

# Қазіргі программалау технологиялары

(C# тілінің элементтері негізінде)

## Бағалау жүйесі

<b>Лекцияларға қатынасу және белсенділік көрсету</b>	<b>5% = 2,5% + 2,5%</b>
<b>Лабораториялық жұмыстар орындау</b>	<b>30% = 15% + 15%</b>
<b>1 межелік бақылау</b>	<b>10% = 5% + 5%</b>
<b>2 межелік бақылау</b>	<b>10% = 5% + 5%</b>
<b>Авторлық жоба жасау (программа)</b>	<b>5%</b>
<b>Нәтижелік емтихан</b>	<b>40%</b>
<b>Барлығы</b>	<b>100%</b>

# Оқулықтар мен әдебиеттер

1. Павловская Т.А. С # (Си Шарп). Язык программирования на платформе .NET. Учебник для вузов. –СПб.: Питер, 2010. -432 с.
2. Прайс Дж , Гандерлой М . Visual C# 2.0 .NET. Полное руководство: Пер.с англ. –К.ВЕК+, СПб.: КОРОНА-ВЕК, М.: Энтроп, 2007. -736 с.
3. Нэш Трей. С# 2008: ускоренный курс. Пер.с англ. –М.: ООО “И.Д.Вильямс”, 2008. -576 с.
4. Культин Н.Б. С# в задачах и примерах. –СПб.: БХВ-Петербург, 2007. -240 с.
5. Петцольд Ч. Программирование для MS Windows на С#. Т. 1. и Т.2. –М.: Издательско-торговый дом”Русская редакция”, 2002. -576 с.

# 1-лекция

## Microsoft Visual C# програмаулау ортасында жұмыс істеу негіздері. Алғашқы программа құру

# Сұрақтар

1. C# программалау ортасы туралы жалпы мәліметтер
2. C# тіліндегі ең қарапайым программа мысалы
3. Программаны компиляциядан өткізу және орындау
  - 3.1. Программаны Visual Studio.NET ортасында орындау
  - 3.2. Программаны орындау

# 1. C# программалау ортасы туралы жалпы мәліметтер

C# тілін («Си-шарп» деп айтылады, мұнда Бетховеннің «Ай сонатасындағы» музыкалық нотация — до-диез таңбасы (#) пайдаланылған; ағылшындар оны «си-шарп» деп оқиды) Microsoft компаниясының маманы **Андерс Хейлзберг** (Anders Hejlsberg) жаңа объектіге бағытталға программалау тілі ретінде C, C++, Java және BASIC тілдерінің негізінде жасап шығарған. Бұл пәнде осы тілдің негізгі элементтері қарастырылады.

С # тілі **Microsoft** фирмасының **.NET** (дот-нет) архитектурасы үшін программалар жазуға арналған тілі. **.NET** – программалау технологиясындағы жаңа платформа, ол желіге қосылған компьютерлерге арналып жасалған.

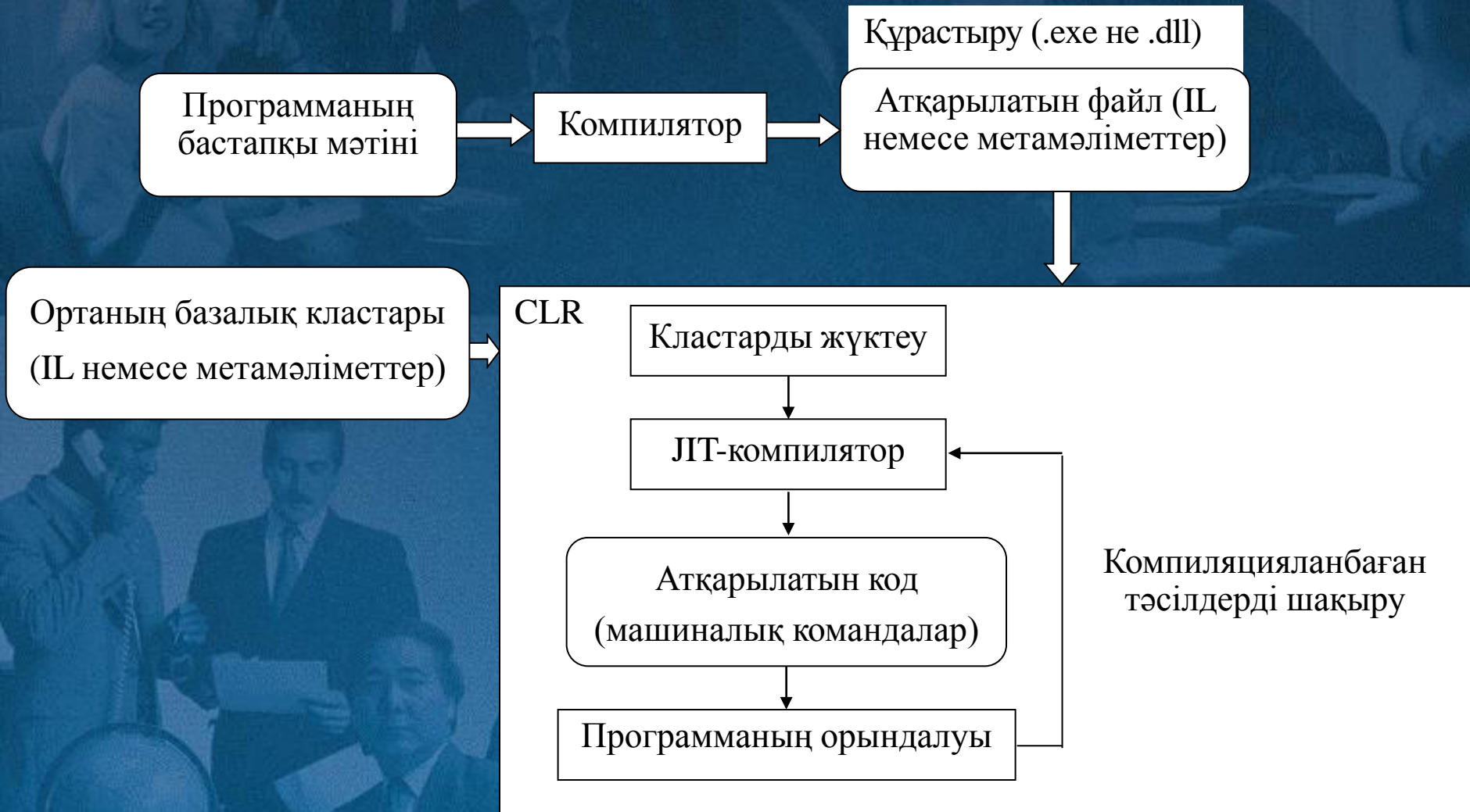
**.NET** көмегімен шағын мобильді компьютерлерде орындалатын программалар жазу ыңғайлы деп есептеледі. Бұл технология **Visual Studio.NET** деп аталады да, бұл ортада Visual Basic, басқарылатын C++ және C # тілдерінде программалар жасау қарастырылған, бірақ ол осылармен ғана шектеліп қалмайды.

C # тілінің мүмкіндігі **Java** тілімен қатарлас, осы екі тіл қазіргі алдыңғы қатарлы технологияларға жатады.

C# тілін сүйемелдейтін **.NET** платформасының программа жасау ортасы бастапқы программаны бірден машиналық кодқа емес, **Microsoft Intermediate Language (MSIL** не **IL**) атты аралық тілге аударады. Ол ОЖ-дан, компьютер типінен тәуелсіз командалардан тұрады да, бірден орындалмай, **тілдің жалпы орындалу ортасы Common Language Runtime – CLR** деп аталатын жүйе арқылы атқарылады. **CLR** ортасы кез келген ОЖ-да орындала береді. Программаны орындау кезінде, **CLR JIT-компиляторын (just in time** – дер кезінде) шақырады. Компилятор программаны бөліктерге бөле отырып, оның тек осы сәтте керек бөлігін ғана орындайды.



Сонымен **.NET** платформасын пайдалану кезіндегі программаның орындалу схемасы суретте көрсетілгендей түрде болады.



Компилятор жұмысы нәтижесінде **құрастыру** (сборка) деп аталатын **exe** немесе **dll** типті файл жасалады, онда **IL** тіліндегі код пен метамәліметтер болады.

*Метамәліметтер программада қолданылатын объектілер және құрастыру туралы ақпарат сақтайды. Ол тіларалық әрекеттесу, қауіпсіздік, сенімділік сияқты мүмкіндіктерді қамтамасыз етеді.*

**.NET** платформасында көлемді кластар кітапханасы бар, оны осы ортадағы кез келген тілде пайдалана беруге болады.

# 2. C# тіліндегі ең қарапайым программа

```
/* Example1_1.cs файлы  
"Hello world!" программасы және ағымдағы  
мерзім мен уақытты экранға шығару */  
class Example1_1  
{  
    public static void Main()  
    { // "Hello World!" сөзін экранға шығару  
      System.Console.WriteLine("Hello World!");  
      // ағымдағы мерзім мен уақытты шығару  
      System.Console.WriteLine("The current date  
and time is " + System.DateTime.Now);  
    }  
}
```

Бастапқы **Example1\_1.cs** файлы блокнотта немесе кез келген редакторда теріледі. Файл кеңейтілуі (типi) – **cs**. Осы тип барлық C# программаларына беріледі.

Программаны TurboC# консолдық компиляторы (MS DOS) арқылы орындау үшін:

**csc Example1\_1.cs ↵**

жолын тереміз. Компилятор бұл файлды машиналық кодқа түрлендіріп, **Example1\_1.exe** файлын жасайды. Оның ішкі мазмұнын, яғни программа нәтижесін көру үшін:

**Example1\_1 ↵**

жолын тереміз.

Программа нәтижесі:

**Hello World!**

**The current date and time is  
28.08.2010 12:22:44 PM**

Программа нәтижесі аздап басқаша да болуы мүмкін, мысалы, мынадай:

**Hello World!**

**The current date and time is  
28.08.2010 12:22:44**

Енді программаның әрбір жолын жеке-жеке қарастырып шығайық.

**/\* Example1\_1.cs** файлы "Hello world!"  
программасы және ағымдағы мерзім мен  
уақытты экранға шығару **\*/**

Бұл C тілдері стиліндегі комментарий,  
компилятор мұны тек бос орын таңбасы  
деп есептейді, ол бізге оның жұмысын  
түсіну үшін ғана керек.

**class Example1\_1**  
{

Жүйелі жақша жаңа **Example1\_1** класының  
сипатталуы басталғанын көрсетеді. Жақша  
жабылғанда сипатталу бітеді. Класс ішінде  
айнымалылар, функциялар – тәсілдер  
болады.

# public static void Main()

Бұл программаның басты функциясы **Main ()**, ол бас әріппен жазылады. Программа осы функцияны орындаудан басталады.

**public** – бұл қатынасу деңгейінің модификаторы, ол осы тәсілге бұл кластан тыс қатынасу мүмкіндігін көрсетеді,

**static** – бұл сөз **Main()** тәсілінің нақты бір объектіге емес, класқа қатысты екенін білдіреді. Егер **static** сөзі болмаса, бұл функцияны шақырмас бұрын осы класс объектісін жасап алу керек еді.

```
System.Console.WriteLine("Hello World!");
```

Бұл функция аты мына бөліктерден тұрады:

- **System** — атаулар кеңістігі;
- **Console** — сол атаулар кеңістігінде анықталған стандартты класс;
- **WriteLine** — осы класта анықталған тәсіл.

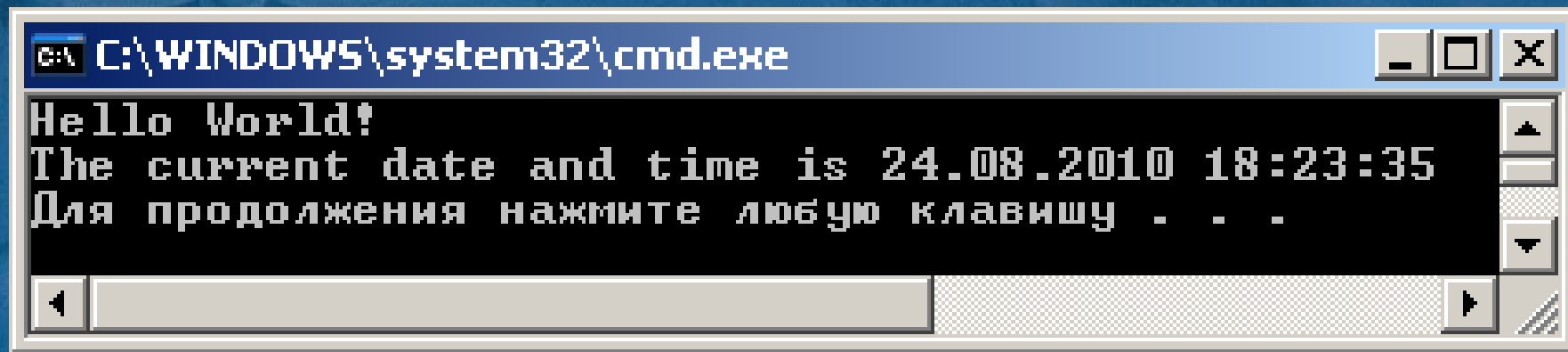
Программалардағы **WriteLine** және **Write** тәсілдері — **Console** класын экранға шығару ісін атқарады. **Write** тәсілі экранға (консольға) мәлімет шығарып, курсорды осы жолда қалдырады. Ал **WriteLine** тәсілі курсорды жаңа жолға көшіреді. Бұл ортада әртүрлі параметрлермен қолданылатын **Write** тәсілінің 18 сипаттамасы, ал **WriteLine** тәсілінің 19 сипаттамасы бар екен.



```
System.Console.WriteLine("The current date  
and time is " + System.DateTime.Now);
```

Мұнда экранға күн-ай мерзімі мен уақытты беретін тәсіл мәлімет шығарады. **Now** — **DateTime** объектісінің қасиеті, ол жүйелік дата мен уақытты береді. **Now** — статикалық қасиет, сондықтан оны **DateTime** объектісін жасамай-ақ қолдана беру ге болады.

Программа жұмысы нәтижесі:



```
C:\WINDOWS\system32\cmd.exe
Hello World!
The current date and time is 24.08.2010 18:23:35
Для продолжения нажмите любую клавишу . . .
```

## Тағы бір мысал

// Listing1\_1.cs файлы

```
Using System;
```

```
namespace ConsoleApplication1
```

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

```
/// Summary description for Class1
```

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

```
class Class1
```

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

```
/// The main entry point for the application
```

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

```
[STAThread]
```

```
public static void Main(string[] args)
```

```
{ //
```

```
/// TODO: Add code to start application
```

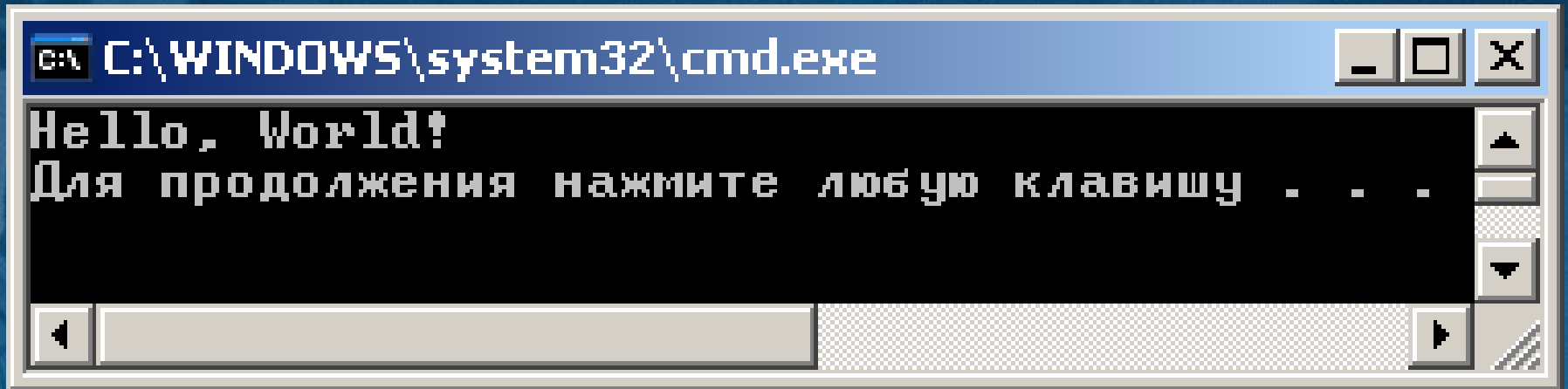
```
//
```

```
Console.WriteLine("Hello, World!");
```

```
}
```

```
}
```

```
}
```



**Using System** директивасы **System** атаулар кеңістігінен тікелей стандартты класс аттарын (кеңістік атын көрсетпей) пайдалануға рұхсат береді.

**namespace** түйінді сөзі осы программа үшін **ConsoleApplication1** атты өз атаулар кеңістігін жасайды. Бұл программа объектілеріне ат беруге болатынын көрсетеді.

2 не 3 қиғаш сызықтан басталатын жолдар комментарий болып саналады, 3 сызық программа комментарийлерінен құжат жасауға мүмкіндік береді.

C# тілі объектіге бағытталған тіл, сондықтан оның программасы бір-бірімен өзара байланысқан кластардан тұрады. Бұл программада тек бір ғана класс бар, оған өзіміз **Class1** деп ат бердік, класс сипаттамасы **class** түйінді сөзінен басталады да, аты жазылады. Сонан соң жүйелі жақша ішінде класс элементтері көрсетіледі. Олардағы функциялар мен мәліметтер **тәсіл – метод** деп аталады.

Жоғарыдағы класс ішінде тек бір элемент – **Main** методы, яғни тәсілі бар. Әр программада осындай бір басты тәсіл болады, программаны орындау содан басталады.

### 3. Программаны компиляциядан өткізу және орындау

C# тіліндегі программа мәтін түрінде жазылады. Кез келген мәтіндік редактор арқылы программаны компьютерде терміз де, артынан оны компиляциядан өткізіп, сонан соң орындап, нәтижесін аламыз.

Сонымен, компьютердің C# тіліндегі программаны орындауы үшін, оны машина тіліне аударатын *компилятор* керек екені түсінікті шығар.

Компилятор программа мәтіні жазылған файлды оқып, оны талдайды, қателерін тексереді. Қателері түзетілген соң, программаны орындап, нәтиже беретін атқарылатын файл (исполняемый) жасайды.

Программаны бір рет компиляциядан өткізген соң, оны әр түрлі бастапқы мәліметтер үшін атқарылатын файлды қайталап орындай отырып, нәтижелер аламыз.

Microsoft Windows NT, Windows 2000, Windows XP, Windows 2007 операциялық жүйелерінде **Visual Studio.NET** ортасы қолданылады, ол *программалаудың біріктірілген ортасы* болып табылады, яғни құрамында **C#** тәрізді мәтіндік редакторды, компиляторды, т.б. қосымша программаларды біріктіріп тұрады.

*Бұл ортада* кез келген программа **жоба** түрінде қарастырылады.

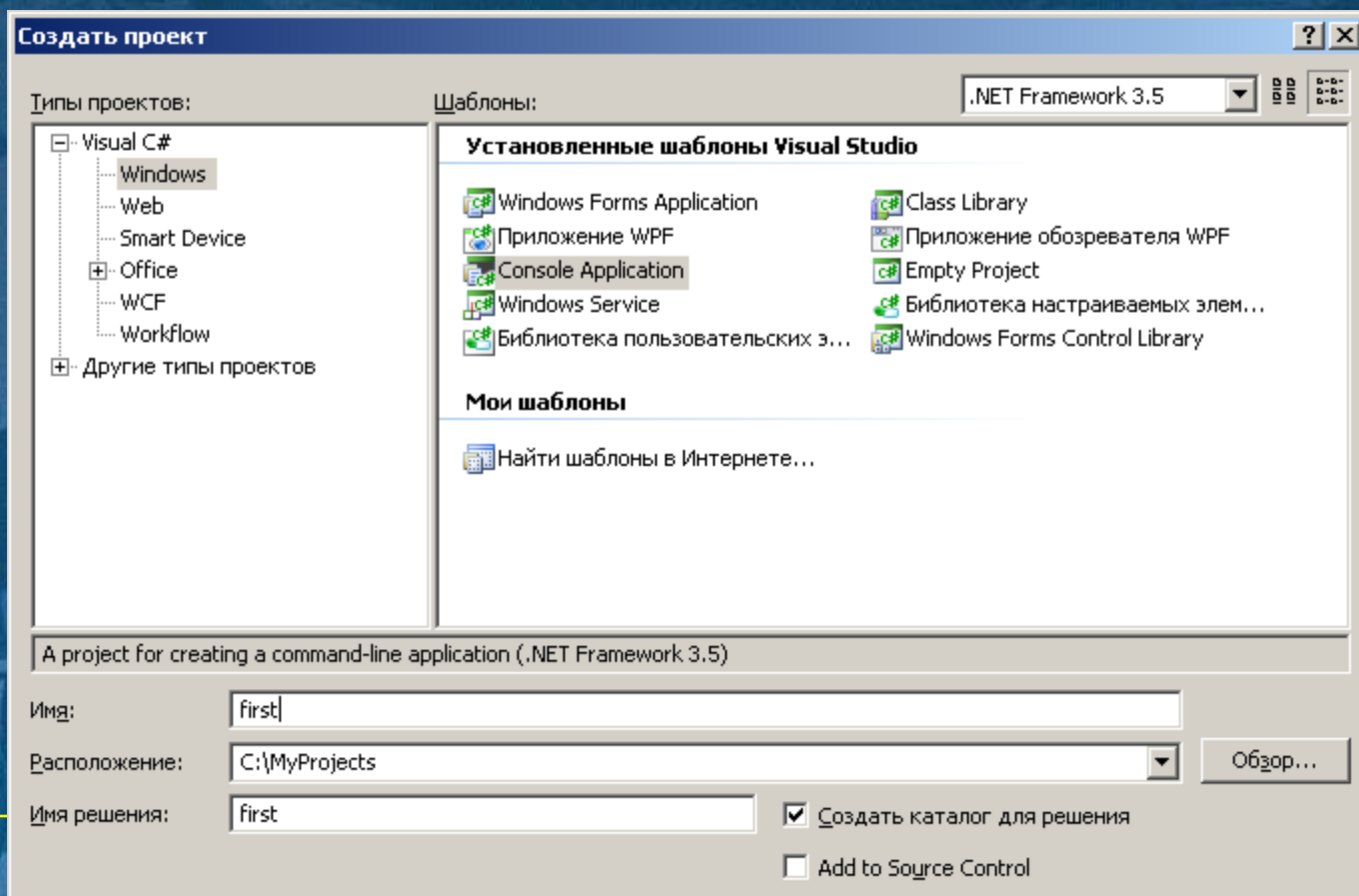


# 3.1. Программаны Visual Studio.NET ортасында орындау

.NET ортасында жасалған кез келген программа жоба (*project*) болып саналады. Жоба — бір-бірімен байланысқан бастапқы немесе соларға қосымша файлдар жиыны, соларды компиляциядан өткізіп біріктірсек, атқарылатын бір файл шығуы мүмкін. Бұл көлемді программалар жасау үшін өте ыңғайлы. Visual Studio.NET ортасында жұмыс былай басталады:

1) Пуск ► Программы ► Microsoft Visual.NET ► Microsoft Visual C#. Сонда .NET ортасының бас терезесі ашылады.

2) Енді жаңа жоба жасау үшін **File ► New ► Project...** командасын орындаймыз немесе **Ctrl + Shift + N** пернелерін басамыз немесе сол жақ шетте жоғарғы батырманы шертеміз. Жаңа сұхбат (диалог) терезесі шығады.



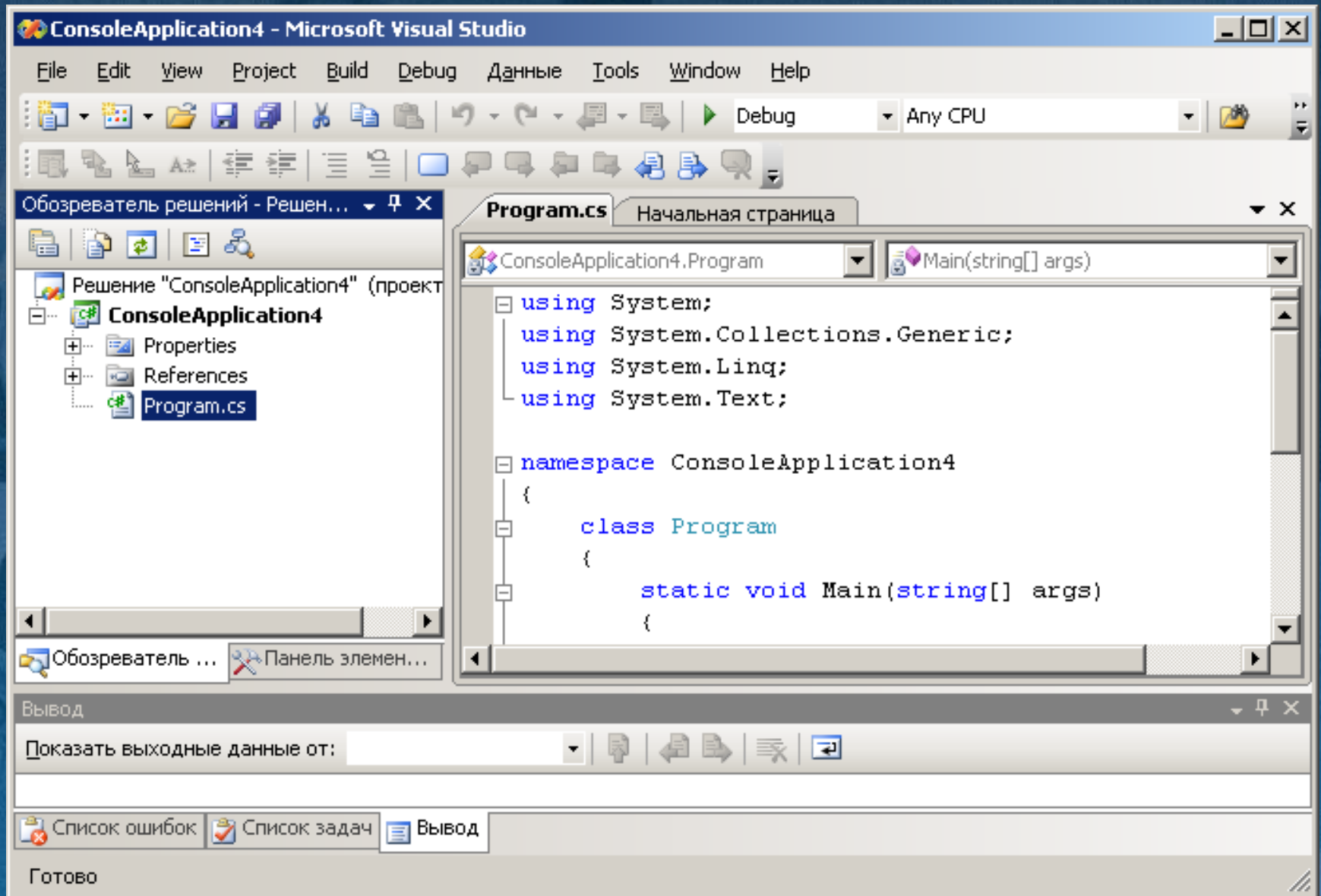
3) Осы терезенің сол жағындағы **Visual C# Projects** жолын, ал оң жағынан - **Console Application** жолын таңдаймыз.

- ◆ **Name** өрісіне жоба атын (**first**) енгізу керек;
- ◆ **Location** өрісіне файлдарды сақтайтын бума атын енгізу қажет (егер ондай бума жоқ болса, ол автоматты түрде ашылады). Ол үшін арнайы бума болуы қажет, мыс.

**C:\MyProjects;**

- ◆ **OK** батырмасын шерту керек.

Осы әрекеттерден кейін экранда келесі суреттегідей көрініс пайда болады.



Visual C# 2008 ортасының бас терезесі (жұмыс үстелі)

Казахский национальный университет им. Аль-Фараби

Жоба жасау басында Visual Studio ортасы программаның алғашқы үлгісін ұсынады. Оны программаның нұсқасы ретінде пайдаланып, ары қарай түрлендіре беруге болады.

Керегін қалдырып, керек емесін өшіріп, арасына буфер арқылы программа мәтінін басқа ортадан көшіріп алуға болады.

Программа коды арасына комментарий қосып отырған өте пайдалы іс болып саналады, өйткені біраз уақыт өткен соң, бұрынғы істегендеріңіз тез ұмытылып кетеді.

4) Терезенің сол жақ жоғарғы бұрышында – **Solution Explorer** – жобаны басқару терезесі орналасқан (егер ол жоқ болса, **View – Solution Explorer** командасын орындау керек). Терезеде жобаға кіретін барлық ресурстар: кітапханаға сілтеме (**System, System.Data, System.XML**), жарлық файлы (**App.ico**), кластың бастапқы мәтіні бар файлы (**Clas1.cs**) және құрастыру жайлы ақпарат (**Assemblyinfo.cs**).

Егер төменгі жақтағы жарлықты шертіп, **Class View** ішкі бетіне көшсек, осы терезеден басқа да ақпаратты көруге болады. **Class View** ішкі бетінде программаға кіретін кластың барлық тізімдері тұрады.

Экранның басты бөлігін редактор терезесі алып тұр, онда программа мәтіні тұрады. Түйінді сөздер көк, комментарий – жасыл, қалған мәтін – қара түске боялып тұрады.

Сол жақ шетте құрылым символдары орналасқан, солардың минус таңбасы бар квадратын шертіп, код бөлігін жасырын ұстауға болады.

Сонда плюс таңбасы шығып, оны шертсек, программа бөлігі қайта көрінеді.

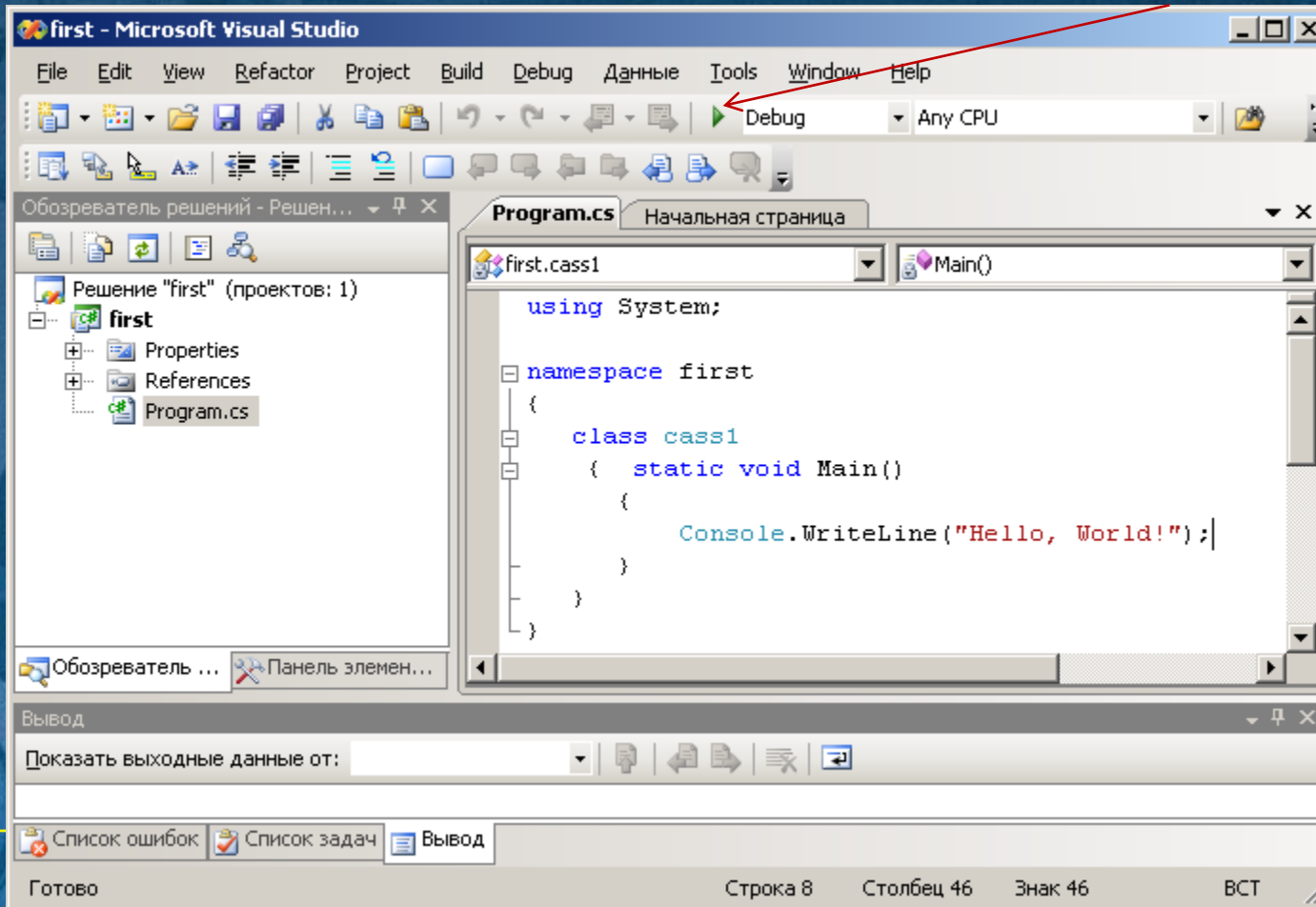
Енді редакциялау терезесіне программа мәтінін енгізу қажет. Біздің **Program.cs** атты алғашқы программамыздың коды төмендегідей болсын делік.

```
using System;  
namespace first  
{ class class1  
  { static void Main()  
    {  
      Console.WriteLine("Hello, World!");  
    }  
  }  
}
```



## 3.2. Программаны орындау

Программаны орындау үшін, төменгі суреттегі аспаптар тақтасының тілсызық бағыттауыш батырмасын (▶) шертеміз немесе **F5** (жөндеуді бастау) пернесін басамыз:



Программада қате болмаса, нәтижесі төмендегі суреттегідей болуы тиіс. Бірақ ол **нәтижелік терезе** немесе **тұтынушы терезесі** бірден жабылып қалады.

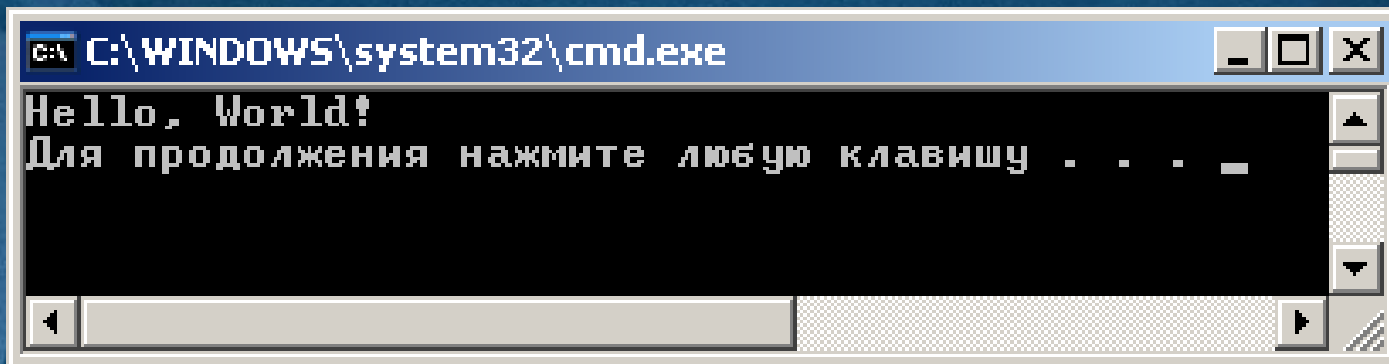
Ал егер қате болса, оларды түзету керек.



file:///C:/MyProjects/first/first/bin/Debug/first.EXE

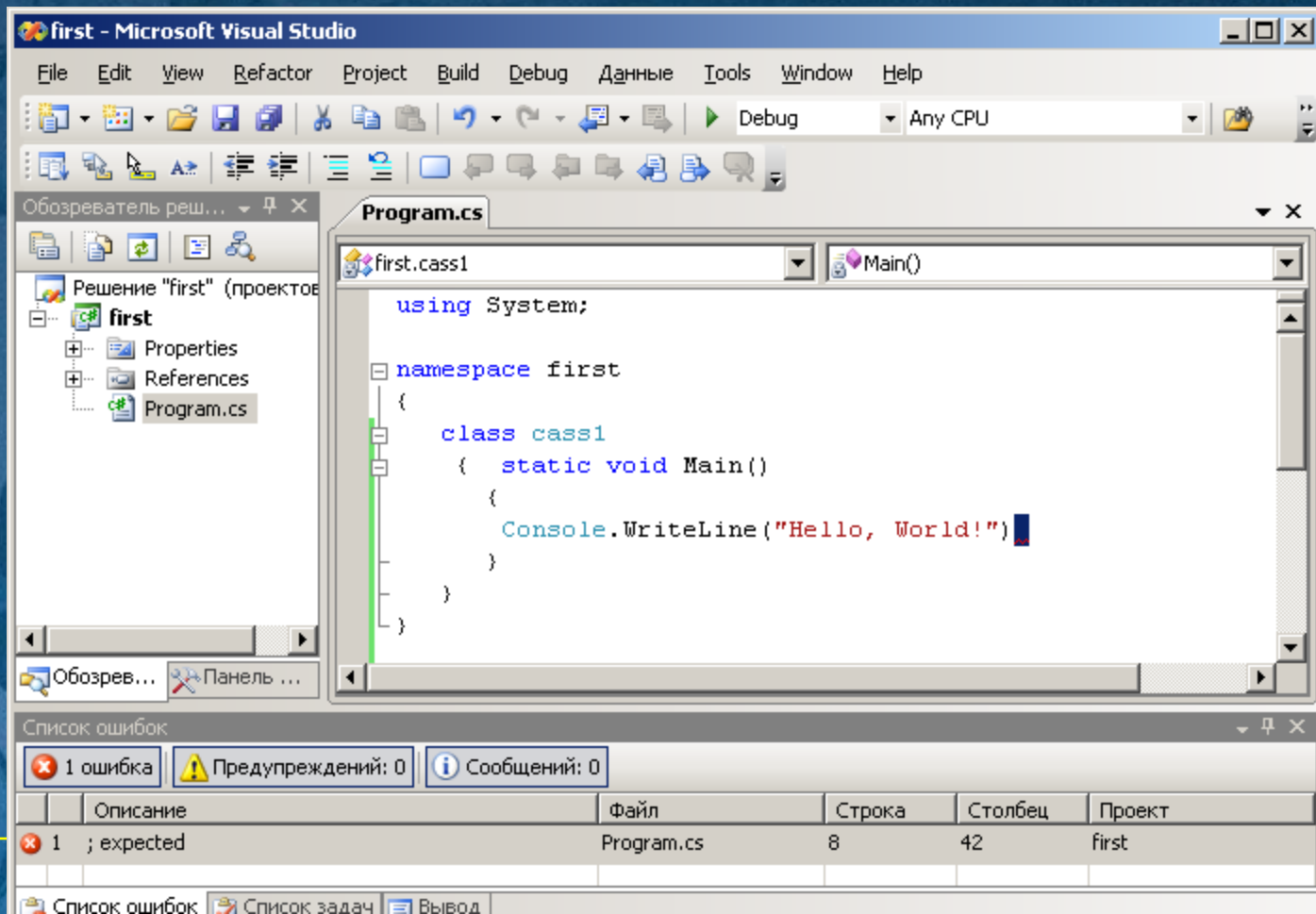
Hello, World!

Енді дайын программаны қайта орындап, нәтижесін көру үшін **Debug ► Start Without Debugging** меню командаларын орындау қажет немесе **Ctrl+F5** пернелерін басу керек. Сонда программа нәтижесі экранға шығады да, біздің оны толық көруімізге мүмкіндік береді.

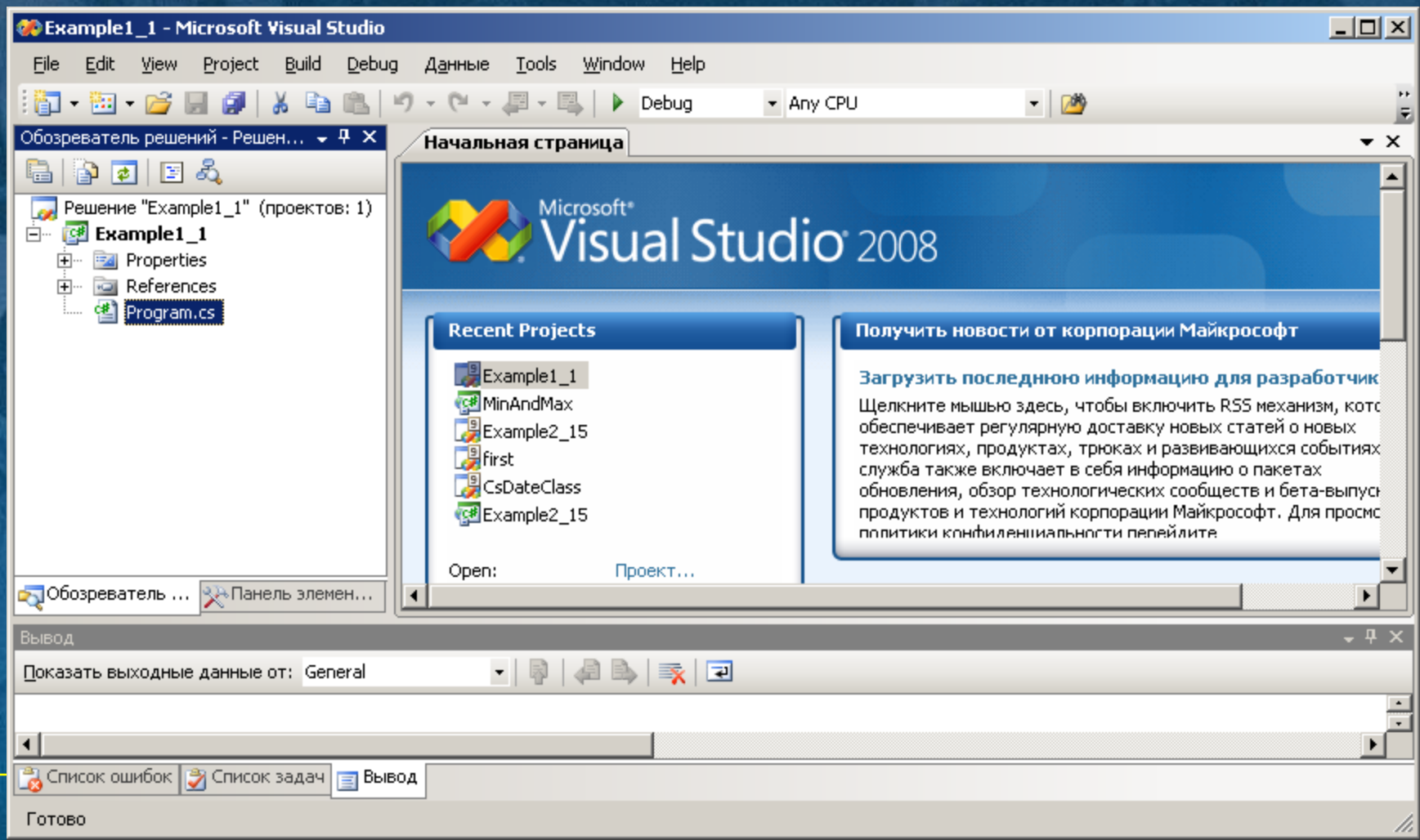
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window content displays the text 'Hello, World!' followed by a Russian instruction: 'Для продолжения нажмите любую клавишу . . .'. The window has standard Windows window controls (minimize, maximize, close) and a scroll bar at the bottom.

Жұмысты аяқтау үшін, **File ► Exit** командасын орындаймыз немесе **Visual C#** жүйесін жауып тастаймыз.

Егер соңғы нүктелі үтір таңбасы (;) болмай, қате шықса, онда төменгі жаққа қате жайлы мәлімет шығады. Сол мәліметті екі шертсек, қате жолда қара белгі пайда болады.



Visual C# ортасын жауып қайта ашсақ, төмендегі сурет шығады. Керекті жобаның атын шертсек, соның программасы экранға шығып, ары қарай жұмыс істеуді жалғастырамыз.



# Қосымша әдебиеттер

1. Биллиг В.А. Основы программирования на С#. –М.: Изд-во “Интернет-университет информационных технологий ИНТУИТ.ру”, 2006. -488 с.
2. Шилдт Г. Полный справочник по С#. –М.: Издательский дом “Вильямс”, 2004. -752 с.
3. Ватсон К. С#. –М.: Лори, 2004. -880 с.
4. Вирт Н. Алгоритмы + структуры данных = программы / Пер. с англ. -М.: Мир, 2006. - 406 с., ил.
5. Секунов Н. Самоучитель С#. Серия “Самоучитель”. -СПб.: БХВ-Петербург, 2001. - 576 с.
6. Microsoft Corporation. Разработка Web-приложений на Microsoft Visual Basic.NET и Microsoft Visual C#.NET Учебный курс MCAD/MCSD. -М.: Издательско-торговый дом “Русская Редакция”, 2003. -704 с.

## **C# тілінде программа жазуға мыналар керек:**

Microsoft .NET Framework 3.5 пакеті. C# программасын құруға қажеттінің барлығы бар платформа. Оны тегін көшіріп алуға болатын парақ:

<http://msdn.microsoft.com/ru-ru/netframework/aa569263.aspx>.

.Net Framework үшін C# тілінде программа теру, орындау ортасы керек, олар жұмыс істеу процесін барынша ыңғайлы етуге арналған. Сіздерге ол үшін Microsoft Visual C# 2008 Express Edition пакеті ұсынылады, оны тегін көшіріп алуға болатын Майкрософт фирмасының сайты:

<http://www.microsoft.com/rus/express/download/>. Егер сіз .NET Framework пакеті бар Visual C# 2008 Express алсаңыз, онда алдыңғы пунктті орындау қажет емес.

Өз компьютеріңізге алдын ала дайындалған мысалдар файлдарын көшіріп алғыңыз келсе, төмендегі парақтағы архивті алып ашыңыз:

<http://intuit.ru/software/school/cs2kids/examples.zip>,

# Интернеттегі С# ресурстары

- <http://msdn.microsoft.com/vstudio/express/visualCsharp>
- <http://msdn.microsoft.com/downloads>
- <http://ips.ifmo.ru>
- [www. ieee.org](http://www.ieee.org)
- <http://www.vek-publisher.com.ua>
- <http://www.helloworld.ru/>
- <http://www.emanual.ru>
- <http://www.sources1.ru>
- <http://www.rusdoc.ru>



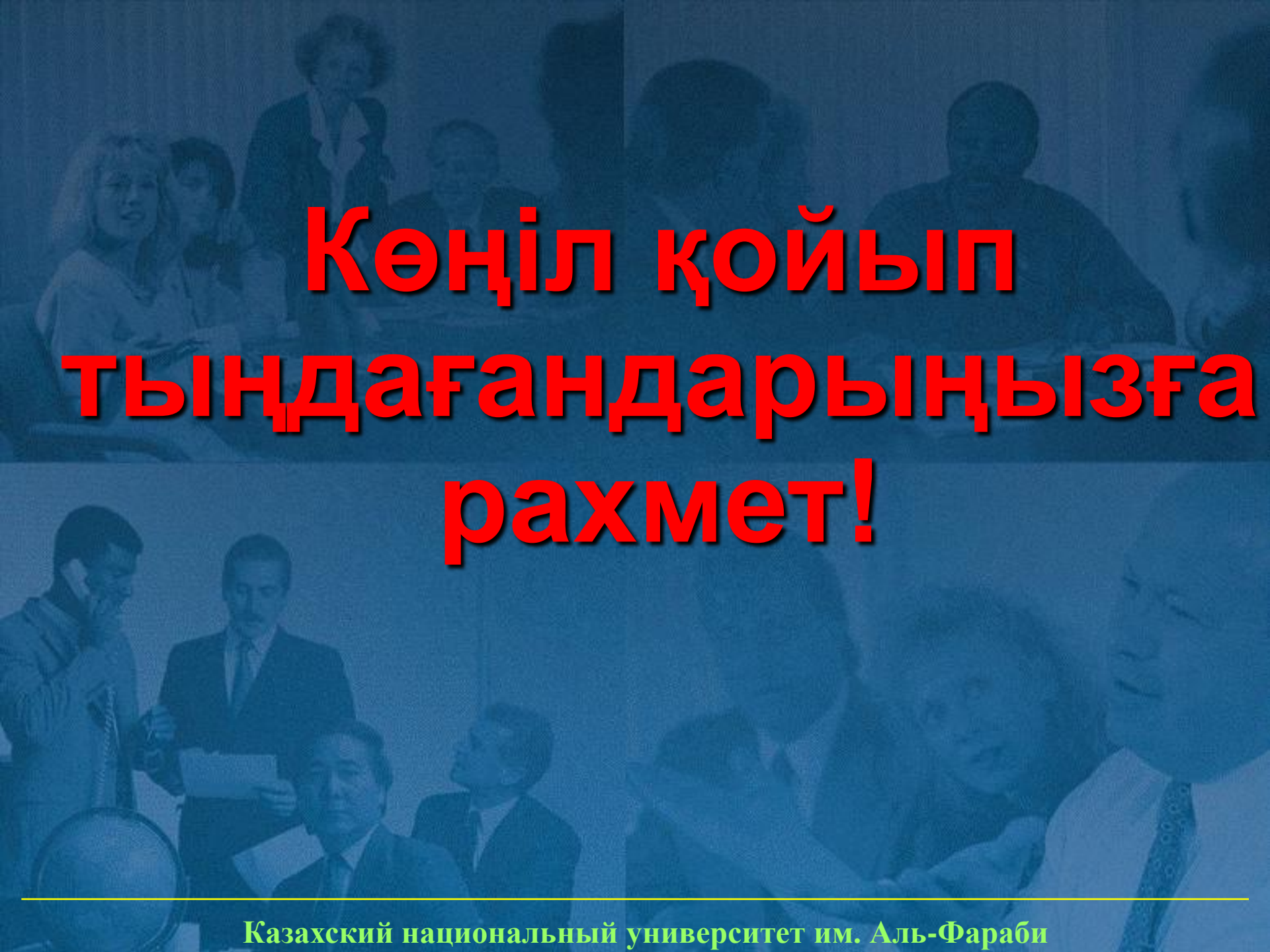
Енді өздерің бір қысқаша программа жазып шығындар, ол экранға студенттің аты-жөнін

Азат Омарұлы Ахметов

Азат  
Омарұлы  
Ахметов

Азат  
Омарұлы  
Ахметов

деп шығаратын болсын.



**Көңіл қойып  
тыңдағандарыңызға  
рахмет!**

C# тілінде енгізу-шығару операторлары жоқ, оның орнына стандартты объектілер қолданылады.

Консольмен жұмыс істеу үшін **Console** класы пайдаланылады, ол **System** атаулар кеңістігінде анықталған. Осы кластың **Write** және **WriteLine** тәсілдері мысалдарда қолданылған болатын.

Енді осы класқа тағы бір мысал қарастырайық.

```
using System;
namespace ConsoleApplication1
{ class Class1
  {static void Main()
    { int i = 3;
      double y = 4.12;
      decimal d = 600m;
      string s = "Азат";
      Console.WriteLine( "i = " + i );           // 1
      Console.WriteLine( "y = {0} \nd = {1}", y, d ); // 2
      Console.WriteLine( "s = " + s );           // 3
    }
  }
}
```

**Нәтижесі:**

```
i = 3
y = 4.12
d = 600
s = Азат
```

# Мәлімет шығару

Программада **int** типті **myInt** айнымалысы анықталған делік.

```
int myInt = 12345;
```

Бұл айнымалы былай шығарылады:

```
System.Console.Write("myInt = " + myInt);
```

Экрандағы нәтижесі былай болды:

```
myInt = 12345;
```

Мұнда ешқандай безендіру форматы жоқ.

Берілген формула бойынша есептеу мысалы:

$$y = \sqrt{\pi \cdot x} - e^{0.2\sqrt{a}} + 2\text{tg}2\alpha + 1,6 \cdot 10^3 \cdot \log_{10} x^2$$

Формулаға екі бастапқы мәндер –  $x$  және  $\alpha$  енгізіледі. Олардың типі айтылмағандықтан, `double` типін таңдап алайық. Енді программасын құрайық.

**// Listing3\_14.cs программасы**

```
using System;
```

```
using System.IO;
```

```
namespace ConsoleApplication1
```

```
{ class Class1
```

```
{ static void Main()
```

```
{ Console.Write(" x engiz: " );
```

```
string buf = Console.ReadLine ();
```

```
double x = Convert.ToDouble( buf );
```

```
Console.Write(" alfa engiz: " );
```

```
string buf = Console.ReadLine ();
```

```
double a = double.Parse( buf );
```

```
double y = Math.Sqrt(Math.PI*x)-Math.Exp(0.2 *  
    Math.Sqrt(a)) + 2+Math.Tan(2*a) +  
    1.6e3*Math.Log10(Math.Pow(x,2));  
Console.WriteLine(" x={0} ; alfa={1} ", x, a);  
Console.WriteLine(" y = " +y);  
}  
}  
}
```

**Нәтижесі:**

**x engiz: 3,2 ↵**

**alfa engiz: 0,5 ↵**

**y = 1622,05609027627**

## Енгізу тәсілдеріне мысал:

```
// Listing3_10.cs программасы
using System;
namespace ConsoleApplication1
{
    class Class1
    { static void Main()
      { Console.WriteLine( "Coz tirkesin engiz : " );
        string s = Console.ReadLine();      // 1
        Console.WriteLine( "s = " + s );

        Console.WriteLine( "Symbol engiz : " );
        char c = (char)Console.Read();      // 2
        Console.ReadLine( );              // 3
        Console.WriteLine( "c = " + c );
        string buf; // тіркес - сан енгізілетін буфер
        Console.WriteLine( "Butin san engiz : " );
        buf = Console.ReadLine( );
        int i = Convert.ToInt32(buf);      // 4
        Console.WriteLine( i );
      }
    }
}
```



```
Console.WriteLine( "Nakty san engiz : " );  
buf = Console.ReadLine( );  
double x = Convert.ToDouble ( buf );           // 5  
Console.WriteLine( x );
```

```
Console.WriteLine( "Nakty san engiz : " );  
buf = Console.ReadLine( );  
double y = double.Parse( buf );              // 6  
Console.WriteLine( y );
```

```
Console.WriteLine( "Nakty san engiz : " );  
buf = Console.ReadLine( );  
decimal z = decimal.Parse( buf );  
// 7  
Console.WriteLine( z );  
}  
}  
}
```