

C#

|

Introduction

MEGAPlan:

1. Syntax
2. Strings
3. Arrays
4. Classes
5. OOP
6. OOP2
7. Collections
8. Generics
9. Interfaces
10. Delegates and Events
11. Files
12. ALL IN

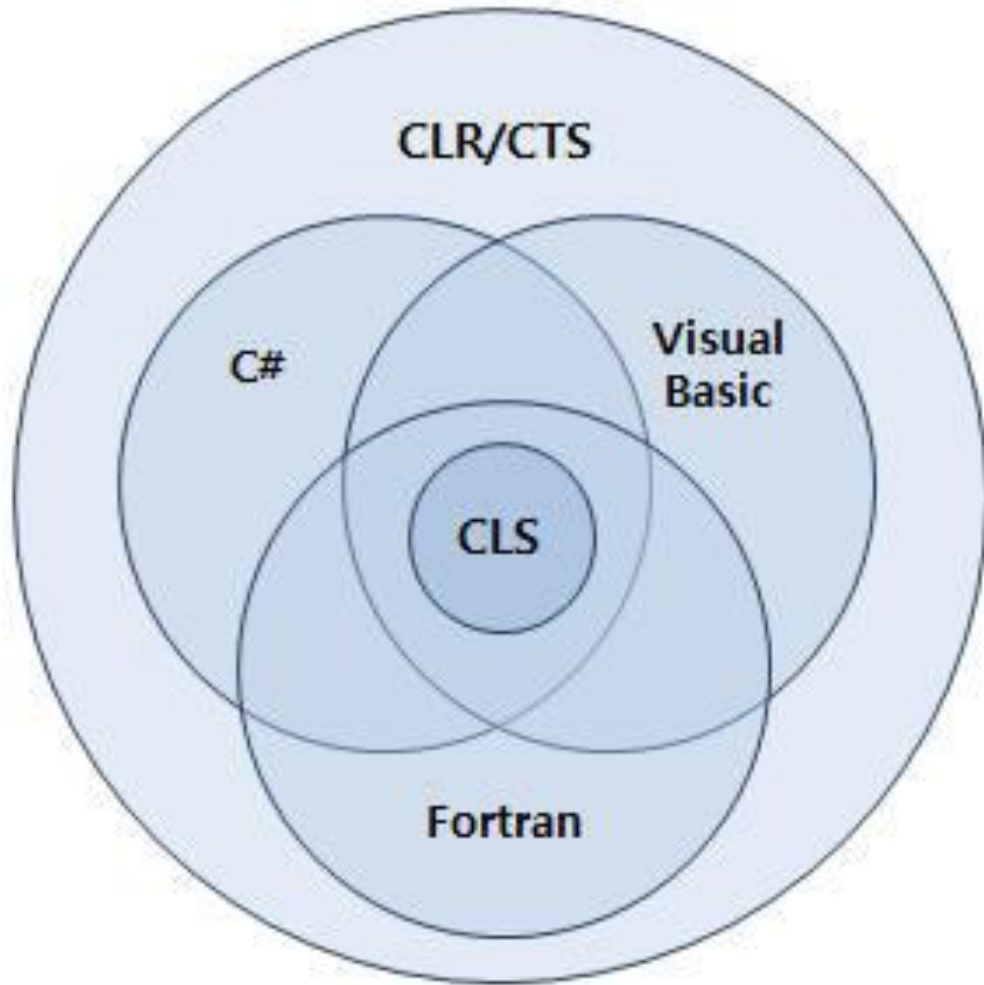
Plan

- Environment
- C# Types
- Basic Constructs

Environment

- C# - язык программирования
- .Net Framework Class Library (FCL) – куча созданного кода
- CLR – среда выполнения приложений
 - **C#,C++,Java >> IL >> Assembler >> 10010101101010**
- Visual Studio – интегрированная среда разработки

Environment



- **CLR** (*Common Language Runtime*) – общеязыковая среда выполнения приложений
- **CTS** (*Common Type System*) – система общих типов
- **CLS** (*Common Language Specification*) – общая спецификация для языков программирования

Types C#

- Язык строго типизирован – всё есть тип
 - Строгое соответствие между типами

Примеры типов:

System.Int32 – целое число

System.IO.FileInfo – тип для работы с файлом

System.IO.Directory – тип для работы с директориями

System.Console – тип для работы с консолью

```
namespace System
{
    public struct Int32 {
        //...
    }
}
```

```
using System;
using System.IO;
```

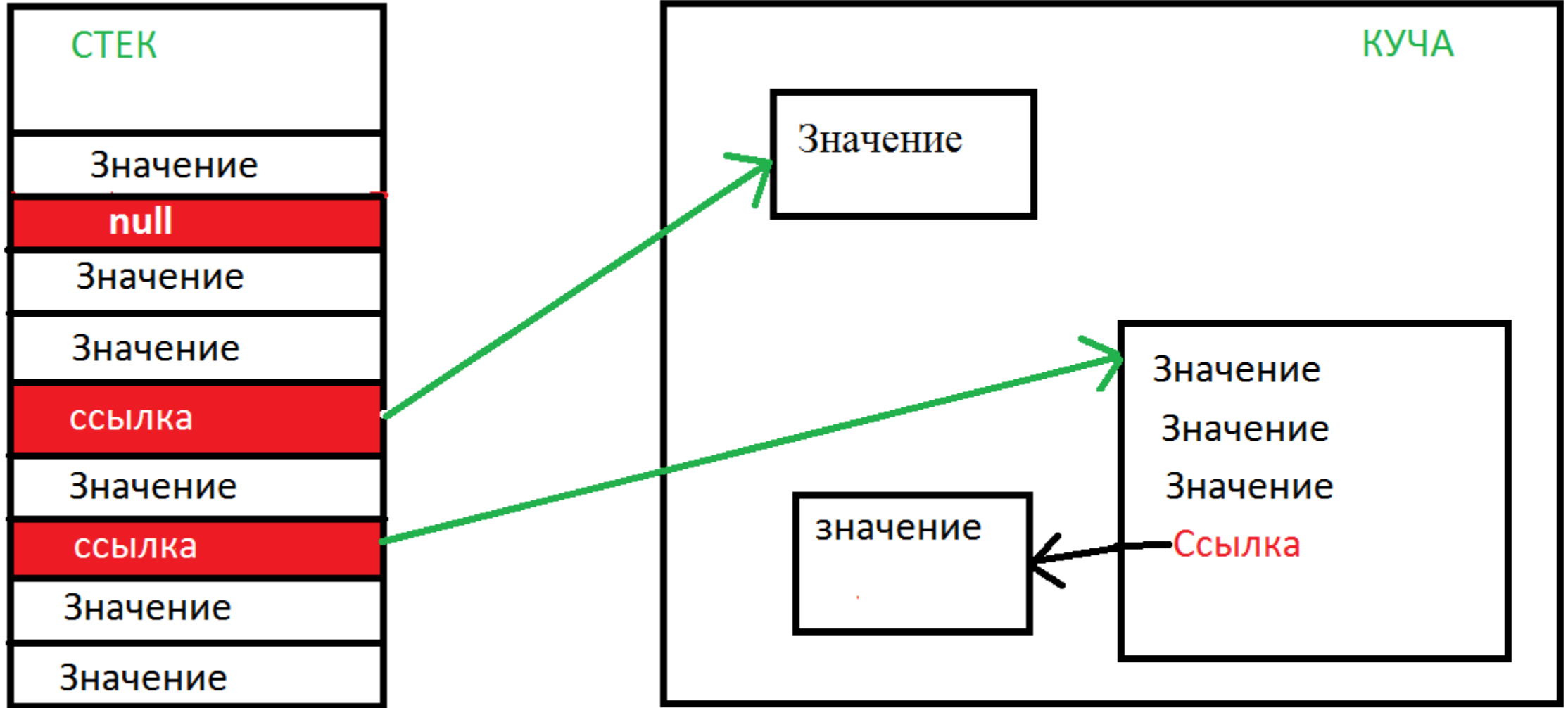
Types C#

- Значимая переменная (в стеке)
- Ссылочная переменная (в куче, а в стеке ссылка на адрес в куче)

Базовый тип `Object`

Все люди и не только произошли от `Object`

Stack & Heap



Basic Value Types

Полное имя	Короткое имя	Описание
<code>System.Int32</code>	<code>int</code>	Целые числа 0, -1, 123, 1002390
<code>System.Double</code>	<code>double</code>	Вещественные числа (с точкой) 0, -1.1, 12.3, 100.2390
<code>System.Boolean</code>	<code>bool</code>	Булевский тип (логика) <code>true</code> либо <code>false</code>
<code>System.Char</code>	<code>char</code>	Символ 'с', '0', '1'
<code>System.String</code>	<code>string</code>	Строка "Привет", "123"

Operators

Arithmetic	Action
+ - * /	Арифметич
++i	Префикс. Инкремент
i++	Постинкремент
--j	Префикс. декремент
j--	Пост. декремент
a % b	Остаток от деления
+=, *=, /=, -=	a *= 4 тоже что a = a * 4

Condition	Action
!	Отрицание (true >> false)
>, <, <=, >=	Без комментариев
==, !=	Равенство, не равенство
,	Строгое и не строгое ИЛИ
&, &&	Строгое и не строгое И
b ? a : c	Условный оператор
^	XOR
??, is, as	Объединения с null, .., ..

Ветвления

```
if(/*условие*/)
{
    /*код*/
}
```

```
if(Weather.IsRain)
{
    MySelf.TakeUmbrella();
}
else
{
    MySelf.TakeHat();
}
```

```
if(Weather.IsRain == true)
{
    MySelf.TakeUmbrella();
}
```

```
if(moon == Moon.Half)
{
    GoToStreet();
}
else if(moon == Moon.Full)
{
    GoToForest();
}
else
{
    SitAtHome();
}
```

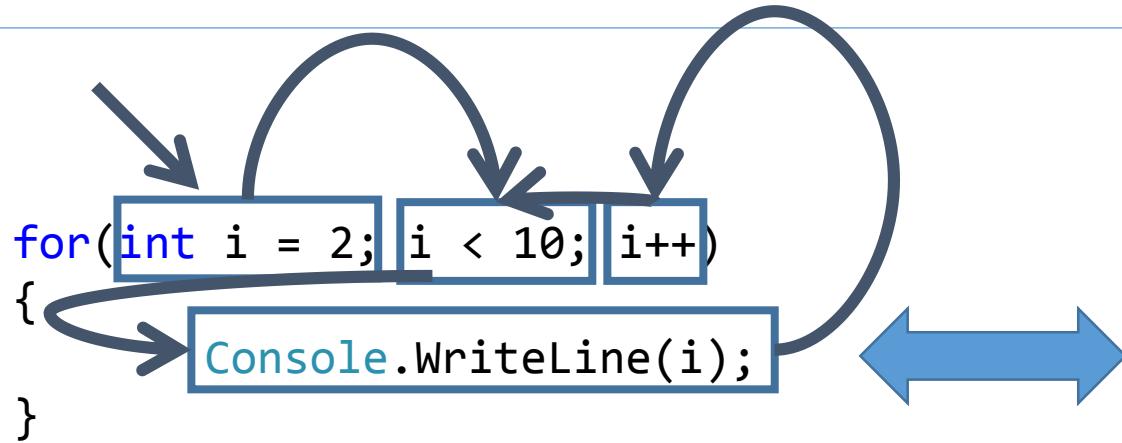
Ветвления (оператор switch)

```
if(/**/)
{
    //...
}else if(/**/)
{
    //...
}else if(/**/)
{
    //...
}else if(/**/)
{
    //...
}else if(/**/)
{
    //...
}else if(/**/)
{
    //...
}else if(/**/)
{
    //...
}
```

```
string meeting = "date";

switch(meeting)
{
    case "bussines":
        Suit();
        break;
    case "date":
        Dress();
        break;
    default:
        Casual();
        break;
}
```

Cycles



```
int i = 2;  
while (i < 10)  
{  
    Console.WriteLine(i);  
    i++;  
}
```

```
int i = 2;  
do  
{  
    Console.WriteLine(i);  
    i++;  
}while (i < 10);
```

```
string array[] = { "e10", "e11", "e12", "e13" };  
foreach(item in array)  
{  
    Console.WriteLine(item);  
}
```

Enum (Перечисления)

```
public enum Days
{
    Sunday = 0,
    Monday = 1,
    Tuesday = 2,
    Wednesday = 3,
    Thursday = 4,
    Friday = 5,
    Saturday = 6
}
```

```
public enum Days
{
    Sunday,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
}
```

```
[Flags] // БИТОВЫЕ ФЛАГИ
public enum FileActs
{
    None = 0,
    Write = 1,
    Read = 2,
    ReadWrite = Read | Write,
    Append = 4,
    WriteAppend = Write|Append, // =5
    ReadAppend = Read|Append, // = 6,
    WRA = Write|Read|Append, // = 7
    Truncate = 8,
    //Write|Truncate
    //..
    //Write|Read|Append|Truncate
    All = Write | Read | Append | Truncate
    // = 15
}
```

Function >> method's signature

```
тип_возвращаемого_значения Имя_функции(тип_1 имя_1, тип_2 имя_2, ...)  
{  
    /*  
        некоторый код  
    */  
    return (тип_возвращаемого_значения) значение;  
}
```

Напомнить про строгую типизацию

```
double Pow(double chislo, double pow)  
{  
    int result = 0;  
    //...  
    return result;  
}
```

```
void DoSmthingMain()  
{  
    /*  
        Do something main  
    */  
}
```

Catch Exceptions (обработка исключений)

```
try
{
    int a = 10 / 0;
}
catch (DivideByZeroException ex)
{
    Console.WriteLine(ex.Message);
}
//finally
//{
// Execute always
//}
```


Literature

1. Эндрю Стиллмэн - Изучаем C#
2. Дж. Рихтер - CLR via C#
3. [**msdn.com**](https://msdn.com)
4. stackoverflow.com
5. cyberforum.ru



Спасибо за внимание!