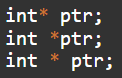
**3. Указатели. Объявление указателей. Операции с указателями. Адресная арифметика. Преобразование указателей. Сравнение указателя на фундаментальный тип и указателя на класс.**

**Указатель** – переменная, значением которой является адрес ячейки памяти.

**Объявление указателей**

Происходит с использованием символа звездочки (\*) перед именем переменной: 

Количество пробелов слева и справа от звёздочки значения не имеет.



При объявлении нескольких сразу, звёздочку нужно ставить возле имени каждой переменной

**Операции с указателями:**

* **Присваивание адреса**:

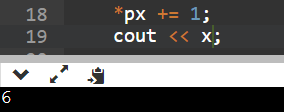
Указателю можно присвоить адрес объекта того же типа, либо значение другого указателя.

Для получения адреса переменной используется оператор &:

* **Разыменование**:

*\*имя\_указателя* позволяет получить объект по адресу, который хранится в указателе.



* **Логические операции.**

При операциях сравнения (==, !=, <, > и т.д.) сравниваются ссылки в указателе

**Адресная арифметика**

При сложении и вычитании адресу в указателе прибавляется/вычитается размер типа указателя.

Например, если указатель типа *int*, то при *pointer++* к ссылке прибавится 4 байта, т.е. размер типа int.

Операции \* и / к указателям не применимы.

**Преобразование указателей**

Преобразование типа указателя можно сделать двумя способами:



* В стиле C: y = (int\*) x;
* В стиле C++: y = static\_cast<int\*> (x);

Преобразование указателя типа A к указателю типа B – небезопасная операция, поэтому неявно не выполняется.

Преобразование типа не изменяет хранимый адрес в указателе

**Сравнение указателя на фундаментальный тип и указателя на класс.**