

**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: “Issiqlik mashinasining foydali ish koeffitsiyentini topish”
dasturini yaratish.**

Bajardi:

912-15 guruh talabasi
Romanov Otanazar

Ilmiy rahbar:

Dasturiy injiniring kafedrasii
assistenti
Aliyev Oybek

MUNDARIJA

KIRISH	
I BOB. Nazariy qism	
1.1-§. Masalani qo`yilishi	4
II BOB. Asosiy qism	
2.1-§. Microsoft Visual C++ da ilova dastur yaratish	5
2.2-§. Dasturda foydalanilgan komponentalar.....	16
2.3-§. Loyihani Visual Studio 2012 dasturida C++ tilida yaratish.....	19
2.4-§. Dasturdan foydalalnish uchun yo`riqnoma.....	22
XULOSA	
FOYDALANILGAN ADABIYOTLAR	
ILOVA	

KIRISH

Keyingi yillarda amaliy dasturchilarga juda ko'p integratsion dastur muhitlari taklif etilyapti. Bu muhitlar u yoki bu imkoniyatlari bilan bir-biridan farqlanadi.

Aksariyat dasturlashtirish muhitlarining fundamental asosi c++ tiliga borib taqaladi. C++ tili B. Strastrup tomonidan yaratilgan.

C++ standarti ayni vaqtda ISO-International Standards Organizatsion (Standartlash bo'yicha xalqaro tashkilot) standarti deb ham nomlanadi.

Vaqt o'tishi bilan dasturchilar oldiga qo'yilgan masalalar o'zgarib boryapti. Bundan yigirma yil oldin dasturlar katta hajmdagi ma'lumotlarni qayta ishlash uchun tuzilar edi. Bunda dasturni yozuvchi ham foydalanuvchisi ham kompyuter sohasi bo'yicha talab etilardi. Hozirda esa ko'pgina o'zgarishlar ro'y berdi. Kompyuter odamlar tomonidan uni chuqur o'rganish vositasi emas, ko'proq o'zlarining oldilariga qo'yilgan, o'zlarining ishlariga tegishli bo'lgan muammolarni yechish vositasi bo'lib qoldi.

Axborot texnologiyalarining takomillashtirilishi jamiyatni axborotlashda 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.

Texnologiyani quyidagicha ta'riflash mumkin. Texnologiya – bu sun'iy obyektlarni 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 yerda 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.

Katta hajmdagi ma'lumotlar bazasi va axborotlar ustida ishlashga to'g'ri kelyapti. Jamiyat taraqqiyotida yuz berayotgan jadal o'zgarishlar uning bir qismi bo'lgan informatika soxasiga xam o'z ta'sirini ko'rsatmoqda. Bu ta'sir shunchalik kuchliki, axborot texnologiyalarida bo'layotgan o'zgarishlar yillar ichida emas, balki oylar ichida o'zgarib va boyib bormoqda. Axborot texnologiyalarida juda katta yutuq va o'zgarishlar amalga oshdi. Yangi axborot texnologiyalarining yoki kompyuter texnologiyalarining paydo bo'lishi bu soxadagi xizmat qilish uslubini tubdan o'zgartirdi. Tasavvur qilib ko'ring, yaqin-yaqinlargacha biror masalani kompyuterda yechish uchun algoritmlarni, dasturlash tillaridan birini, dasturni kompyuter xotirasiga kiritishni, uning xatolariyu natijani taxlil qilishni talab qilar edi. Bu ishni faqat mutaxassislargina amalga oshirar edilar. Ammo bu soxada mutaxassis bo'lmagan foydalanuvchilar sonining kundan-kunga ko'payishi jamiyat oldida ma'lum qiyinchiliklarni tug'dirdi.

I. NAZARIY QISM

1.1 Masalaning qo`yilishi

Ushbu kurs ishi “Issiqlik mashinasining foydali ish koeffitsiyentini topish” dasturini tuzishga mo’ljallangan bo’lib, dasturni tuzish jarayonida quyidagi ishlar amalga oshirilishi lozim:

1. Visual Studio muhitida yechish.

2. Bu dasturni Visual Studio Windows form application orqali C++ dasturida tuzish.

Dastur quyidagi ishlarni amalga oshirishi lozim:

1. Issiqlik miqdori berilganda issiqlik mashinasining foydali ish koeffitsiyentini hisoblaydi.

2. Temperatura berilganda issiqlik mashinasining foydali ish koeffitsiyentini hisoblaydi.

II. ASOSIY QISM

2.1. Microsoft Visual C++ da ilova dastur yaratish

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

Visual C++ 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 va 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'lanishi.

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.

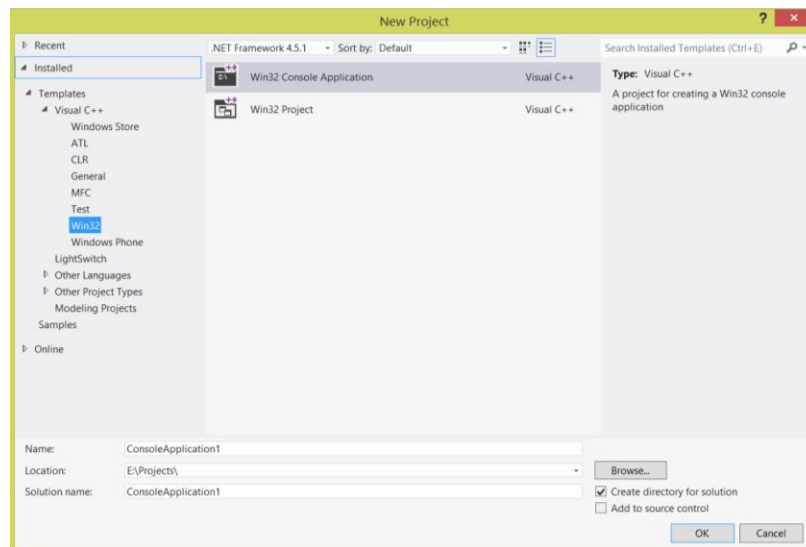
Konsol ilovalar shabloni yaratiluvchi ilovaga barcha zarur elementlarni qo'shadi, shundan so'ng dasturchi bu shablona o'zining C/C++ tilidagi operatorlarini qo'shadi. Keyin ilova avtonom ishga tushuvchi fayl sifatida kompilyatsiyalanadi va bajarish uchun ishga tushirilishi mumkin. Foydalanuvchi bilan 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 **Debug** buyrug'I yordamida amalga oshiriladi.

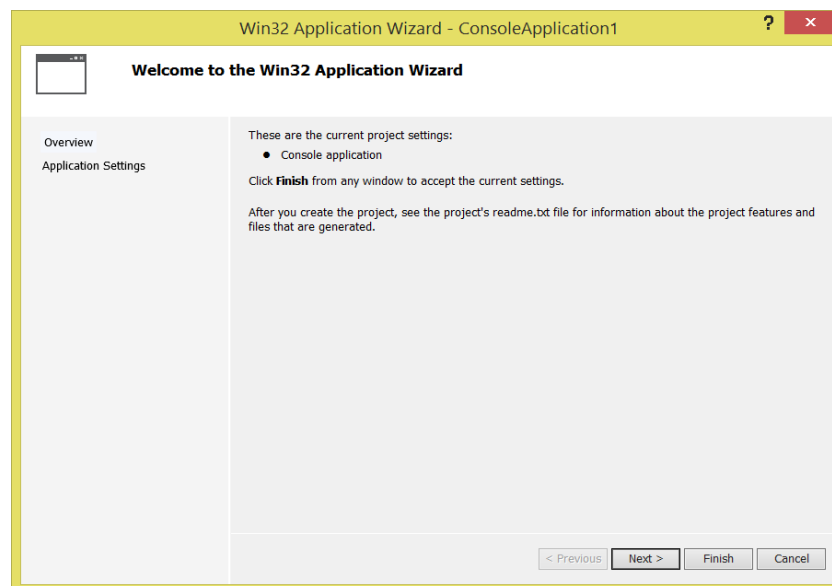
Konsol ilovalar yaratish uchun quyidagi qadamlarni bajarish zarur:

1. Visual C++ 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

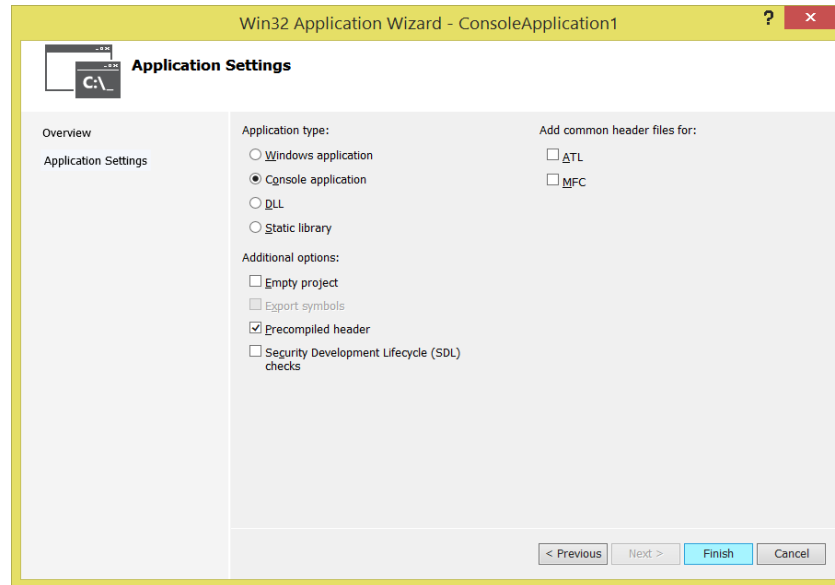
foydalaning; *Name* maydoniga hosil qilinuvchi ilova nomini kiriting; *OK* tugmasini bosing.



2.1 – rasm. New Project dialog oynasi.



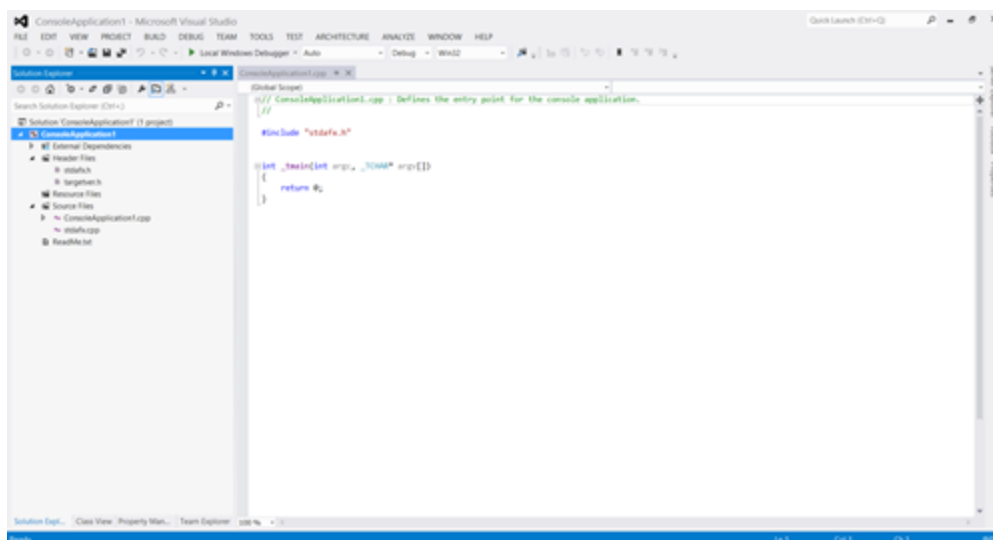
2.2 – rasm. Yaratiluvchi loyiha turini belgilash dialog oynasi.



2.3 – 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).



2.4 – rasm. Oldindan tayyorlangan konsol ilovasi.

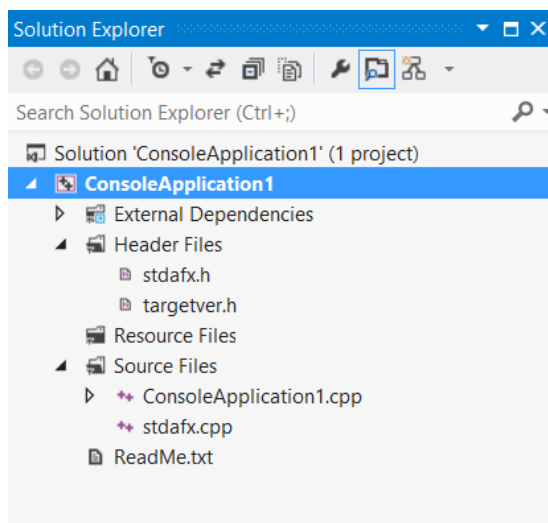
4. Oldindan tayyorlangan konsol ilovasi asosiy funksiya boshi `int _tmain(int argc, _TCHAR* argv[])` va figurali qavslar bilan ajratilgan tanasidan iborat. Funksiyaning boshidagi `_tmain` ni `_tmain()` ga o'zgatiramyiz va dastur tanasidan `return 0` ni olib tashlaymiz.

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.

Visual C++ 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.



2.5 – rasm. Ilova loyihasining ko'rinishi.

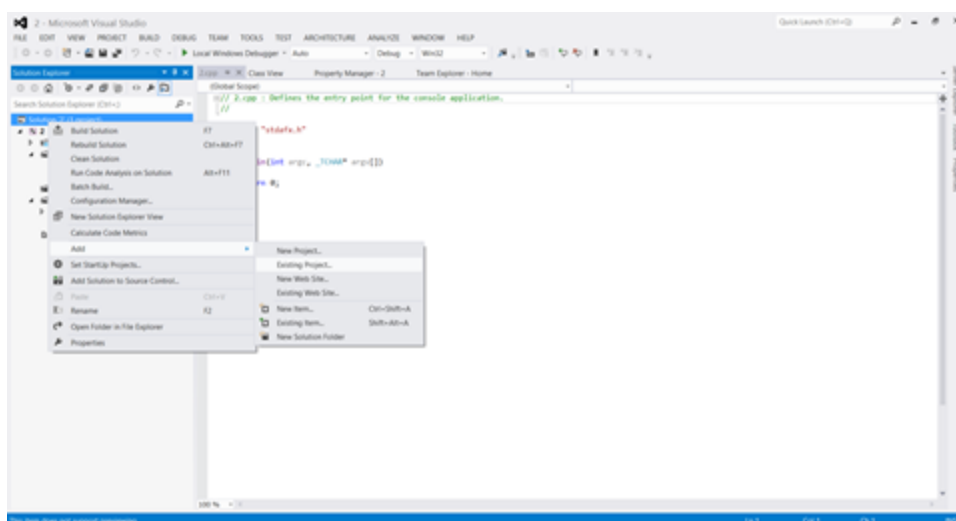
Loyiha doirasida kompilyatsiyalangan va qurilgan, bajarilishi mumkin bo'lgan dastur yaratiladi. Har qanday loyiha kamida ikkita podkonfiguratsiyaga ega: **otladochniy** va oddiy (bajariluvchi). Loyihalar boshqa karkas, boshqa konteynerning qismi hisoblanadi. U **Solution (Yechim)** deb nomlanadi va loyihalar 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 toppish 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

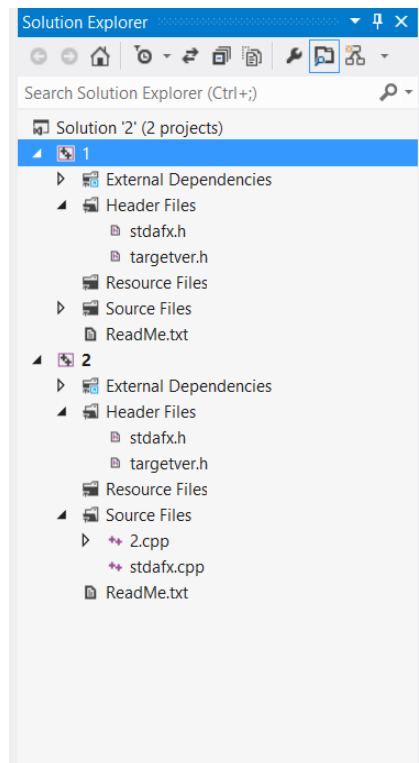
tezlashtiradi. Ta'klidlash joizki, loyihalarning barcha fayllari ham **Solution Explorer** da tasvirlanavermaydi.

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** loyihagini ochamiz, buning natijasida u **Solution '2'** ga qo'shiladi (3.7 – rasm).



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



2.7 – 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>
```

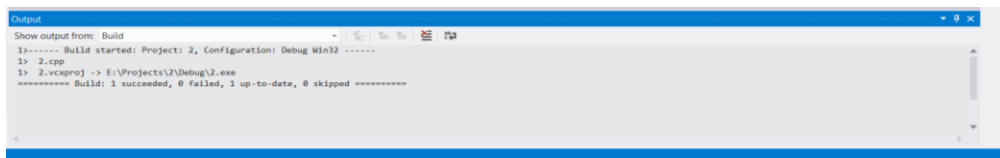
```
#include <tchar.h>
```

- **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:

```
C:\WINDOWS\system32\cmd.exe
0 -17.8
20 -6.7
40 4.4
60 15.6
80 26.7
100 37.8
120 48.9
140 60.0
160 71.1
180 82.2
200 93.3
220 104.4
240 115.6
260 126.7
280 137.8
300 148.9
```

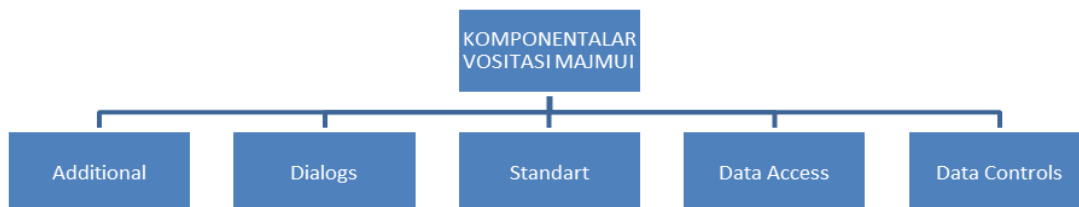
For operatori

While operatoridan tashqari takrorlanish jarayonini for operatori bilan ham tashkil qilishimiz mumkin. Yuqoridagi temperaturani hisoblash dasturini boshqacha ko'rinishda tashkil qilamiz:

```
#include "stdafx.h"
#include <conio.h> // _getch funksiyasi uchun
int _tmain(int argc, _TCHAR* argv[])
{
    int fahr;
    for (fahr=0; fahr <= 300; fahr=fahr+20)
        printf("%4d %6.1f\n", fahr, (5.0/9.0)*(fahr-32.0));
    _getch();
}
```


2.2. 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.

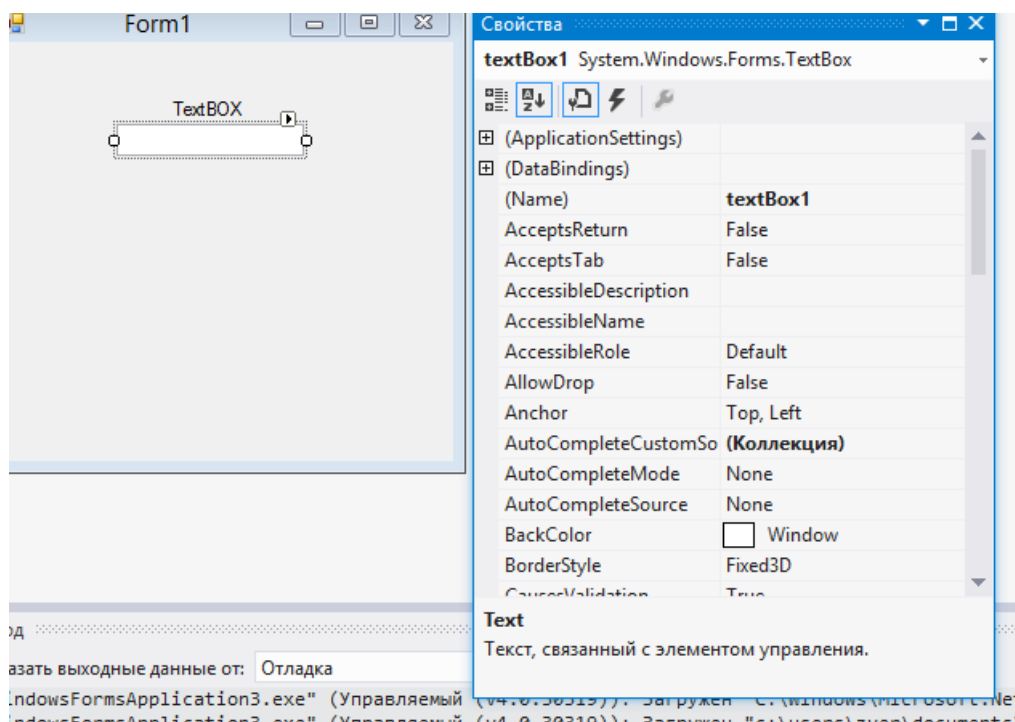
TextBox komponentasi



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)



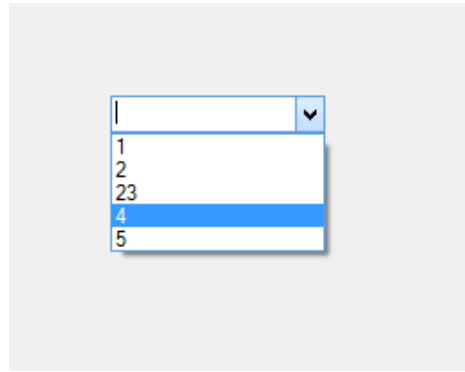
2.8-rasm

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. (3.9-rasm)



2.9-rasm

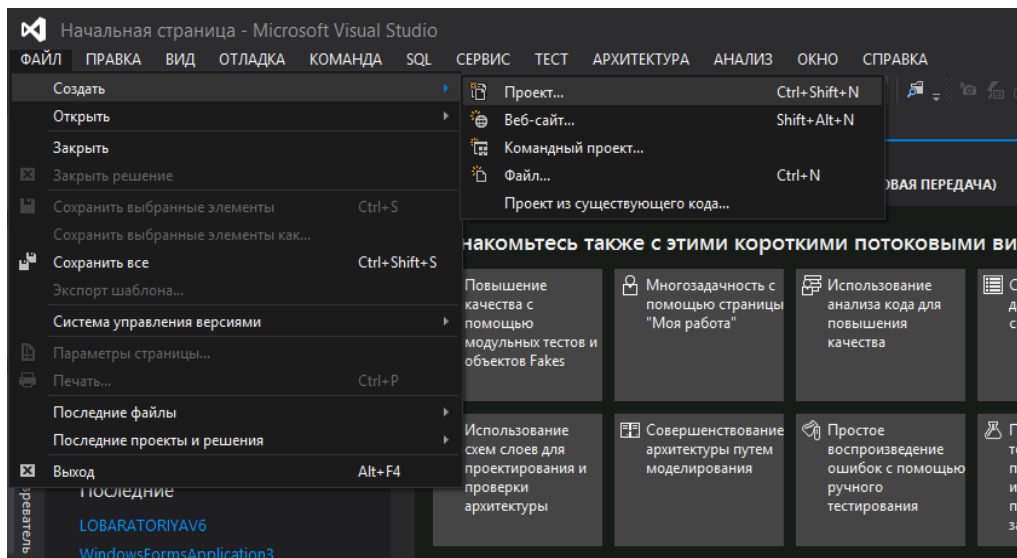
2.3. Loyihani Visual Studio 2012 dasturida C++ tilida yaratish

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

Forma oynasi

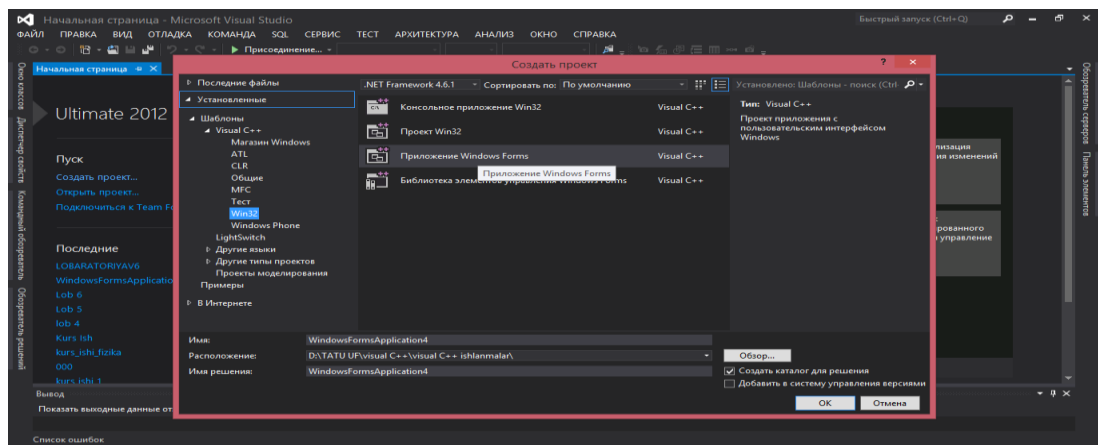
Visual Studio dasturida forma yaratish.

File ->Project(3.10-rasm)



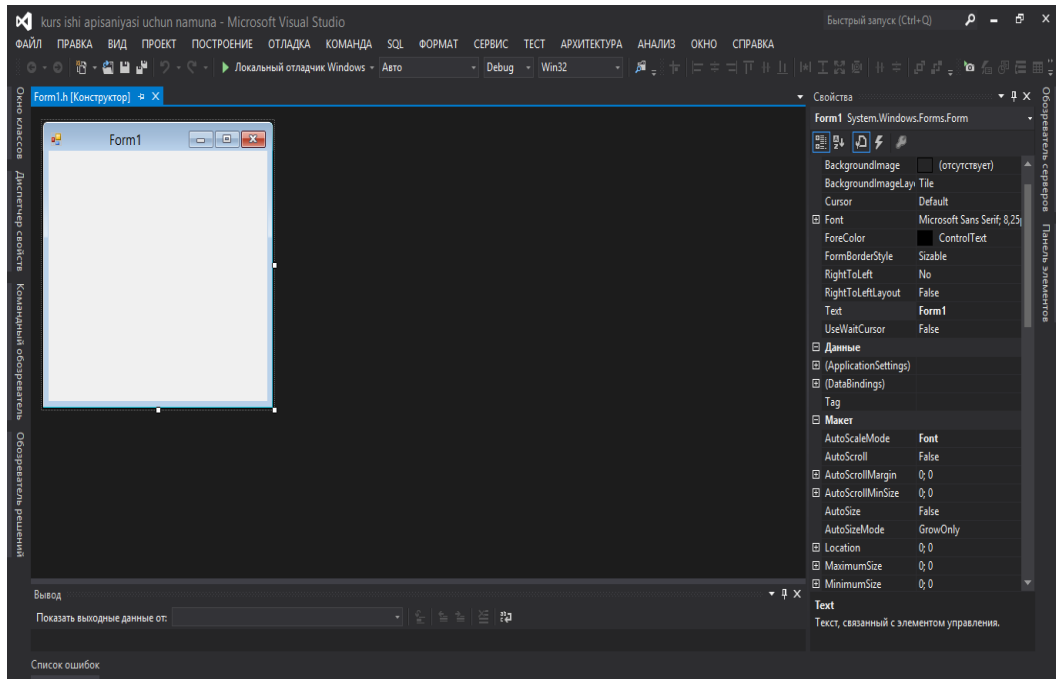
2.10-rasm

Visual C++ ->Win32->Windows Forms



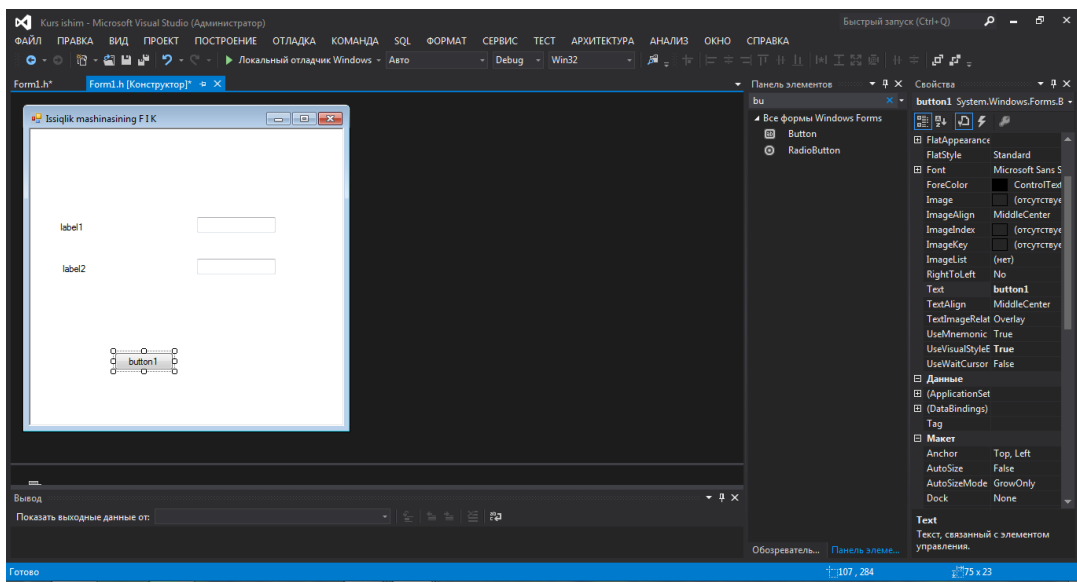
2.11-rasm

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

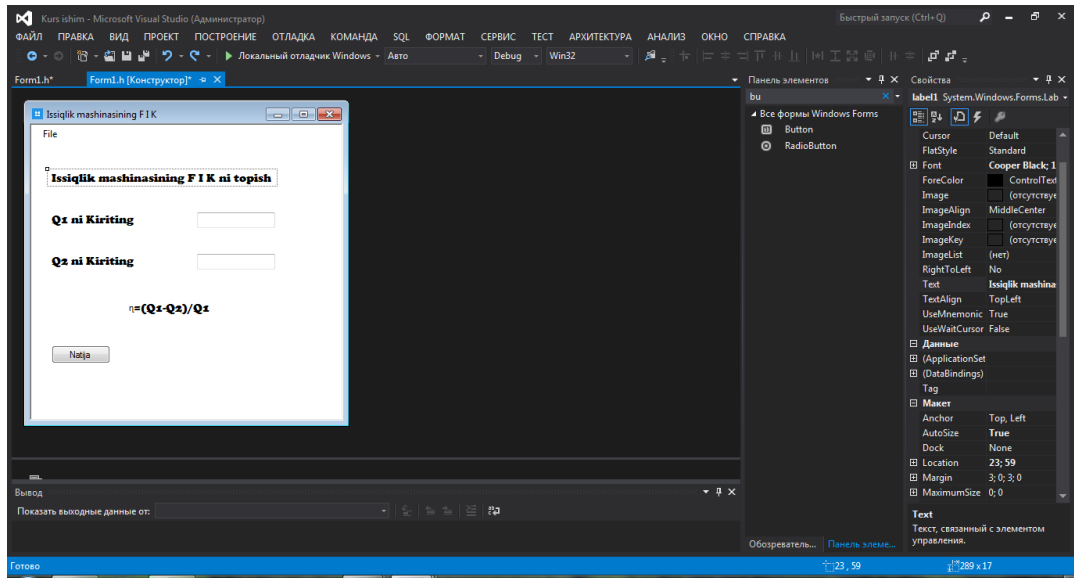


2.12-rasm

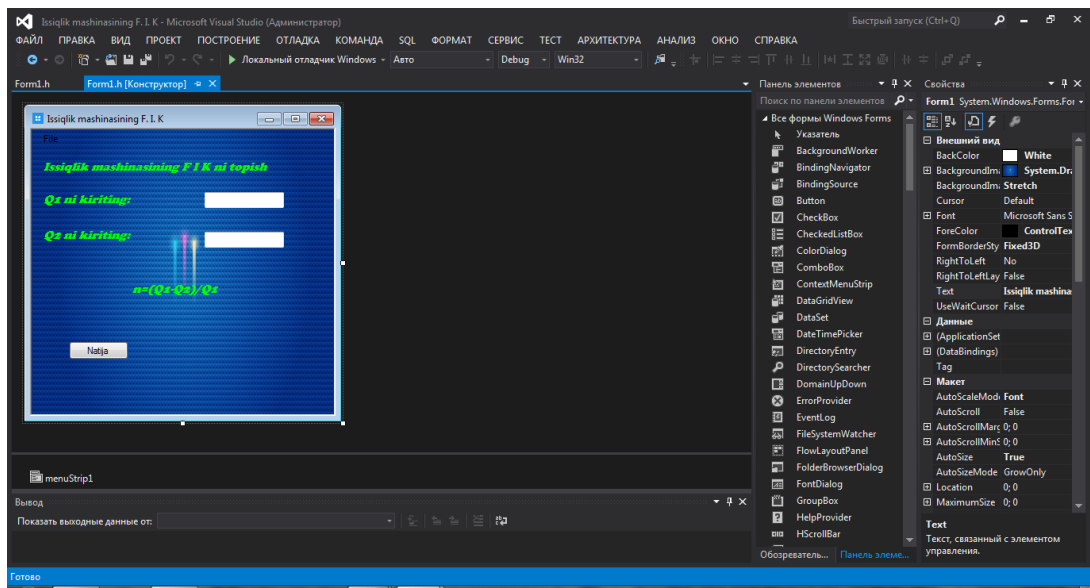
Dasturda zarur bo`lgan komponentalar tanlanadi va form oynasiga tashlanadi.



2.13-rasm



2.14-rasm

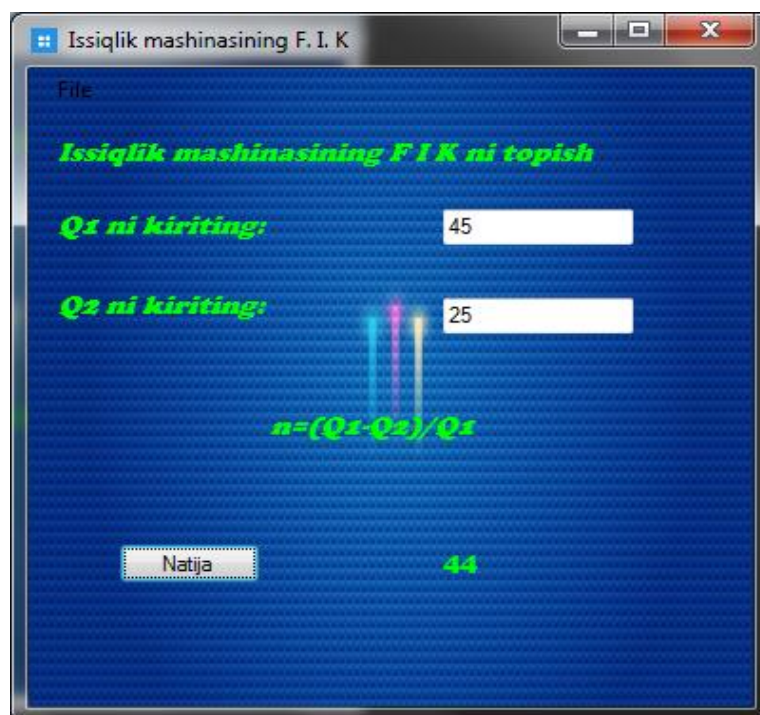


2.15-rasm

2.4. Dasturdan foydalalnish uchun yo`riqnoma



2.16-rasm



2.17-rasm

Xulosa

Ushbu kurs ishini bajarish davomida quyidagi ishlar amalga oshirildi:

1. Visual Studio muhitida windows form oynasi ochildi.
2. Windows form oynasiga kerakli kompanetalar joylashtirildi.

Quyidagi bilim va ko'nikmalarga ega bo'ldim:

1. Visual C++ muhitida mukammal dastur tuzish o'rganidim.
2. Visual C++ da komponentalar bilan ishlashni o'rgandim.
3. Issiqlik miqdori berilganda issiqlik mashinasining foydali ish koefitsiyentini hisoblandi.
4. Temperatura berilganda issiqlik mashinasining foydali ish koefitsiyentini hisoblandi.

Dastur quyidagi qulayliklarga ega:

1. Dastur juda soda tuzilgan, bu esa oddiy foydalanuvchi uchun ham dasturdan foydalanishda qiyinchilik tug'dirmaydi.
2. Dastur ishlash tezligi yuqori.

Foydalanilgan adabiyotlar:

I. Asosiy adabiyotlar:

1. Zohidov, R.A. Issiqlik texnikasi: o‘quv qo‘llanma/ R.A. Zohidov, M.M. Alimova, Sh.S.Mavjudova; O‘zbekiston Respublikasi Oliy va o‘na maxsus ta’lim vazirligi. Toshkent: «O‘zbekiston faylasuflari milliy jamiyati» nashriyoti, 2010. - 200 b.
2. Зиборов В. - MS Visual C ++2010 в среде NET Питер, 2012

II. Qo‘shimcha adabiyotlar:

1. Б. Пахомов - СС++ и MS VisualC++ 2008 для начинающих БХВ-Петербург» 2008

III. Internet-resurslar:

1. <http://referat.arxiv.uz/index.php?do=files&op=download&fileid=46577>
2. <http://n.ziyouz.com/kutubxona/category/151-texnika-va-texnologiya?download=7471:issiqlik-texnikasi-r-zohidov-va-b>
3. <http://library.ziyouz.com/ru/book/download/27608>

ILOVA

```
#pragma once
#include"stdafx.h";
#include"Temperaturada1.h";
#include<math.h>

namespace IssiqlikmashinasiningFIK {

    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::Label^ label1;
```

```
private: System::Windows::Forms::MenuStrip^ menuStrip1;
private: System::Windows::Forms::ToolStripMenuItem^
newToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
exitToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
newToolStripMenuItem1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Label^ label4;
```

```
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::ToolStripMenuItem^
temperaturaToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
refreshToolStripMenuItem;
```

```
private: System::ComponentModel::IContainer^ components;
protected:
```

```
private:
    /// <summary>
    /// Требуется переменная конструктора.
    /// </summary>
```

```
#pragma region Windows Form Designer generated code
    /// <summary>
    /// Обязательный метод для поддержки конструктора - не
изменяйте
    /// содержимое данного метода при помощи редактора
кода.
    /// </summary>
void InitializeComponent(void)
```

```

    {

        System::ComponentModel::ComponentResourceManager^ resources =
(gcnew
System::ComponentModel::ComponentResourceManager(Form1::typeid));
        this->label1 = (gcnew
System::Windows::Forms::Label());
        this->menuStrip1 = (gcnew
System::Windows::Forms::MenuStrip());
        this->newToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->newToolStripMenuItem1 = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->temperaturaToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->exitToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->label2 = (gcnew
System::Windows::Forms::Label());
        this->label3 = (gcnew
System::Windows::Forms::Label());
        this->textBox1 = (gcnew
System::Windows::Forms::TextBox());
        this->textBox2 = (gcnew
System::Windows::Forms::TextBox());
        this->button1 = (gcnew
System::Windows::Forms::Button());
        this->label4 = (gcnew
System::Windows::Forms::Label());
        this->label5 = (gcnew
System::Windows::Forms::Label());
        this->refreshToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->BackColor =
System::Drawing::Color::Transparent;

```

```

        this->label1->Font = (gcnew
System::Drawing::Font(L"Cooper Black", 11.25F,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label1->ForeColor =
System::Drawing::Color::Lime;
        this->label1->Location = System::Drawing::Point(12,
38);

        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(335, 38);
        this->label1->TabIndex = 0;
        this->label1->Text = L"Issiqlik mashinasining F I K ni
topish";

        this->label1->Click += gcnew
System::EventHandler(this, &Form1::label1_Click);
        //
        // menuStrip1
        //
        this->menuStrip1->BackColor =
System::Drawing::Color::Transparent;
        this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) { this-
>newToolStripMenuItem});
        this->menuStrip1->Location =
System::Drawing::Point(0, 0);
        this->menuStrip1->Name = L"menuStrip1";
        this->menuStrip1->Size = System::Drawing::Size(381,
24);

        this->menuStrip1->TabIndex = 1;
        this->menuStrip1->Text = L"menuStrip1";
        //
        // newToolStripMenuItem
        //
        this->newToolStripMenuItem->DropDownItems-
>AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(4)
{ this->refreshToolStripMenuItem,
        this->newToolStripMenuItem1, this-
>temperaturaToolStripMenuItem, this->exitToolStripMenuItem});
        this->newToolStripMenuItem->Name =
L"newToolStripMenuItem";

```

```

        this->newToolStripMenuItem->Size =
System::Drawing::Size(37, 20);
        this->newToolStripMenuItem->Text = L"File";
        this->newToolStripMenuItem->Click += gnew
System::EventHandler(this, &Form1::newToolStripMenuItem_Click);
        //
        // newToolStripMenuItem1
        //
        this->newToolStripMenuItem1->Name =
L"newToolStripMenuItem1";
        this->newToolStripMenuItem1->Size =
System::Drawing::Size(152, 22);
        this->newToolStripMenuItem1->Text = L"Issiqlik";
        this->newToolStripMenuItem1->Click += gnew
System::EventHandler(this, &Form1::newToolStripMenuItem1_Click);
        //
        // temperaturaToolStripMenuItem
        //
        this->temperaturaToolStripMenuItem->Name =
L"temperaturaToolStripMenuItem";
        this->temperaturaToolStripMenuItem->Size =
System::Drawing::Size(152, 22);
        this->temperaturaToolStripMenuItem->Text =
L"Temperatura";
        this->temperaturaToolStripMenuItem->Click += gnew
System::EventHandler(this, &Form1::temperaturaToolStripMenuItem_Click);
        //
        // exitToolStripMenuItem
        //
        this->exitToolStripMenuItem->Name =
L"exitToolStripMenuItem";
        this->exitToolStripMenuItem->Size =
System::Drawing::Size(152, 22);
        this->exitToolStripMenuItem->Text = L"Exit";
        this->exitToolStripMenuItem->Click += gnew
System::EventHandler(this, &Form1::exitToolStripMenuItem_Click);
        //
        // label2
        //
        this->label2->BackColor =
System::Drawing::Color::Transparent;

```

```

        this->label2->Font = (gcnew
System::Drawing::Font(L"Cooper Black", 11.25F,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label2->ForeColor =
System::Drawing::Color::Lime;
        this->label2->Location = System::Drawing::Point(12,
79);

        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(144, 29);
        this->label2->TabIndex = 2;
        this->label2->Text = L"Q1 ni kiriting:";
        //
        // label3
        //
        this->label3->BackColor =
System::Drawing::Color::Transparent;
        this->label3->Font = (gcnew
System::Drawing::Font(L"Cooper Black", 11.25F,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label3->ForeColor =
System::Drawing::Color::Lime;
        this->label3->Location = System::Drawing::Point(12,
124);

        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(144, 29);
        this->label3->TabIndex = 3;
        this->label3->Text = L"Q2 ni kiriting:";
        //
        // textBox1
        //
        this->textBox1->BackColor =
System::Drawing::Color::White;
        this->textBox1->Location =
System::Drawing::Point(218, 79);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(100,
20);

        this->textBox1->TabIndex = 4;

```

```

        this->textBox1->TextChanged += gnew
System::EventHandler(this, &Form1::textBox1_TextChanged);
        //
        // textBox2
        //
        this->textBox2->Location =
System::Drawing::Point(218, 129);
        this->textBox2->Name = L"textBox2";
        this->textBox2->Size = System::Drawing::Size(100,
20);

        this->textBox2->TabIndex = 5;
        //
        // button1
        //
        this->button1->Location = System::Drawing::Point(47,
267);

        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(75, 23);
        this->button1->TabIndex = 6;
        this->button1->Text = L"Natija";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gnew
System::EventHandler(this, &Form1::button1_Click);
        //
        // label4
        //
        this->label4->BackColor =
System::Drawing::Color::Transparent;
        this->label4->Font = (gnew
System::Drawing::Font(L"Cooper Black", 11.25F,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label4->ForeColor =
System::Drawing::Color::Lime;
        this->label4->Location = System::Drawing::Point(124,
192);

        this->label4->Name = L"label4";
        this->label4->Size = System::Drawing::Size(144, 26);
        this->label4->TabIndex = 7;
        this->label4->Text = L"n=(Q1-Q2)/Q1";

```



```

        this->label4->Click += gnew
System::EventHandler(this, &Form1::label4_Click);
        //
        // label5
        //
        this->label5->AutoSize = true;
        this->label5->BackColor =
System::Drawing::Color::Transparent;
        this->label5->Font = (gnew
System::Drawing::Font(L"Cooper Black", 11.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->label5->ForeColor =
System::Drawing::Color::Lime;
        this->label5->Location = System::Drawing::Point(215,
269);

        this->label5->Name = L"label5";
        this->label5->Size = System::Drawing::Size(0, 17);
        this->label5->TabIndex = 9;
        //
        // refreshToolStripMenuItem
        //
        this->refreshToolStripMenuItem->Name =
L"refreshToolStripMenuItem";
        this->refreshToolStripMenuItem->Size =
System::Drawing::Size(152, 22);
        this->refreshToolStripMenuItem->Text = L"Refresh";
        this->refreshToolStripMenuItem->Click += gnew
System::EventHandler(this, &Form1::refreshToolStripMenuItem_Click);
        //
        // Form1
        //
        this->AutoSizeDimensions =
System::Drawing::SizeF(6, 13);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->AutoSize = true;
        this->BackColor = System::Drawing::Color::White;
        this->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"$this.BackgroundImage")));

```

```

        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(381, 359);
        this->Controls->Add(this->label5);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->menuStrip1);
        this->ForeColor =
System::Drawing::SystemColors::ControlText;
        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::Fixed3D;
        this->Icon = (cli::safe_cast<System::Drawing::Icon^
>(resources->GetObject(L"$this.Icon")));
        this->MainMenuStrip = this->menuStrip1;
        this->Name = L"Form1";
        this->Text = L"Issiqlik mashinasining F. I. K";
        this->Load += gnew System::EventHandler(this,
&Form1::Form1_Load);
        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e) {
    }
    private: System::Void label1_Click(System::Object^ sender,
System::EventArgs^ e) {
    }
    private: System::Void
newToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
    }

```

```

private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    double q1,q2;
    int n;
    if(textBox1->Text==" " || textBox2->Text==" ")
    {
        MessageBox::Show("Qiymat toliq kiritilmadi");
    }
    else
    {
        q1=Convert::ToDouble(textBox1->Text);
        q2=Convert::ToDouble(textBox2->Text);
        n=((q1-q2)*1.0/q1)*100;
        label5->Text=Convert::ToString(n);
    }
}

private: System::Void label4_Click(System::Object^ sender,
System::EventArgs^ e) {
}

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
}

private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
}

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

private: System::Void newToolStripMenuItem1_Click(System::Object^
sender, System::EventArgs^ e) {
    label1->Text=Convert::ToString("Issiqlikda ishlash");
    label2->Text=Convert::ToString("Q1 ni kiriting");
    label3->Text=Convert::ToString("Q2 ni kiriting");
    label4->Text=Convert::ToString("n=(Q1-Q2)/Q1");
    textBox1->Text=Convert::ToString("");
}

```

```

        textBox2->Text=Convert::ToString("");
        label5->Text=Convert::ToString("");

    }
    private: System::Void
    temperaturaToolStripMenuItem_Click(System::Object^ sender,
    System::EventArgs^ e) {
        label1->Text=Convert::ToString("Temperaturada
    ishlash");

        label2->Text=Convert::ToString("T1 ni kiriting");
        label3->Text=Convert::ToString("T2 ni kiriting");
        label4->Text=Convert::ToString("n=(T1-T2)/T1");
        textBox1->Text=Convert::ToString("");
        textBox2->Text=Convert::ToString("");
        label5->Text=Convert::ToString("");

    }
    private: System::Void refreshToolStripMenuItem_Click(System::Object^
    sender, System::EventArgs^ e) {
        textBox1->Text=Convert::ToString("");
        textBox2->Text=Convert::ToString("");
        label5->Text=Convert::ToString("");
    }
};
}

```