**Практическая работа 4**

**Тема: «Разработка и отладка модуля обработки элементов массива»**

# Постановка задачи

Для конкретного варианта ввести массив исходных данных и выполнить над ним указанные действия. Изучив алгоритмы упорядочения, выбрать один из них. Написать программу, которая работает с любым набором данных. Входную информацию и результаты счета вывести на печать, снабдив их соответствующими заголовками.

# Общие сведения

Массив – это совокупность переменных одного типа, к которым обращаются с помощью общего имени. Доступ к отдельному элементу массива может осуществляться с помощью индекса. В языке С# все массивы состоят из соприкасающихся участков памяти. Наименьший адрес соответствует первому элементу, наибольший адрес соответствует последнему элементу. Массивы могут иметь одну или несколько размерностей.

**Одномерные массивы**

Одномерный массив – это фиксированное количество элементов одного и того же типа, объединенных общим именем, где каждый элемент имеет свой номер. Нумерация элементов массива в С# начинается с нуля, то есть, если массив состоит из 10 элементов, то его элементы будут иметь следующие номера: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Одномерный массив в С# реализуется как объект, по этому его создание представляет собой двухступенчатый процесс. Сначала объявляется ссылочная переменная на массив, затем выделяется память под требуемое количество элементов базового типа, и ссылочной переменной присваивается адрес нулевого элемента в массиве. Базовый тип определяет тип данных каждого элемента массива. Количество элементов, которые будут храниться в массиве, определяется размер массива.

В общем случае процесс объявления переменной типа массив, и выделение необходимого объема памяти может быть разделено. Кроме того, на этапе объявления массива можно произвести

его инициализацию. Поэтому для объявления одномерного массива может использоваться одна из следующих форм записи:

|  |  |
| --- | --- |
| ***Форма записи*** | ***Пояснения*** |
| базовый\_тип [] имя\_\_массива;    *Например*:  int [] a; | Описана ссылка на одномерный массив, которая в дальнейшем может быть использована:   1. для адресации на уже существующий массив; 2. передачи массива в метод в качестве параметра 3. отсроченного выделения памяти под элементы массива. |
| базовый\_тип [] имя\_\_массива = new базовый\_тип [размер];    *Например*:  int []a=new int [10]; | Объявлен одномерный массив заданного типа и выделена память под одномерный массив указанной размерности. Адрес данной области памяти записан в ссылочную переменную. Элементы массива равны нулю.  ***Замечание***. Надо отметить, что в C# элементам массива присваиваются начальные значения по умолчанию в зависимости от базового типа. Для арифметических типов – нули, для ссылочных типов – null, для символов - пробел. |
| базовый\_тип [] имя\_\_массива={список инициализации};    *Например*:  int []a={0, 1, 2, 3}; | Выделена память под одномерный массив, размерность которого соответствует количеству элементов в списке инициализации. Адрес этой области памяти записан в ссылочную переменную. Значение элементов массива соответствует списку инициализации. |

Обращения к элементам массива происходи с помощью индекса, для этого нужно указать имя массива и в квадратных скобках его номер. Например, a[0], b[10], c[i].

***Замечание****. В C# индексация массивов начинается с нуля.*

Так как массив представляет собой набор элементов, объединенных общим именем, то обработка массива обычно производится в цикле. Рассмотрим несколько простых примеров работы с одномерными массивами.

***Пример 1.***

static void Main()

{

int[] myArray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }; int i;

for (i = 0; i < 10; ++i)

Console.WriteLine(myArray[i]);

}

**Задание**. Измените программу так, чтобы числа выводились в строчку.

***Пример 2.***

static void Main()

{

int[] myArray = new int[10]; int i;

for (i = 0; i < 10; i++) myArray[i] = i \* i; for (i = 0; i < 10; i++)

Console.WriteLine(myArray[i]); }

**Задание**. Измените программу так, чтобы обрабатывался массив из n чисел.

Хотя при инициализации массива нет необходимости использовать операцию new, все же массив можно инициализировать следующим образом:

int [ ] myArray = new int [ ] { 99, 10, 100, 18, 78, 23, 163, 9, 87, 49 };

Несмотря на избыточность, данная форма инициализации массива может оказаться полезной в том случае, когда уже существующей ссылке на одномерный массив присваивается ссылка на новый массив. Например:

static void Main()

{

int[] myArray = { 0, 1, 2, 3, 4, 5}; int i;

for (i = 0; i < 10; i++)

Console.Write(" "+myArray[i]);

Console.WriteLine("\nНовый массив: ");

myArray = new int[] { 99, 10, 100, 18, 78, 23, 163, 9, 87, 49 }; // 1 for (i = 0; i < 10; i++)

Console.Write(" " + myArray[i]);

}

Следует отметить, что первоначально переменная myArray ссылалась на 6-ти элементный массив. В строке 1 переменной myArray была присвоена ссылка на новый 10элементный массив, в результате чего исходный массив оказался неиспользуемым, т.к. на него теперь не ссылается ни один объект. Поэтому он автоматически будет удален сборщиком мусора.

*Массивы и исключения* Выход за границы массива в C# расценивается как ошибка, в ответ на которую генерируется исключение - IndexOutOfRangeException.

Рассмотрим следующий пример:

static void Main()

{

int[] myArray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }; int i; try

{

for (i = 0; i <= 10; i++) Console.WriteLine(myArray[i]);

}

catch (IndexOutOfRangeException)

{

Console.WriteLine("Exception: Выход за границу диапазона");

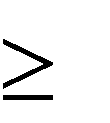
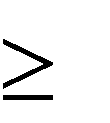
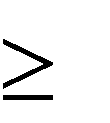
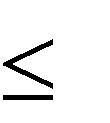
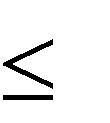
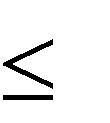
}

}

**Задание**. Добавьте в программу обработчики исключений FormatException и OutOfMemoryException. Вспомните, что они контролируют.

## 5.4 Сортировка

Рассмотрим массив целых или вещественных чисел *a1, a2, …, an* . Пусть требуется переставить элементы этого массива так, чтобы после перестановки они были упорядочены по неубыванию *a1* *a2* … *an* или по невозрастанию *a1* *a2* … *an* . Эта задача называется задачей сортировки или упорядочения массива. Для решения этой задачи можно воспользоваться, например, следующими алгоритмами:



а) Найти элемент массива, имеющий наименьшее (наибольшее) значение, переставить его с первым элементом. Затем проделать то же самое, начав со второго элемента и так далее.

(Сортировка выбором)

б) Последовательным просмотром чисел *a1, a2, …, an* найти наименьшее *i* такое, что *ai* > *ai+1* или *ai* < *ai+1* . Поменять *ai* и *ai+1* местами, возобновить просмотр с элемента *ai+1* и так далее. Тем самым самое наибольшее или наименьшее число передвинется на последнее место. Следующие просмотры следует начинать опять сначала, уменьшая на единицу количество просматриваемых элементов. Массив будет упорядочен после просмотра, в котором участвовали только первый и второй элементы. (Сортировка обменами)

в) Просматривать последовательно *a2, …, an* и каждый новый элемент вставлять на подходящее место в уже упорядоченную последовательность *a1*, …, *ai-1* . Это место определяется последовательным сравнением *ai* с упорядоченными элементами *a1*, …, *ai-1*

. (Сортировка простыми вставками)

г) Сравнить элементы *a1* и *a2*и, если *a1* > *a2* (или *a1*< *a2*), то эти элементы переставить. Далее сравнить элементы *а2*и *а3* и, если *а2* > *a3* (или *а2* < *a3*), то их переставить. Далее сравнить элементы *а3* и *а4* и так далее до элементов *an-1* и *an* включительно. Далее эти действия повторить, начиная опять с первого элемента. Последним является контрольный проход, при котором не будет перестановок элементов. (Сортировка по методу пузырька)

**Самостоятельная работа**

# Вариант 1

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) сумму отрицательных элементов массива;

2) произведение элементов массива, расположенных между максимальным и минимальным элементами. Упорядочить элементы массива по возрастанию.

# Вариант 2

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. сумму положительных элементов массива;
2. произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами. Упорядочить элементы массива по убыванию.

# Вариант 3

В одномерном массиве, состоящем из *n* целых элементов, вычислить:

1. произведение элементов массива с четными номерами;
2. сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные 0, считать положительными).

# Вариант 4

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) сумму элементов массива с нечетными номерами;

2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

# Вариант 5

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. максимальный элемент массива;
2. сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале [a,b]. Освободившиеся в конце массива элементы заполнить нулями.

# Вариант 6

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) минимальный элемент массива;

2) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.

# Вариант 7

В одномерном массиве, состоящем из *n* целых элементов, вычислить:

1. номер максимального элемента массива;
2. произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.

# Вариант 8

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. номер минимального элемента массива;
2. сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом — все остальные.

# Вариант 9

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. максимальный по модулю элемент массива;
2. сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

# Вариант 10

В одномерном массиве, состоящем из *n* целых элементов, вычислить:

1. минимальный по модулю элемент массива;
2. сумму модулей элементов массива, расположенных после первого элемента, равного нулю. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине — элементы, стоявшие в нечетных позициях.

# Вариант 11

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. номер минимального по модулю элемента массива;
2. сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Сжать массив, удалив из него все элементы, величина которых находится в интервале [a,b]. Освободившиеся в конце массива элементы заполнить нулями.

# Вариант 12

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. номер максимального по модулю элемента массива;
2. сумму элементов массива, расположенных после первого положительного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале [a,b], а потом — все остальные.

# Вариант 13

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. количество элементов массива, лежащих в диапазоне от А до В;
2. сумму элементов массива, расположенных после максимального элемента. Упорядочить элементы массива по убыванию модулей элементов.

# Вариант 14

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. количество элементов массива, равных 0;
2. сумму элементов массива, расположенных после минимального элемента. Упорядочить элементы массива по возрастанию модулей элементов.

# Вариант 15

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) количество элементов массива, больших С;

2) произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом — все положительные (элементы, равные 0, считать положительными).

# Вариант 16

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) количество отрицательных элементов массива;

2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

# Вариант 17

В одномерном массиве, состоящем из *n* целых элементов, вычислить:

1. количество положительных элементов массива;
2. сумму элементов массива, расположенных после последнего элемента, равного нулю. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом — все остальные.

# Вариант 18

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить: 1) количество элементов массива, меньших С;

2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом — все остальные. **Вариант 19**

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. произведение отрицательных элементов массива;
2. сумму положительных элементов массива, расположенных до максимального элемента. Изменить порядок следования элементов в массиве на обратный.

# Вариант 20

В одномерном массиве, состоящем из *n* вещественных элементов, вычислить:

1. произведение положительных элементов массива;
2. сумму элементов массива, расположенных до минимального элемента.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.