

МГКИП

Лабораторная работа № 4

по курсу
«Технические средства информатизации»

Специальность 2203
«Программное обеспечение ВТ и АС»

2001

Тема: Работа с массивами

Цель работы: Изучение использования массивов в C++

Технические средства: IBM PC 80486, HDD 200 Мб, ОЗУ 8 Мб

Программные средства: MS-DOS 6.22
Norton Commander 5.0 Rus
Borland C++ 3.1

Краткие теоретические сведения

Массив – это переменная, в которой можно хранить множество значений одного и того же типа. Для объявления массива необходимо указать требуемый тип данных и размер массива. Для указания размера массива количество его элементов помещается в квадратные скобки. Количество квадратных скобок указывает мерность массива. Массивы можно объявлять глобально и локально, а также статически и динамически.

Поскольку размер оперативной памяти ограничен моделью памяти, необходимо быть уверенным, что модель памяти позволит разместить массив в ОП. Для этого используется оператор `sizeof`.

Инициализация массива для символьных массивов может быть проведена с помощью строки.

Для инициализации массива любого типа необходимо указать присвоение заключенного в фигурных скобках значений. Для многомерных массивов присвоение происходит построчно (по первому измерению), причем каждую строку и измерение заключают в отдельные фигурные скобки.

Доступ к элементам массива указывается имя массива, а затем номер требуемого элемента в квадратных скобках (для многомерных массивов указываются индексы по всем измерениям в отдельных квадратных скобках). Нумерация элементов массива начинается с нуля и номер максимального элемента на единицу меньше числа элементов по данному измерению.

Структуры и массивы позволяют хранить информацию и объединять ее в группы. Массив может входить в состав структуры, а та же структура может являться типом массива. Пример:

```
struct Employee {
    char name [64];
    int age;
    char ssan [];
    int pay_grade;
    float salary;
    unsigned employee_number;
    struct Date {
        int month;
        int day;
        int year;
    } emp_dates [3];
} stuff [100];
```

Доступ:

```
stuff[10].emp_dates[0].month = 7;
stuff[10].emp_dates[0].day = 7;
```

```
staff[10].emp_dates[0].year = 7;
```

Если в программе используется только то или иное поле в зависимости от значения выгоднее использовать объединения. Пример:

```
struct EmpDates {  
    int days_worked;  
    struct LastDate {  
        int month;  
        int day;  
        int year;  
    } last_day;  
};
```

заменяется на

```
union EmpDates {  
    int days_worked;  
    struct LastDate {  
        int month ;  
        int day;  
        int year;  
    } last_day;  
};
```

Для доступа к массиву можно использовать указатель (объявление `int *p`);
Операция нахождения адреса переменной.

Пример:

```
int p=5;  
int *q;  
q=&p;
```

Доступ с помощью указателя применяется операция звездочка (*):

```
p = *q;  
или  
*q = 7;
```

Арифметические операции над указателями приведут к увеличению значения на размер типа данных, на которые указывает указатель.

Задание

1. Создать одномерный статический локальный массив стандартного типа. Произвести его инициализацию различными методами.
2. Создать двухмерный статический локальный массив стандартного типа. Произвести его инициализацию различными методами.
3. Создать многомерный статический локальный массив стандартного типа. Инициализировать его.
4. Для всех статических массивов обеспечить вывод на экран через унифицированную функцию используя передачу массива по указателю.
5. Создать динамический массив стандартного типа. Инициализировать его, вывести на экран через отдельную функцию.
6. Предложить формат структуры для записи информации о студенте.
7. Реализовать структуру информации о студенте, как тип данных всех массивов.
8. Инициализировать все массивы на базе типа структура через отдельную функцию.
9. Для всех массивов вывести размер массива и адреса массивов.

Контрольные вопросы

1. Как определить количество элементов массива программным путем?
2. Чем ограничена размерность массива?
3. Максимальное значение измерений массива?
4. Как влияет модель памяти на работу с массивами?

Содержание отчета

1. Название работы
2. Цель работы
3. Алгоритм работы программы
4. Листинг программы
5. Отчет работы программы
6. Ответы на контрольные вопросы