

## Практическое занятие № 1 «Описание и работа с одномерными массивами в языке C++»

### Учебные цели:

- получение умений и навыков создания и отладки программ для обработки одномерных массивов;
- получение умений и навыков создания и отладки программ с циклической структурой.

### Воспитательные цели:

- формировать диалектико-материалистическое мировоззрение;
- формировать навыки самостоятельности и дисциплинированности;
- стимулировать активную познавательную деятельность обучающихся, способствовать формированию у них творческого мышления.

**Категория слушателей:** 2,3 курс.

**Время:** 90 мин.

**Место проведения:** компьютерный класс.

### Материально-техническое обеспечение:

1) персональный компьютер *IBM PC* с операционной системой Windows XP; 2) среда разработки приложений *Visual C++.NET*.

### ПЛАН ПРАКТИЧЕСКОГО ЗАНЯТИЯ

Учебные вопросы	Время, мин
Вступительная часть . . . . .	5
1. Одномерные массивы в C++. . . . .	20
2. Выполнение индивидуального задания . . . . .	60
Заключительная часть . . . . .	5

### Элементы теории

**Массив** представляет собой переменную, содержащую упорядоченный набор данных одного типа. К каждому элементу массива можно получить доступ по его адресу. В языках C/C++ массив не является стандартным типом данных. Напротив, он сам имеет тип: char, int., float, double и т. д. Существует возможность создавать массивы массивов, указателей, структур и др.

Объявляя массив, необходимо сначала указать тип хранимых данных, а затем имя массива и его размер. *Размером массива* называется количество его элементов, указываемое в квадратных скобках.

Например, `int I_Array[25]` – здесь, объявлен массив `I_Array` из 25-ти элементов типа `int`. Обнаружив это объявление, компилятор зарезервирует область памяти достаточную для хранения 25-ти переменных типа `Int`.

**Инициализация по умолчанию.** В соответствии со стандартом ANSI глобальные массивы (расположенные вне любой функции), а также массивы, объявленные статическими внутри функции, по умолчанию заполняются нулями, если не заданы начальные значения элементов массива.

**Явная инициализация.** В соответствии со стандартом ANSI элементам глобальных и локальных массивов можно явно присваивать начальные значения. Небольшой массив переменных встроенных типов (например, `int` или `char`) можно инициализировать при объявлении. Для этого после имени массива помещают знак равенства (=) и заключенный в фигурные скобки список значений, отделяемые запятой.

Например: `int I_Array[5] = {1, 2, 3, 4, 5};`

Здесь объявлен массив `I_Array` из пяти целочисленных элементов, которым присвоены значения `I_Array [0] – 1, I_Array [1] – 2` и т.д.

Если размер массива не указан, но список значений присутствует, то будет создан и инициализирован массив достаточного размера, чтобы содержать все перечисленные значения. Таким образом, эта строка аналогична предыдущей: `int I_Array[] = {1, 2, 3, 4, 5};`

Нельзя инициализировать количество элементов, превосходящее объявленный размер массива: `int I_Array[5] = {1, 2, 3, 4, 5, 6};`

Такая строка приведет к ошибке во время компиляции, поскольку объявлен массив для пяти элементов, а инициализировать пытались шесть. Но следующая запись вполне допустима: `int I_Array[5] = {1, 2};`

В данном случае объявлен массив из пяти элементов, а инициализированы только первые два: `I_Array[0]` и `I_Array[1]`.

**Доступ к элементам массива.** К каждому из элементов можно обратиться по его номеру, расположенному в квадратных скобках после имени массива. Номера элементов массива начинаются с *нуля*. Следовательно, первым элементом массива `I_Array` будет `I_Array [0]`. вторым – `I_Array [1]` и т.д.

### Вопросы для самопроверки

1. Дайте определение понятию массив.
2. Как обратиться к элементу массива?
3. Как присвоить элементу массива новое значение?
4. Инициализация массива по умолчанию.

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОТРАБОТКЕ УЧЕБНЫХ ВОПРОСОВ

## Практическое задание

**Задание 1.** В соответствии с Вашим вариантом составить и отладить программу для решения следующей задачи обработки одномерных массивов:

1. Исключить из массива  $A_1..A_N$  первый отрицательный элемент.
2. Исключить из массива  $A_1..A_N$  первый четный элемент, следующий за максимальным.
3. Дан массив целых чисел  $A_1..A_N$ . Выяснить, какая из трех ситуаций имеет место: все числа  $A_1..A_N$  равны нулю, в последовательности  $A_1..A_N$  первое ненулевое число - положительное, первое ненулевое число - отрицательное.
4. Дан массив целых чисел  $A_1..A_N$ . Выяснить, какая из трех ситуаций имеет место: все числа  $A_1..A_N$  равны нулю, в последовательности  $A_1..A_N$  первое ненулевое число - положительное, первое ненулевое число - отрицательное.
5. Даны целые числа  $A_1..A_N$ . Определить количество целых чисел, входящих в последовательность  $A_1..A_N$  по одному разу.
6. Дан массив действительных чисел  $A_1..A_N$ . Найти максимальный элемент среди отрицательных элементов и поменять его местами с минимальным положительным.
7. Перенести в начало одномерного массива второй нулевой элемент.
8. В одномерном массиве перенести в конец минимальный элемент.
9. Перенести в хвост одномерного массива все отрицательные элементы.
10. Перенести в начало одномерного массива все нечетные элементы.

*Примечание:* задание 1 решить двумя способами без привлечения дополнительного массива и при помощи дополнительного массива.

Составим программный код на языке C++ для решения следующей задачи обработки массива: все отрицательные элементы массива меньше -10 заменить 0 и полученный массив вывести на печать. Для решения поставленной задачи воспользуемся дополнительным массивом (см. Листинг П.1).

## Листинг П. 1.

```
#include "stdafx.h"
#include <iostream>

int main()
{
    using namespace std;
    int i,k,M[15],M_new[15];
    cout << "Vvedite kolichestvo eleventov massiva M. k=";
    cin >> k;
    for (i=1;i<=k;i++)
    {
        cout << "Vvedite M["<<i<<"]="; // Vvod elementov odnomernogo
        cin >> M[i];                    // massiva
    }
    cout << endl;
    for (i=1;i<=k;i++)
        cout << M[i]<<endl; // Vyvod elementov odnomernogo
                                // massiva kak vektor-stolbec
    cout << endl;
    for (i=1;i<=k;i++)
        cout << M[i]<<" \t"; // Vyvod elementov odnomernogo
                                // massiva kak vektor-stroka
    cout << endl;

    for (i=1;i<=k;i++)
        if (M[i]<-10) M_new[i]=0;
        else M_new[i]=M[i];
    cout << "Obrabotanny massiv" << endl;
    for (i=1;i<=k;i++)
        cout << M_new[i] << "\t"; // Vyvod elementov novogo
    cout << endl;                    // massiva
    return 0;
}
```

## Контрольные вопросы

1. Как узнать размерность массива?
2. Явная инициализация массива.
3. Как в одномерном массиве посчитать количество нулевых элементов?
4. Как в одномерном массиве посчитать сумму положительных элементов?
5. Как в одномерном массиве поменять первый и последний элемент местами?
6. Как из массива исключить максимальный элемент?
7. Как в массиве посчитать произведение всех нечетных элементов?