## Указатели на массив

Распределение массива во время компиляции называется ***статическим связыванием*** и означает, что массив встраивается в программу во время компиляции.

***Динамическое связывание*** - выделение памяти во время выполнения программы при помощи функции new.

Массив для которого память выделяется при помощи функции new называется ***динамическим массивом.***

Создать динамический массив на C++ легко; вы сообщаете операции new тип элементов массива и требуемое количество элементов. Синтаксис, необходимый для этого, предусматривает указание имени типа с количеством элементов в квадратных скобках. Например, если необходим массив из 10 элементов int, следует записать так:

*int \* psome = new int [10] ; // получение блока памяти из 10 элементов типа int*

Операция **new** возвращает адрес первого элемента в блоке. В данном примере это значение присваивается указателю psome.

Как всегда, вы должны сбалансировать каждый вызов new соответствующим вызовом delete, когда программа завершает работу с этим блоком памяти. Однако использование new с квадратными скобками для создания массива требует применения альтернативной формы delete при освобождении массива:

*delete [ ] psome; // освобождение динамического массива*

Присутствие квадратных скобок сообщает программе, что она должна освободить весь массив, а не только один элемент, на который указывает указатель. Обратите внимание, что скобки расположены между delete и указателем.

*int \* pt = new int;*

*short \* ps = new short [500] ;*

*delete [] pt; // эффект не определен, не делайте так*

*delete ps; // эффект не определен, не делайте так*

**Общая форма** выделения и назначения памяти для массива выглядит следующим образом:

*имя\_типа имя\_указателя = new имя\_типа [количество\_элементов] ;*

*int \* psome = new int [10]; // получить блок для 10 элементов типа int*

**Листинг 4.18. arraynew.срр**

*// arraynew.ерр -- использование операции new для массивов*

*#include <iostream>*

*int main()*

*using namespace std;*

*double \* p3 = new double [3]; // пространство для З значений double*

*p3[0] = 0.2; // трактовать рЗ как имя массива*

*p3[l] = 0.5*

*p3[2] = 0.8*

*cout « “рЗ[1] is " « рЗ[1] « ".\n”; // вывод рЗ[1]*

*рЗ = рЗ + 1; // увеличение указателя*

*cout << "Now p3[0] is " << р3[0] « " and // вывод рЗ[0]*

*cout << рЗ[1] is " « рЗ[1] « " .\n" ; // вывод рЗ[1]*

*рЗ = рЗ - 1; // возврат указателя в начало*

*delete [] рЗ; // освобождение памяти*

*return 0;*

*}*

**Ниже показан вывод программы из листинга 4.18:**

*рЗ[1] is 0.5.*

*Now p3[0] is 0.5 and p3[l] is 0.8.*

***Указатели, массивы и арифметика указателей***

**Листинг 4.19. addpntrs. ерр**

*// addpntrs.ерр -- сложение указателей*

*#include <iostream>*

*int main()*

*{*

*using namespace std;*

*double wages[3] = {10000.0, 20000.0, 30000.0};*

*short stacks[3] = {3, 2, 1};*

*// Два способа получить адрес массива*

*double \* pw = wages; // имя массива равно адресу*

*short \* ps = &stacks[0]; // либо использование операции взятия адреса*

*//с элементом массива*

*cout « "pw = " << pw << ", \*pw = " << \*pw << endl;*

*pw = pw + 1;*

*cout << "add 1 to the pw pointer: \n"; // добавление 1 к указателю pw*

*cout « "pw = " « pw « ", \*pw = " « \*pw « "\n\n" ;*

*cout « "ps = " « ps « ", \*ps = " « \*ps « endl;*

*ps = ps + 1;*

*cout << "add 1 to the ps pointer: \n"; // добавление 1 к указателю ps*

*cout « "ps = " << ps << ", \*ps = " << \*ps << "\n\n";*

*// Доступ к двум элементам с помощью нотации массивов*

*cout « "access two elements with array notation\n";*

*cout << "stacks[0] = " << stacks[0]*

*« ", stacks[l] = " << stacks[l] << endl;*

*// Доступ к двум элементам с помощью нотации указателей*

*cout << "access two elements with pointer notation\n";*

*cout << "\*stacks = " << \*stacks*

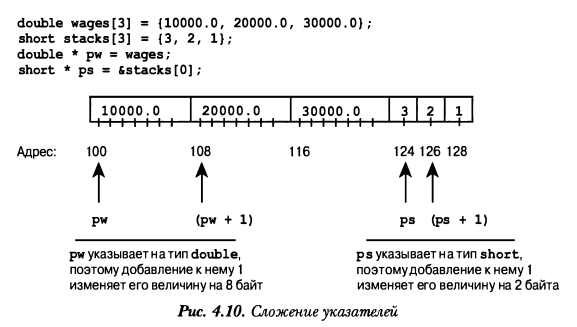
*« ", \* (stacks + 1) = " « \* (stacks + 1) « endl;*

*cout « sizeof (wages) « " = size of wages array\n"; // размер массива wages*

*cout « sizeof (pw) « " = size of pw pointer\n"; // размер указателя pw*

*return 0;*

*}*

*Ниже приведен вывод программы из листинга 4.19:*

*pw = 0x28ccf0, \*pw = 10000*

*add 1 to the pw pointer:*

*pw = 0x28ccf8, \*pw = 20000*

*ps = 0x28ccea, \*ps = 3*

*add 1 to the ps pointer:*

*ps = 0x28ccec, \*ps = 2*

*access two elements with array notation*

*stacks[0] = 3, stacks[l] = 2*

*access two elements with pointer notation*

*\*stacks = 3, \*(stacks + 1) = 2*

*24 = size of wages array*

*4 = size of pw pointer*

На заметку!

Добавление единицы к переменной указателя увеличивает его значение на количество байт, представляющее размер типа, на который он указывает.