C# programming language. The beginning

Course

Programming Languages

Semester 2, FIIT

Mayer S.F.

Mikhalkovich S.S

LECTURE # 7, 8. Single-dimensional arrays

One-dimensional arrays

```
// How to allocate memory to an array
int[] a = new int[5];

// How to fill an array
int[] b = { 1, 2, 3, 4, 5 };

// ArrGen functions are absent!
for (var i = 0; i < a.Length - 1; i++)
    a[i] = i * i;

// Output of an array
foreach (var x in a)
    Console.Write($"{x} ");

// Return an allocated memory
a = null;

// The .Print extension method is absent
```

How to fill an array

```
// 1 3 5 7 9 11 13 15 17 19 – arithmetic progression
int[] a = new int[10];
a[0] = 1;
for (var i = 1; i < a.Length; i++)
     a[i] = a[i - 1] + 2;
```

```
// 1 2 4 8 16 32 64 128 256 512 – geometric progression
int[] a = new int[10];
a[0] = 1;
for (var i = 1; i < a.Length; i++)
    a[i] = a[i - 1] * 2;
```

```
// 1 1 2 3 5 8 13 21 34 55 – Fibonacci sequence
int[] a = new int[10];
a[0] = 1;
a[1] = 1;
for (var i = 2; i < a.Length; i++)
     a[i] = a[i - 1] + a[i - 2];
```

Some algorithms

```
int[] a = new int[10] { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 };

// sum
var sum = 0;
for (var i = 0; i < a.Length; i++)
     sum += a[i];

// sum in foreach loop
var sum = 0;
foreach (var x in a)
     sum += x;

// Transformation
for (var i = 0; i < a.Length; i++)
     a[i] = a[i] * a[i];
```

Some algorithms

```
// Min/Max
var min = a[0];
for (int i = 1; i < a.Length; i++)
   if (a[i] < min)
       min = a[i];
```

```
 // MinIndex/MaxIndex
var min = a[0];
var ind = 0;
for (int i = 1; i < a.Length; i++)
    if (a[i] < min)
     {
          min = a[i];
          ind = i;
     }
Console.WriteLine($"min = {a[ind]}
```

```
 // MinIndex/MaxIndex - 2
var min = int.MaxValue;
var ind = -1;
for (int i = 0; i < a.Length; i++)
     if (a[i] < min)
      {
           min = a[i];
           ind = i;
      }
```

Some algorithms

```
int[] a = { 1, 2, 7, -3, 4 };
```

```
// Shift left
for (int i = 1; i < a.Length; i++)
        a[i - 1] = a[i];
a[a.Length - 1] = 0;
```

```
a = { 2, 7, -3, 4, 0 };
```

```
// Shift right
for (int i = a.Length - 1; i > 0; i--)
    a[i] = a[i - 1];
a[0] = 0;
```

```
a = { 0, 1, 2, 7, -3 };
```

```
 // Circular Shift left
var x = a[0];
for (int i = 1; i < a.Length; i++)
    a[i - 1] = a[i];
a[a.Length - 1] = x;
```

```
a = { 2, 7, -3, 4, 1 };
```

```
// Circular Shift right
var x = a[a.Length - 1];
for (int i = a.Length - 1; i > 0; i--)
    a[i] = a[i - 1];
a[0] = x;
```

```
a = { 4, 1, 2, 7, -3 };
```

Sorting

```
int[] a = { 1, 2, 7, -3, 4 };
```

-3 1 2 4 7

```
// Bubble sort – 0(n²)
for (var i = 0; i < a.Length - 2; i++)
      for (var j = a.Length - 1; j >= i + 1; j--)
            if (a[j] < a[j - 1])
                  Swap(ref a[j], ref a[j - 1]);
```

```
 // changing the values (swapping)
static void Swap<T>(ref T a, ref T b)
{
     var x = a;
     a = b;
     b = x;
 }
```

```
// Standard sort – 0 (n*log n)
System.Array.Sort(a);

a = a.OrderBy(x => x).ToArray();

a = a.OrderByDescending(x => x).ToArray();
```

Procedure of **Array** class

Methods of sequences

Lecture task

Lesson #7, tasks

https://labs-org.ru/c-sharp7-eng/

User methods

Lesson 8

Extension method Print

```
// How to write your own .Print method

public static class MyFuncs
{
    public static void Print<T>(this T[] a)
    {
        foreach (var x in a)
            Console.Write($"{x} ");
    }
}

class Program
{
    static void Main(string[] args)
    {
        int[] b = { 1, 2, 3, 4, 5 };
        b.Print();
    }
}
```

In static class

The "this" keyword means that we extend this type by this method

One-dimensional arrays (3)

```
public static class MyFuncs
{
    public static T[] ArrGen<T>(int n, Func<int,T> fun)
    {
        var a = new T[n];
        for (var i = 0; i < a.Length; i++)
            a[i] = fun(i);
        return a;
    }
}

class Program
{
    static void Main(string[] args)
    {
        int[] a = MyFuncs.ArrGen(10, i => i * i);
    }
}
```

Type of function
int -> T

Lambda-expression

Lecture task

Lesson #8, task 3

https://labs-org.ru/c-sharp8-eng/

Q & A