

## 3.2. Видимость и инициализация переменных

### 3.2.1. Видимость переменной

Ссылаться на переменную (и использовать) можно только после того, как она была объявлена:

```
cout << x;  
int x = 5;
```

Такой код даже не скомпилируется. Компилятор выдаст ошибку «'x' was not declared in this scope». Если же поменять строчки местами, программа заработает.

Другой пример:

```
{  
    int x = 5;  
    {  
        cout << x;  
    }  
    cout << x;  
}  
cout << x;
```

Здесь переменная `x` определена внутри операторных скобок, и в трех местах кода имеет место попытка вывести ее на экран. При этом программа не компилируется: компилятор указывает на ошибку при попытке вывести на экран `x` за пределами первых операторных скобок. Если эту строчку закомментировать, программа успешно скомпилируется:

```
{  
    int x = 5;  
    {  
        cout << x;  
    }  
    cout << x;  
}  
//cout << x;
```

Таким образом, переменные в C++ видны только после своего объявления и до конца блока, в котором были объявлены. Например, следующий код с условным оператором не скомпилируется:

```
if (1 > 0) {  
    int x = 5;  
}  
cout << x;
```

Это связано с тем, что переменная объявлена внутри тела условного оператора.

То же самое имеет место для цикла `while`:

```
while (1 > 0) {  
    int x = 5;  
}  
cout << x;
```

И для цикла `for`:

```
for (int i = 0; i < 10; ++i) {  
    int x = 5;  
}  
cout << x;
```

Может возникнуть вопрос: видна ли переменная `i`, которая была объявлена как счетчик цикла:

```
for (int i = 0; i < 10; ++i) {  
    int x = 5;  
}  
cout << i;
```

Оказывается, что она также не видна.

Еще один пример:

```
string s = "hello";  
{  
    string s = "world";  
    cout << s << endl;  
}  
cout << s << endl;
```

Здесь переменная `s` определена как внутри операторных скобок, так и вне их. Программа компилируется и выводит:

```
world  
hello
```

Однако использование одинаковых имен, хоть не вызывает ошибку компиляции, считается плохим стилем, так как усложняет понимание кода и увеличивает вероятность ошибиться.