардан кўплаб марта фойдаланишади.

2-мисол. Факториални ҳисоблайдиган модул ёзинг ва ундан фойдаланиб $n!$, $m!$, $(n-m)!$ ни қийматини ҳисоблаш учун дастур тузинг.

```

unit modul1;
interface
function fact(n: integer): integer;
implementation
function fact;
begin
var nf,mf,nm,f,n,m pf,i:integer;
begin
if k<0 then fact:=0
else if k=0 then fact:=1
else
begin
pf:=1;
for i:=2 to k do
pf:=pf*I;
fact:=pf
end
end; End.

```

```

program modulFact;
uses modul1;
Var n1,m1,nf1,mf1,nmf1:integer;
Begin

```

```

readln(n1,m1);
nf1:=fact(n1);
mf1:=fact(m1);
nmf1:=fact(n1*m1);
write('nf1=',nf1,' ':4,'mf1=',mf1,' ':4,'nmf1=',nmf1);
End.

```

6- лаборатория иши.

Мавзу: Объектга йўналтирилган дастурлаш технологияси

Режа:

1. C++ class лари ва уларнинг таърифи, асосий тушунчалари.
2. Class формати.
3. Class лардан фойдаланиб дастурлаш.

1. C++ Class лари ва уларнинг таърифи, асосий тушунчалари

Жуда кўп ҳолларда дастурлашда берилмалар тўпламлари билан ишлашга тўғри келади. Масалан, ёзувлар, ёзувлар бинар дарахти билан, тармоқ берилмалари билан ва ҳ.з. лар билан ишлашга тўғри келади. Ёзувлар ҳар турдаги берилмаларни бирлаштиради. Ҳар бир тавсифланаётган ахборот алоҳида ҳеч нарсани билдирмайди. Улар биргаликда эса бирон бир объектни тавсифлаши мумкин. Энг содда таркиб масалан, массив алоҳида бир хил турдаги қийматларни яхши сақлайди, аммо ҳар хил турдаги катталикларни сақлашга мутлақо ярамайди. Бундай ахборотлар комплексларини объектлар деб атаيمиз. Шу вақтгача биз дастурлаш тилида аниқланган турлар билан ишлаш дастурларини яратдик.

Синфлар – бу маълум бир дастурда аниқланган турдир. Синфни аниқлаш уни ташкил этувчи атрибутлар ва улар устида қандай операцияларни бажарилишини аниқланишидир.

Бир нечта ҳар хил турдаги катталикларни битта объектга бирлаштирувчи конструкция синф деб аталади.

Синф объектлар турини ва унинг модули бирлигини ҳам тавсифлайди. ОЙД да бош дастур тушунчаси йўқолади, шунингдек қисм дастурлар ҳам йўқолади. Улар фақат синфлар қисмлари шаклида қатнашишлари мумкин. Пакетлар тушунчаси ҳам йўқ. Ҳатто Integer ва Real каби базавий турлар ҳам базавий синфларга кирувчи синф сифатида қаралади. ОЙД да фақат битта базавий ҳисоблаш механизми бор. Маълум бир объект борки, у ҳамиша бирон бир синф экземплярлари ҳисобланади, ҳисоблаш эса ушбу объект чақирувчи синф компоненти амали ҳисобланади. Масалан, ойнани экранда кўрсатиш учун ойнани ифодаловчи объектнинг Window объекти экземплярлари бўлган display компоненти чақирилади.

Инкапсуляция

Синфларни яратишда кўпинча фақат синфнинг ўз ичида ишлатилувчи компоненталарини ишлатишга тўғри келади. Бундай компоненталардан фойдаланувчилар фойдаланишмайди ва биз уларни инкапсуляция қилишимиз уларни ёпишимиз лозим бўлади. Синфлар мутлақо компоненталарини чақириш орқали ахборот алмашади.

Ворислик

ДВ ларни яратишда кўплаб синфлар яратилади. Уларнинг кўплари олдин яратилган синфларнинг вариантлари бўлиши мумкин. Бундай мураккаб тизимларни соддароқ бошқариш учун синфилаштириш –ворислик механизми ишлатилади. А синфи В синфи вориси ҳисобланади, агарда у В синфининг баъзи компоненталарини меросласа.

Полиморфизм

Полиморфизм (polymorphism) – бу мумкин бўлган ҳар хил турдаги объектларни синф моҳиятига бирлаштира олиш имкониятидир. Полиморфизм ворислик усули билан назорат қилинади. Полиморфизм - энг универсал ва радикал воситадир. У инкапсуляциялаш ва ворислик билан чамбарчас боғлиқ. Полиморфизм шундай ҳолатки, бунда қандайдир битта нарсга кўп шаклларга эга бўлади. Дастурлаш тилида «кўп шакллар» дейилганда, битта ном автоматик механизм томонидан танлаб олинган турли кодларнинг номидан иш кўриш тушунилади. Шундай қилиб, полиморфизм ёрдамида битта ном турли хулқ-атвори билдириши мумкин.

2. Class *formati*

Class ni умумий ҳолда қуйидагича аниқлаш мумкин:

Class **className**

{ **private:** //қисмий

<privat маълумотлар аъзолари, конструкторлар, усуллар>

protected: //himoyalangan

<ҳимояланган маълумот аъзолари, конструкторлар ва усуллар>

public: //умумий

<оммавий хусусиятлар> <оммавий маълумот аъзолари > <оммавий конструкторлар ва деструкторлар > <оммавий усуллар> }

Мисол. Кубикларни ташлаш ўйини class ни яратиш.

//Класни аниқлаш (кубикларни ташлаш):

class Tdice

{ public:

int LastValue;

intGetTDiceValue(); };

Metodini aniqlash:

Int Tdice::GetDiceValue()

```
{   LastValue=random(n)+1;
return LastValue; }
```

Class ни тавсифлаш калит сўз **class** дан бошланади, ундан кейин class номи ва иккита фигурали қавс class танасини очади ва ёпади.

Class аниқланишидаги очилган фигурали қавсдан кейин **public** калит сўзи ёзилади ва бу калит сўздан кейин синфнинг объектлари ёзилади. Юқоридаги дастурда KreditKarta турига эга бўлган **nds** ўзгарувчиси эълон қилинмоқда.

3. Class лардан фойдаланиб дастурлаш

Комплекс сонлар алгебраси учун class ва ундан фойдаланиб дастур яратиш.

1-Мисол.

Комплекс сонлар ҳақиқий ва мавҳум қисмлардан ташкил топган ва улар устида арифметик амаллар бажариш мумкин. Комплекс сонлар арифметикаси учун Complex class ини яратиш ва ундан фойдаланиб бутун арифметикани амалга оширувчи дастур яратиш талаб қилинади.

Ечиш.

Биз кўйидаги Complex class ини аниқлаймиз:

```
class Complex
{   public:
    int haq; // ҳақиқий қисм
    int mav; // мавҳум қисм
    void Add(Complex x);
    // комплекс сонни қўшимча қилиш
};
```

Комплекс сон ҳақиқий қисм **haq** бутун сондан ва мавҳум қисм **mav** дан ташкил топади – улар синф атрибутлари дейилади. Complex class и учун битта усул Add усули аниқланган.

Complex class и аниқлангач биз Complex туридаги ўзгарувчиларни аниқлаймиз:

```
Complex number;
```

Энди, number номли ўзгарувчи Complex туридаги қийматни сақлайди, яъни Complex classi объектнини сақлайди. Объектлар яратилгач, биз унинг қийматларини ўрнатишимиз мумкин:

```
number.haq = 1;
number.mav= 2;
```

". "операцияси объект атрибутига мурожат қилишни билдиради. Complex classi нинг яна бир объектини яратамиз ва уни биринчи объектга қўшимча қиламиз:

```
Complex num2;
number.Add(num2);
```

Бундан кўриниб турибдики, **Add** усули объект билан амалга оширилади. Объект номи (ёки объектни сақловчи ўзгарувчи аслида бир

нарсадир), бу ҳолда **number**, биринчи ёзилган. Шундан сўнг нуқта билан ажратилган ҳолда **Add** усули аргументи билан ёзилган. Усуллар кўпинча хабарлар деб ҳам аталади. Аммо хабарни жўнатиш учун уни қабул қилиб олувчи бўлиши лозим. Шундай қилиб **number** объектига num2 аргументи **Add** хабари жўнатади. **Number** объекти бу хабарни қабул қилиб олади ва ўз қийматига аргумент қийматини қўшади.

Complex classi ning усулини аниқлаш:

```
void
Complex::Add(Complex x)
{
    this->haq = this->haq + x.haq;
    this->mav = this->mav + x.mav;
}
```

Бу усулда биринчи иккита қатор Complex klassi ning Add усули эканлигини билдиради. Фигурали кавслар ичида эса Add усули операциялари аниқланган. Бу аниқланиш қуйидагича тушунилади: Complex объекти қийматини қаралаётган объект қийматига қўшиш учун ҳақиқий қисмлар қўшилади ва бу қиймат жорий объект ҳақиқий қисми атрибутда сақланади. Худди шунингдек, икки complex сонлар мавҳум қисмлари қўшилади ва натижа жорий объект мавҳум қисмини аниқловчи атрибутда сақланади.

this-> ёзуви кўпинча туширилиб қолдирилади:

```
// complex сонлар синфини аниқлаш.
class Complex {
public:
    int mav;
    int haq; // ҳақиқий қисм
    int mav; // мавҳум қисм
    void Add(Complex x); // комплекс сонни қўшимча қилиш };
// void ning қўшиш усулини аниқлаш
Complex::Add(Complex x)
{    haq = haq + x.haq ;
    mav = mav + x.mav ;}
Int main()
{
    Complex number;
    number.haq = 1;
        // Complex синфининг биринчи объекти
    number.mav = 3;
    Complex num2;
        // Complex синфининг иккинчи объекти
    num2.haq = 2;
    num2.mav = 1;
    number.Add(num2);
```

```
return 1;
```

Бу синфни соддарок кўринишда аниқлаймиз:

```
class Kompl {  
public:  
    int haq;  
    int mav;  
    Kompl ()  
    { haq=0; mav=0; }  
};
```

Бу синф учун усуллар аниқланади, дастурни келтирамиз:

```
#include <iostream>  
#include "kompl.h"  
using namespace std;  
  
void main () {  
    Kompl x;  
    x.haq=4; x.mav=5; Kompl y; y.haq=6; y.mav=3;  
    cout<<"Berilgan kompleks sonlar\n"<<x.haq<<"+"<<x.mav<<"i\n";  
    cout<<y.haq<<"+"<<y.mav<<"i\n";  
    Kompl z=x-y;  
    cout<<"Ayirmasi="<<z.haq<<"+"<<z.mav<<"i"<<endl;  
    z=x+y;  
    cout<<"Yig'indisi="<<z.haq<<"+"<<z.mav<<"i"<<endl;  
    z=x*x;  
    cout<<"Ko'payitmasi="<<z.haq<<"+"<<z.mav<<"i"<<endl;  
    int a=8; z.mav=4;  
    z=x-a;  
    cout<<z.haq<<"+"<<z.mav<<"i"<<endl;  
    z=x+a;  
    cout<<z.haq<<"+"<<z.mav<<"i"<<endl;  
    z=x*a;  
    cout<<z.haq<<"+"<<z.mav<<"i"<<endl;  
}  
class Kompl {  
public:  
    int haq;  
    int mav;  
    Kompl () {  
        haq=0; mav=0;  
    }  
};
```

```

Kompl operator-(const Kompl x)const;
Kompl operator-(long x)const;
Kompl operator+(long x) const;
Kompl operator+(const Kompl x)const;
Kompl operator*(long x) const;
Kompl operator*(const Kompl x)const;
};
Kompl Kompl::operator-(const Kompl x)const {
    Kompl y;
    y.haq=haq-x.haq;
    y.mav=mav-x.mav;
    return y;
}
Kompl.h
Kompl Kompl::operator-(long x)const {
    Kompl y;
    y.haq=haq-x;
    return y;
}
Kompl Kompl::operator+(long x)const {
    Kompl y;
    y.haq=haq+x;
    return y;
}
Kompl Kompl::operator+(const Kompl x)const {
    Kompl y;
    y.haq=haq+x.haq;
    y.mav=mav+x.mav;
    return y;
}
Kompl Kompl::operator*(long x)const {
    Kompl y;
    y.haq=haq*x;
    return y;
}
Kompl Kompl::operator*(const Kompl x)const {
    Kompl y;
    y.haq=haq*x.haq-mav*x.mav;
    y.mav=haq*x.mav+mav*x.haq;
    return y;
}

```

Дастур ишлаши натижаси:

```

Berilgan kompleks sonlar
7+9i
11+13i
Ayirmasi=-4+-4i
Yig'indisi=18+22i
Ko'payitmasi=-32+126i
-1+0i
15+0i
56+0i
Press any key to continue

```

2-мисол

Ўзида исми, фамилияси, кредит карточка номери майдонларидан таркибланган объектни тавсифловчи синфни аниқлаш ва ундан фойдаланиб дастур ёзиш талаб қилинади:

Қуйида C++Builder 6.0 даги дастур келтирилади.

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    class NameDataSet
    {
    public:
        int kreditKartochkaNomeri[20];
        float summasi[20];
        String ismi[20];
        String Familiyasi[20];
    };
    NameDataSet nds;
    int i,n;
    n=StrToInt(Edit1->Text);
    int Max= 20;

```

The screenshot shows a graphical user interface for a C++ application. At the top, there are two memo fields. The left memo field, labeled 'Memo1', contains a list of names: Sermat, salimbek, Ko'chim, Odil, To'lqin, Sarimsoq, Sherxon, Sherzod, Bekzod, To'raqul, Karim, Botir, to'rabek, Sharif, and Sunnat. The right memo field, labeled 'Memo2', contains a list of names: Begimquliv, Karimov, Salimjonov, Ko'chimov, Botiraliyev, Shrinqulov, Sindorov, mamadaliev, Toshpolatov, Tursunqulo, Berdiyrov, Sariko'zov, koratoyev, Kalandarov, and Begaliev. Below these memos, there is a text box containing the number '15' and a button labeled 'Button1'. On the right side of the window, there are four labels: 'Sermat', 'Begimquliv', 'K/Kartochka N 12355', and 'summa= 123546.5'.

```

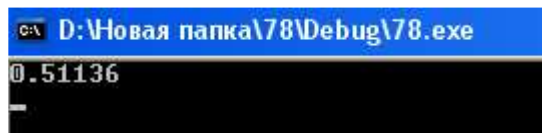
    for(i=1;i<=Max-1;i=i+1)
    { nds.ismi[i]=Memo1->Lines->Strings[Memo1->Lines->Count-i];
      nds.Familiyasi[i]=Memo2->Lines->Strings[Memo2->Lines->Count-i];} ;
    for(i=1;i<=Max-1;i=i+1)
    { nds.kreditKartochkaNomeri[i]=12340+i;
      nds.summasi[i]=123456.5+6*i; } ;
    Label1->Caption=String(nds.ismi[n]);
    Label2->Caption=String(nds.Familiyasi[n]);
    Label3->Caption="K/Kartochka N "+IntToStr(nds.kreditKartochkaNomeri[n]);
    Label4->Caption="summa= "+FloatToStr(nds.summasi[n]);
  }

```

```

#include <iostream.h>
#include <Math.h>
// синфни аниклаш
class Func
{public:
  double Value;
  //усулни эълон килиш
  FuncValue(double a,double b,double s,double x);};
//усулни аниклаш
Func::FuncValue(double a,double b,double s,double x)
{a=cos(x);b=sqrt(b*x);s=sin(-s);
  Value=(a*x*x+b*x +s);}
// асосий дастур
int main()
{Func Fun;
  double x,a,b,s,z;
  x=3.2;a=6.2; b=4.2; s=1.5;z=0.0;
  // синфга мурожат килиш
  Fun.FuncValue(a,b,s,x);
  cout<<Fun.Value<<endl;
  cin >>z;}

```



Синфга икки марта мурожат килиш:

```

#include <iostream.h>
#include <Math.h>
class Func
{public:
  double Value;
  FuncValue(double x,double a,double b,double s);};

```

```

Func::FuncValue(double x,double a,double b,double s)
{a=cos(x);b=sqrt(b*x);s=sin(-s);
  Value=(a*x*x+b*x +s);}
int main()
{Func Fun;
  Func Fun1;
  double x,a,b,s,z;
  x=5.2;a=6.2; b=5.2; s=4.5;z=0.0;
  Fun.FuncValue(a,b,s,x);
  cout<<Fun.Value<<endl;
  x=15.2;a=16.2; b=5.2; s=4.5;z=0.0;
  cout<<Fun1.Value<<endl;
  cin >>z;}

```

```

D:\Новая папка\78\Debug\78.exe
71.9392
-9.25596e+061

```

```

#include <iostream.h>
#include <Math.h>
class Func
{public:
  int Value;
  FuncValue(int a,int b,int s,int x);};
Func::FuncValue(int a,int b,int s,int x)
{  Value=(s+a+b*x);}
int main()
{Func Fun;
  int x,a,b,s,z;
  x=1;a=4; b=4; s=2;z=2;
  z=Fun.FuncValue(a,b,s,x);
  cout<<z<<endl;}

```

```

D:\оютр яряър\и1\оютр яряър
10
Press any key to continue

```

```

include <iostream.h>
#include <Math.h>
class Func
{public:
  float Value;
  FuncValue(float a,float b,float s,float x);};
Func::FuncValue(float a,float b,float s,float x)
{a=cos(x);b=sqrt(b*x);s=sin(-s);

```



```

        Value=(a*x*x+b*x +s);}
int main()
{Func Fun;
float x,a,b,s,z;
x=4.2;a=6.2; b=4.2; s=2.5;z=0.0;
Fun.FuncValue(a,b,s,x);
cout<<Fun.Value<<endl;
cin >>z;}

```



Текширилган варианты

$Y=a*x*x+b*x+s$ ($a=\cos z$, $b=\sqrt{b*x}$; $s=\sin(-s)$;) функциясига синф яратамиз ва натижалар оламиз:

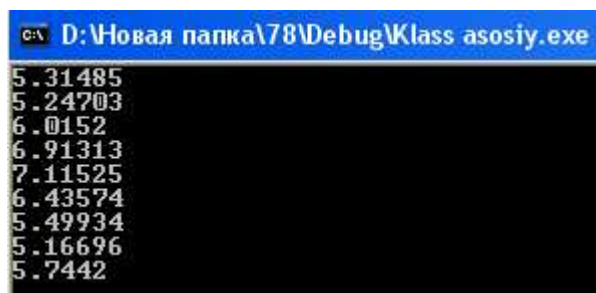
функциясига синф яратамиз ва натижалар оламиз:

1. Синф номини <Func> деб атаймиз.
2. Бу синфнинг экземплярини қуйидагича яратамиз: Func Func ; .
3. Синфга мурожат қилишни қуйидагича ёзамиз: Fun.FuncValue (a,b,s,x).

```

#include <iostream.h>
#include <Math.h>
class Func
{public:
float Value;
FuncValue(int x,float a,float b,float s);};
Func::FuncValue(int x,float a,float b,float s)
{a=cos(x);b=sqrt(b*x);s=sin(-s);
Value=a+b+s;}
int main()
{Func Fun;
float a,b,s; int x;
a=6.2;
b=5.2;
s=4.5;
for (x=1;x<10;x++)
{Fun.FuncValue(a,b,s,x);
cout<<Fun.Value<<endl;}
cin >>x;}

```



```

D:\Новая папка\78\Debug\Klass asosiy.exe
5.31485
5.24703
6.0152
6.91313
7.11525
6.43574
5.49934
5.16696
5.7442

```

Фойдаланилган адабиётлар рўйхати

1. Биллиг В.А.
Основы программирования на C#
Интернет-университет информационных технологий - INTUIT.ru, 2006
2. Фридман А.Л.
Язык программирования Си++
Интернет-университет информационных технологий - INTUIT. ru, 2004
3. Ёрбеков Я. Объектга йўналтирилган дастурлаш тиллари, маърузалар матни, ТАТУ, 2007 йил.