# C#
# |
# Syntax

# MEGAPlan:

I. Syntax
II. Arrays
III. Strings
IV. Classes
V. OOP
VI. OOP2
VII. Collections
VIII. Generics
IX. Interfaces
X. Delegates and Events
XI. Files
XII. ALL IN

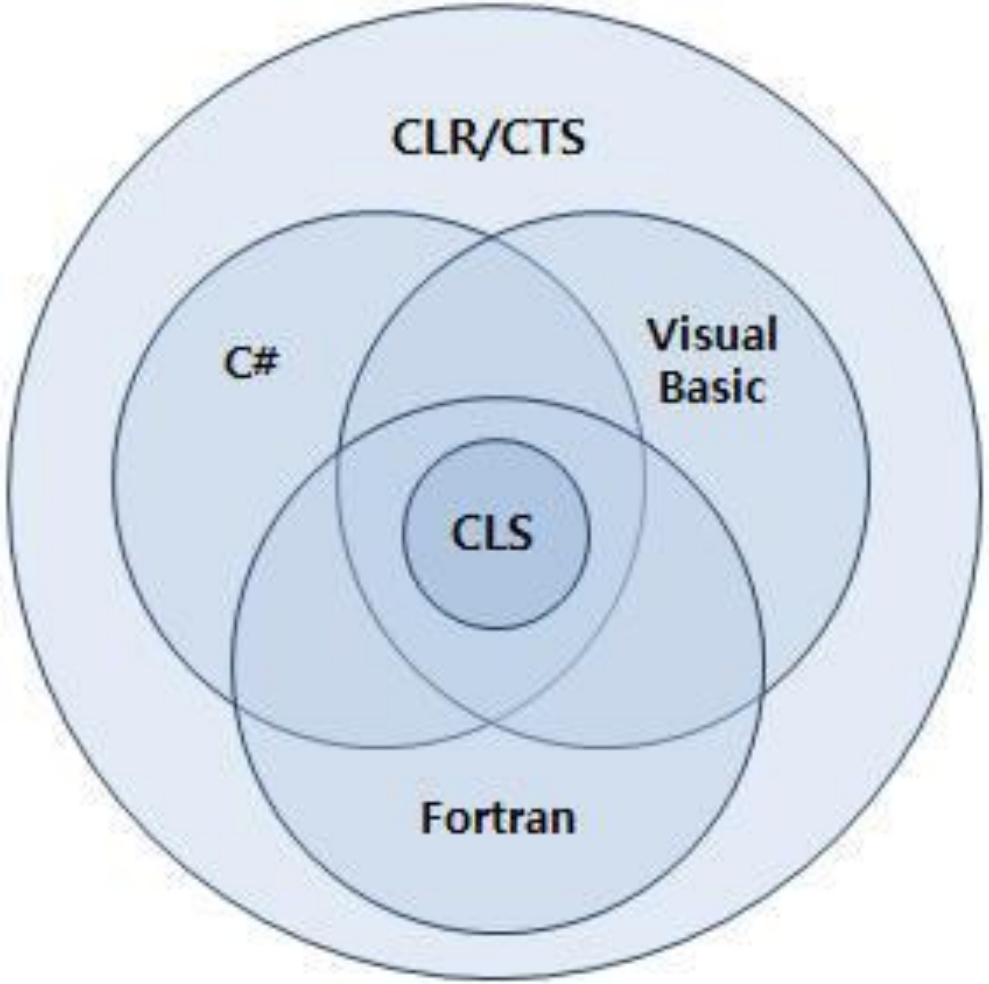
# Plan

- 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

# Variables

**Переменная** :

- Ячейка для хранения информации
- Именованная область в оперативной памяти
  - внутри { блока кода } имена не должны повторяться
- Относится к определенному типу (число, строка, картинка, …)

# 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;
```

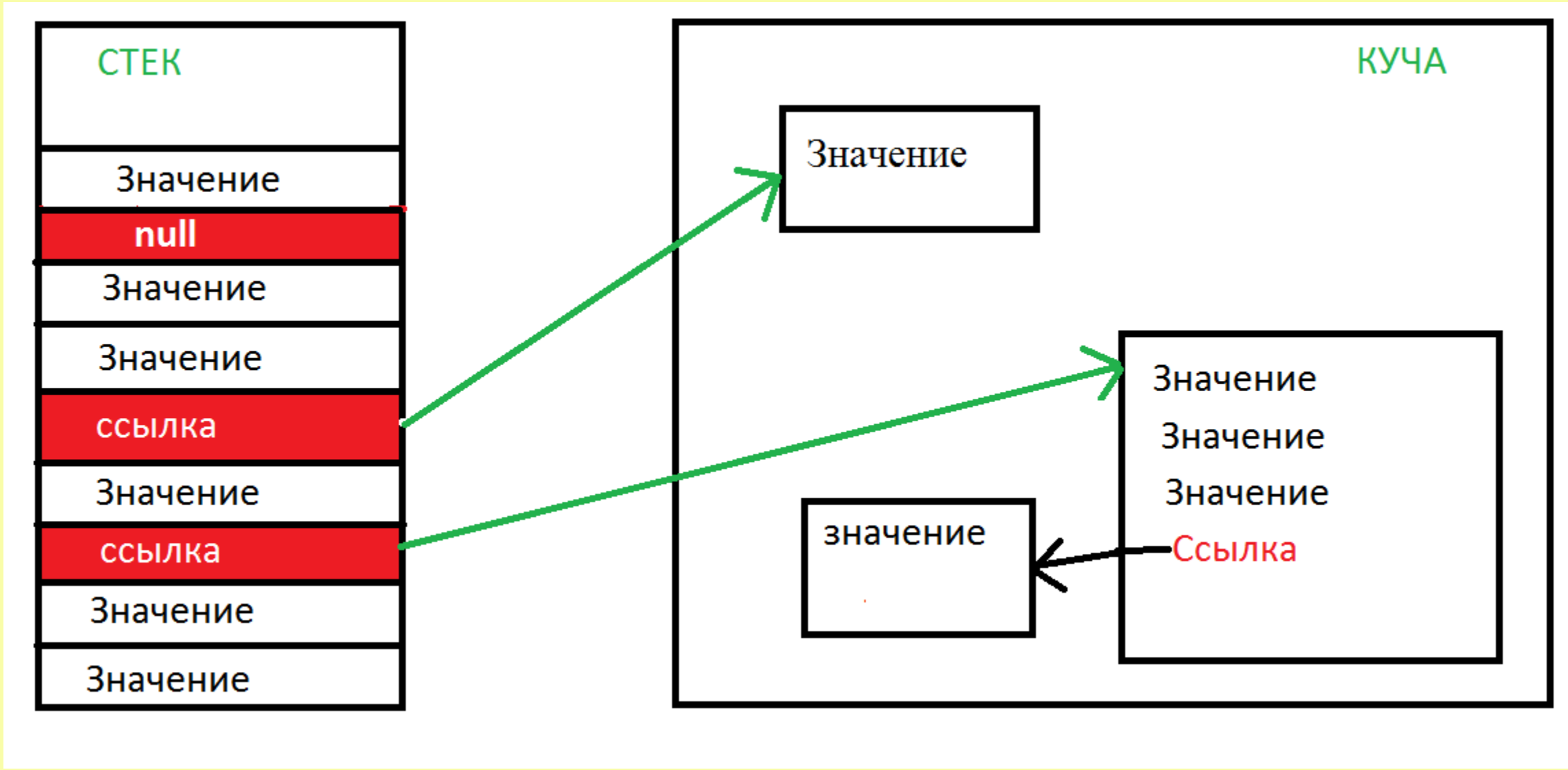
# Types C#

- 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


СТЕК
Значение
null
Значение
Значение
ссылка
Значение
ссылка
Значение
Значение
КУЧА
Значение
Значение
Значение
Значение
значение
Ссылка

# Basic Value Types

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

https://msdn.microsoft.com/ru-ru/library/6a71f45d.aspx?f=255&MSPPError=-2147217396

# Ветвления

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

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

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

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

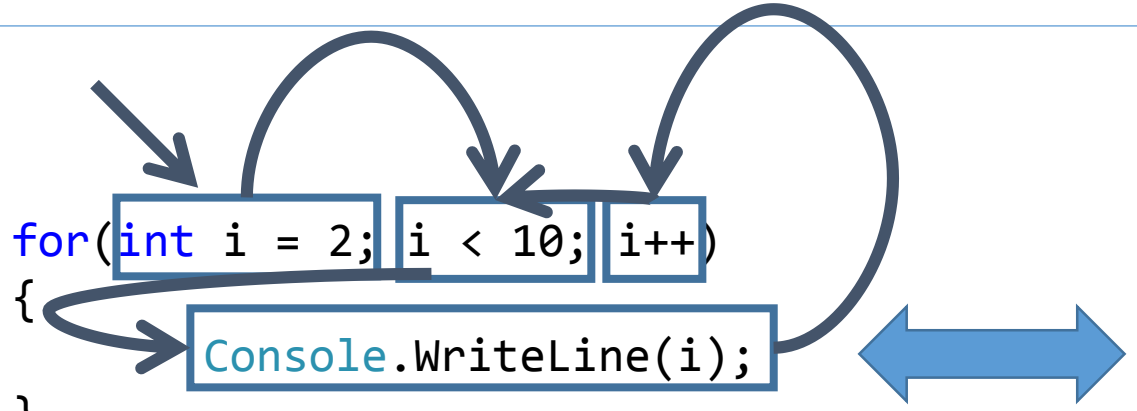
# Ветвления (оператор switch)

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

```
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[] = { "el0", "el1", "el2", "el3" };

foreach(var 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**
4. stackoverflow.com
5. cyberforum.ru


Спасибо за внимание!