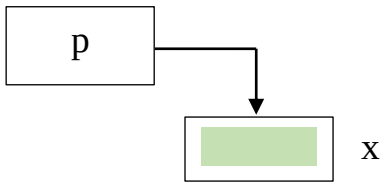
Преобразование типов указателей

Напомним, что указатель – это *переменная*, содержащая в качестве своего значения адрес другой переменной.



Как и всякая переменная, указатель имеет свой тип. Если переменная x (на которую указывает указатель) имеет тип int, то сам указатель p имеет тип int*.

Пример.

Тип переменной x	Тип указателя p
int	int *
char	char *
float	float *
double	double *

Как и обычные переменные, указатели можно преобразовывать из одного типа в другой.

Преобразование типов указателей

явное

1. если указатели разных типов (Пр.3)

2. преобразование типа указателя в целое число (Пр.1)

неявное

1. если указатели одного типа (Пр.2)

2. Любой указатель может быть преобразован к указателю на void * (Пр.4)

Пример 1.

Мы уже говорили о том, что по умолчанию значение указателя выводится в 16-ричной системе счисления. Если мы хотим его вывести в 10-тичной системе счисления, то нужно преобразовать указатель в целое число:

```
cout << (unsigned int)p;
```

Пример 2.

```
float * p;
float *q;
p = q;   // можно
q = p;   // можно
```

Пример 3.

```
float * p;
int *q;
p = q;   // нельзя
q = p;   // нельзя
p = (float *)q;   // можно
q = (int *)p;   // можно
```

Из примера 3 видно, что принудительно можно преобразовать любой тип указателя в любой другой. Это приведет к тому, что содержимое памяти по этому адресу начнет трактоваться по-другому. Операция преобразования типов указателей никак не меняет содержимого памяти по данному адресу; она лишь меняет способ трактовки этого содержимого, который тоже может быть важен. Например, и число типа int, и число типа float оба занимают по 4 байта. Но способ представления конкретных чисел, например 5 для int и 5.0 для float, существенно отличается один от другого. Поэтому следующий фрагмент выдаст совсем не 5.0, а нечто другое (что именно?):

```
int x = 5;
int *q = &x;
float * p;
p = (float *)q;
cout << *p << endl;
```

Однако, такое преобразование переменных из одного типа в другой, сохраняющее не значение переменной, а содержимое памяти по данному адресу, иногда

бывает важно. Например, таким образом можно превратить какое-либо значение в набор байтов, чтобы потом обрабатывать эти байты по-отдельности.

Пример 4.

В C++ также имеется некоторый специальный тип указателя <<неизвестно на что>>. Такие указатели определяются с ключевым словом void:

```
void *p;
```

Такому указателю можно присвоить любой указатель (т. е. указатель на любой тип), но пользоваться им непосредственно нельзя – сначала нужно явно привести его к типу указателя на конкретный тип, написав перед ним (тип *).

```
float * p;
int *q;
void *r;
r = p;  // можно
r = q; // можно
```