**УРОК 22-27: Едномерен масив**

1. **Въведение**

Да се опитаме да решим с програма следната задача:

*Дадени са средните дневни температури, измерени в град София, за 2011 г. Да се намерят десетте най-високи стойности.*

Вече сме решавали задача, при която се въвеждат определен брой числа и се намира най-голямото от тях, чрез преглеждане на всички числа. В случая обаче се иска да изведем не най-голямото число, а десетте най-големи, което налага да се прегледат дадените 365 числа 10 пъти. **Не можем да искаме от потребителя да въвежда данни за програмата повече от веднъж!** Ако някои от въведените данни ни трябват повече от един път, те трябва да се съхраняват в паметта. Абсолютно неразумно би било също така, да декларираме 365 променливи – a1, a2, …, a365 и да пишем отделен код за обработката на всяка от тях.

За решаването на тази, и подобни на нея задачи, в езиците за програмиране се използват **масиви.**

**2. Масив**

**а) същност**

Масивът е област от паметта, съставена от еднотипни полета с еднакъв размер, в които могат да се съхранят редица от стойности от един и същ тип. Масивът има **име**, което е общо за всички полета и **дължина** – броят на полетата. Всяко поле от съответния на масива тип, наричано **елемент** на масива, може да се разглежда и използва в програмата като променлива от същия тип.

Елементите на масива се различават един от друг по своя **индекс** – това поредния номер на елемента в редицата. Номерирането на елементите в C# започва от нула. На Фиг. 1 е показан масив a с 8 елемента от тип int.

В езика С# името на елемент на масив се образува, като след името на масива се постави индексът му, ограден в квадратни скоби: a[0], a[1], …, a[i], …



**б) деклариране на масив**

*<тип>*[] *<име на масива>;*

**Примери:** int[] a;

double[] point;

string[] names;.

int[] a, b, c;.

**в) заделяне на памет за деклариран масив**

*< име на масива > = new <тип>*[*<размер>*]*;*

**Примери:** a = new int[8];

point = new double[2];

Подобно на числените променливи, заделянето на памет може да стане в оператора за деклариране на масива.

**Пример:** int[] a = new int[8];

**г) Инициализация на масив, въвеждане и извеждане на елементите**

Инициализацията на масив се състои в инициализация на елементите му. По подразбиране, елементите на целочислените масиви се инициализират с нули, масивите от тип char – с малки латински букви 'a', а от тип string – с празни низове. Ако потребителят иска друга инициализация, има няколко начина да се направи това:

* **В оператора за деклариране**. При този начин, стойностите на елементите се разделят един от друг със запетая и се заграждат в къдрави скоби. В този случай няма нужда от служебната дума new и задаване на размера. За масива се отделя толкова място в паметта, колкото са елементите в инициализиращия списък.

**Пример:** int[] а = {1, -3, 5, 5, 2, -6, 4, 11};

* **В кода на програмата.** В този случай даването на стойност става с оператор за присвояване за всеки елемент поотделно и не се различава от даването на стойност на променлива.

**Пример:** а[0] = 1; а[1] = -3; а[2] = 5; и т.н. или в цикъл

for(i=0;i< n; i++)

{

array[i] = int.Parse(Console.ReadLine());

}

* **Въвеждат се от потребителя**, по време на изпълнение на програмата.

**Пример:** int n, i;

n = int.Parse(Console.ReadLine());

int[] array = new int[n];

for (int i = 0; i < n; i++)

{

array[i] = int.Parse(Console.ReadLine());

}

В примера, броят на елементите на масива се въвежда в променливата n и след това се декларира самият масив, с размер n.

* **Извеждането на елементите на масив** - не се различава от извеждането на коя да е променлива от същия тип.

**Пример:** for (int i = 0; i < 5; i++)

{

Console.WriteLine(a[i]);

}

**Задача.** *(стр.123, задача)* - Зададени са 10 цели числа в определен ред. Да се въведат в компютъра и да се изведат на конзолата в ред, обратен на реда на въвеждането им.

**Решение:** Очевидно е, че числата трябва да се съхранят в масив като се въведат числата в масива с цикъл, в който управляващата променлива се мени от 0 до 9, а при извеждането – да намалява от 9 до 0.

Програмният фрагмент, с който реализираме първият подход е:

int i, n=10;

int[] array = new int[n];

for (i = 0; i < n; i++)

{

array[i] = int.Parse(Console.ReadLine());

}

for (i = n-1; i >= 0; i--)

{

Console.WriteLine(array[i]);

}

**Дом.: 1)** Дадени са цифрите от 0 до 9. Да се въведат в компютъра и да се изведат на конзолата в ред, обратен на реда на въвеждането им.

**2)** стр.124, зад. 1 - За всеки от програмните фрагменти:

а) int[] a = {1,2,3,4,5,6}, b; б) int[] a = {1,2,3,4,5,6}, b;

b = a[2] + a[3]; b = a[2] + a[6];

определете стойността на b след изпълнението на фрагмента.

**Упражнение 1 (Console App)**

**Задача 1.** стр.123, задача – променяме задачата така, че да се въвежда от клавиатурата броя на числата и самите числа.

**Задача 2.** стр.123, задача – добавяме да извежда само положителните числа в ред обратен на въвеждането им.

**Задача 3.** стр.123, задача – добавяме да извежда броя на положителните числа.

**Задача 4.** стр.123, задача – добавяме да извежда сумата на положителните числа.

**Решение за зад. 1-4:**

int i, br=0, sum=0;

Console.Write("Enter the number of numbers: ");

int n=int.Parse(Console.ReadLine());

int[] array = new int[n];

for (i = 0; i < n; i++)

{

array[i] = int.Parse(Console.ReadLine());

}

for (i = n-1; i >= 0; i--)

{

if (array[i]>0)

{

Console.WriteLine(array[i]);

br=br+1;

sum=sum+array[i];

}

}

Console.WriteLine("Broi: " + br);

Console.WriteLine("Sumata e: " + sum);

**Дом.:** Дадени са n цели числа. Напишете конзолно приложение, което въвежда от клавиатурата броя на числата и самите числа и извежда в конзолата само отрицателните числа, техният брой и сумата им.

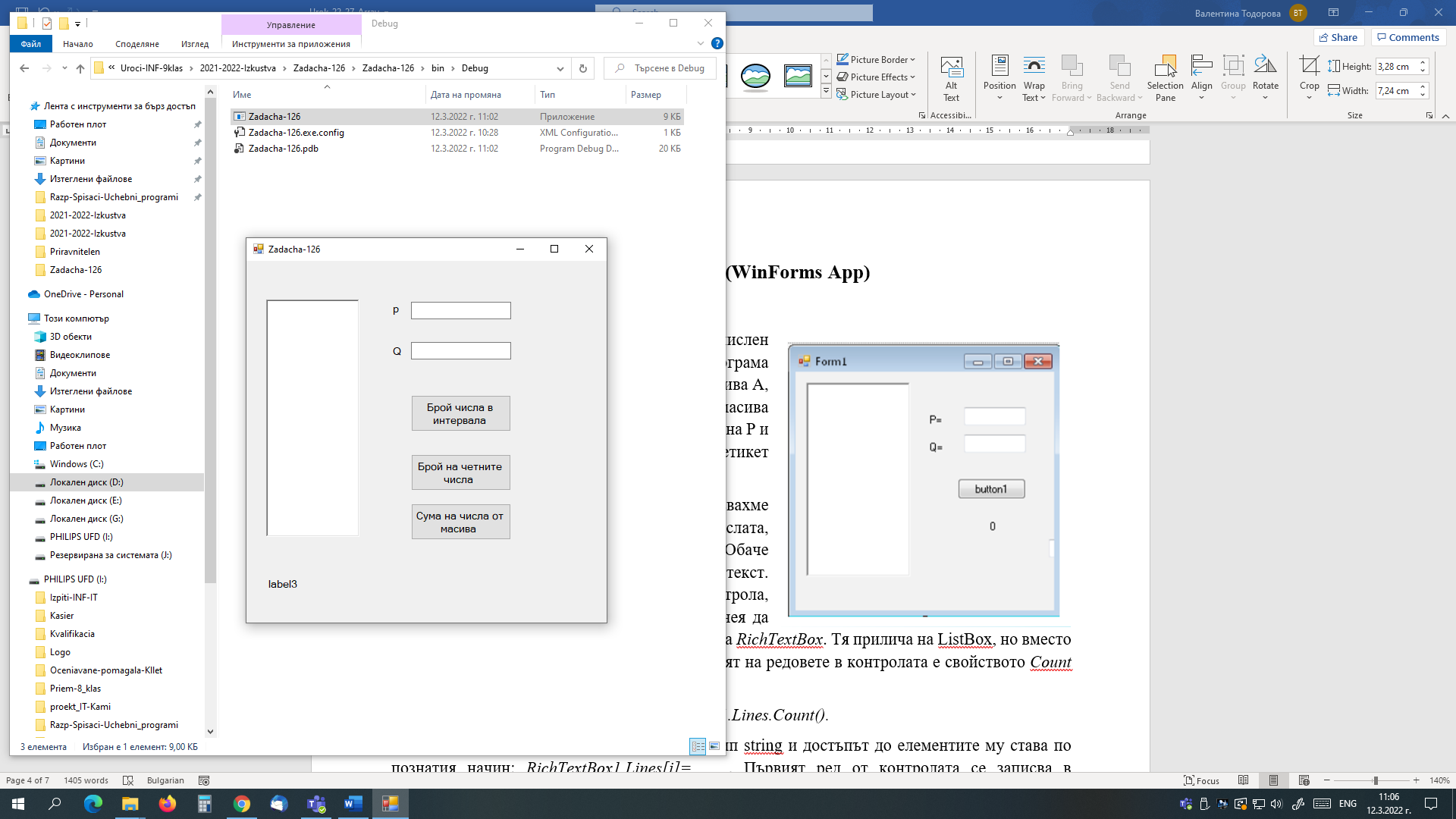
**Упражнение 2 (WinForms App)**

**Задача 1.** стр. 126, задача - Даден е целочислен масив А и две цели числа P и Q. Напишете програма на С#, която да намери броя на числата от масива А, които са в интервала [P;Q]. Стойностите на масива да се въведат в списъчна кутия, а стойностите на P и Q – в TextBox. Резултатът да се изведе в етикет Label.

***Анализ***. В предишни часове използвахме списъчната кутия ListBox за въвеждане на числата, като броят им беше предварително фиксиран. Обаче в ListBox може само да се изписва текст. Следователно, трябва да използваме друга контрола, която е подобна на списъчната кутия, но в нея да може да въвеждаме данни. Такава е контролата *RichTextBox*. Тя прилича на ListBox, но вместо Items, в нея се използва свойството *Lines*. Броят на редовете в контролата е свойството *Count* на Lines и се изписва по този начин:

*RichTextBox1.Lines.Count().*

Lines, както и Items, е едномерен масив от тип string и достъпът до елементите му става по познатия начин: *RichTextBox1.Lines[i]= …* . Първият ред от контролата се записва в RichTextBox1.Lines[0], следователно ако контролата има 10 реда, т.е. RichTextBox1.Lines.Count()=10, то последният ред ще се запише в RichTextBox1.Lines[9].

**Задача 2.** Да се изведе броят на четните числа в масива.

**Задача 3.** Да се изведе сумата на числата в масива.

**Задача 4.** Да се изведе броят на отрицателните числа в масива.

**Задача 5.** Да се изведат числата в масива, които са извън интервала [P;Q].

**Задача 6.** Да се изведат числата в масива, които са равни на P.

**Задача 7.** Да се изведе най-малкият индекс на елемент от масива, чиято стойност е равна на Q. Ако няма такава, в етикета да се изведе –1.

**Решение: зад. 1 – 7**

|  |  |
| --- | --- |
|  |  |
|  |

**Задача 8.** Да се изведе най-малкият елемент на масива.

***Указание***. За да спестите време, ви препоръчваме да копирате част от програмния код за обработката на първия бутон и да го поставите в кода за обработката на втори бутон и след това да направите нужните промени в кода така, че да решават новата задача. Освен това, не е необходимо при натискане на всеки бутон да се създава масивът. По-добре е да се добави бутон "Масив", който след като се въведат стойности в списъчното поле RichTextBox, да ги запише в масива А. След това може да се напише обработката на събитието Click за бутоните на задачите. По този начин програмата ще е по-кратка

**Дом.4: 1.** Да се изведе броят на нечетните числа в масива.

**2.** Да се изведе произведението на числата в масива.

**3.** Да се изведе броят на положителните числа в масива.

**Дом.5: 1.** Да се изведат числата в масива, които са равни на Q.

**2.** Да се изведе най-малкият индекс на елемент от масива, чиято стойност е равна на P. Ако няма такава, в етикета да се изведе –1.

**3.** Да се изведе най-големият елемент на масива.