**Динамические массивы**

Когда мы объявляем массив, место для него выделяется во время трансляции, т. е. до выполнения программы. Такой массив называется **статическим**.

В то же время иногда размер данных заранее неизвестен. Например, пусть в файле записан массив чисел, которые нужно отсортировать. Их количество неизвестно, но известно, что такой массив помещается в оперативную память.

**В этом случае есть два варианта:**  
  
1) выделить заранее максимально большой блок памяти;  
2) выделять память уже во время выполнения программы (т. е. **динамически**), когда станет известен необходимый размер массива.

**Другой пример** — задача составления алфавитно-частотного словаря. В файле находится список слов. Нужно вывести в другой файл все различные слова, которые встречаются в файле, и определить, сколько раз встречается каждое слово. Здесь проблема состоит в том, что нужный размер массива можно узнать только тогда, когда все различные слова будут найдены и, таким образом, задача решена. Поэтому нужно сделать так, чтобы массив мог «расширяться» в ходе работы программы.

Эти задачи приводят к **понятию динамических структур данных**, которые позволяют во время выполнения программы:  
  
• создавать новые объекты в памяти;  
• изменять их размер;  
• удалять их из памяти, когда они не нужны.

**Память под эти объекты выделяется в специальной области**, которую обычно называют «кучей» (англ. heap).

**Создание динамического массива**

**#include <iostream>**

**using** **namespace** std;

**int** **main**()

{

**int** num; *// размер массива*

cout << "Enter integer value: ";

cin >> num; *// получение от пользователя размера массива*

**int** \*p\_darr = **new** **int**[num]; *// Выделение памяти для массива*

**for** (**int** i = 0; i < num; i++) {

*// Заполнение массива и вывод значений его элементов*

p\_darr[i] = i;

cout << "Value of " << i << " element is " << p\_darr[i] << endl;

}

**delete** [] p\_darr; *// очистка памяти*

**return** 0;

}

Синтаксис выделения памяти для массива имеет вид указатель = new тип[размер]. В качестве размера массива может выступать любое целое положительное значение.

**Вопросы и задания для самостоятельной работы**

1. Приведите примеры задач, в которых использование динамических массивов даёт преимущества (какие именно?).  
2. Что такое динамические структуры данных? Где выделяется память под эти данные?  
3. Как объявить в программе динамический массив и задать его размер?  
4. Как расширить массив в ходе работы программы? Не потеряются ли при этом уже записанные в нём данные?  
5. Как определить границы изменения индексов динамического массива? Нужно ли хранить его размер в отдельной переменной?  
6. Как удалить массив из памяти?  
7. Как разместить в памяти динамическую матрицу?  
8. Как передать динамический массив в подпрограмму?  
9. Какие проблемы могут возникнуть при сохранении динамических массивов и матриц в файлах? Как вы предлагаете их решать?