

Unsafe, fixed, P/Invoke

Амарский Артем Олегович

Введение

- Сборщик мусора
 - NET's garbage collector manages the allocation and release of memory for your application
- Managed code
- Unmanaged code
 - Использование нативных библиотек

Unsafe

- Пишется при объявлении методов/полей/структур/классов
 - Либо `unsafe { }`
- Небезопасный код
 - Указатели
- Синтаксис как в C
- `void*`

```
int* p1, p2, p3;    // Ok
int *p1, *p2, *p3;  // Invalid in C#
```

Fixed

- Только внутри unsafe
- Фиксируется указатель
 - Недоступны сборщику мусора

Примеры

```
unsafe
{
    string str = "Strings are immutable!";

    for (int i = 0; i < str.Length / 2; i++)
    {
        fixed (char* strPtr = str)
        {
            char tmp = strPtr[i];
            strPtr[i] = str[str.Length - i - 1];
            strPtr[str.Length - i - 1] = tmp;
        }
    }

    Console.WriteLine("Strings are immutable!");
}
```

```
string str = "Strings are immutable!";

LibReverseString.Reverse(str);

Console.WriteLine("Strings are immutable!");
```

P/Invoke

- Библиотеки с неуправляемым кодом в управляемом
- Маршалинг
 - default marshaling type
- Атрибут `DLLImport` – точка входа
- Наоборот – с помощью делегатов

Пример

```
using System;
using System.Runtime.InteropServices;

public class Program
{
    // Import user32.dll (containing the function we need) and define
    // the method corresponding to the native function.
    [DllImport("user32.dll", CharSet = CharSet.Unicode, SetLastError = true)]
    private static extern int MessageBox(IntPtr hWnd, string lpText, string lpCaption, uint uType);

    public static void Main(string[] args)
    {
        // Invoke the function as a regular managed method.
        MessageBox(IntPtr.Zero, "Command-line message box", "Attention!", 0);
    }
}
```