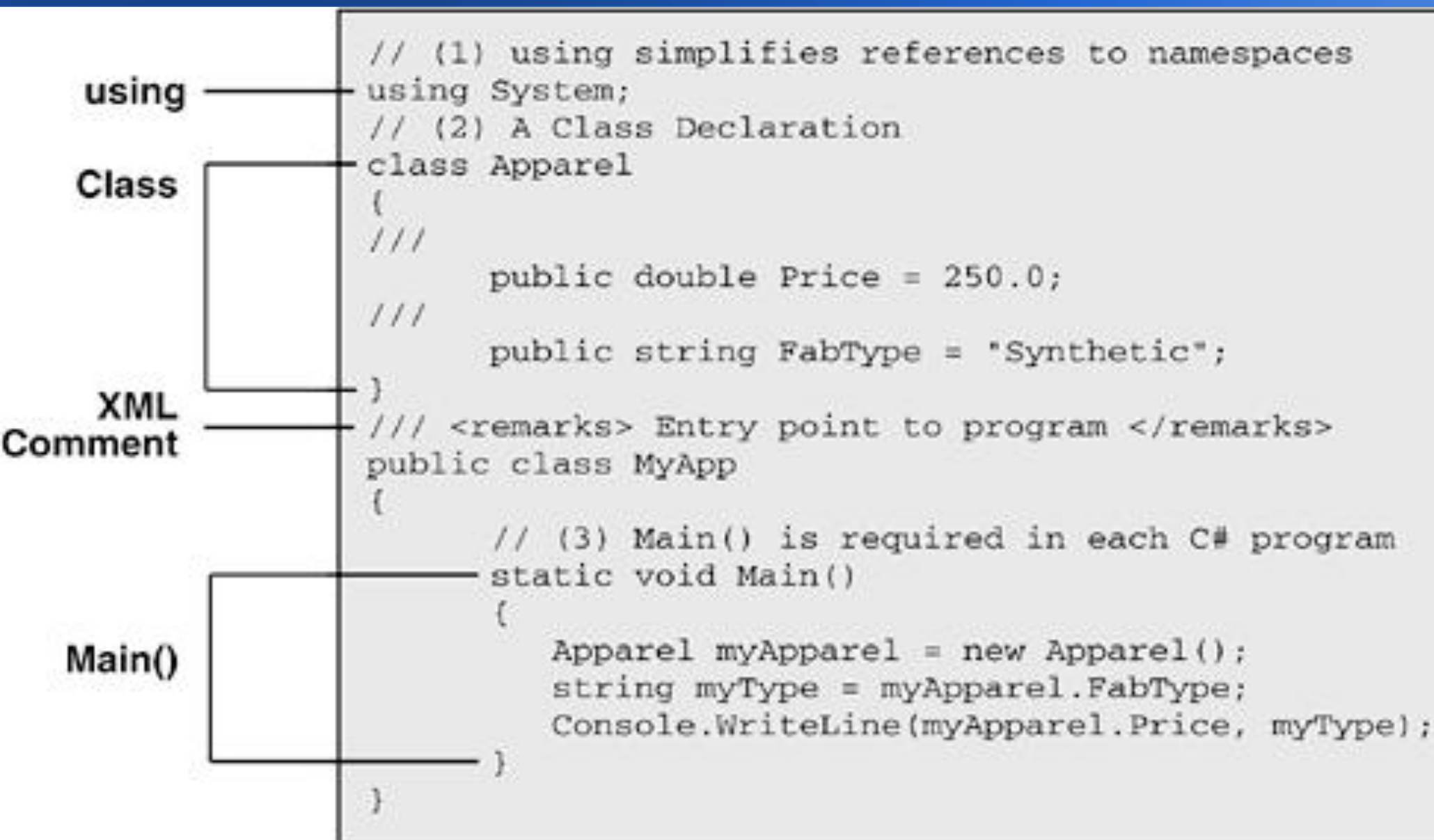


C# хэлний үндэс

C# програмын бүтэц



```
static int Main()
{
    return 0; // бүхэл тоон утга буцаах ёстой

    static int Main(string[] args)
    {
        // аргументуудаар гүйх
        foreach(string myArg in args)
            Console.WriteLine(myArg);
        return 0;
    }
}
```

```
C:\> MyApparel 5 6
```

C# хэлний мөрдлөг

Хүснэгт 2-1. C# нэрлэх мөрдлөг

Төрөл	Тэмдэглэгээ	Жишээ, тайлбар
Класс	Паскаль	<ul style="list-style-type: none"> Нэр үг, нэр үгийн хослолууд I үсгээр эхлэхгүй байхыг хичээх хэрэгтэй, учир нь энэ нь интерфэйсийн нэрэнд хэрэглэгддэг. Доогуур зураасыг хэрэглэх хэрэггүй.
Тогтмол	Паскаль	<code>public const double GramToPound = 454.0 ;</code>
enum төрөл	Паскаль	<ul style="list-style-type: none"> Тоочих утгуудад Паскаль тэмдэглэгээг хэрэглэнэ Use singular name for enums. <code>public enum WarmColor { Orange, Yellow, Brown}</code>
Үзэгдэл	Паскаль	<ul style="list-style-type: none"> Үзэгдэл боловсруулагчийн нэр EventHandler дагавартай байдаг Үзэгдлийн аргумент нь EventArgs дагавартай байдаг
Иксепшн	Паскаль	<ul style="list-style-type: none"> Exception дагавартай
Интерфэйс	Паскаль	<ul style="list-style-type: none"> I утвартай, жнь, IDisposable
Локал хувьсагч	Камел	<ul style="list-style-type: none"> Хувьсагчдад Паскаль тэмдэглэгээ хэрэглэнэ, <code>int myIndex.</code>

Хүснэгт 2-1. C# нэрлэх мөрдлөг

Төрөл	Тэмдэглэгээ	Жишээ, тайлбар
Метод	Паскаль	<ul style="list-style-type: none"> Үйл үг эсвэл үйл үгийн үеүүдийг ашиглах хэрэгтэй
Нэйmspэйс	Паскаль	<ul style="list-style-type: none"> Ижил нэртэй нэйmspэйсээс зайлсийхийн тулд зохиогч компаны нэрийг өмнө нь оруулах нь зүйтэй. <p>Acme.GraphicsLib</p>
Проперти	Паскаль	<ul style="list-style-type: none"> Нэр үг ашиглах хэрэгтэй
Параметр	Камел	<ul style="list-style-type: none"> Параметрийн зорилгыг илэрхийлэх утга бүхий нэр сонгох хэрэгтэй.

Pascal & Camel

- `MyClassAdder`
- `myClassAdder`

C# програмын тайлбар

- мөр тайлбар (//)
- олон мөр тайлбар (/* */).

/// <remarks> классын XML тайлбар </remarks>

XML тайлбар нь 3 налуу зураасаар (///) эхэлдэг ба ихэвчлэн бүтэц, класс, классын гишүүдийг тайлбарласан XML таагыг агуулдаг.

/// <remarks> классын XML тайлбар </remarks>

XML тайлбар нь 3 налуу зураасаар (///) эхэлдэг ба ихэвчлэн бүтэц, класс, классын гишүүдийг тайлбарласан XML таагыг агуулдаг.

C# төрөл	FCL төрөл	Тайлбар
object	System.Object	Бүх төрлүүдийн суурь төрөл
string	System.String	Юникод мөр
decimal	System.Decimal	28 цифрийн нарийвчлалтай аравтын тоо
bool	System.Boolean	true эсвэл false утга
char	System.Char	16 битийн юникод тэмдэгт
byte	System.Byte	8-бит тэмдэггүй бүхэл тоо
sbyte	System.SByte	8-бит бүхэл тоо
short	System.Int16	16-бит бүхэл тоо
int	System.Int32	32-бит бүхэл тоо
long	System.Int64	64-бит бүхэл тоо
ushort	System.UInt16	16-бит тэмдэггүй бүхэл тоо
uint	System.UInt32	32-бит тэмдэггүй бүхэл тоо
ulong	System.UInt64	64-бит тэмдэггүй бүхэл тоо
single (float)	System.Single	Дан-нарийвчлалтай бодит тоо
double	System.Double	Давхар-нарийвчлалтай бодит тоо

```
System.Int32 age = new System.Int32(17);  
int age = 17;  
System.Int32 age = 17;
```

Санамж

```
int iMax = int.MaxValue; // хамгийн их 32 бит бүхэл тоо  
int pVal = int.Parse("100"); // тэмдэгт мөр рүү хувиргах
```

```
short i16 = 50; // 16-бит бүхэл тоо
```

```
int i32 = i16; // Зөв: int нь том хэмжээтэй
```

```
i16 = i32; // Алдаа: short нь 16 бит, int нь 32 бит
```

```
i16 = (short) i32; // Төрөл хувиргалт ашиглаж байгаа учраас зөв
```

```
decimal pct = .15M; // M нь тогтмол утгад хэрэглэгдэнэ
```

. decimal

```
decimal iRate = 3.9834M;      // decimal тоо  
iRate = decimal.Round(iRate,2); // 3.98 буцаана  
decimal dividend = 512.0M;  
decimal divisor = 51.0M;  
  
decimal p = decimal.Parse("100.05");  
  
// дараагийн заавар нь 2 үлэгдэл буцаана  
decimal rem = decimal.Remainder(dividend,divisor);
```

. Bool & char

```
bool bt = true;

string bStr = bt.ToString(); // "true" буцаана

bt = (bool) 1;           // алдаа
```

```
myChar = 'B';           // 'B' нь ASCII 66 утгатай

myChar = (char) 66; // 'B'-тэй адил

myChar = '\u0042'; // Юникод тэмдэгт

myChar = '\x0042'; // 16-тын тэмдэгт

myChar = '\t';           // хэвтээ tab тэмдэгт
```

```
bool bt;

string pattern = "123abcd?";

myChar = pattern[0];           // '1'

bt = char.IsLetter(pattern,3); // үнэн ('a')

bt = char.IsNumber(pattern,3); // худал

bt = char.IsLower(pattern,0);  // худал ('1')

bt = char.IsPunctuation(pattern,7); // үнэн ('?')

bt = char.IsLetterOrDigit(pattern,1); // үнэн

bt = char.IsNumber(pattern,2);  // үнэн ('3')

string kstr="K";

char k = char.Parse(kstr);
```

• byte, sbyte

```
byte[] b = {0x00, 0x12, 0x34, 0x56, 0xAA, 0x55, 0xFF};  
string s = b[4].ToString(); // 170 буцаана  
char myChar = (char) b[3];
```

• short, int, long

```
short i16 = 200;  
i16 = 0xC8 ;    // 16-тын утга 200  
int i32 = i16;  // төрөл хувиргах шаардлагагүй
```

. single, double

- single төрөл нь 1.5×10^{-45} ээс 3.4×10^{38} хүртэл утга бүхий 7 цифрийн нарийвчлалтай бодит тоо.
- double нь 5×10^{-324} ээс 1.7×10^{308} хүртэл 15-с 16 цифрийн нарийвчлалтай бодит тоо.
- Бодит тоон үйлдлийн үр дүн тодорхой бус үед NaN (Not a Number) утгыг буцааж болно. Жишээлбэл, 0.0 тоог 0.0 –д хуваахад үр дүн нь NaN болно.
- Бодит тоог өөр төрөл рүү хувиргахдаа System.Convert методыг ашигладаг.

cont.

```
float xFloat = 24567.66F;  
int xInt = Convert.ToInt32(xFloat); // 24567 буцаана  
int xInt2 = (int) xFloat;  
if(xInt == xInt2) { }           // Худал  
string xStr = Convert.ToString(xFloat);  
single zero = 0;  
if (Single.IsNaN(0 / zero)) { } // ҮНЭН  
double xDouble = 124.56D;
```

. Тоон мөрийг тоон утга руу хувиргах Parse ба TryParse

```
short shParse = Int16.Parse("100");
```

```
int iParse = Int32.Parse("100");
```

```
long lparse = Int64.Parse("100");
```

```
decimal dParse = decimal.Parse("99.99");
```

```
float sParse = float.Parse("99.99");
```

```
double dbParse = double.Parse("99.99");
```


cont.

```
int result;
```

```
// мөрийг хувиргаж үр дүнг result параметерт хадгалах
```

```
bool ok = Int32.TryParse("100", out result);
```

```
bool ok = Int32.TryParse("100", NumberStyles.Integer, null, out result);
```

• Үйлдлүүд: Арифметик, Логик, Нөхцөлт

• Арифметик үйлдлүүд

Хүснэгт 2-4. Тоон үйлдлүүд			
Үйлдэл		Тодорхойлолт	Жишээ
+	(3)	Нэмэх	<code>int x = y + 10;</code>
-		Хасах	
*	(2)	Үржэх	<code>int x = 60;</code> <code>int y = 15;</code> <code>int z = x * y / 2; // 450</code> <code>y = x % 29; // үлдэгдэл нь 2</code>
/		Хуваах	
%		Үлдэгдэл олох	
++	(1)	Угтвар/дагавар	<code>x = 5;</code> <code>Console.WriteLine(x++) // x = 5</code> <code>Console.WriteLine(++x) // x = 6</code>
--		Нэмэх/хасах	
~	(1)	Бит гүйцээлт	<code>int x = ~127; // -128 буцаана</code>
>>	(4)	Баруун шилжих	<code>byte x = 10; // бинари 10 нь 01010 болно</code> <code>int result = x << 1; // 20 = 10100</code> <code>result = x >> 2; // 5 = 00101</code> <code>byte, char, short, int, long дээр хэрэглэнэ</code>
<<		Зүүн шилжих	
&	(5-6-7)	Бит AND	<code>byte x = 12; // 001100</code> <code>byte y = 11; // 001011</code> <code>int result = x & y; // 8 = 001000</code> <code>result = x ^ y; // 7 = 000111</code>
		Бит OR	
^		Бит XOR	

Хүснэгт 2-5. Нөхцөлт ба Харьцуулах үйлдлүүд

Үйлдэл	Тодорхойлолт	Жишээ
==	Тэнцүү	<code>if (x == y) { ... }</code>
!=	Тэнцүү бус	
<	Бага	<code>if (x < y) { ... }</code>
<=	Бага буюу тэнцүү	
>	Их	
>=	Их буюу тэнцүү	
&&	Логик AND	<code>if (x == y && y < 30) { ... }</code>
	Логик OR	Хэрэв эхний илэрхийлэл худал бол дараагийн илэрхийллийг үнэлэхгүй
&	Логик AND	<code>if (x == y y < 30) { ... }</code>
	Логик OR	Үргэлж хоёр дахь илэрхийллийг үнэлнэ
!	Логик үгүйсгэл	<code>if !(x == y && y < 30) { ... }</code>

. Салаалах зааврууд

Хүснэгт 2-6. Салаалах зааврууд

Заавар	Жишээ
<pre>if (буулийн илэрхийлэл) { // зааврууд } else { // зааврууд }</pre>	<pre>if (bmi < 24.9) { weight = "хэвийн"; riskFactor = 2; } else { weight = "илүүдэл"; riskFactor=6; }</pre>
<pre>switch (илэрхийлэл) { case тогтмол: // зааврууд; // break/goto/return() case тогтмол: // зааврууд; // break/goto/return() default: // зааврууд; // break/goto/return() }</pre> <ul style="list-style-type: none"> Тогтмол нь бүхэл тоо, тоочсон утга, тэмдэгт мөр зэрэг байж болно. Си хэлнээс ялгаатай нь "доош унагах" бичиглэлийг зөвшөөрөхгүй. case блок бүр удирдлага шилжүүлэх заавраар төгсөх ёстой. 	<pre>switch (ndx){ case 1: fabric = "хөвөн"; blend = "100%"; break; case 2: // 2 & 3 нэгтгэх case 3: fabric = "хөвөн"; blend = "60%"; break; default: fabric = "хөвөн"; blend = "50%"; break; }</pre>

. if-else

if (буулийн илэрхийлэл) заавар

if (буулийн илэрхийлэл) заавар1 else заавар2

// давхар if зааврууд

if (age > 16)

if (age > 16)

if (sex == "M")

{ if (sex == "M")

type = "Эрэгтэй";

{ type = "Эрэгтэй";

else type = "Эмэгтэй" ;

} else {

else

type = "Эмэгтэй" ;

type = "хүүхэд";

}

} else { type = "хүүхэд"; }

```
switch( илэрхийлэл ) {switch block}
```

```
// тэмдэгт мөр дээр switch хэрэглэх
using System;

public class MyApp {
    static void Main(String[] args) {
        switch (args[0]) {
            case "COTTON":    // том, жижиг үсэг ялгаатай
            case "cotton":
                Console.WriteLine("Сайн цэвэр утас.");
                goto case "natural";
            case "polyester":
                Console.WriteLine("Нийлэг эдийн утас.");
                break;
            case "natural":
                Console.WriteLine("Цэвэр утас.");
                break;
            default:
                Console.WriteLine("Мэдэхгүй утас.");
                break;
        }
    }
}
```

• Давталтууд

- While

while (буулийн илэрхийлэл) { бие код }

- Do

do { бие } while (буулийн илэрхийлэл);

- For

for ([initialization]; [termination condition]; [iteration])
{ бие }

- foreach давталт

foreach (type identifier in collection)
{ бие код }

. Давталтаас удирдлагыг шилжүүлэх

Хүснэгт 2-7. Давталтаас удирдлагыг шилжүүлэх буюу таслах зааврууд

Заавар	Тодорхойлолт	Жишээ
<code>break</code>	Давталтыг таслан удирдлагыг дараагийн зааварт шилжүүлнэ.	<pre>while (true) { ndx+=1; if (ndx >10) break; }</pre>
<code>continue</code>	Давталтыг шинээр эхлүүлнэ. Түүний ард бичигдсэн зааврууд биелэхгүй.	<pre>while (ndx <10) { ndx +=1; if(ndx %2 ==1) continue; totVal += ndx; }</pre>
<code>goto</code> <code>identifier;</code> <code>goto case exp;</code> <code>goto default;</code>	Заасан лэйбел рүү удирдлагыг шилжүүлнэ. <code>goto</code> зааврыг үүрлэсэн блок дотор шилжилт хийхэд ашиглах хэрэггүй, жнь, давталт дотор.	<pre>public int FindMatch(string myColor) { string[] colorsAvail("hex", "улаан", "ногоор", "гүн хөх"); int loc; int matches=0; foreach (colorType in colorsAvail) { loc = colortype.IndexOf(myColor); if (loc >=0) goto Found; continue; Found: matches += 1; } return(matches); }</pre>
<code>return</code> <code>[expression];</code>	Тухайн метоодоос түүнийг дуудсан метод руу удирдлагыг шилжүүлнэ. Аргумент заагаагүй тохиолдолд хоосон утга буцна.	<pre>public double Area(double w, double l){ return w * l; }</pre>

Препроцессор команды

Хүснэгт 2-8. Препроцессор команды

Команд	Тодорхойлолт
<code>#define</code> <code>#undef</code>	Тэмдэг тавих, цуцлах команд. Тэмдэг тавигдсан үед <code>#if</code> командаар шалгахад үнэн утгатай байна.
<code>#if</code> <code>#elif</code> <code>#else</code> <code>#endif</code>	C# хэлний <code>if, else if, else</code> заавруудтай төстэй.
<code>#line</code>	Мөрийн дугаарлалтыг өөрчлөх болон мэдэхэд хэрэглэгдэнэ.
<code>#region</code> <code>#endregion</code>	Visual Studio.NET ашиглах үед хэсэг кодыг бүлэглэж хураах, задлах боломжтой болдог.
<code>#error</code> <code>#warning</code>	<code>#error</code> гадаад алдаа үүсгэнэ. <code>#warning</code> анхааруулга үүсгэнэ.