

ФИЗИКАЛЫҚ ПРОЦЕССТЕРДІ КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ КУРСЫ

Бұл курс ҚҰУ - нің физика-техникалық факультетінің студенттеріне арналған. Лектор - Хасанов Манас Кабылтайевич.

Тел: +7 (707) 612 66 92
Почта: manas_hasanov@mail.ru

I-бөлім

C-SHARP АҚПАРАТТЫҚ БАҒДАРЛАМАЛАУ ТІЛІ

§1. Компьютерлік модельдеу ұғымы. C-sharp ақпараттық бағдарламалау тілінің «Visual studio» платформасын орнату. C-sharp бағдарламалау тілінің негіздері. Айнымалылар, негізгі типтер, шартты операторлар.

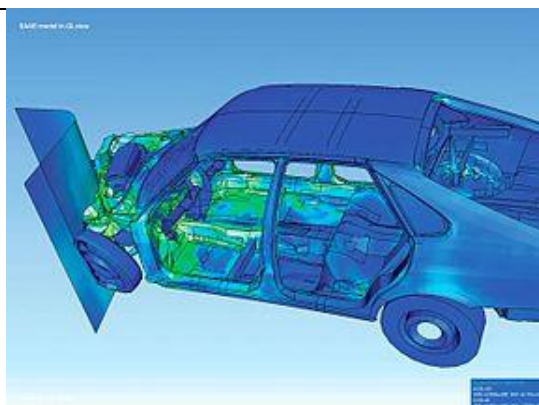
Компьютерлік модельдеу ұғымы.

Компьютерлік модель немесе **сандық модель** деп жеке компьютерде немесе суперкомпьютерде орындалатын, белгілі нысананы (объект) және оның қасиеттерін алгоритмдер арқылы сипаттайтын компьютерлік бағдарламаны айтады.

Компьютерлік модельдеу бізге мынадай мүмкіндіктер береді:

- Зерттеу нысаналар жиының арттырады - яғни шың жағдайда жүзеге асыру мүмкін емес нысаналарды зерттеуге.
- Кез келген нысаналарды визуализациялауға. Соның ішінде абстрактты нысаналарды да.
- Құбылыстар мен процестердің даму динамикасын зерттеуге.
- Уақытты басқаруға (яғни тездету, артқа айналдыру, т.б.).
- Нысана модельін бірнеше рет сынауға.(яғни калпына келтіріп қайта сынауға).
- Нысананың әр түрлі сипаттамаларын болжауда.
- Нысананы дайындамай тұрып оның ең тиімді конструкцияларын табуда.
- Адам және қоршаған ортаға зақым тигізбей эксперименттер қоюда.

«**SOLIDWORKS**» бағдарламасы арқылы автомобильдің соқтығысу процессін компьютерлік модельдеу.



Сонымен қазіргі таңда физикалық процесстерді модельдейтін көптеген дайын программалар «пакеттер» бар. Олар: Механика саласындағы - «**Solidworks**», Ядролық реакторларды модельдейтін - «**MCNP**», Элементар бөлшектердің әсерлесуін модельдейтін - «**GEANT-4**», және т.б.

Алайда, бұл программалар физикадағы барлық процесстерді қамти алмайды және де бұл программалар қолданылатын алгоритмдер көп жағдайда «жабық» қолданыста болады. Сол себепті біздің курсымыз осындай дайын программаларды қолдануға емес, физикалық процестерге қатысты алгоритмдерді құрып үйренуге бағытталған.

Компьютерлік алгоритмдерді әр түрлі ақпараттық бағдарламалау тілдерінде жазуға болады. Солардың бірі **C-sharp** ақпараттық бағдарламалау тілі. Біз осы курс барысында **C-sharp** ақпараттық бағдарламалау тілін қолданып, физикаға қатысы бар алгоритмдерді қарастырамыз.

C-sharp ақпараттық бағдарламалау тілін қолданып программа жазу үшін бізге осы тілдің платформасын орнату қажет. Бұл платформаның аты «**Microsoft Visual studio**» деп аталады. «**Microsoft Visual studio**» платформасы **C-sharp** ақпараттық бағдарламалау тілін қолданып программа жазуға арналған программа болып табылады.

Сонымен, келесі қадам осы «**Microsoft Visual studio**» платформасын өзіміздің компьютерге орнату болады.

«**Microsoft Visual studio**» платформасын орнату.

1. Алдымен «**Microsoft**» компаниясының ресми сайтына барып, тіркеліңіз.

Сілтеме: <https://visualstudio.microsoft.com/ru/downloads/?rr=https%3A%2F%2Fwww.google.by%2F>

2. «**Microsoft Visual studio**» платформасының «**Community**» нұсқасын (яғни студенттерге арналған ақысыз түрі) таңдаңыз.

Microsoft | Visual Studio Продукты Скачать Marketplace Больше Бесплатная версия Visual Studio Все продукты Microsoft

Скачать

Windows macOS

Visual Studio 2019

Полнофункциональная интегрированная среда разработки (IDE) для приложений Android, iOS, Windows, веб- и облачных приложений

Версия: 16.2
[Заметки о выпуске](#)

Community Эффективная среда IDE предоставляется бесплатно для студентов, участников создания открытого исходного кода и отдельных пользователей	Professional Профессиональная интегрированная среда разработки, оптимальная для небольших команд	Enterprise Масштабируемое, комплексное решение для команд любого размера
Загружается бесплатно ↓	Бесплатная пробная версия ↓	Бесплатная пробная версия ↓
Скачать	Скачать	Скачать

Visual Studio Code

Быстрый, бесплатный редактор с открытым исходным кодом, адаптирующийся под ваши потребности

[Заметки о выпуске](#)

[Загружается бесплатно ↓](#)

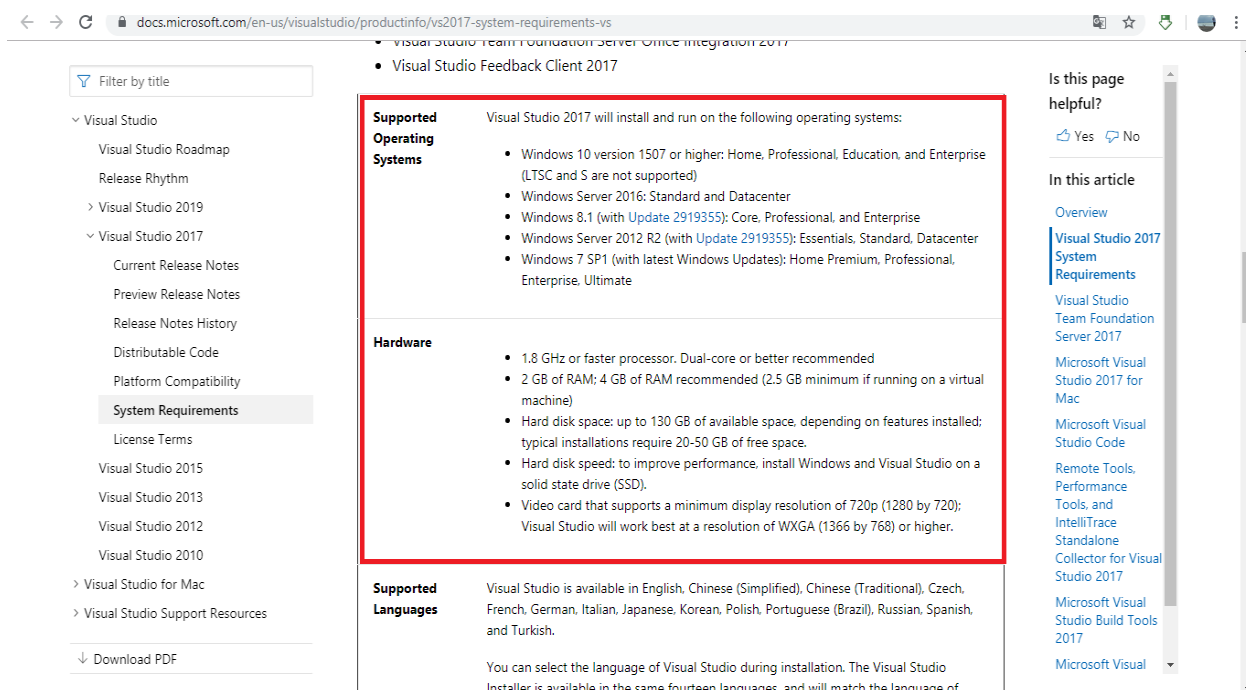
Загружая и используя Visual Studio Code, вы принимаете условия

[Feedback](#)

3. Жүктеу және орнату алдында «*Microsoft Visual studio*» платформасының өзіңіздің компьютеріңізбен үйлесімділігін тексерген жөн.

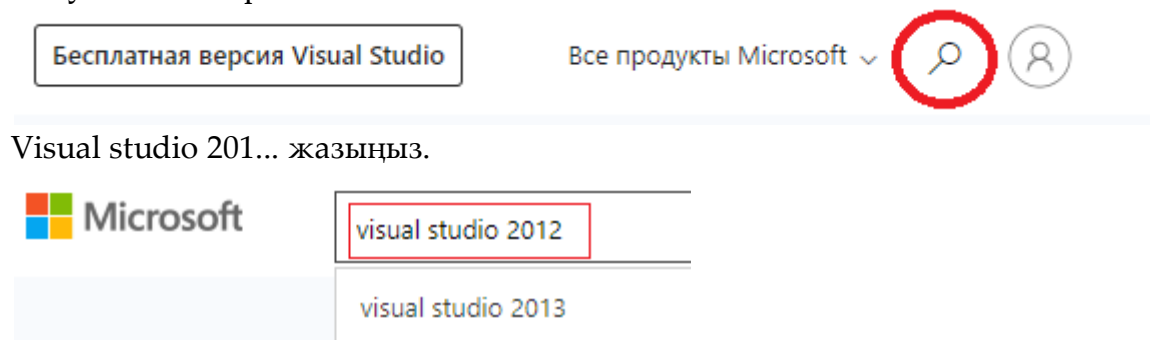
Үйлесімділігі тексеру сілтемесі:

<https://docs.microsoft.com/en-us/visualstudio/productinfo/vs2017-system-requirements-vs>

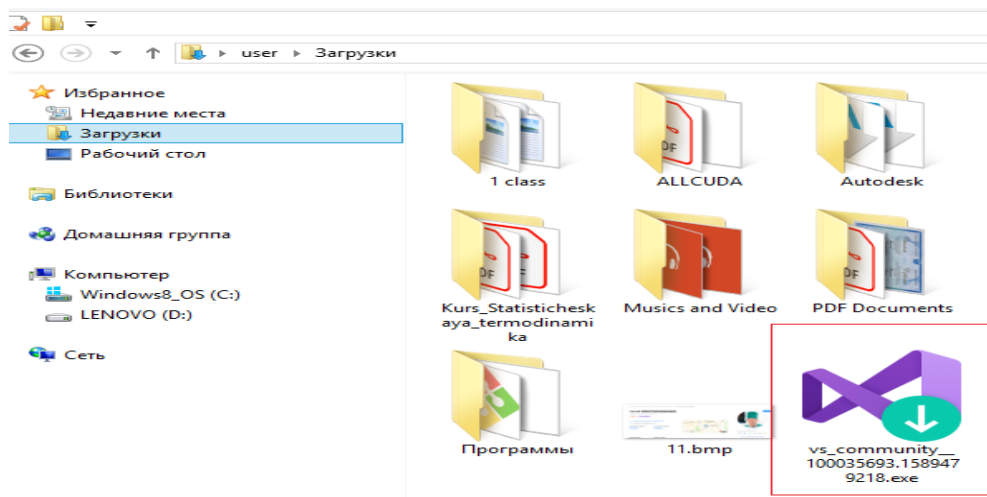


4. Егерде әлде бір себептермен сіздің компьютеріңізбен платформа арасында үйлеспеушілік туындаса (компьютер ескі немесе ескі windows орнатылған себебінен), онда төменде көрсетілгендей платформаның ескі түрін таңдасаңыз болады. Мысалы: *Visual studio-12, Visual studio-13 немесе Visual studio-15 пен Visual studio-17*.

Іздеу дегенге барыңыз

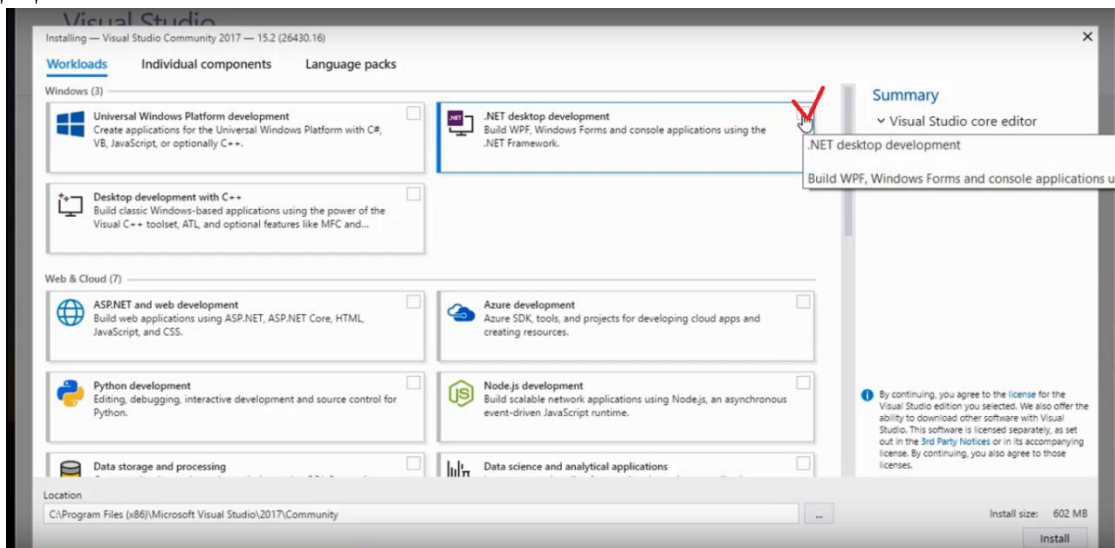


5. «Скачать» немесе «Download» басыңыз. Сол кезде жүктемеде мынадай файл болуы керек. (онлайн жүктеу жүргізетін программа)

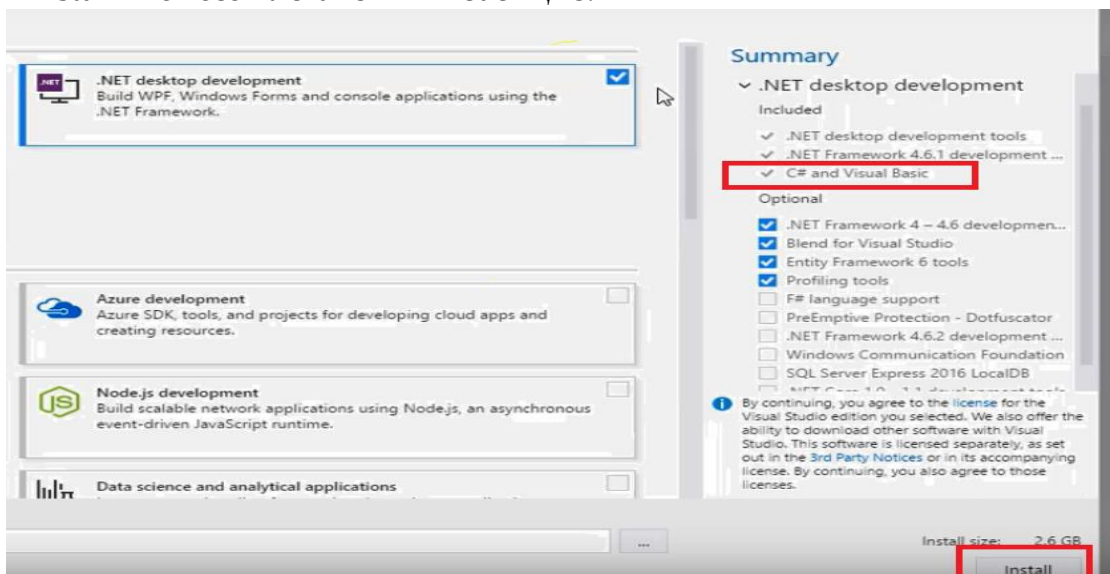


6. Интернет байланысы орнықты екенің тексеріңіз де жүктелген файлды атқаруға жіберіңіз.

7. Біраз уақыт күтіңіз. Мына сурет пайда болғанда «.NETdesktopdevelopment» таңдаңыз.



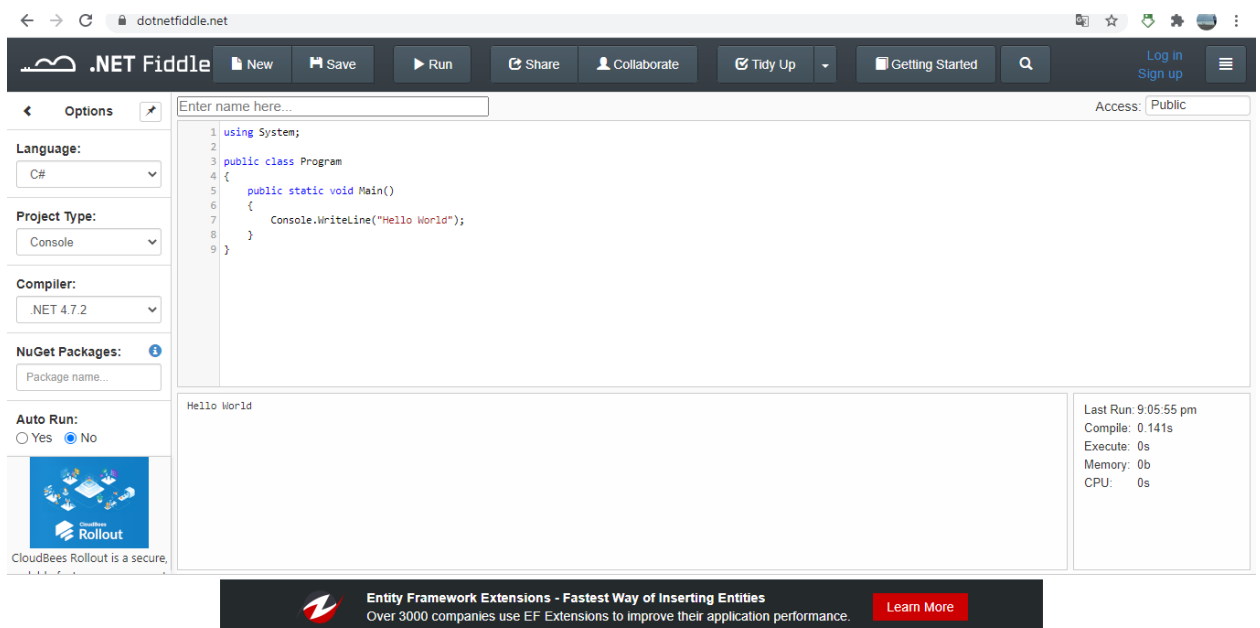
8. «Install» немесе «Установить» басыңыз.



9. Қалай орнату керек екені туралы видео мына сілтеме бойынша:

<https://www.youtube.com/watch?v=L0kjB17fSY>

10. Алайда c-sharp ақпараттық барғаарлама тіліндегі программаны тек *Visual studio* ғана емес, сондай ақ онлайн редакторларда да жазуға болады.



Алайда бұл жерде кей библиотекалар жасамай қалуы мүмкін. Сол себепті онлайн компиляторды қолдану алдағы лекцияларда қиындықтар туғызуы мүмкін.

C-sharp бағдарламалау тілінің негіздері.

Айнымалылар.

Программалау барысында әлде бір информацияны жадыда сақтау мақсатында айнымалыларды қолданады. Кез келген айнымалының типі, аты және мәні болады. Айнымалының типі осы айнымалы қандай ақпаратты сақтай алатынын анықтайды.

Мысалы:

int x = 5;

Бұл жерде `x`-атты айнымалы келтірілген. Оның типі `int` (бұл тип тек - 2147483648 до 2147483647 аралығында жататын бүтін сандарды қабылдай алады). Бұл айнымалыға `5` деген бүтін сан меншіктеліңген.

Сонымен, кез келген айнымалыны еңгізер алдында оның типін анықтау қажет. Яғни

Айнымалы типі айнымалы аты;

Ал айнымалының мәнін кейін меншіктесек те болады.

Мысалы:

```
int y;  
код ...код  
.....  
y = 7;
```

Назар аударыңыз: айнымалының аты еңгізілгеннен кейін немесе айнымалыға мәң меншіктелгеннен кейін міндетті түрде үтір нүкте қойылады.

Айнымалылардың айрықша бір ерекшелігі бар. Бұл программа барысында олардың мәндерін бірнеше рет өзгертуге болады. Мысалы, айнымалыны анықтайтын, оның мәнін өзгертетін және консольға шығаратын шағын программа құрайық:

```
using System;  
namespace App  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            // x_ainymaly атты айнымалыны анықтайық та, оған 10 деген мәң меншіктейік.  
            int x_ainymaly = 10;  
            // енді осы айнымалыны консольге шығарайық.  
            Console.WriteLine(x_ainymaly);    //10 деген жазу шығады  
  
            // x_ainymaly атты айнымалының мәнін 20 деген мәңге өзгертейік.  
            x_ainymaly = 20;  
  
            // x_ainymaly тағы бір рет консольге шығарайық.  
            Console.WriteLine(x_ainymaly);    //20 деген жазу шығады  
  
            Console.Read(); // Бұл консоль жабылып қалмауы үшін қажет. Алайда  
            //кейбір редакторлар бұны қажет етпейді (мысалы кейбір онлайн редакторларда)  
        }  
    }  
}
```

Қарапайым типтер.

Көптеген басқа да программалау тілдеріндегідей C# тілінде де ең қарапайым типтер жиыны бар. Олар мыналар:

1. `bool`: `true` немесе `false` мәндерін қабылдайды.

Мысалы: `bool IsActive = true;`

Бұл жерде аты `IsActive` деп аталатын мәні `true` тең `bool` типті айнымалы анықталған.

2. `byte`: `0` мен `255` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 1 байт.

Мысалы: `byte b = 13;`

Бұл жерде аты `b` деп аталатын мәні `13` тең `byte` типті айнымалы анықталған.

3. `sbyte`: `-128` мен `127` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 1 байт.

Мысалы: `sbyte sb = -12;`

Бұл жерде аты `sb` деп аталатын мәні `-12` тең `sbyte` типті айнымалы анықталған.

4. `short`: `-32768` мен `32767` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 2 байт.

Мысалы: `short sh = 777;`

Бұл жерде аты `sh` деп аталатын мәні `777` тең `short` типті айнымалы анықталған.

5. `int`: `-2147483648` мен `2147483647` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 4 байт.

Мысалы: `int a = 10;`

Бұл жерде аты `a` деп аталатын мәні `10` тең `int` типті айнымалы анықталған.

Мысалы: `int b = 0b101;` //Бұл жерде сан бинарды түрде меңшіктелген.

Бұл жерде аты `b` деп аталатын мәні `5` тең `int` типті айнымалы анықталған.

Мысалы: `int c = 0xFF;` //Бұл жерде сан оңалтылық сандар жүйесінде меңшіктелген.

Бұл жерде аты `c` деп аталатын мәні `255` тең `int` типті айнымалы анықталған.

6. `uint`: `0` мен `4294967295` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 4 байт.

Мысалы: `uint ui = 15;`

Бұл жерде аты `ui` деп аталатын мәні `15` тең `uint` типті айнымалы анықталған.

7. `long`: `-9 223 372 036 854 775 808` мен `9 223 372 036 854 775 807` аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 8 байт.

Мысалы: long lg = 14;

Бұл жерде аты lg деп аталатын мәні 14 тең long типті айнымалы анықталған.

8. ulong: 0 мен 18 446 744 073 709 551 615 аралығындағы бүтін сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 8 байт.

Мысалы: ulong ul = 20;

Бұл жерде аты ul деп аталатын мәні 20 тең ulong типті айнымалы анықталған.

9. float: $-3.4 \cdot 10^{38}$ мен $3.4 \cdot 10^{38}$ аралығындағы бүтін және бөлшек сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 4 байт. Мантиссасы: 6 - 7.

Мысалы: float f = 3.14;

Бұл жерде аты f деп аталатын мәні 3.14 тең float типті айнымалы анықталған.

10. double: $-5.0 \cdot 10^{324}$ мен $1.7 \cdot 10^{308}$ аралығындағы бүтін және бөлшек сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 8 байт. Мантиссасы: 15 - 16.

Мысалы: double d = 5.72;

Бұл жерде аты d деп аталатын мәні 5.72 тең double типті айнымалы анықталған.

11. decimal: $\pm 1.0 \cdot 10^{-28}$ мен $\pm 7.9228 \cdot 10^{28}$ аралығындағы бүтін және бөлшек сандарды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 16 байт. Мантиссасы: 28-29.

Мысалы: decimal dec = 5.0000072;

Бұл жерде аты dec деп аталатын мәні 5.0000072 тең decimal типті айнымалы анықталған.

12. char: Unicode кодировкасындағы бір ғана символды мәң ретінде меңшіктеуге болады. Жадыда алатын көлемі 2 байт.

Мысалы: char ch = 'A';

Бұл жерде аты ch деп аталатын мәні 'A' тең char типті айнымалы анықталған.

13. string: Unicode кодировкасындағы бірнеше символдарды мәң ретінде меңшіктеуге болады.

Мысалы: string str = "This is a string";

Бұл жерде аты str деп аталатын мәні "This is a string" тең string типті айнымалы анықталған.

14. Object: Кез-келген типтегі мәнді сақтай алады. Жадыда алатын көлемі 4 байт 32 разрядты платформада және 8 байт 64 разрядты платформада. Барлық басқа типтер мен кластар үшін негіз тип болып табылады.

Мысалы: object obj1 = 22;

Бұл жерде аты obj1 деп аталатын мәні 22 тең object типті айнымалы анықталған.

Мысалы: `object obj2 = 3.14;`

Бұл жерде аты `obj2` деп аталатын мәні 3.14 тең `object` типті айнымалы анықталған.

Мысалы: `object obj3 = "Hello";`

Бұл жерде аты `obj3` деп аталатын мәні "Hello" тең `object` типті айнымалы анықталған.

Мысал ретінде әр түрлі типті бірнеше айнымалыларды анықтап, олардың мәндерін консольға шығарайық.

```
using System;
namespace App
{
    class Program
    {
        static void Main(string[] args)
        {
            string name = "Tom";
            int age = 33;
            bool isEmployed = false;
            double weight = 78.65;

            Console.WriteLine("Аты:" + name);
            Console.WriteLine("Жасы:" + age);
            Console.WriteLine("Салмағы:" + weight);
            Console.WriteLine("Жұмыскер ме?:" + isEmployed);

            Console.ReadLine();
        }
    }
}
```

Шартты операторлар.

if / else конструкциясы.

`if / else` конструкциясы белгілі бір шарттың ақиқаттығын тексереді және тест нәтижесіне байланысты белгілі бір кодты орындайды.

Мысал.

```
using System;
namespace App
{
    class Program
    {
        static void Main(string[] args)
        {
            int num_1 = 8;
            int num_2 = 6;
            if(num_1 > num_2){
                Console.WriteLine(num_1 + "more than" + num_2);
            }
            Console.ReadLine();
        }
    }
}
```

Шарт **if** деген кілт сөзінен кейін жақша ішінде жазылады: **if**(num_1 > num_2)

Егер жақша ішіндегі шарт орындалатын болса, онда фигурлы жақша ішінде жазылған код орындалуға жіберіледі.

```
{  
    Console.WriteLine(num_1 + "more than" + num_2);  
}
```

Егер жақша ішіндегі шарт орындалмайтын болса, онда фигурлы жақша ішінде жазылған код орындалуға жіберілмейді.

Біздің қарастырған мысалда num_1 > num_2 болып тұр (шынымен 8>6), сол себепті фигурлы жақша ішінде жазылған код

```
Console.WriteLine(num_1 + "more than" + num_2);
```

орындалуға жіберіледі.

Ал егер де шарт орындалмаған жағдайда біз басқа бір код қолданылғанын қаласақ ше? Бұл жағдайда **else** операторынан кейін тағы бір блокты қосып, сол блоктың ішінде қажетті кодты жазсақ болады:

Мысал.

```
using System;  
namespace App  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            int num_1 = 8;  
            int num_2 = 6;  
  
            if(num_1 > num_2){  
                Console.WriteLine(num_1 + "more than" + num_2);  
            }  
            else  
            {  
                Console.WriteLine(num_1 + "less than" + num_2);  
            }  
  
            Console.ReadLine();  
        }  
    }  
}
```

switch конструкциясы.

switch конструкциясы мағынасы жағынан **if/else** конструкциясына ұқсайды. Алайда, ол бірден бірнеше шартты өңдеуге мүмкіндік береді:

Мысал.

```
using System;
namespace App
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Press Y or N");
            //төменде типі string болатын selection атты айнымалыға
            //консолдан еңгізілген жазу меңшіктеледі.
            string selection = Console.ReadLine();

            //switch операторы selection айнымалысын тексеріске қосады
            switch (selection)
            {
                //төменде егер selection "Y" қа тең болса деп тұр
                case "Y":
                    //жоғарыдағы шарт орындалса онда мына кодты орында деп тұр
                    Console.WriteLine( "Y is pressed " );
                    break; //сонымен тоқтап switch-тан шық деп тұр
                case "N":
                    Console.WriteLine( "N is pressed " );
                    break;

                //Егер жоғарыдағы барлық шарттар орындалмаса онда
                default:
                    Console.WriteLine( "Another symbol is pressed" );
                    break;
            }
            Console.ReadLine();
        }
    }
}
```