

Лекция 14.

Массивы - операции с массивами (размер изменяемый)

Понятие массива

2

- Массив - структура данных, позволяющая хранить под одним именем совокупность данных любого, но только одного какого-то типа
- С массивом связаны следующие его свойства:
 - имя;
 - тип;
 - размерность;
 - размер.

Виды массивов

3

- Массивы в C#:
 - одномерные массивы;
 - многомерные массивы;
 - массивы массивов (ступенчатые, зубчатые массивы).

Одномерные массивы

Объявление. Инициализация

4

- Объявление:

```
<тип>[ ] <объявители>;
```

```
int[ ] a, b, c;
```

- Объявление с инициализацией:

- Явная инициализация:

```
double[ ] x = {5.5, 6.6, 7.7};
```

- Инициализация с вызовом конструктора массива (ноль для арифметики, пустая строка для строковых массивов):

```
int[ ] d = new int[5];
```

Одномерные массивы

Инициализация

5

- Инициализация массива:

- с использованием ключевого слова **new**:

```
int[] myArr = new int[] {10,20,30,40,50};
```

- без использования ключевого слова **new**:

```
string[] info = { "Фамилия", "Имя", "Отчество" };
```

- с использованием **new** и указанием **размера**:

```
char[] symbol = new char[4] { 'X','Y','Z','M' };
```

Этапы решения

6

- Сокращенная форма:
 - ввод исходных данных;
 - решение задачи;
 - вывод результата на экран.
- Развернутая форма для задач с массивами:
 - объявление массива;
 - выделение памяти для необходимого количества элементов массива;
 - определение значений элементов массива;
 - вывод исходного массива на экран;
 - выполнение алгоритма;
 - вывод результата на экран.

3. Удаление элементов массива

7

3.1 Удалить последний нулевой элемент массива

Алгоритм:

А) в цикле for просматриваем все элементы массива с конца (т.к. нужно найти последний элемент) и если текущий элемент равен нулю, то прерываем выполнение цикла

Б) в цикле for начиная с текущего элемента массива ($i = i$ или ничего не указываем) и до предпоследнего ($< n-1$) присваиваем значению текущего элемента массива значение следующего элемента

В) изменить размер массива

Значение	1	0	-2	5	1	3	7
Индекс	0	i	$i+1$			$n-2$	$n-1$



Козьминых Н.М.

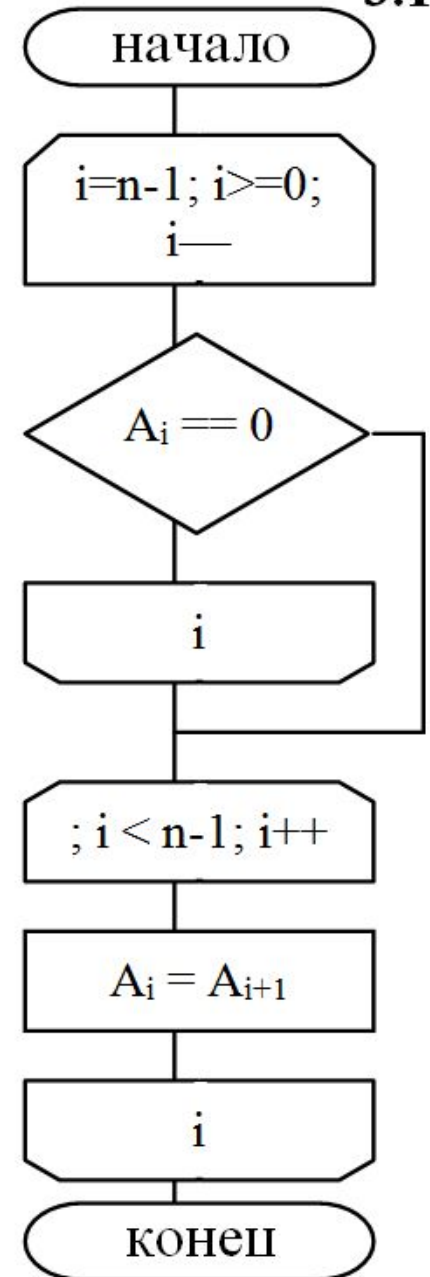
3. Удаление элементов массива

8

3.1 Удалить последний нулевой элемент массива

```
void task() {  
    int i;  
    for (i = A.Length - 1; i >= 0; i--)  
        if (A[i] == 0)  
            break;  
    if (i == -1)  
        return;  
    for (; i < A.Length; i++)  
        A[i] = A[i+1];  
    Array.Resize(ref A, A.Length - 1);  
}
```

3.1



3. Удаление элементов массива

9

3.2 Удалить минимальный положительный элемент массива

Исходный массив

	1	2	3	4	5	6
▶	-7	1	5	8	9	-8

Результат

	1	2	3	4	5
▶	-7	5	8	9	-8

3. Удаление элементов массива

10

3.2 Удалить минимальный положительный элемент массива

Алгоритм:

А) находим минимальный положительный элемент

1) переменной признаку k присваиваем -1 : нет элемента

2) в цикле `for` начиная с начала;

если текущий элемент меньше нуля, переходим к следующей итерации цикла (т.к. нам нужно только положительные);

если положительных элементов не было ($k == -1$) или текущий элемент меньше минимального из рассмотренных, то в переменную индекса записываем индекс текущего минимального элемента

Б) в цикле `for` начиная с найденного элемента массива ($i = k$) и до предпоследнего ($< n-1$) присваиваем значению текущего элемента массива значение следующего элемента

В) изменить размер массива

3. Удаление элементов массива

11

3.2 Удалить минимальный положительный элемент массива

```
void task() {  
    int i, k;  
    k = -1;  
    for (i = 0; i < A.Length ; i++) {  
        if (A[i] < 0) continue;  
        if (k == -1 || A[i] < A [k]) k = i;  
    }  
    if (k == -1)  
        return;  
    for (i = k; i < A.Length - 1; i++)  
        A[i] = A[i+1];  
    Array.Resize(ref A, A.Length - 1);  
}
```

3. Удаление элементов массива

12

3.3 Удалить минимальный положительный элемент массива

Исходный массив						
	1	2	3	4	5	6
▶	-7	1	5	8	9	-8

Результат					
	1	2	3	4	5
▶	-7	5	8	9	-8

3. Удаление элементов массива

13

3.3 Удалить минимальный положительный элемент массива

Алгоритм:

А) найти первый положительный элемент

Б) найти минимальный положительный, начиная со следующего после найденного положительного элемента

В) удалить найденный элемент

Г) изменить размер массива

3. Удаление элементов массива

14

3.3 Удалить минимальный положительный элемент массива

```
private void task(ref int[] b){
    int imin = -1;
    for(int i = 0; i < b.Length; i++){
        if (b[i] > 0){
            imin = i;
            break;
        }
    }
    if (imin == -1)
        return;
    for(int i = imin + 1; i < b.Length; i++){
        if (b[i] > 0 && b[i] < b[imin]){
            imin = i;
        }
    }
    for(int i = imin; i < b.Length - 1; i++){
        b[i] = b[i + 1];
    }
    Array.Resize(ref b, b.Length - 1);
}
```

3. Удаление элементов массива

15

3.4 Удалить все положительные элементы массива

Исходный массив

	1	2	3	4	5	6
▶	-7	1	5	8	9	-8

Результат

	1	2
▶	-7	-8

3. Удаление элементов массива

16

3.4 Удалить все положительные элементы массива

Алгоритм:

А) внешний цикл - найти положительный элемент

Б) удалить найденный элемент

Г) изменить размер массива

3. Удаление элементов массива

17

3.4 Удалить все положительные элементы массива

```
private void task(ref int[] b) {  
    int k = 0;  
    for(int i = 0; i < b.Length - k; i++)  
        if (b[i] > 0) {  
            for (int j = i; j < b.Length - 1 - k; j++)  
                b[j] = b[j + 1];  
            k++;  
            i--;  
        }  
    Array.Resize(ref b, b.Length - k);  
}
```

3. Удаление элементов массива

18

3.4 Удалить все положительные элементы массива (вариант №2)

```
private void task (ref int[] b){
    for(int i = 0; i < b.Length; i++)
        if (b[i] > 0) {
            for (int j = i; j < b.Length - 1; j++)
                b[j] = b[j + 1];
            Array.Resize(ref b, b.Length - 1);
            i--;
        }
}
```

4. Вставка элементов массива

19

4.1. Вставить заданное число перед последним положительным элементом ($z = 6$)

Исходный массив

	1	2	3	4	5	6	
▶	-9	-9	5	-6	-4	-10	

Результат

	1	2	3	4	5	6	7	
▶	-9	-9	6	5	-6	-4	-10	

4. Вставка элементов массива

4.1. Вставить заданное число перед последним положительным элементом

Алгоритм:

А) вводим значение заданного числа, которое будем вставлять

Б) в цикле for просматриваем все элементы массива с конца (т.к. нужно найти последний элемент) и если текущий элемент больше нуля (положительный), то прерываем выполнение цикла

В) если элемент не найден, преждевременно выходим из цикла

Г) изменяем размер массива

Д) в цикле for, начиная с позиции после последней ($i = n-1$) и до позиции последнего положительного элемента, не включая его ($> i$), с шагом -1 присваиваем значению текущего элемента массива значение предыдущего элемента

Е) вставляем в позицию перед последним положительным элементом заданное число ($A_i = z$)

4. Вставка элементов массива

21

4.1. Вставить заданное число перед последним положительным элементом

```
private void task (ref int[] b)    {
    int i, j, z;
    z = Convert.ToInt32(txtZ.Text);
    for (i = b.Length- 1; i >= 0; i--)
        if (b[i] > 0)
            break;
    if (i == -1)
        return;
    Array.Resize(ref b, b.Length + 1);

    for (j = b.Length - 1; j > i; j--)
        b[j] = b[j-1];
    b[i] = z;
}
```

4. Вставка элементов массива

22

4.2. Вставить значение минимального элемента перед и после максимального элемент

Исходный массив

	1	2	3	4	5	6		
▶	8	-1	4	-8	8	5		

Результат

	1	2	3	4	5	6	7	8
▶	-8	8	-8	-1	4	-8	8	5

4. Вставка элементов массива

4.2. Вставить значение минимального элемента перед и после максимального элемента

Алгоритм:

- А) присваиваем переменной, в которой будем хранить минимальное значение массива, значение первого элемента массива ($\min = A_0$)
- Б) присваиваем переменной, в которой будем хранить индекс максимального элемента массива, индекс первого элемента ($\max = 0$)
- В) в цикле for просматриваем все элементы массива с начала
 - 1) если значение текущего элемента массива больше значения элемента с индексом \max , то запоминаем текущий индекс как индекс максимального элемента
 - 2) если значение текущего элемента массива меньше переменной, в которой храниться минимальное значение массива (\min), то присваиваем этой переменной значение текущего элемента
- Г) изменяем размер массива
- Д) в цикле for сдвигаем элементы массива на две позиции вправо
- Е) в позицию после максимального элемента записываем значение максимального элемента массива
- Ж) в позицию минимального элемента и через позицию записываем значение минимального элемента массива

4. Вставка элементов массива

4.2. Вставить значение минимального элемента перед и после максимального элемента

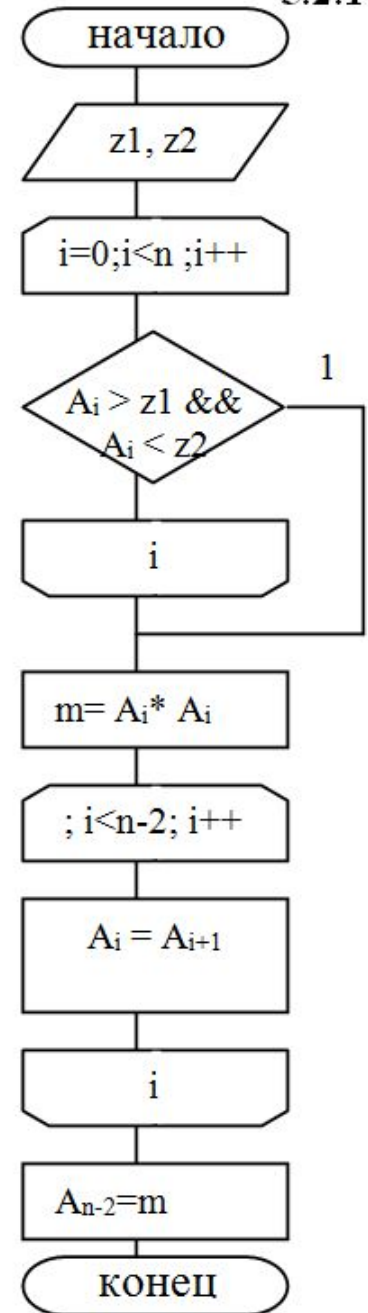
```
private void task(ref int[] b) {
    int i, min, imax;
    min = b[0];
    imax = 0;
    for (i = 0; i < b.Length; i++) {
        if (b[i] > b[imax])
            imax = i;
        if (b[i] < min)
            min = b[i];
    }
    Array.Resize(ref b, b.Length + 2);
    for (i = b.Length - 1; i >= imax + 2; i--)
        b[i] = b[i - 2];
    b[imax + 1] = b[imax];
    b[imax] = min;
    b[imax + 2] = min;
}
```


5. Изменение местонахождения элемента с помощью сдвига

25

5.1 Найти первый элемент, находящийся в заданных пределах, возвести его в квадрат и поставить перед последним элементом

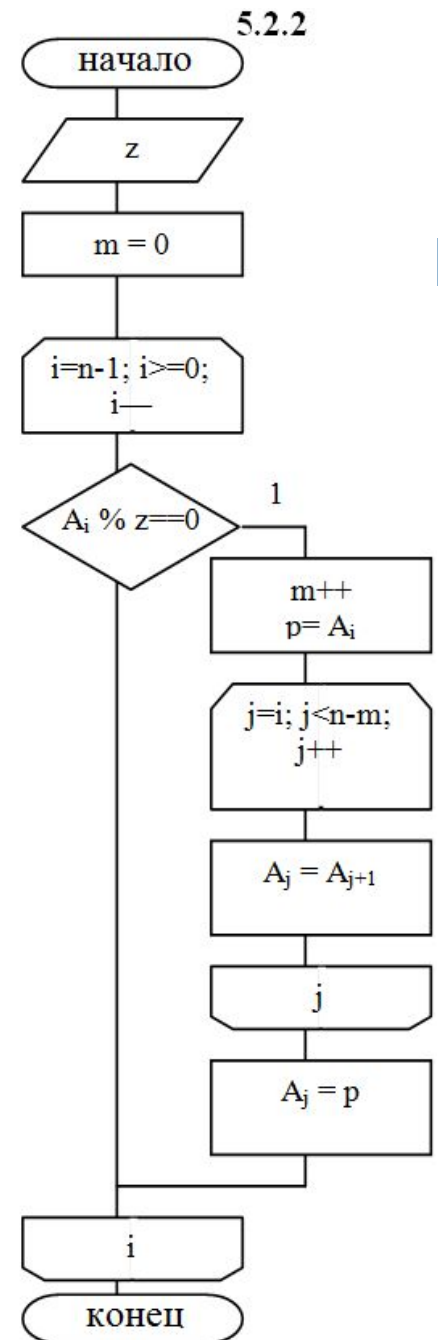
5.2.1



5. Изменение местонахождения элемента с помощью сдвига

26

5.2 Переставить в конец массива элементы, кратные заданному числу



27

Спасибо за внимание

Вопросы...