

Лекція 1.

Основи мови програмування C#

Література

1. Шилдт, Герберт. С# 4.0: полное руководство.: Пер. с англ. — М.: 000 "И.Д. Вильяме", 2011. — 1056 с.
2. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник — СПб.: ПИТЕР, 2010. — 432 с.
3. С/С++ : Теорія та практика. Навч.-методичний посібник / В.В.Войтенко, А.В.Морозов. — Житомир: ЖДТУ, 2004. — 324 стор.
4. Томашевський В.В. Технології програмування. Мова С#: навч.посібн. /В.В. Томашевський. — Житомир: ЖВІ НАУ, 2012. — 484 с.

Найпростіша програма на C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication6
{
    ссылка: 0
    class Program
    {
        ссылка: 0
        static void Main(string[] args)
        {
        }
    }
}
```

Тип даних – множина можливих значень та набір операцій над цими значеннями

Змінна – це область пам'яті, яка має ім'я і в якій зберігається значення певного типу даних



Оперативна пам'ять

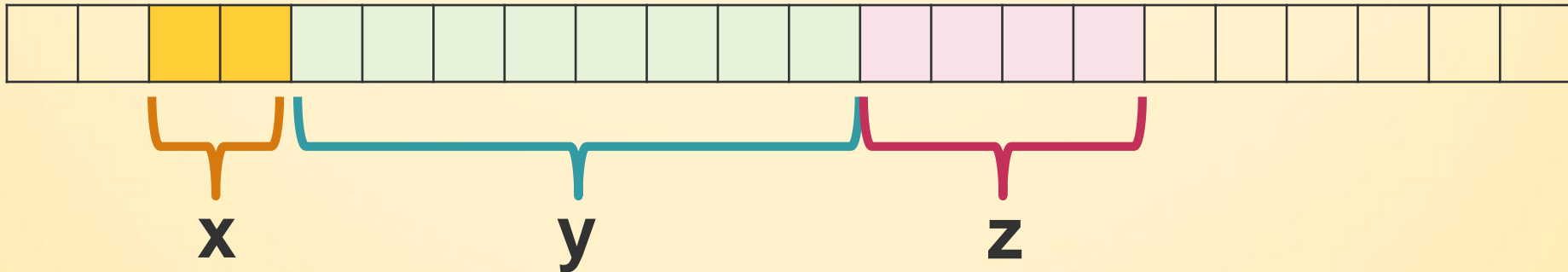
Для того, щоб створити змінну, її потрібно **оголосити**, вказавши тип даних:

тип даних

ім'я змінної ;

Приклад:

```
short x;  
double y;  
int z;
```

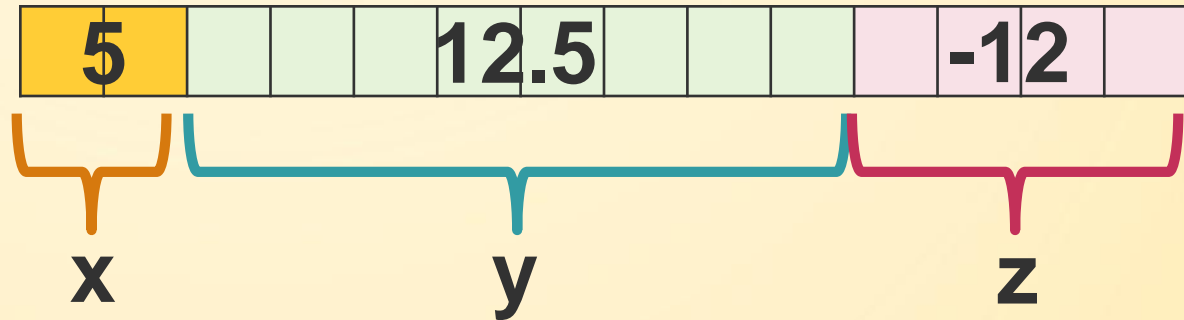


Для запису конкретного значення у змінну потрібно використовувати операцію **присвоювання значення змінній**:

`ім'я змінної` = `значення` ;

Приклад:

```
short x;  
double y;  
int z;  
x = 5;  
y = 12.5;  
z = -12;
```



В одному рядку можна оголошувати кілька змінних та одразу присвоювати їм початкові значення.

Приклад 1:

```
short x;  
double y;  
int z;  
short a;  
short b;  
x = 5;  
y = 12.5;  
z = -12;
```

Приклад 2:

```
short x = 5, a, b;  
double y = 12.5;  
int z = -12;
```

Типи даних мови C#

Цілі

Тип даних	Тип .NET	Діапазон значень	Кількість байтів
byte	System.Byte	0 ... 255	1
sbyte	System.SByte	-128 ... 127	1
short	System.Int16	-32768 ... 32767	2
ushort	System.UInt16	0 ... 65535	2
int	System.Int32	$-2 \cdot 10^9 \dots 2 \cdot 10^9$	4
uint	System.UInt32	0 ... $4 \cdot 10^9$	4
long	System.Int64	$-9 \cdot 10^{18} \dots 9 \cdot 10^{18}$	8
ulong	System.UInt64	0 ... $1.8 \cdot 10^{19}$	8

Типи даних мови C#

Дробові

Тип даних	Тип .NET	Кількість знаків	Кількість байтів
float	System.Single	7	4
double	System.Double	15-16	8
decimal	System.Decimal	28-29	16

Рядковий

Тип даних	Тип .NET
string	System.String

Типи даних мови C#

Символьний

Тип даних	Тип .NET	Кількість байтів
char	System.Char	2

Логічний (булевий)

Тип даних	Тип .NET	Кількість байтів
bool	System.Boolean	1

Є лише два можливих булевих значення:

true – істина

false – хиба

Оголошення змінних

```
bool isEnabled = true;  
int x;  
double y = 3.0;  
string hello = "Hello World";  
char c = 's';  
int a = 4;  
int z = a + 5;  
System.Int64 r = z * 2;  
System.String s = "Some string";  
System.Double d = 1.5;
```

Дробові константи

- Дробові константи по замовчуванню мають тип `double`;
- Щоб задати константу типу `float` треба після числа дописати літеру F:
`float someFloatValue = 1.5F;`
- Щоб задати константу типу `decimal` треба після числа дописати літеру M:
`decimal someDecValue = 1.5M;`
- Для констант типу `double` можна вказувати літеру D, але необов'язково:
`double someDb1Value = 1.5D;`

Операції мови C#

Арифметичні:

бінарні:

+	-	*	/	%
---	---	---	---	---

унарні:

+	-	++	--
---	---	----	----

Префіксний та
постфіксний
інкремент та
декремент

Порівняння:

бінарні:

>	>=	<	<=	==	!=
---	----	---	----	----	----

Операції мови C#

Присвоювання:

бінарні:

=	+=	-=	*=	і т.д.
---	----	----	----	--------

Операції мови C#

Логічні:

бінарні:

&&	
----	--

унарні:

!

Тернарна операція:

умова ? вираз1 : вираз2

Операції мови C#

Відмінність від мови C: логічні операції та операції порівняння повертають значення типу **bool** (а не цілі числа, як у мові C)

Мова C#	Мова C
<pre>int x = 1; if (x) { ... }</pre>	<pre>int x = 1; if (x) { ... }</pre>

Помилка компіляції, тут потрібне значення булевого типу

У мові C код працює, оскільки істиною є будь-яке ненульове значення

Операції мови C#

Мова C#	Мова C
<pre>bool x = true; if (x) { ... }</pre>	<pre>int x = 1; if (x) { ... }</pre>

Вираз	Результат у мові C#	Результат у мові C
<code>1 > 2</code>	<code>false</code>	<code>0</code>
<code>100 > 3</code>	<code>true</code>	<code>1</code>
<code>(2 > 1) (3 > 2)</code>	<code>true</code>	<code>1</code>

Оператори мови C#

Аналогічні мові C:

розгалуження: `if`, `switch`

цикли: `for`, `while`, `do ... while`,
`break`, `continue`

Відмінності:

- умова – це значення типу `bool`
- `switch` підтримує стандартні типи, у тому числі і `string`
- у `switch` оператори `break` є обов'язковими

Виведення у консоль

- `Console.Write()` – виводить і залишає курсор у тому ж рядку
- `Console.WriteLine()` – виводить і переводить курсор на наступний рядок

Синтаксис:

```
Console.Write("рядок формату", зм1,  
зм2, ...);
```

Рядок формату може містити специфікатори форматування

Виведення у консоль

Специфікатори форматування

задаються у с

вказується номер символу...



```
C:\Windows\system32\cmd.exe
```

```
Значення x = 10, d = 1,5, f = 1
```

```
int x = 10;
double d = 1.5;
float f = 1.3f;
Console.WriteLine(
    "Значення x = {0}, d = {1}, f = {2}",
    x, d, f);
Console.WriteLine("Значення x = " + x +
"Значення d = " +d);
```

Виведення у консоль

Дробові числа можуть виводитись з роздільником «кома». Щоб встановити роздільник «крапка» використовуйте код:

```
static void Main(string[] args)
{
    System.Globalization.CultureInfo customCulture = (System.Globalization.CultureInfo)
        System.Threading.Thread.CurrentThread.CurrentCulture.Clone();
    customCulture.NumberFormat.NumberDecimalSeparator = ".";
    System.Threading.Thread.CurrentThread.CurrentCulture = customCulture;

    int x = 10;
    double d = 1.5;
    float f = 1.3f;
    Console.WriteLine("Значення x = {0}, d = {1}, f = {2}", x, d, f);
}
```

Для підтримки української абетки треба додати такий код:

```
Console.OutputEncoding = Encoding.Unicode;
Console.InputEncoding = Encoding.Unicode;
```

Читання даних з консолі

- Мова С# дозволяє читати лише рядки або окремі символи.
- Неможливо прочитати з клавіатури значення інших типів (наприклад, цілих).
- `Console.ReadKey()` – читає один символ
- `Console.ReadLine()` – читає рядок

Читання даних з консолі

Щоб працювати з цілими або дробовими числами виконують такі дії:

- 1) Читають з клавіатури рядок (тип `string`)
- 2) Виконують перетворення типу з `string` до потрібного (наприклад, `int`)

Читання даних з консолі

Перетворення типу можна виконати такими способами:

1) `тип x = тип.Parse(Console.ReadLine());`

2) `тип x = Convert.ToТипDotNet(Console.ReadLine());`

Приклад:

```
int x = int.Parse(Console.ReadLine());
```

```
int x = Convert.ToInt32(Console.ReadLine());
```


Читання даних з консолі

Приклад:

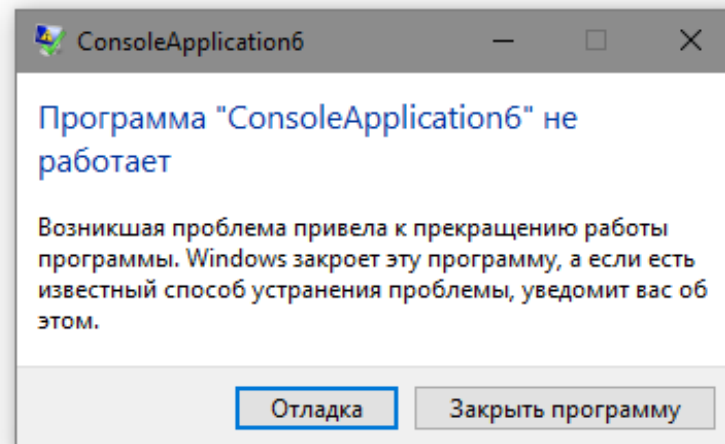
```
class Program
{
    static void Main(string[] args)
    {
        int x;
        Console.WriteLine("Введіть x = ");
        x = int.Parse(Console.ReadLine());
    }
}
```

Читання даних з консолі

При введенні некоректних даних отримуємо помилку:

```
C:\Windows\system32\cmd.exe
Введіть x = hdfgdg

Необработанное исключение: System.FormatException: Входная строка имела неверный ф
в System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer&
fo info, Boolean parseDecimal)
в System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info)
в System.Int32.Parse(String s)
в ConsoleApplication6.Program.Main(String[] args) в D:\MegaUpload\Проекты Visual
TeApplication6\ConsoleApplication6\Program.cs:строка 15
```



Читання даних з консолі

Щоб не отримувати помилки
потрібно використовувати

TryParse:

```
int x;  
Console.Write("Введіть x = ");  
if (int.TryParse(Console.ReadLine(), out x))  
{  
    ... x введено правильно ...  
}
```

Для опрацювання вдома:

- 1) <https://metanit.com/sharp/tutorial/1.1.php>
- 2) <https://metanit.com/sharp/tutorial/1.2.php>
- 3) <https://metanit.com/sharp/tutorial/2.1.php>
- 4) <https://metanit.com/sharp/tutorial/2.2.php>
- 5) <https://metanit.com/sharp/tutorial/2.3.php>