

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA  
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI  
URGANCH FILIALI**

**KOMPYUTER INJINIRINGI FAKULTETI**

**C++ DA DASTURLASH FANIDAN**

# **Kurs ishi**

**MAVZU: Berilgan jismning ko`rsatilgan suyuqlikdagi holatini  
aniqlovchi dasturiy ta`minot yaratish**

**Bajardi:**

912-15 guruh talabasi  
Nazarova Nilufar

**Ilmiy rahbar:**

Dasturiy injiniring kafedراسي  
assistenti  
Aliyev Oybek

## MUNDARIJA

<b>KIRISH</b> .....	2
<b>I. NAZARIY QISM</b> .....	5
2.1. Masalani qo`yilishi.....	5
<b>II. ASOSIY QISM</b> .....	6
3.1. Microsoft Visual Studioda ilova yaratish .....	6
3.2. Dasturda foydalanilgan komponentalar.....	16
3.3. Loyihani Microsoft Visual Studioda yaratish.....	18
3.4. Dasturdan foydalalnish uchun yo`riqnoma.....	21
<b>XULOSA</b> .....	23
<b>FOYDALANILGAN ADABIYOTLAR</b> .....	24
<b>ILOVA</b> .....	25

## **KIRISH**

Axborot texnologiyalarining takomillashtirilishi jamiyatni axborotlashtirishda muhim omil hisoblanadi. Ma'lumki, axborot texnologiyalari informatika qonun – qoidalari asosida takomillashtiriladi.

Texnologiya so'zi grekchadan tarjima qilinganda san'at, ustalik, malaka ma'nosini anglatadi. Texnikada texnologiya deganda ma'lum kerakli material mahsulotni hosil qilish uchun usullar, metodlar va vositalar yig'indisidan foydalanadigan jarayon tushuniladi. Texnologiya ob'ektining dastlabki, boshlang'ich holatini o'zgartirib, yangi, oldindan belgilangan talabga javob beradigan holatga keltiradi. Misol uchun sutdan turli texnologiyalar orqali qatiq, tvorog, smetana, yog' va boshqa sut mahsulotlarini olish mumkin. Agar boshlang'ich xom ashyo sifatida axborot olinsa, ushbu axborotga ishlov berish natijasida axborot mahsulotinigina olish mumkin. Ushbu holda ham "texnologiya" tushunchasining ma'nosi saqlanib qolinadi. Faqat unga "axborot" so'zini qo'shish mumkin. Bu narsa axborotni qayta ishlash natijasida moddiy mahsulotni emas, balki axborotnigina olish mumkinligini aniqlab turadi.

Texnologiyani quyidagicha ta'riflash mumkin. Texnologiya - bu sun'iy ob'ektlarni yaratishga yo'naltirilgan jarayonlarni boshqarishdir. Kerakli jarayonlarni kerakli yo'nalishda borishini ta'minlash uchun yaratilgan shart-sharoitlar qanchalik yaxshi tashkil etilganligi texnologiyaning samaradorligini bildiradi. Bu erda tabiiy jarayonlar nafaqat moddaning tarkibi, tuzilishi va shaklini o'zgartirish maqsadida, balki axborotni qayta ishlash va yangi axborot hosil qilish maqsadida ham boshqariladi. Shuning uchun axborot texnologiyasini quyidagicha ta'riflash mumkin.

Axborot texnologiyasi - bu axborot ma'lumotni bir ko'rinishdan ikkinchi, sifat jihatidan yangi ko'rinishga keltirish, axborotni yig'ish, qayta

ishlash va uzatishning usul va vositalari majmuasidan foydalanish jarayonidir.

Moddiy ishlab chiqarish texnologiyasining maqsadi insonning talabini qondiradigan yangi mahsulot ishlab chiqarishdan iborat. Axborot texnologiyasining maqsadi esa insonning biror-bir ishni bajarishi uchun zarur bo'lgan, uni tahlil etish va u asosida qaror qabul qilishi kerak bo'lgan yangi axborotni ishlab chiqarishdan iborat. Turli texnologiyalarni qo'llab, bitta moddiy resurslardan turli mahsulotlar olish mumkin. Xuddi shu narsani axborot texnologiyalariga nisbatan ham aytish mumkin.

Misol: matematikadan nazorat ishini bajarganda har bir o'quvchi boshlang'ich axborotni qayta ishlash uchun o'zining bilimini qo'llaydi. Masalaning echimi bo'lgan yangi axborot mahsuloti, o'quvchi tanlay olgan masalani echish texnologiyasi, usuliga bog'liq.

Moddiy ishlab chiqarishda turli maxsus jihozlar, stanoklar, uskunalar va boshqalar ishlatiladi. Axborot texnologiyalari uchun ham o'zining "uskunalari", vositalari mavjud. Bular kseroks, telefaks, faks, skaner va boshka vositalardir. Bu vositalar orkali axborotga ishlov berilib, o'zgartiriladi. Hozirgi paytda axborotga ishlov berish uchun kompyuterlar va kompyuter tarmoqlari keng qo'llanilmoqda. Axborot texnologiyasida kompyuterlar va kompyuter tarmoqlarining qo'llanishiga urg'u berish maqsadida ko'pincha kompyuter va kommunikatsion texnologiya haqida gapirishadi.

Axborot texnologiyasi o'zi uchun asosiy muhit bo'lgan axborot tizimlari bilan bevosita bog'liqdir. Chunki axborot texnologiyasi axborot tizimlarida mavjud bo'lgan ma'lumotlar ustida bajariladigan turli xil murakkablikdagi operatsiyalar, amallar va algoritmlarni bajarishdan iborat bo'lgan tartiblashtirilgan jarayondir.

Axborot texnologiyalari faqat fan va texnika hodisasi bo'lmasdan, iqtisodiy rivojlanishning muhim omiliga aylanmoqda. Axborot bilan qamrab

olinmagan biror muhim xo'jalik sektorini (ishlab chiqarish, transport, kredit-moliya sohasi, savdo) misol keltirish qiyin. Ayni paytda kompyuterlar va aloqa vositalari asosida axborotni to'plash, saqlash va taqdim etishning zamonaviy usullari, yangi axborot texnologiyalari va xizmatlarni sotish (tarqatish) maqsadlarida ishlab chiqarish mustaqil tarmoq sifatida shakllandi va ajralib chiqdi. Shunday qilib, xalq xo'jaligini axborotlashtirish kelgusiga yorib o'tish demakdir.

Axborot texnologiyalari, shu jumladan, kompyuterlar keng qo'llanilib kelayotgan ayrim sohalarni ko'rib chiqamiz.

#### Biznes sohasi

Axborot texnologiyalari yangi korxonalar yoki firmaning muvaffaqiyat bilan faoliyat ko'rsatishida katta rol o'ynashi mumkin.

Masalan:

biznes reja tuzishda va kutilayotgan daromad va chiqimlarni hisoblashda;

## II. NAZARIY QISM

### 2.1 Masalaning qo`yilishi

Ushbu kurs ishi Berilgan jismning ko`rsatilgan suyuqlikdagi holatini aniqlovchi dasturiy ta`minot yaratishga qaratilgan bo`lib, bu dasturiy ta`minotning vazifasi fizika kursida suyuqliklarda jismning holatlarini aniqlashga qaratilgan.

Ushbu dasturiy vositani ishlab chiqishda o`ziga hos talablar mavjud. Bu talablar quyida sanab o`tiladi.

Dasturiy vositani ishlab chiqish uchun masalaning qo`yilishi haqida yoritib beramiz.

Dastur kompleksi quyidagi vazifalarni bajarishi talab etiladi.

- Suyuqlik turlarini tanlash uchun uchta suyuqlik turi va qo`shimcha boshqa bir suyuqlikni qo`lda kiritish.

- Parametr kiritish uchun bo`sh maydonlar.
- Arximed Kuchini aniqlash.
- Jismning og`irlik kuchini aniqlash.
- Kiritilgan jismning suyuqlikdagi holatini aniqlash.

### III. ASOSIY QISM

#### 3.1. Microsoft Visual Studioda ilova yaratish

Microsoft Visual C++ tilida ilovalar yozishning boshlang'ich ko'nikmalariga ega bo'lish.

VC++ da dasturlar ilovalar deb nomlanadi. Ilovalar maxsus konstruksiya ko'rinishidagi muhitda – foydalanuvchi uchun bir qancha fayllar majmui ko'rinishidai *loyihada* yaratiladi.

C tilidagi dastur bu – belgilangan talablarga javob beruvchi, maxsus dasturlash tuzilishiga ega funksiyalar majmuidir. Ilova – bu ichida ilova algoritmini amalga oshiruvchi operatorlar joylashtirilgan asosiy funksiyadir. Operatorlar ichida shundaylari ham borki, ular algoritmni amalga oshishida talab qilinadigan boshqa funksiyalarni chaqirishga xizmat qiladi. Ixtiyoriy dasturni ishga tushirish, dasturning qolgan barcha qismini o'zida jamlagan *asosiy funksiyani* ishga tushirishdan boshlanadi. Funksiyaning bir qismi dasturchi tomonidan, qolgan qismi – kutubxona funksiyalari – foydalanuvchiga dasturiy muhit tomonidan taqdim qilinadi va dasturni ishlab chiqish jarayonida foydalaniladi.

Yaratiluvchi ilovalarni 2 ta katta kategoriyaga ajratish mumkin. Ular *ishchi stol ilovalari* va *Windows 8 ilovalari*dir. Ishchi stol ilovalari bu biz bilgan a sevadigan ilovalardir. Ular menyu paneli, instrumentlar paneli va ko'pincha ilova oynasining pastki qismida joylashgan holatlar satriga ega bo'ladilar. Biz ushbu fanda ishchi stol ilovalarini yaratishni maqsad qilib qo'ydik.

Windows 8 ilovalari ishchi stol ilovalaridan ancha farqli. Ularda foydalanuvchi interfeysi mavjud bo'lib u ishchi stol ilovalaridan butunlay farq qiladi. Foydalanuvchining to'g'ridan to'g'ri ma'lumotlar bilan bog'lanishi diqqat markazidadir, misol uchun menyu belgilari va instrumentlar tugmalari

orasidagi boshqaruvga bog'lanisi.

C/C++ ni o'rganishda biz maxsus ilovalar ko'rinishidan foydalanamiz – shablonlarni tayyorlash muhitida oldindan tayyorlangan konsol ilovalar asosida shakllantiriladi.

Konsol ilovalar – bu grafik interfeysiz ilovalar bo'lib, ular foydalanuvchi bilan maxsus buyruqlar satri orqali o'zaro aloqada bo'ladi yoki asosiy menyu muhitidagi maxsus buyruqlar bilan ishga tushiriladi. Bunday ilovalar **File/New Project** buyrug'i bajarilishidan so'ng ochiluvchi dialog oynasiga ega bo'lgan maxsus shablonlar yordamida yaratiladi.

Konsol ilovalar shabloni yaratiluvchi ilovaga barcha zarur elementlarni qo'shadi, shundan so'ng dasturchi bu shablonga o'zining C/C++ tilidagi operatorlarini qo'shadi. Keyin ilova avtonom ishga tushuvchi fayl sifatida kompilyatsiyalanadi va bajarish uchun ishga tushirilishi mumkin. Foydalanuvchi bilan muloqot ilova ishga tushirilgandan so'ng ochiluvchi maxsus konsol oynasi orqali amalga oshiriladi (shu oynada dastur habarlari tasvirlanadi, u orqali hisoblash uchun ma'lumotlar kiritiladi va uning o'ziga hisoblangan natijalar chop qilinadi).

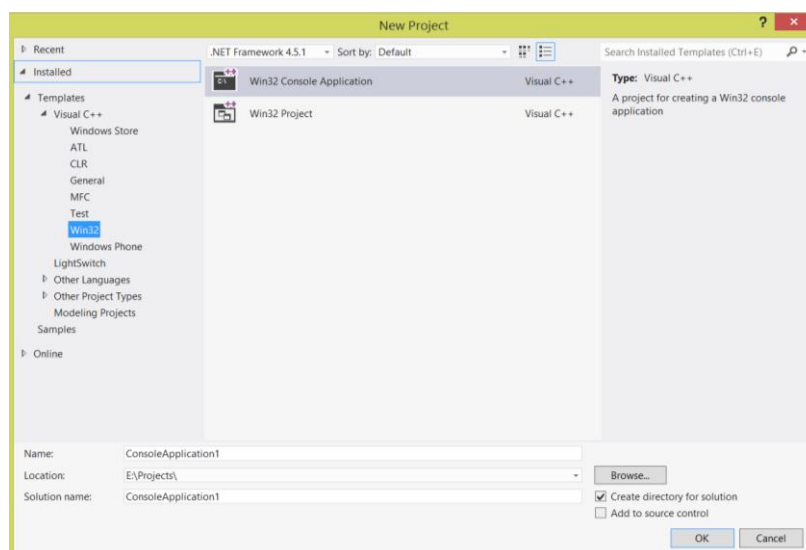
Kompilyatsiya va proyektni bitta joyga jamlash asosiy menyuning **Build** buyrug'I orqali amalga oshiriladi. Kompilyatsiya va proyektni bir joyga yig'ishdan so'ng uni bajarish uchun ishga tushirish mumkin. Bajarilish uchun ishga tushurish asosiy menyuning **Debbug** buyrug'I yordamida amalga oshiriladi.

Konsol ilovalar yaratish uchun quyidagi qadamlarni bajarish zarur:

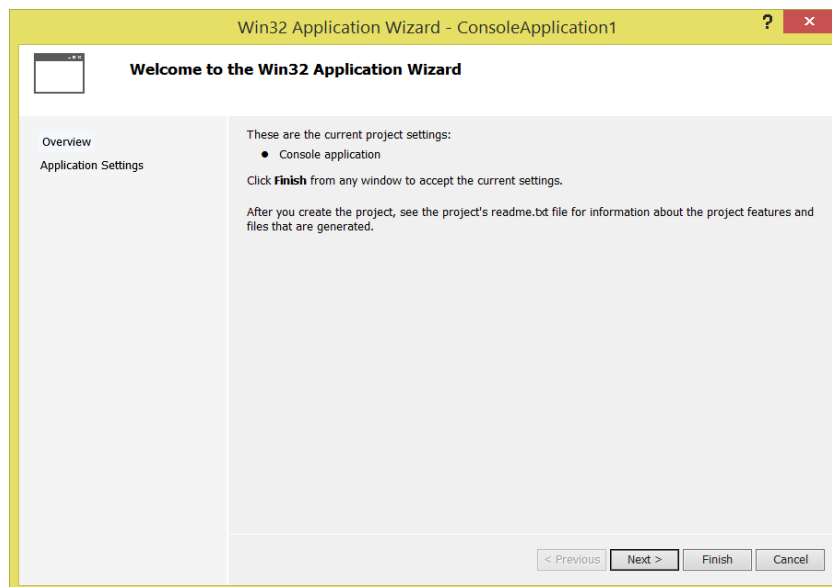
1. VC++ ni ishga tushirish.
2. Asosiy menyuning **File/New/Project** buyrug'ini bajarish. 3.1 – rasmdagi dialog oynasi hosil bo'ladi. Unda quyidagilarni ketma – ket bajaring: **Win32** bandini tanlang; **Win32 Console Application** qismini



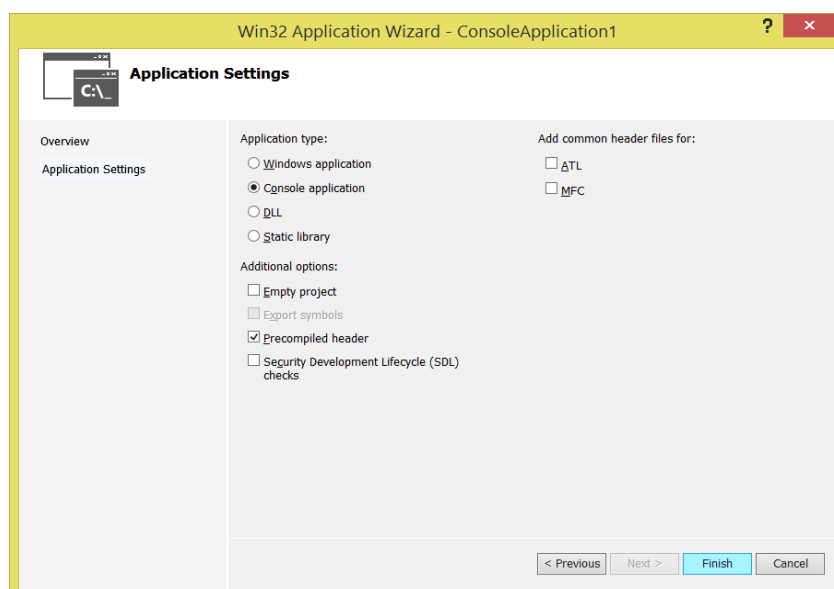
tanlang; Ilovaning saqlanish joyini ko'rsatish uchun "**Browse...**" tugmasidan foydalaning; **Name** maydoniga hosil qilinuvchi ilova nomini kiriting; **OK** tugmasini bosing.



3.1 – rasm. New Project dialog oynasi.



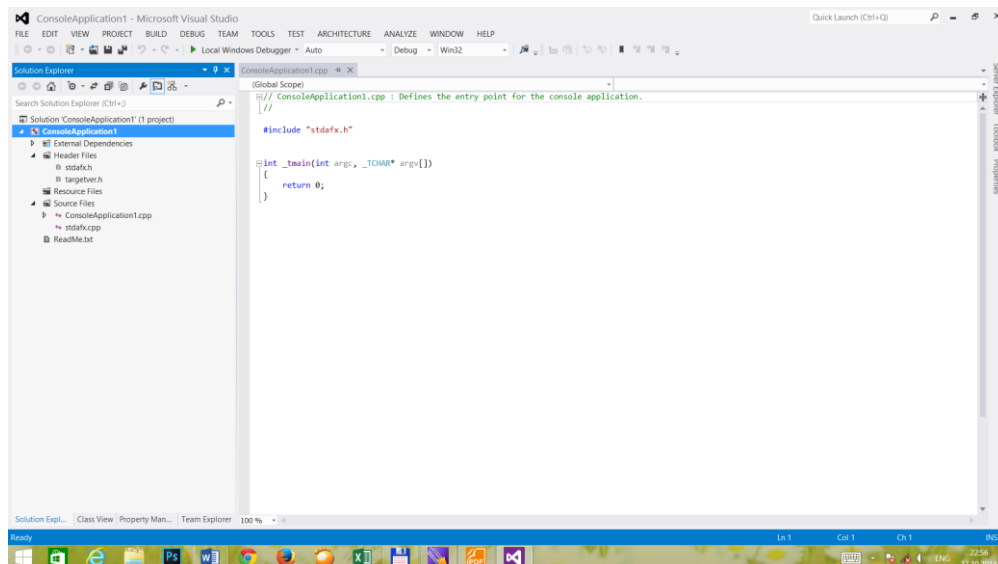
3.2 – rasm. Yaratiluvchi loyiha turini belgilash dialog oynasi.



3.3 – rasm. Yaratiluvchi loyihaning turini aniqlashtirish dialog oynasi.

3.2 – rasmda yaratiluvchi loyihaning turini tanlash taklif etiladi (aniqrogʻi, siz konsol loyiha yaratayotganingizni tasdiqlash). Buning uchun Finish tugmasini bosish yetarli. Gap shundaki, muhit koʻplab boshqa turdagi loyihalarni yaratishga va unga qoʻshimcha rejimlarni berishga imkon beradi, shuning uchun agar siz Finish tugmasi oʻrniga Next tugmasini bossangiz, boshqa dialog oynasi hosil boʻladi (3.3 – rasm).

3. 3.3 – rasmda tasvirlangan oynada tanlash radiotugmalarni belgilash bilan amalga oshiriladi (bu tugmalar atrofidagilar bilan oʻzaro aloqador, agar tugmalardan birortasi belgilangan boʻlsa, qolganlari oʻchirilgan holda boʻladi, bu bir qiymatli tanlash imkonini beradi). Tasvirda konsol ilovasi tugmasi yoqilgan. Agar endi Finish tugmasini bossak, u holda quyidagi natijaga ega boʻlamiz (huddi oldingi qadamda finishni bosganimizday). Natijada oldindan tayyorlangan konsol ilovasiga ega boʻlamiz (3.4 – rasm).



### 3.4 – rasm. Oldindan tayyorlangan konsol ilovasi.

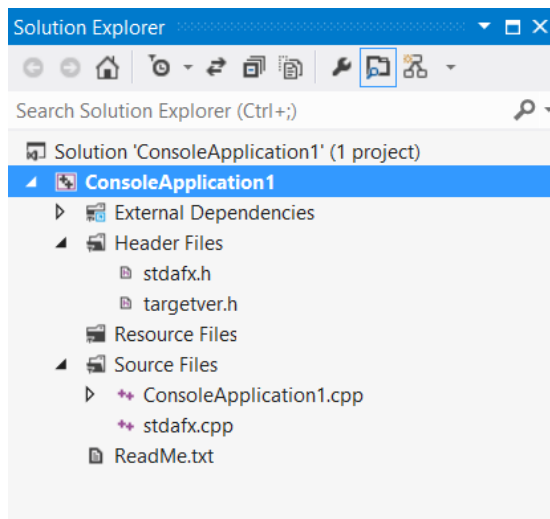
4. Oldindan tayyorlangan konsol ilovasi asosiy funksiya boshi `int _tmain(int argc, _TCHAR* argv[])` va figurali qavslar bilan ajratilgan tanasidan iborat. Funksiyaning boshidagi `_t main` ni `_tmain()` ga o'zgatiramyiz va dastur tanasidan `return 0` ni olib tashlaymiz.

Bularning barchasini Kod redaktori oynasida amalga oshiramiz. Haqiqatdan ham Redaktor oynasida joylashganligingizni tekshirish uchun sichqonchani Redaktor oynasining ixtiyoriy biror joyida bir marta bossangiz, kursor siz ko'rsatgan joyga o'rnatiladi (Redaktor shu nuqtada sizning keying harakatlaringizni kutadi). Endilikda siz xohlagan matningizni matn redaktoridagi kabi kiritishingiz, **Delete** yoki **Backspace**, strelka tugmalarini, kiritish va boshqa tahrirlash tugmalaridan joydalanishingiz mumkin.

Funksiya boshlanishi `_tmain` ko'rinishidan `_tmain()` ko'rinishga o'tkazganimiz bizning asosiy funksiyamiz konsol ilovalarini bog'lash uchun xizmat qiluvchi argumentlarga ega emas ekanligini anglatadi.

VC++ yaratiluvchi ilovani ikkita biri ikkinchisiga bog'langan konteyner ko'rinishida rasmiylashtiriladi. Bittasi (asosiy konteyner) **Yechim (Solution)** deb, ikkinchisi **Loyiha (Project)** deb nomlanadi. Loyiha fayllar guruhini

birlashtiruvchi konfiguratsiya (konteyner, karkas) shaklida belgilangan.



3.5 – rasm. Ilova loyihasining ko’rinishi.

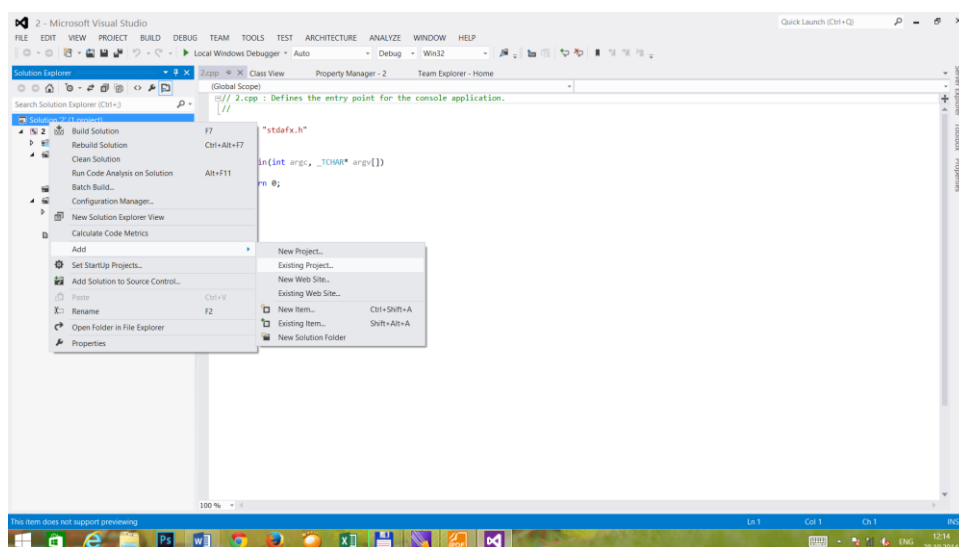
Loyiha doirasida kompilyatsiyalangan va qurilgan, bajarilishi mumkin bo’lgan dastur yaratiladi. Har qanday loyiha kamida ikkita podkonfiguratsiyaga ega: **otladochniy** va oddiy (bajariluvchi). Loyihalar boshqa karkas, boshqa konteynerning qismi hisoblanadi. U **Solution (Yechim)** deb nomlanadi va loyihalar orasidagi o’zaro aloqani tasvirlaydi: bitta **Yechim** ko’plab loyihalarni o’z ichiga olishi mumkin, bitta loyiha esa ilovaning joriy holatini ta’minlovchi ko’plab elementlarga ega bo’lishi mumkin. Aytish mumkinki, **Yechim** – birlashgan loyihalar guruhidan iborat o’zgachabir narsadir. Uni oddiygina qilib: Yechim termini bizni chalg’itmasligi uchun Guruh loyihasi deb ataymiz. Guruh loyihalari bilan ishlovchi maxsus instrument mavjud bo’lib, biz uni **Solution Explorer** deb ataymiz. Uni topish uchun **View** menyusiga murojaat qilish zarur. Dastur muhitining o’zi avtomatik tarzda yaratilayotgan ilovani guruh loyihasi shaklida rasmiylashtiradi.

Ilovalarni rasmiylashtirishga bunday yondashish loyihalar guruhi bilan bir butun holda ishlashga imkon beradi, bu ilova qayta ishlanishi jarayonini tezlashtiradi. Ta’klidlash joizki, loyihalarning barcha fayllari ham **Solution**

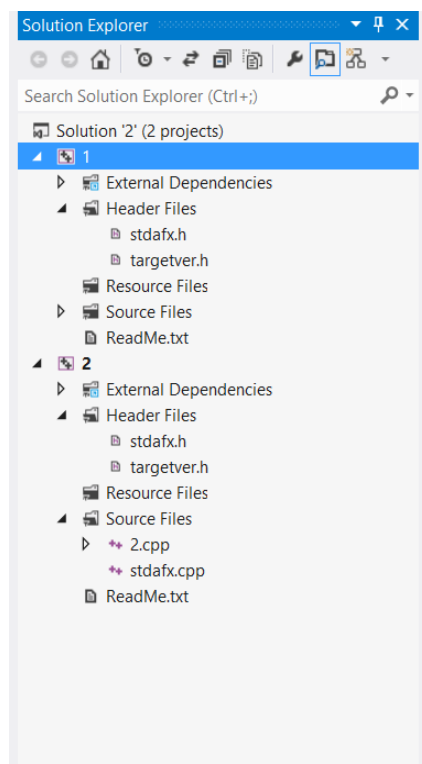
**Explorer** da tasvirlanavermaydi.

Misol uchun **1** nomli loyiha yaratamiz va uni **2** guruhiga qo'shamiz. Buning uchun sichqonchanning o'ng tugmasini **Solution '2'** qatori ustida bir marta bosamiz va paydo bo'lgan konteks menyudan **Add/Existing Project...** tugmasini tanlaymiz (3.6 – rasm).

Bunda loyihani qidirish uchun dialog oynasi hosil bo'ladi, keyin oddiy usul bilan **1** loyhasini ochamiz, buning natijasida u **Solution '2'** ga qo'shiladi (3.7 – rasm).



3.6 – rasm. Oldindan yaratilgan loyihani loyihalar guruhi (Solution '2') ga qo'shish jarayoni.



3.7 – rasm. 1 loyihasi qoʻshilgan Solution ‘2’.

Qoʻshilgan loyihalarni boshqalari kabi oʻchirish mumkin, buning uchun sichqonchani shu loyiha ustida bosib konteks menyuni hosil qilish va undan **Remove** buyrugʻini tanlash lozim.

Endi loyihja ichida yaratilgan fayllarni koʻrib chiqamiz:

- **1.cpp** – bu asosiy kiruvchi fayl va yaratiluvchi loyihaga kirish nuqtasi (1 – bu yerda loyiha nomi).

- **stdafx.cpp** – ilovani kompilyatsiyalash uchun maxsus Vs9VimTestCpp.pch faylini ulaydi.

- **stdafx.h** – ilova kompilyatsiyasi uchun maxsus fayllarni ulaydi.

Uning tarkibi:

```
#pragma once //kompilyatsiya uchun buyruqlar:
```

```
#include “targetver.h”
```

```
#include <stdio.h>
```

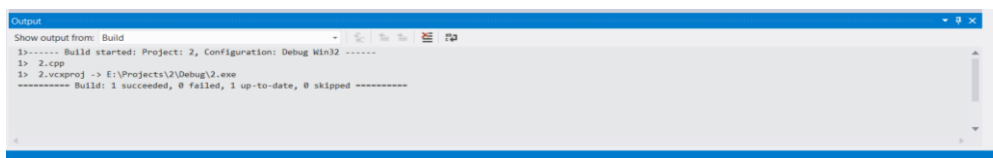
```
#include <tchar.h>
```

- **targetver.h**– Windows Vistaning maxsus xususiyatlaridan foydalanishga imkon beradi. Ushbu faylning tarkibi:

```
#pragma once //kompilyatsiya uchun buyruqlar:
#ifdef _WIN32_WINNT //Allow use of features
    // specific to Windows Vista ore later.
#define _WIN32_WINNT 0x0600 //Change this to the
    // appropriate value to target
    // other versions of Windows.
#endif
```

- **ReadMe.txt** – shablon orqali yaratilgan loyihaning ba’zi konsol ilova fayllarini izohlovchi fayl. Faylning tarkibini konteks menyu orqali **Open** buyrug’ini berish bilan bajarish mumkin.

Yozilgan ilovamiz ishga tushishi uchun uni avval kompilyatsiyalash lozim. Buning uchun <F7> tugmasini bosish yoki asosiy menyudan Build/Build Solution bandini tanlash lozim.



Agar kompilyatsiyada xatolik yuzaga kelsa, kompilyatsiya oynasidagi xatolik ustida sichqoncha tugmasini 2 marta bosilsa, dastur kodida qator belgilanishi orqali qaysi qatorda xatolik bo’lganligini ko’rib olish mumkin.

Muvaffaqiyatli kompilyatsiyalangan dasturni endilikda qurish (bitta joyaga yig’ish) lozim, undan keyin esa ishga tushirishimiz mumkin. Ishga tushirish uchun Debug/ Start Without Debugging (otladchiksiz ishga tushirish) buyrug’ini berishingiz yoki <Ctrl>+<F5> tugmalar kombinatsiyasini bosishingiz mumkin. Yoki katalogga kirib exe modulini ishga tushirishingiz ham mumkin.

MS Visual C++ da tarmoqlanish va takrorlanish operatorlari qo'llanilishini o'rganish. Tarmoqlanish va takrorlanish jarayonlariga oid masalalarni yechish bo'yicha ko'nikmalarga ega bo'lish.

### **While operatori**

Temperaturani Farangeytdan Selsiyga o'tkazuvchi dastur kodi quyida keltirilgan, bunda temperature  $C=(5/9)*(F - 32)$  va C – Selsiy shkalasidagi, F esa Farangeyt shkalasidagi temperature.

```
#include "stdafx.h"
#include <conio.h> // _getch() funksiyasi uchun
int _tmain(int argc, _TCHAR* argv[])
{
    int lower, upper, step;
    float fahr, cels;
    lower=0;
    upper=300;
    step=20;
    fahr=lower;
    while (fahr <= upper)
    {
        cels=(5.0/9.0)*(fahr-32.0);
        printf("%4.0f %6.1f\n", fahr, cels);
        fahr = fahr + step;
    }
    _getch();
}
```

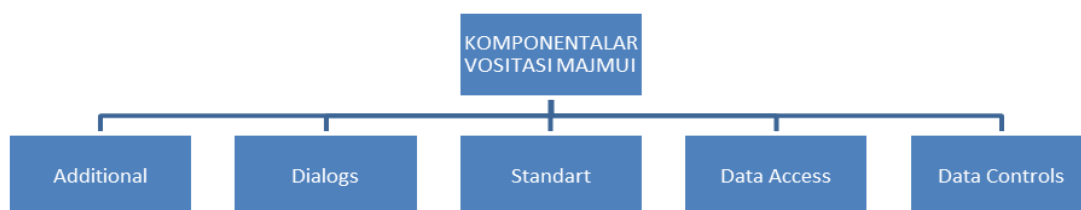
Kompilyator va tuzuvchi bir vaqtda <F7> tugmasi bilan ishga tushiriladi. Dastur kompilyatsiyalanadi va tuziladi. Uning bajarilishi uchun



<Ctrl>+<F5> tugmalar kombinatsiyasini bajaramiz. Oynada quyidagi natija paydo bo'ladi:

### 3.2. Dasturda foydalanilgan komponentalar

#### Visual C++ da forma komponentalari



#### Label komponentasi

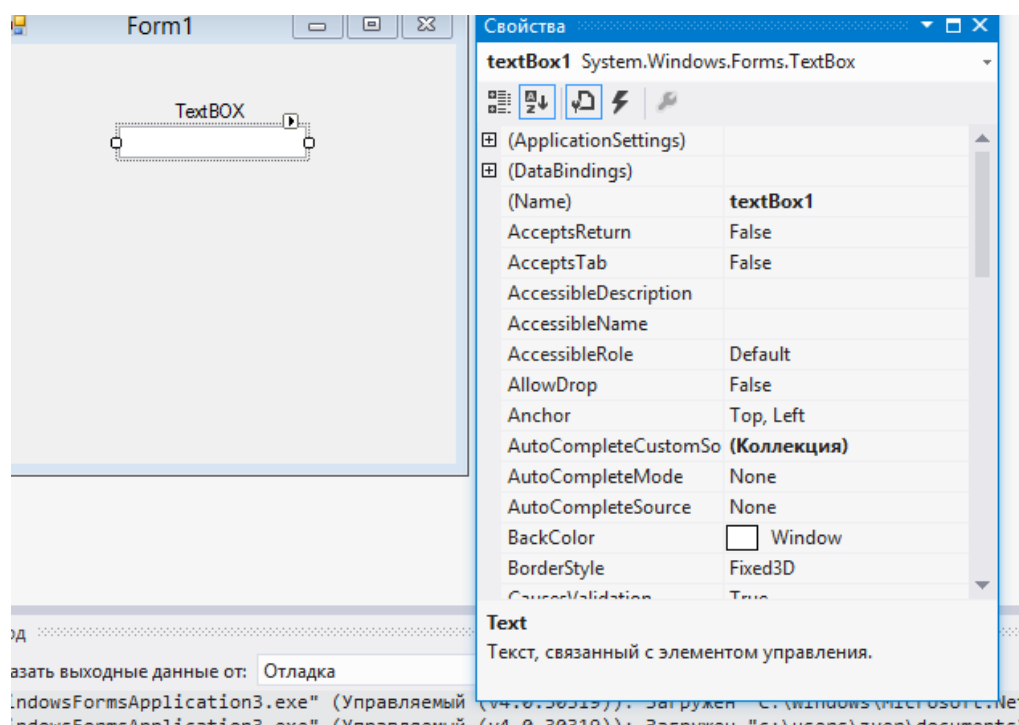
Label Yozuv (label komponenti) yorliqni ifodalaydi va ko`pincha o`zining Caption xususiyatiga ega bo`lmagan boshqa boshqaruv elementlarining sarlavasi sifatida ishlatiladi. Ko`pincha yozuvlarni tasvirlash uchun nishon deb nomlanadigan Label komponenti ishlatiladi. U dastur bajarilish vaqti foydalanuvchi tahrirlashi mumkin bo`lgan oddiy matnni tasvirlash uchun ishlatiladi.

Visual Studio dasturida Label komponentasini ishlatish uchun “Панель элементов” oynasida Label nomli komponenta tanlanadi.

#### TextBox komponentasi

Bir satrli redaktor matn tasvirlanishi va o`zgartirilishi mumkin bo`lgan ma`lumot kiritish maydonini ifodalaydi. Visual Studioda bir qancha bir satrli redaktorlar mavjud bo`lib, ulardan TextBox komponenti ko`p qo`llaniladi.

TextBox komponenti klaviaturadan turli simvollarni kiritish va tahrirlashga imkon beradi. Bunda boshqaruv tugmalaridan foydalanib satr bo`ylab siljish, <BackSpace> va <Delete> tugmalari yordamida simvollarni o`cherish, matn (bo`lagini ) qismini belgilash va boshqa amallarni bajarish mumkin. Ta`kidlash joizki, bir satrli redaktorlar <Enter> va <Esc> boshqaruv tugmalariga javob bermaydi.(3.8-rasm)



3.8-rasm

TextBox komponentidan parolni kiritishda foydalanganda Char turidagi PasswordChar xususiyatidan foydalanish mumkin. U kiritish oynasiga tasvirlanadigan simvolni belgilaydi. Bu simvol matn kiritayotganda haqiqatda kiritilgan ma`lumot o`rnida paydo bo`ladi.

### ComboBox Komponentasi



ComboBox komponentasiga kiritilgan ro`yxatlar bilan ishlashda foydalaniladi. Komponentaning o`ng tomonida ro`yxatni chiqarish tugmasi

bosilganda ro`yxat paydo bo`ladi.

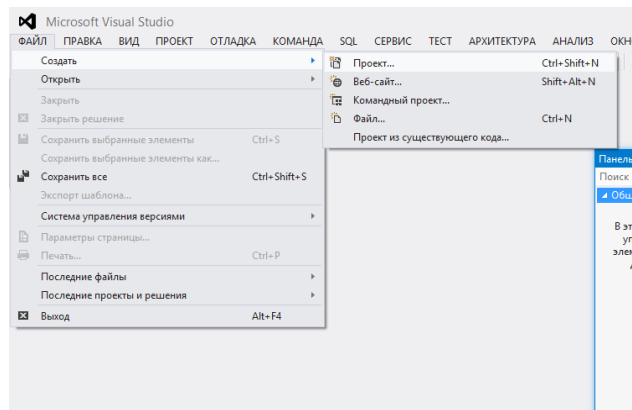
### 3.3. Loyihani Visual Studio 2012 dasturida C++ tilida yaratish

Dasturni yaratishni Visual Studioda ilova yaratish bilan boshlaymiz.  
Visual Studioda ilova quyidagicha yaratiladi.

#### *Forma oynasi*

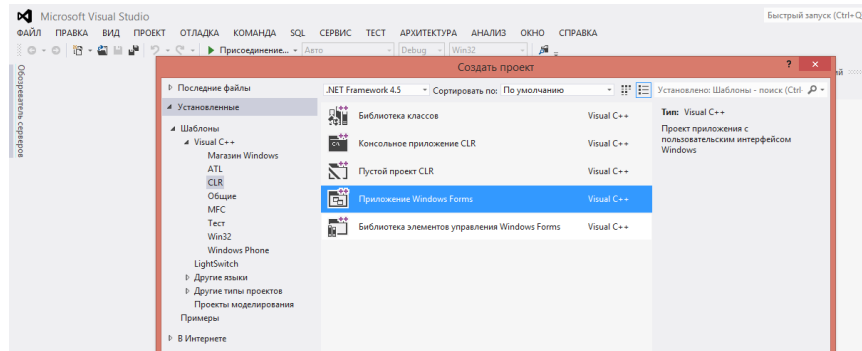
Visual Studio dasturida forma yaratish.

File ->Project(3.3-rasm)



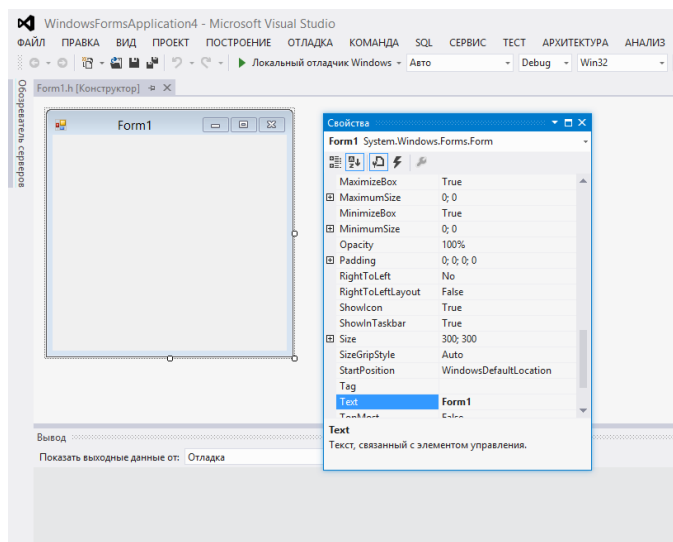
3.10-rasm

Visual C++ ->CLR->Windows Forms



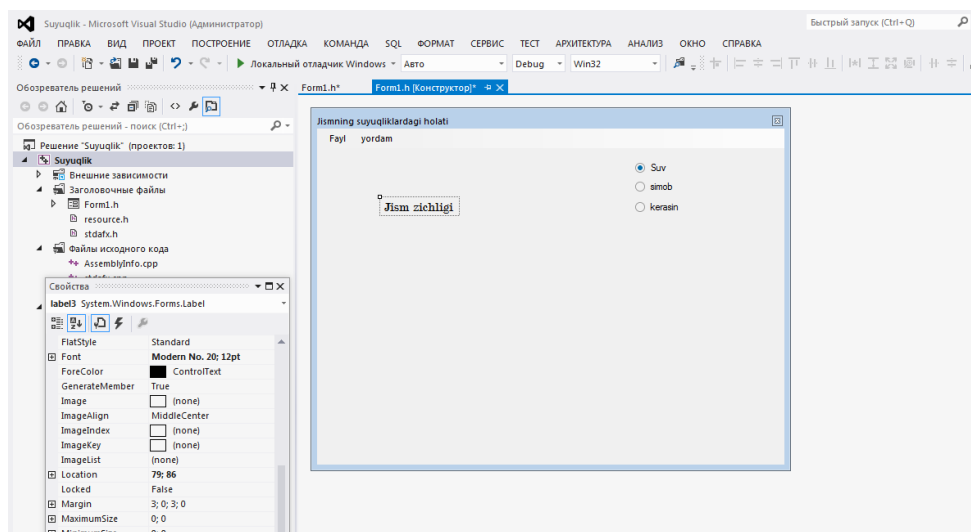
3.11-rasm

Yuqorida keltirilgan ketma – ketliklardan so`ng Form1 oynasi yaratiladi.

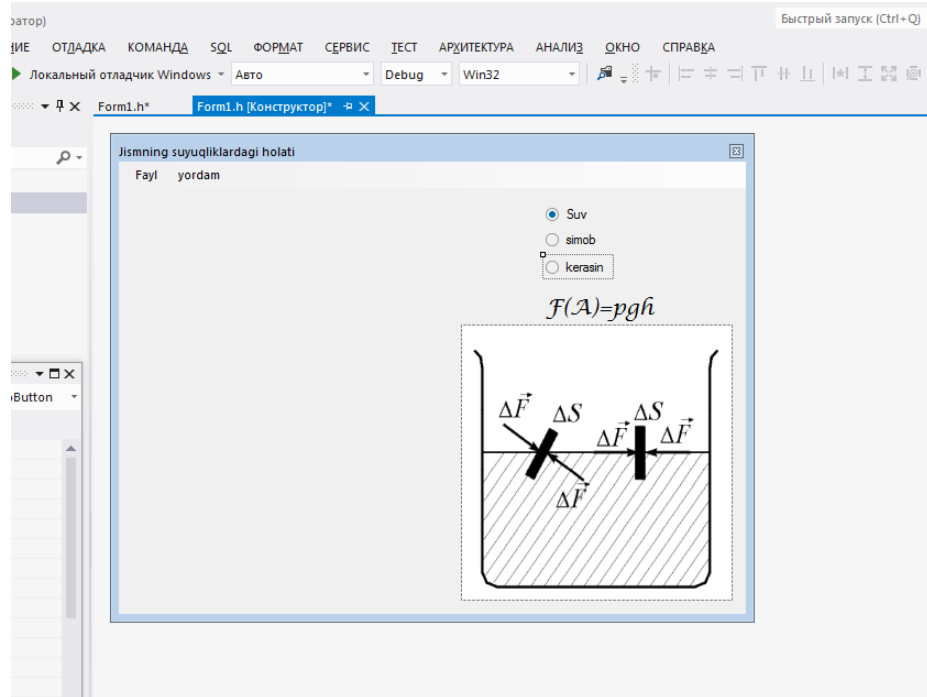


3.12-rasm. Form oynasi yaratildi

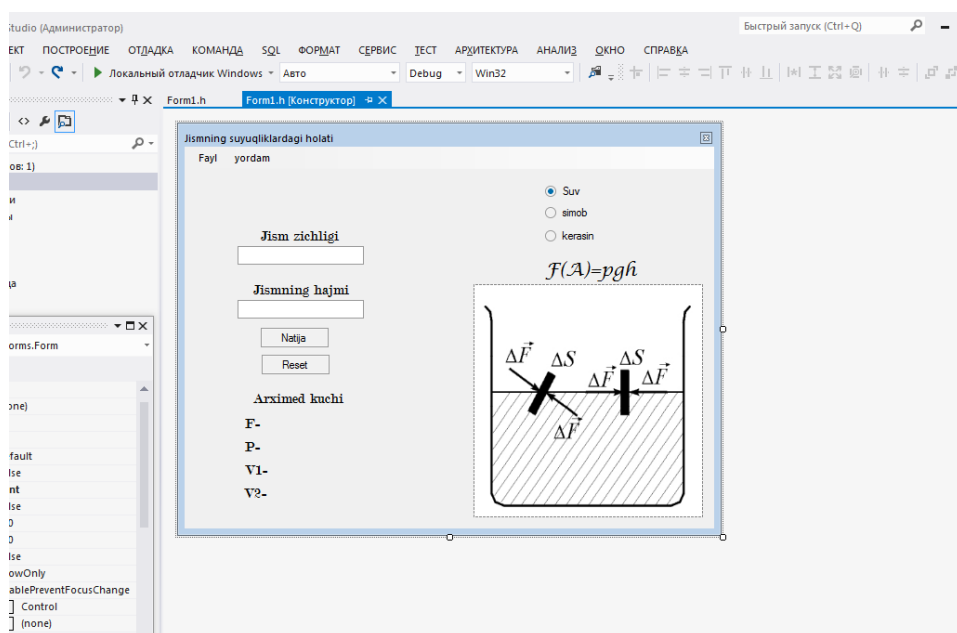
Dasturda zarur bo`lgan komponentalar tanlanadi va form oynasiga tashlanadi.(3.13-rasm)



3.13-rasm.



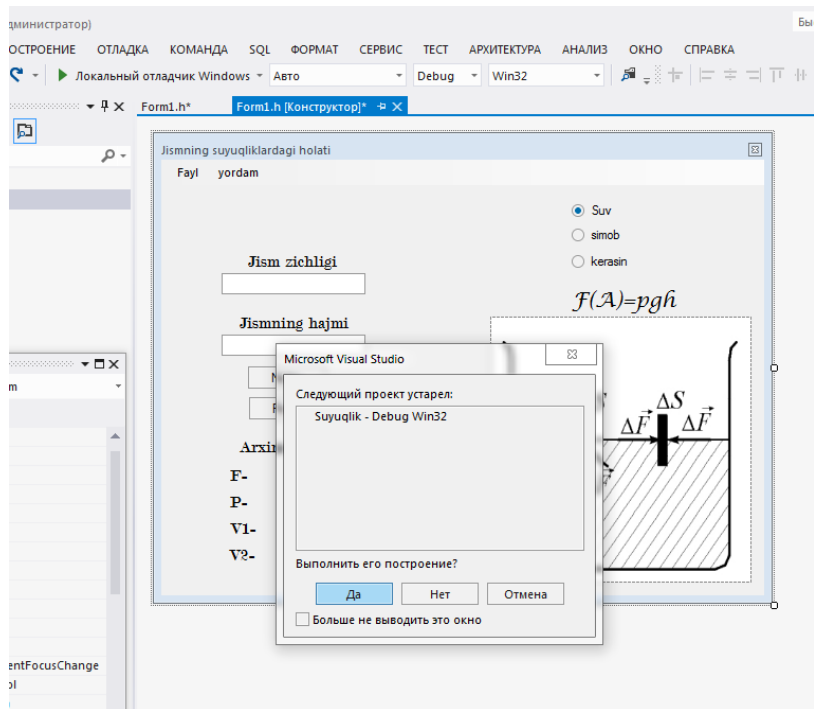
3.14-rasm. Suyuqliklar tanlash uchun radio button



3.15-rasm. Kerakli barcha komponentalar joylashtirildi

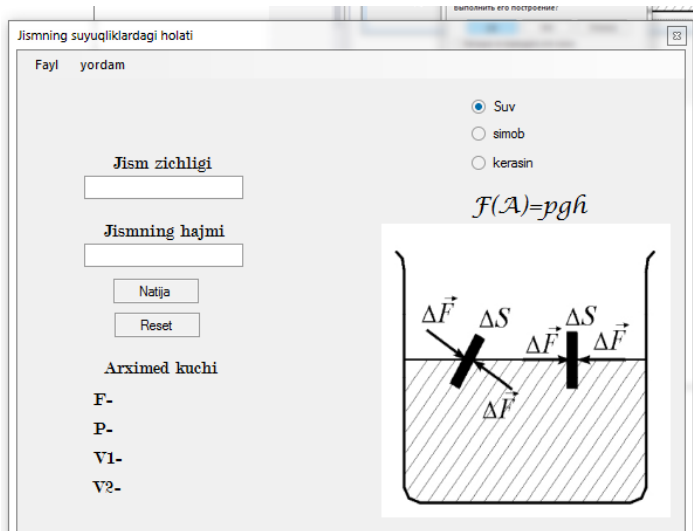
### 3.4. Dasturdan foydalanish uchun yo`riqnoma

#### Dasturni ishga tushirish



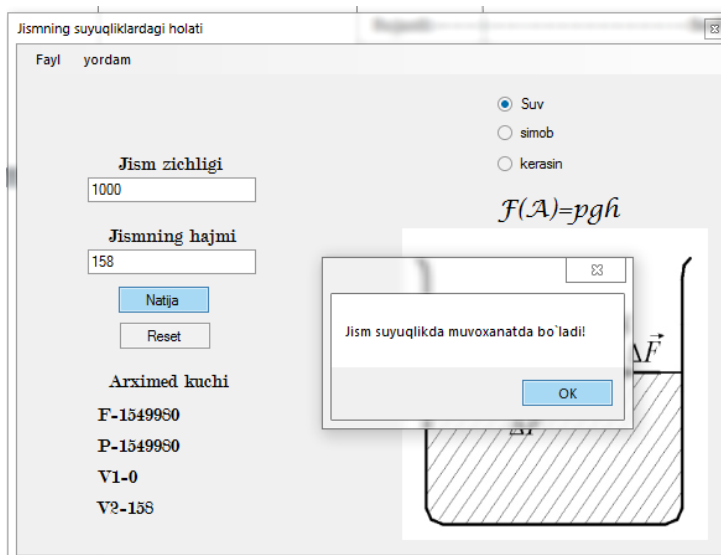
3.16-rasm

#### Dasturning asosiy oynasi



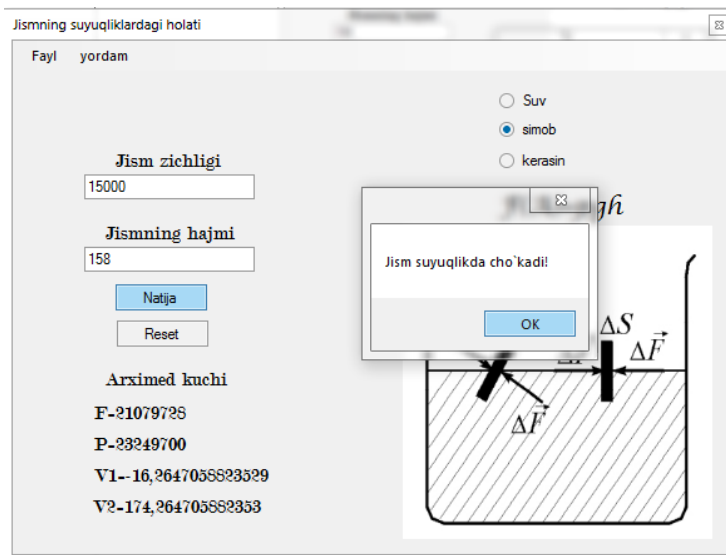
3.17-rasm

#### Qiymat berilgandagi natija



3.17-rasm

2-suyuqlikdagi natija



3.18-rasm



## **Xulosa**

Ushbu kurs ishini bajarish davomida quyidagi ishlar amalga oshirildi:

1. Suyuqlikni tanlash uchun uchta suyuqlik turi va qo'shimcha qo'lda kiritish uchun maydon yaratildi.

2. Arximed kuchi aniqlandi.

3. Og'irlik kuchi aniqlandi.

4. Kiritilgan jismning suyuqlikdagi xolati aniqlandi.

Natijada, kiritilgan jismning suyuqlikdagi holatini aniqlab beruvchi dastur tuzildi.

Kurs ishini bajarish mobaynida quyidagi bilim va ko'nikmalar o'zlashtirildi:

1.Suyuqlikni tanlash uchun uchta suyuqlik turi va qo'shimcha qo'lda kirish.

2.Parametrlarni kiritish uchun bo'sh maydonlar yaratish.

3.Arximed kuchini aniqlash.

4.Jismning suyuqlikdagi holatini aniqlash.

Dastur quyidagi qulayliklarga ega:

1.Dastur bazasida mavjud uchta suyuqlik va qo'lda kiritilgan suyuqlik zichligini aniqlab beradi.

2.Dastur Arximed Kuchini aniqlab beradi.

3.Dastur og'irlik kuchini aniqlab beradi.

4.Dastur jismning suyuqlikdagi holatini aniqlab beradi.

## Foydalanilgan adabiyotlar

### I.Asosiy adabiyotlar

1. Борис Пахомов - C,C++ и MSVisualC++ 2012 для начинающих Санкт-Петербург «БХВ-Петербург» 2013.

2. Зиборов В. - MS Visual C ++2010 в среде NET Питер, 2012

3. Khorton\_Visual\_C\_\_2010\_Polny\_kurs-2011g Санкт-Петербург 2011.

4.Abduqodirov A.A. Xisoblash matematikasi va programmalash. O'qituvchi,1989 y.

5.Horton I. - Beginning Visual C++ 2012 – 2012.

### II.Qo'shimcha adabiyotlar:

1.В.В. Язык C++ – М.: Финансы и статистика, 1996.

2.Ortiqov A., Mamatqulov A. IBM PC kompyuterlaridan foydalanish.- Т.: Komus, 1993 y.

# ILOVA

```
#pragma once
```

```
namespace Suyuqlik {
```

```
    using namespace System;
```

```
    using namespace System::ComponentModel;
```

```
    using namespace System::Collections;
```

```
    using namespace System::Windows::Forms;
```

```
    using namespace System::Data;
```

```
    using namespace System::Drawing;
```

```
    /// <summary>
```

```
    /// Сводка для Form1
```

```
    /// </summary>
```

```
    public ref class Form1 : public System::Windows::Forms::Form
```

```
    {
```

```
    public:
```

```
        Form1(void)
```

```
        {
```

```
            InitializeComponent();
```

```
            //
```

```
            //TODO: добавьте код конструктора
```

```
            //
```

```
        }
```

```
    protected:
```

```
        /// <summary>
```

```
        /// Освободить все используемые ресурсы.
```

```
        /// </summary>
```

```
        ~Form1()
```

```
        {
```

```
            if (components)
```

```
        {  
            delete components;  
        }  
    }  
private: System::Windows::Forms::MenuStrip^ menuStrip1;  
protected:  
private: System::Windows::Forms::ToolStripMenuItem^ faylToolStripMenuItem;  
private: System::Windows::Forms::ToolStripMenuItem^ exitToolStripMenuItem;  
private: System::Windows::Forms::ToolStripMenuItem^ yordamToolStripMenuItem;  
private: System::Windows::Forms::ToolStripMenuItem^  
dasturHaqidaToolStripMenuItem;
```

```
private: System::Windows::Forms::RadioButton^ radioButton1;  
private: System::Windows::Forms::RadioButton^ radioButton2;  
private: System::Windows::Forms::RadioButton^ radioButton3;  
private: System::Windows::Forms::PictureBox^ pictureBox1;  
private: System::Windows::Forms::Label^ label1;  
private: System::Windows::Forms::TextBox^ textBox1;  
private: System::Windows::Forms::TextBox^ textBox2;  
private: System::Windows::Forms::Button^ button1;  
private: System::Windows::Forms::Label^ label2;  
private: System::Windows::Forms::Label^ label3;  
private: System::Windows::Forms::Label^ label4;  
private: System::Windows::Forms::Label^ label5;  
private: System::Windows::Forms::Label^ label6;  
private: System::Windows::Forms::Label^ label7;  
private: System::Windows::Forms::Label^ label8;  
private: System::Windows::Forms::Button^ button2;
```

```
private:  
    /// <summary>  
    /// Требуется переменная конструктора.  
    /// </summary>  
    System::ComponentModel::Container ^components;
```

#pragma region Windows Form Designer generated code

/// <summary>

/// Обязательный метод для поддержки конструктора - не изменяйте

/// содержимое данного метода при помощи редактора кода.

/// </summary>

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

if( textBox1->Text!="" || textBox2->Text!=""){

double P,F,m,v,pj,ps,V1,V2;

pj=Convert::ToDouble(textBox1->Text);

v=Convert::ToDouble(textBox2->Text);

if(radioButton1->Checked==true){

ps=1000;

}

if(radioButton2->Checked==true){

ps=13600;

}

if(radioButton3->Checked==true){

ps=800;

}

P=v\*pj\*9.81;

F=ps\*v\*9.81;

label2->Text+=Convert::ToString(F);

label6->Text+=Convert::ToString(P);

V1=((ps-pj)\*v)/ps;

V2=((pj)\*v)/ps;

label7->Text+=Convert::ToString(V1);

label8->Text+=Convert::ToString(V2);

if(P>F){ MessageBox::Show("Jism suyuqlikda cho`kadi!");}

if(P<F){ MessageBox::Show("Jism suyuqlik yuzasiga qalqib

chiqadi!");}

```

        if(P==F){ MessageBox::Show("Jism suyuqlikda
muvozanatda bo`ladi!");}

        button1->Enabled=false;
    }
    else{
        MessageBox::Show("ma`lumotlar to`liq
kiritilmadi!");
    }

}

private: System::Void dasturHaqidaToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    MessageBox::Show("Kurs ishi mavzusi: Berilgan jismning
ko`rsatilgan suyuqlikdagi holati; \n Dasturchi: Nazarova Nilufar");
}

private: System::Void exitToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Close();
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    button1->Enabled=true;
    label2->Text="F=";
    label6->Text="P=";
    label7->Text="V1=";
    label8->Text="V2=";
}

};
}

```