

Визуал програмчлал

J.IT203

Лекц-1

Сурах бичиг

- C# .NET2009 Г.Гантуяа, М.Отгонбаяр
- Fundamentals of Computer Programming with C# 2013 SvedlinNakov, VeselinKolev & Co.
- Programming Microsoft Windows with C# 2002 Charles Petzold
- Student guide Revision 4.0 2011 Robert J. Oberg and Dana Wyatt

Хичээлийн зорилго

- .NETframework ийн тухай ойлголтууд, технологийн шийдэл, C# хэлийг ашиглан програмчлах талаар судална.

Дүгнэх журам

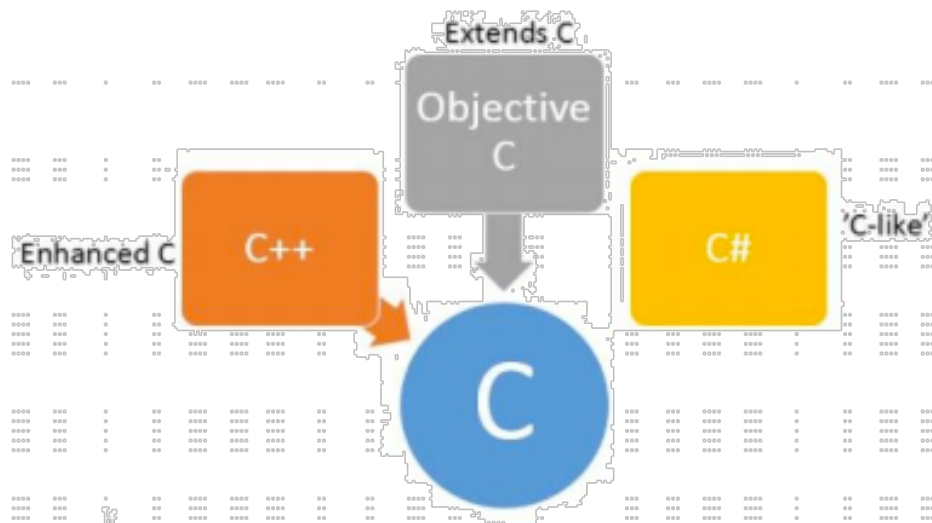
№	Үнэлгээ	Оноо
1	Явцын шалгалт - 2 удаа	20
2	Бие даалт - 2 удаа	25
3	Лаборатори	20
4	Ирц / идэвхи	10
Нийт		70

Агуулга

- .NET Framework
- Framework Class Library
- C# хэлний суурь ойлголтууд
 - Түлхүүр үгс
 - Анхдагч өгөгдлийн төрөл
 - Үндсэн үйлдлүүд
 - Операторууд
 - Хувьсагч

С төрлийн хэлнүүд

- C 1972
- C++ 1983
- Java 1995
- C# 2000



C төрлийн хэлнүүд (Жишээ код)

Java

```
public class Hello {  
    public static void main (String args[]) {  
        System.out.println ("Hello World");  
    }  
}
```

```
public class Hello  
{  
    static void Main()  
    {  
        System.Console.WriteLine("Hello World");  
    }  
}
```

C#

C++

```
#include <iostream>
```

```
int main()  
{  
    std::cout << "Hello World!";  
    return 0;  
}
```

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

```
int main() {  
    /* my first program in C */  
    printf("Hello, World! \n");  
  
    return 0;  
}
```

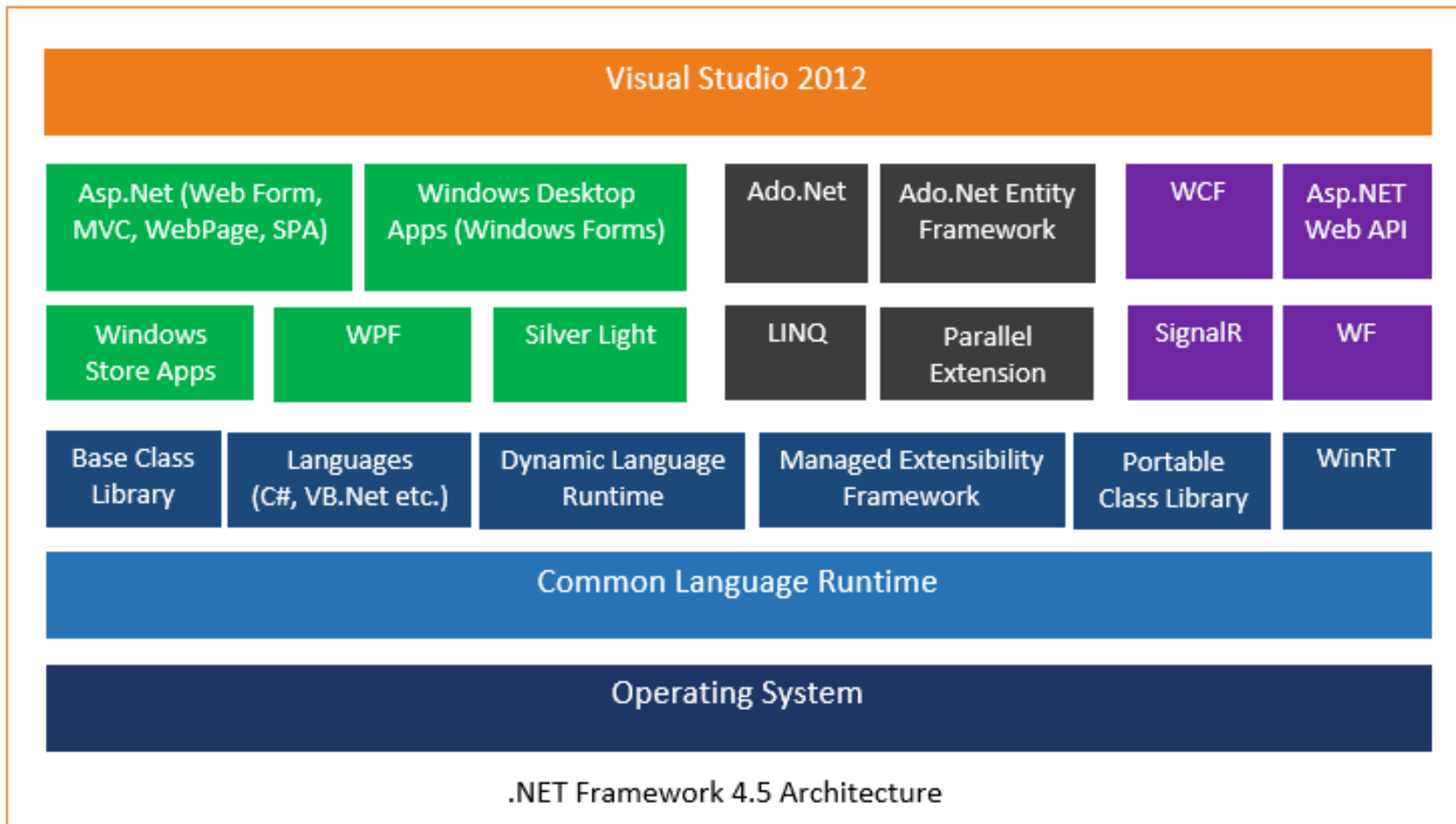
C

.NET Framework тухай

.NET Framework нь интернэт-Вэб програм, Дэсктоп аппликейшн болон мобайл програмуудыг хөгжүүлэх, ажиллуулхад зохиулагдсан програмчлалын орчин юм.

.NET Framework-ын зорилго нь Ямар хэл хэрэглэж байгаа нь гол биш, хамгийн гол нь Интернэт болон Дэсктоп аппликейшны хурдан, хялбар хөгжүүлэх юм.

.NET Framework бүтэц



.NET Framework

- Framework Class Library - .NET програмуудад ашиглагдах классын сан. Үүнд Өгөгдлийн сан, вэб болон графиктай ажиллах олон тооны класс багтана. .NET Framework дээр ажиллах бүх програмууд энэ дундын санг ашиглана.
- Common Language Runtime - кодын ажиллагааг хянах болоод ажиллуулахтай холбоотой бүх үүргийг хүлээнэ: Хөрвүүлэлт, санах ойн хуваарилалт, нууцлал, төрөл ашиглалт гэх мэт

Framework Class Library

- .NET програмд ашиглагдах классын сан ба эдгээр сангууд нь үүрэг зориулалтынхаа дагуу **нэймспэйс (namespace)** гэж нэрлэгдэх бүлгүүдэд хуваагддаг. Жишээ нь ӨС-тай ажиллах классууд нь `System.Data.SqlClient` нэймспэйст байрлана.

Нэйmspэйс /namespace/

- Нэйmspэйс нь тодорхой зүйлсийг нэгэн нэрийн дор хамаатуулан утга учиртай нэрлэх боломж олгоно.
- Framework-н бүх класс Нэйmspэйсэд байрлана.
- Энэ нь эдгээр классуудад хандаж, үр дүнтэйгээр хэрэглэхэд зориулагдана.

Нэйmspэйс /namespace/

Нэйmspэйс	Тайлбар
	Бүх програмуудад ашиглагдах үндсэн төрлүүд байрлана. Мөн атрибут классууд, математикийн сан зэрэг багтана.
System	
System. Data	Өгөгдлийн сантай харьцах класс
System.IO	Файл болон өгөгдлийн урсгалууд
System.NET	Сүлжээний үйлдлүүд
System. Security	Хамгаалалтыг удирдах класс
System. Threading	Асинхрон програмчлал
System. Web	ASP.NET болон интернэттэй холбоотой класс
System. Windows. Forms	Цонхтой програм үүсгэх класс

Түлхүүр үг

- C# хэлэнд дараах хоёр төрлийн түлхүүр үгнүүд ашиглагддаг. Үүнд:
 - 79 reserved keywords
 - 25 contextual keywords

Contextual Keywords
add
var
dynamic
global
set
value

```
class TimePeriod
{
    private double _seconds;
    public double Seconds
    {
        set { _seconds = value; }
    }
}
```

Reserved

Contextual

Түлхүүр үгнүүд

Literal Keywords

null

false

true

value

void

Access keywords

base

this

Namespace Keywords

using

. operator

:: operator

extern alias

Method Parameter Keywords

params

ref

out

Operator Keywords

as

await

is

new

sizeof

typeof

stackalloc

checked

unchecked

Statement Keywords

if

else

switch

case

do

for

foreach

in

while

break

continue

default

goto

return

yield

throw

Type keywords

bool

byte

char

class

decimal

double

enum

float

int

long

sbyte

short

string

struct

uint

ulong

ushort

Modifier keywords

abstract

async

const

event

extern

new

override

partial

readonly

sealed

static

unsafe

virtual

volatile

Анхдагч өгөгдлийн төрлүүд

Short Name	.Net Class	Type	Width	Range (bits)
byte	Byte	Unsigned integer	8	0 to 255
sbyte	SByte	Signed integer	8	-128 to 127
int	Int32	Signed integer	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Unsigned integer	32	0 to 4294967295
short	Int16	Signed integer	16	-32,768 to 32767
ushort	UInt16	Unsigned integer	16	0 to 65535
Long	Int64	Signed integer	64	-9223372036854775808 to 9223372036854775807
ulong	UInt64	Unsigned integer	64	0 to 18446744073709551615
float	Single	Single-precision floating point type	32	-3.402823e38 to 3.402823e38
double	Double	Double-precision floating point type	64	-1.79769313486232e308 to 1.79769313486232e308
char	Char	A single Unicode character	16	Unicode symbols used in text
bool	Boolean	Logical Boolean type	8	True or false
object	Object	Base type of all other types		
string	String	A sequence of characters		
decimal	Decimal	Precise fractional or integral type that can represent decimal numbers with 29 significant digits	128	$\pm 1.0 \times 10e-28$ to $\pm 7.9 \times 10e28$

Үндсэн үйлдүүд

- Арифмерик үйлдлүүд
- Логик үйлдлүүд
- Жиших үйлдлүүд
- Бит үйлдлүүд

Үндсэн үйлдлүүд

Арифметик үйлдэл	Тайлбар	Жишээ
+	Нэмэх операнд	$A+B=30$
-	Хасах	$A-B=-10$
*	Үржүүлэх	$A*B=200$
/	Хуваах	$B/A=2$
%	Үлдэгдэл	$B\%A=0$
++	Бүхэл утгыг нэгээр нэмэгдүүлэх	$A++=11$
--	Бүхэл утгыг нэгээр хорогдуулах	$A--=9$

Үндсэн үйлдлүүд

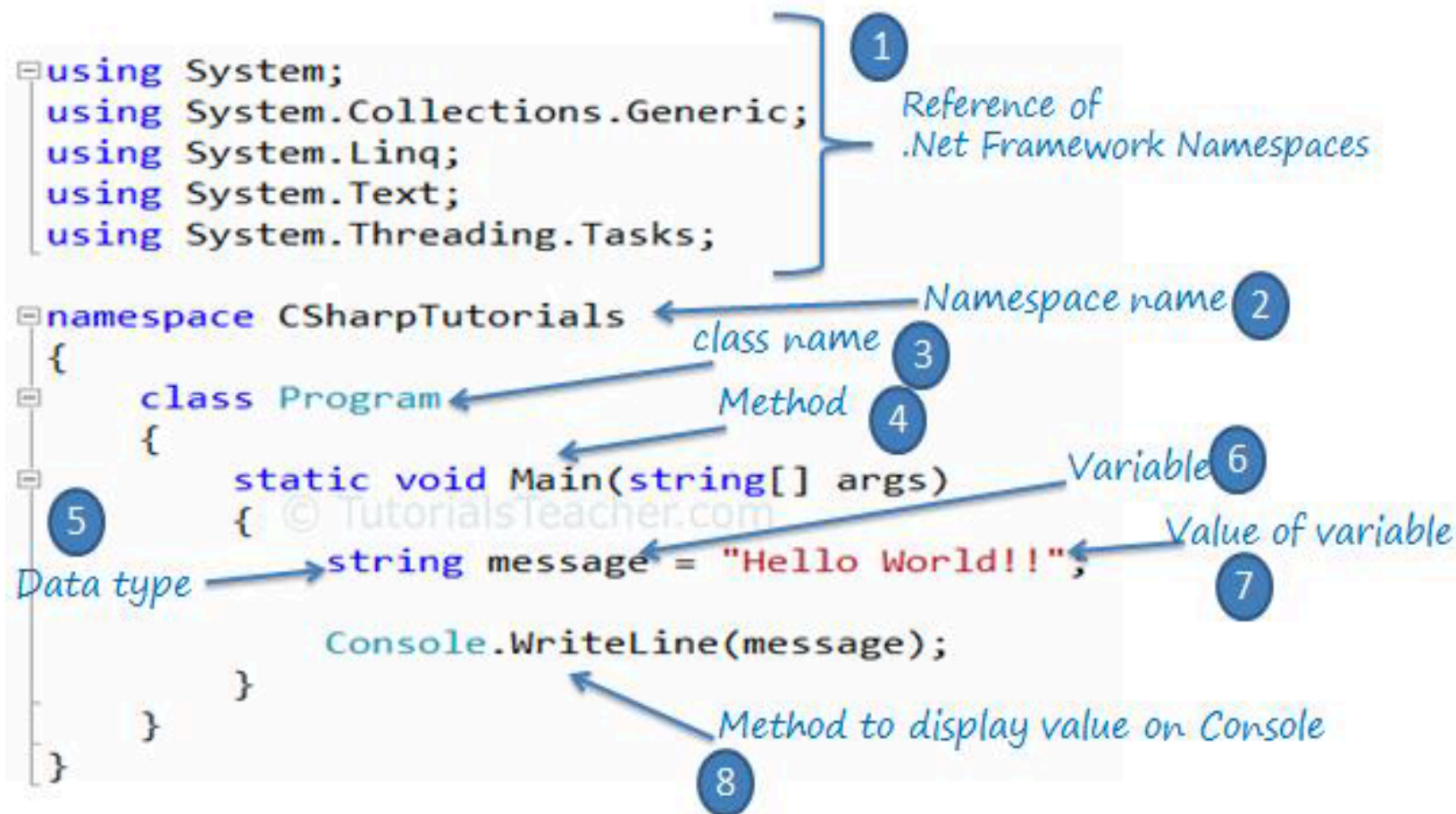
Үйлдэл	Тайлбар	Жишээ
==	2 операндын утга тэнцүү бол үнэн үгүй бол худал	$(A == B)$ худал
!=	2 операнд хоорондоо тэнцүү биш бол үнэн үгүй бол худал	$(A != B)$ үнэн
>	Хэрэв зүүн талын операндын утга баруун гар талын операндын утгаас их бол үнэн үгүй бол худал	$(A > B)$ худал
<	Хэрэв зүүн талын операндын утга баруун гар талын операндын утгаас бага бол үнэн үгүй бол худал	$(A < B)$ үнэн
>=	Хэрэв зүүн талын операндын утга баруун гар талын операндын утгаас их юм уу тэнцүү бол үнэн үгүй бол худал	$(A >= B)$ худал
<=	Хэрэв зүүн талын операндын утга баруун гар талын операндын утгаас бага буюу тэнцүү бол үнэн үгүй бол худал	$(A <= B)$ үнэн

Үндсэн үйлдлүүд

Логик үйлдэл	Тайлбар	Жишээ
&&	AND	A && B худал
	OR	(A B) үнэн
!	NOT	!(A&&B) үнэн

p	q	p&q	p q	p^q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	1
1	0	0	1	1

C# кодын бүтэц



Операторууд

- Сонголтын операторууд
 - If – else
 - Switch - case
- Давталтын операторууд
 - While болон Do – while
 - For
 - Foreach
- Удирдлага шилжүүлэх үйлдэл
 - Goto
 - Continue
 - Break
 - return

IF - else

- Бичигдэх хэлбэр:

```
if (expression)  
{  
    //One or more statements to be executed if the  
    expression evaluates to true  
}  
[else  
{  
    //One or more statements to be executed if the  
    expression evaluates to false  
}]
```

Switch - case

- Бичигдэх хэлбэр:

```
switch(variable)
{
    case value:
        //Statements
        break;
    case value:
        //Statements
        break;
    default:
        //Statements
        break;
}
```


While давталт

- While ()

Бичигдэх хэлбэр:

```
while (condition)
{
    // Statements
}
```

- Do – while()

Бичигдэх хэлбэр:

```
do
{
    //Statements
} while (condition)
```

For давталт

- Бичигдэх хэлбэр:

```
for (initialization; condition; increment/decrement)
{
    //Statements
}
```

- Жишээ нь:

```
For(int ndx=0; ndx <=6; ndx++)
    total = total+ndx;
```

Foreach давталт

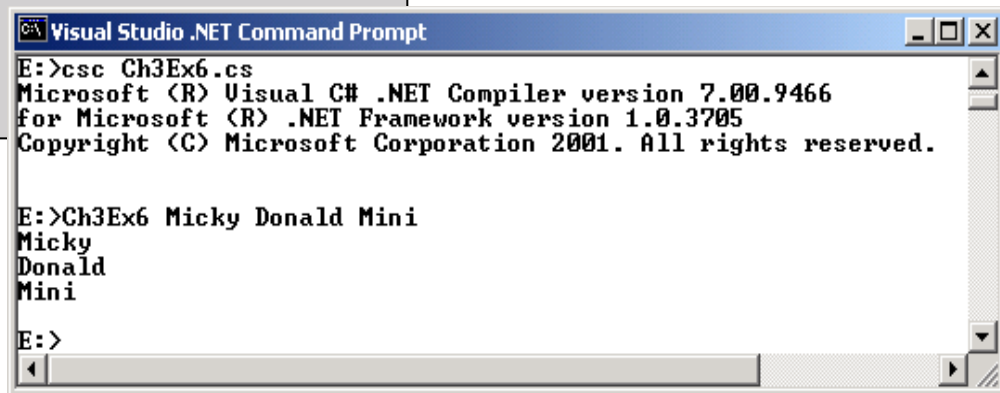
- Бичигдэх хэлбэр:

```
foreach (Type Identifier in expression)  
{  
    //Statements  
}
```

- Энэ давталт нь цуглуулга (жнь: массив) дотроос хувьсагчаар дамжуулан утгуудыг авдаг.
- Энэ давталт нь зөвхөн унших зориултаар ашиглагддаг.

Жишээ нь:

```
using System;
public class ForEachDemo
{
    static void Main (String[] args)
    {
        int index;
        String[] array1=new String[3];
        for (index=0;index<3;index++)
        {
            array1[index] = Console.ReadLine();
        }
        foreach (String strName in array1)
        {
            Console.WriteLine (strName);
        }
    }
}
```



The screenshot shows a Windows command prompt window titled "Visual Studio .NET Command Prompt". The window contains the following text:

```
E:>csc Ch3Ex6.cs
Microsoft (R) Visual C# .NET Compiler version 7.00.9466
for Microsoft (R) .NET Framework version 1.0.3705
Copyright (C) Microsoft Corporation 2001. All rights reserved.

E:>Ch3Ex6 Micky Donald Mini
Micky
Donald
Mini
E:>
```

The output of the program shows the words "Micky", "Donald", and "Mini" on separate lines, which were entered as input in the previous step.

Хувьсагч зарлах

- Хувьсагч бол өгөгдөлд зориулсан энгийн санах ойн байрлал юм.
- Түүнд C#илэрхийллийн хэсэг агуулгыг байрлуулж болно.
- Хувьсагч дахь өгөгдлийн тайлбарыг төрлүүдээр удирддаг.

Хувьсагчийн_төрөл хувьсагчийн_нэр;

Хувьсагчийн_төрөл хувьсагчийн_нэр = Хувьсагчийн_утга;

C#:

```
int x;  
int x=3;
```

Visual C++:

```
int x;  
int x=3;
```

Java:

```
int x;  
int x=3;
```

C#-ийн хувьсагчийн төрлүүд

- C#-ийн төрлүүдийг дараах хоёр хэлбэрт ангилдаг.
 - Value type – энгийн
 - Reference type – заалтын
- C# бол “Хүчирхэг төрөлтэй” хэл юм. Хувьсагчтай холбоотой бүх үйлдлүүдийг гүйцэтгэхэд хувьсагчийн төрөл маш чухал!

C#-ийн төрлүүдийн ангилал

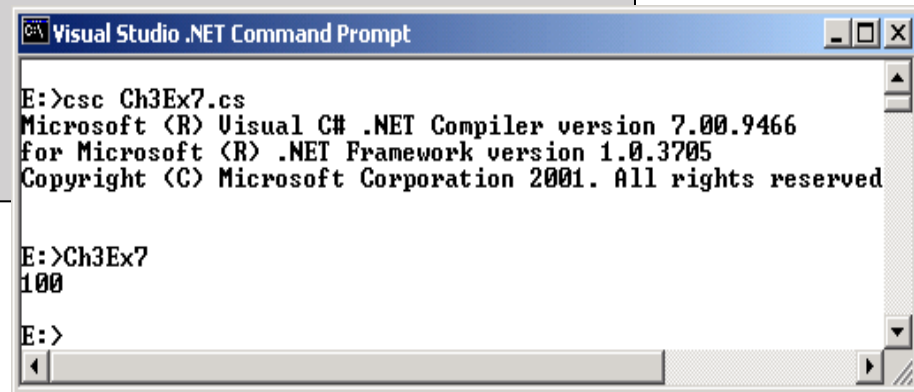
Value Types ■ санах ойд байгаа өгөгдөлтэй ажилладаг.

**Reference
Types**

■ санах ойн хаягийг ажилладаг.

Value Types

```
using System;
class DataTypeTest
{
    public static void Main()
    {
        int variableVal = 100;
        funcTest(variableVal);
        Console.WriteLine("This value of the variable
                           is {0}",variableVal);
    }
    static void funcTest (int variableVal)
    {
        int tempVar = 10;
        variableVal = tempVar*20;
    }
}
```



Visual Studio .NET Command Prompt

```
E:>csc Ch3Ex7.cs
Microsoft (R) Visual C# .NET Compiler version 7.00.9466
for Microsoft (R) .NET Framework version 1.0.3705
Copyright (C) Microsoft Corporation 2001. All rights reserved

E:>Ch3Ex7
100

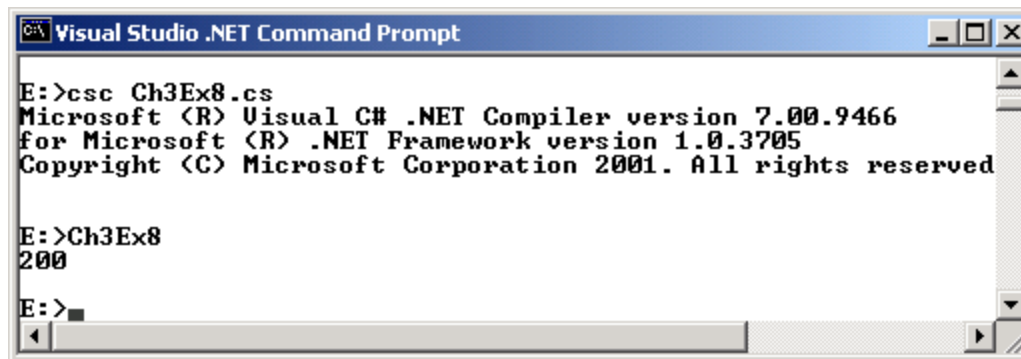
E:>
```


Reference Types

```
using System;
class DataTypeTest
{
    public int variableVal;
}
class DataTypeTestRef
{
    static void Main()
    {
        DataTypeTest dataTest = new
DataTypeTest();
        dataTest.variableVal = 100;
        funcDataTypeTest(dataTest);
        Console.WriteLine (dataTest.variableVal);
    }
}
```

Reference Types - үргэжлэл

```
static void funcDataTypeTest(DataTypeTest dataTest)
{
    int tempVar = 10;
    dataTest.variableVal = tempVar*20;
}
```



The screenshot shows a Visual Studio .NET Command Prompt window. The command prompt displays the following text:

```
E:>csc Ch3Ex8.cs
Microsoft (R) Visual C# .NET Compiler version 7.00.9466
for Microsoft (R) .NET Framework version 1.0.3705
Copyright (C) Microsoft Corporation 2001. All rights reserved

E:>Ch3Ex8
200

E:>
```

Arrays

- Ижил төрлийн олон утгыг хадгалах боломжтой, өгөгдлийн нийлмэл төрлийг массив гэнэ.
- C# -д массивыг дараах байдлаар зарлана:

Өгөгдлийн төрөл [элементийн тоо] Массивын нэр;

`int[6] array1;`



Enumeration

- Enumeration төрөл нь жагсаалт төрөл юм. Энэ төрлийг тодорхойлж өгөхдөө **enum** түлхүүр үгийг ашигладаг.

Жишээлбэл

```
enum Days {Sat, Sun, Mon, Tue, Wed, Thu, Fri};
```

- Энэ жагсаалтанд Sat нь 0, Sun нь 1 гэх мэт дугаарлагдана.

Enumeration

- Жагсаалтын утгыг ахин тодорхойлж болно.
(override)

Жишээлбэл

```
enum Days {Sat=1, Sun, Mon, Tue, Wed, Thu, Fri};
```

- Энэ жагсаалтанд утга нь 1-с эхлэн дугаарлагдана.

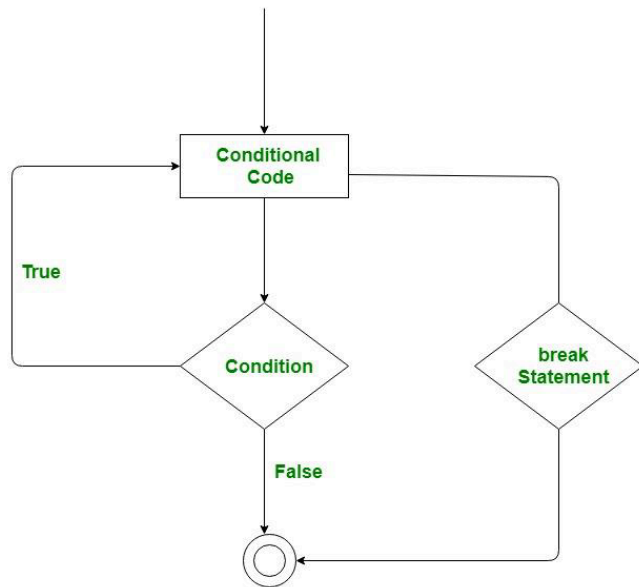
Enumeration

```
using System;
public class EnumTest {
    enum Days { Sat = 1, Sun, Mon, Tue, Wed, Thu, Fri };
    public static void Main() {
        int x = (int)Days.Sun;
        int y = (int)Days.Fri;
        Console.WriteLine("Sun = {0}", x);
        Console.WriteLine("Fri = {0}", y);
    }
}
```

Удирдлага шилжүүлэх үйлдэл

- ▣ Goto - шилжилт
- ▣ Continue - үргэжлүүлэх
- ▣ Break - зогсоох
- ▣ Return – утга буцаах

Break оператор

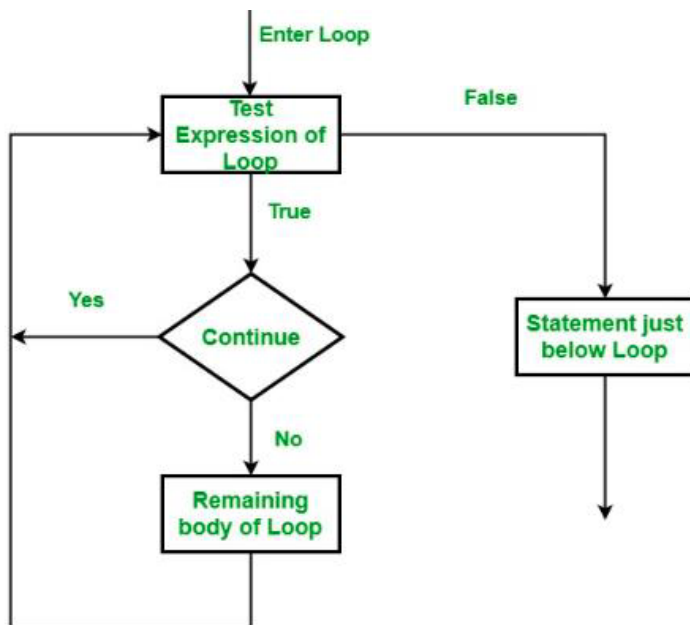


```
for (int i = 1; i < 4; i++)  
{  
    if (i == 3)  
        break;  
  
    Console.WriteLine("GeeksforGeeks");  
}
```

Output:

```
GeeksforGeeks  
GeeksforGeeks
```


Continue оператор



```
// This will skip 4 to print  
for (int i = 1; i <= 10; i++) {
```

```
// if the value of i becomes 4 then  
// it will skip 4 and send the  
// transfer to the for loop and  
// continue with 5
```

```
if (i == 4)  
    continue;
```

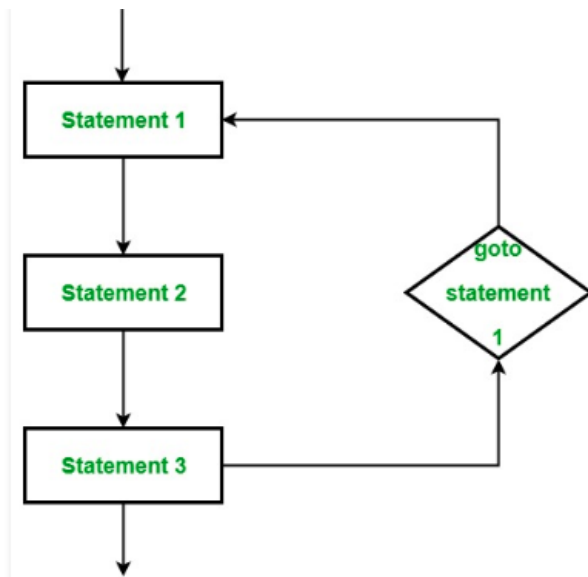
```
    Console.WriteLine(i);
```

```
}
```

Output:

1
2
3
5
6
7
8
9
10

Goto оператор



```
int number = 20;
switch (number) {

case 5:
    Console.WriteLine("case 5");
    break;
case 10:
    Console.WriteLine("case 10");
    break;
case 20:
    Console.WriteLine("case 20");

    // goto statement transfer
    // the control to case 5
    goto case 5;

default:
    Console.WriteLine("No match found");
    break;
}
```

Output:

```
case 20
case 5
```

Return оператор

```
// creating simple addition function
static int Addition(int a)
{
    // add two value and
    // return the result of addition
    int add = a + a;

    // using return statement
    return add;
}

// Main Method
static public void Main()
{
    int number = 2;

    // calling addition funtion
    int result = Addition(number);
    Console.WriteLine("The addition is {0}", result);
}
```

Output:

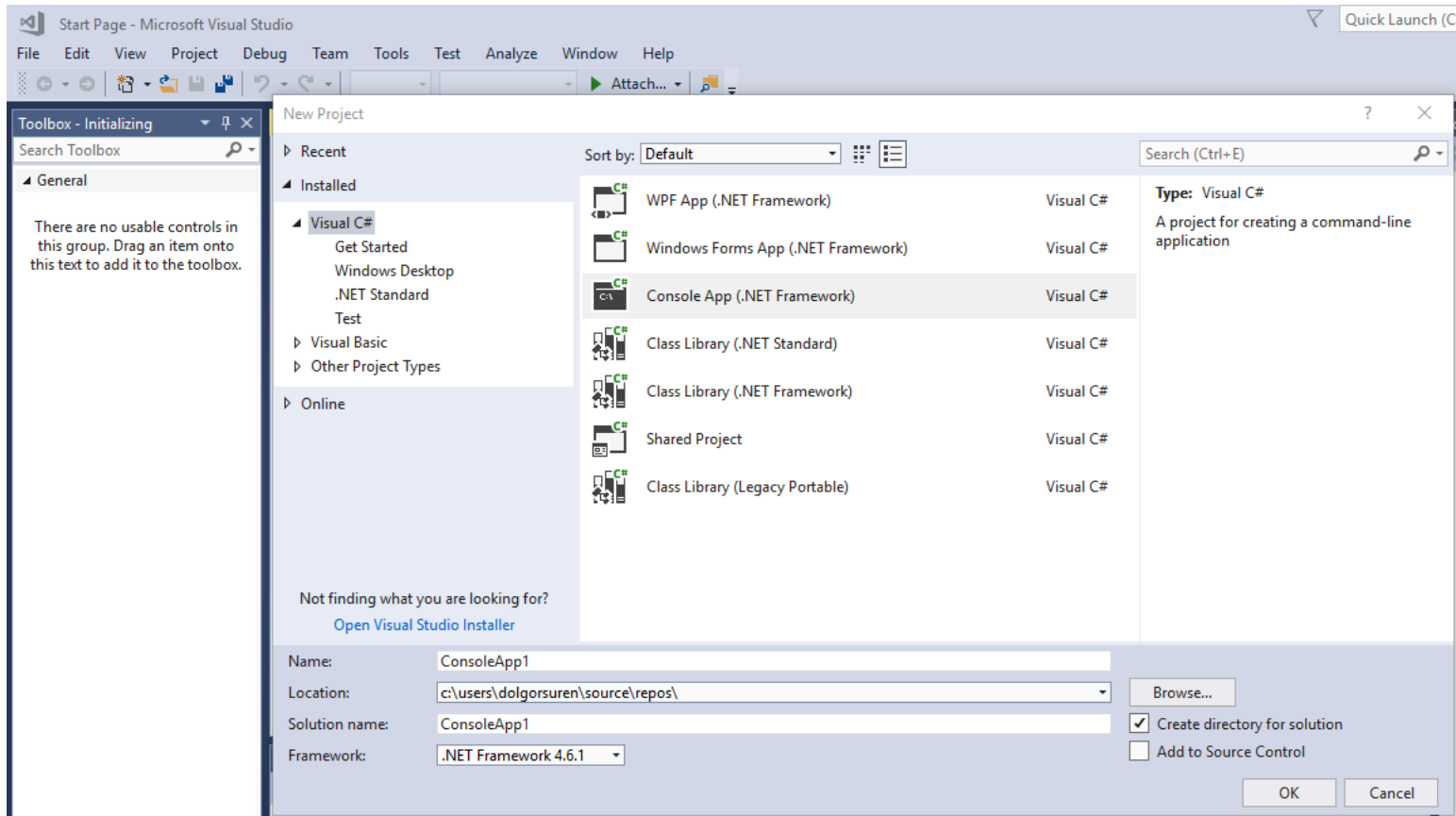
The addition is 4

Console Програм

- Console програм нь **class**-уудаас бүрддэг.
- Console нь:
 - Хөгжүүлэлт хийхэд амар
 - Хэрэглэгчид мэдээллийг командын хэсэгт харуулах
 - Хэрэглэгчээс мэдээлэлийг команд горимоос хүлээн авах
 - no graphical user interface (GUI) – хэрэглэгчийн интерфейс загвар байхгүй.

```
Console.WriteLine();  
Console.ReadLine();  
Console.ReadKey();
```

Console програм үүсгэх



Анхаарал тавьсанд баярлалаа .