Ссылки на функции в c++ с использованием using.

using-объявление

```cpp
#include <iostream>

int main()
{
    using std::cout;          // "using-объявление" сообщает компилятору,
                              // что cout следует обрабатывать, как std::cout
    cout << "Hello, world!";  // std:: здесь уже не нужен

    return 0;
}
```

Строка `using std::cout;` сообщает компилятору, что мы будем использовать объект **cout** из **пространства имен** std. И каждый раз, когда компилятор будет сталкиваться с `cout`, он будет понимать, что это `std::cout`.

Для каждого объекта (`cout`, `cin`, `endl`) нужно использовать отдельное «using-объявление».

using-директива

```cpp
#include <iostream>

int main()
{
    using namespace std;      // "using-директива" сообщает компилятору,
                              // что мы используем все объекты из пространства имен std
    cout << "Hello, world!";  // std:: уже не нужно
    return 0;
}
```

Так как с помощью using-директивы мы подключаем ВСЕ имена из пространства имен std, то вероятность возникновения конфликтов имен значительно возрастает (хотя и остается незначительной). `using namespace std;` сообщает компилятору, что мы хотим использовать всё, что находится в пространстве имен std, так что, если компилятор найдет имя, которое не сможет распознать, он будет проверять его наличие в пространстве имен std.

Пример конфликта с using-директивой

```
1  #include <iostream>
2
3  int cout() // объявляем нашу собственную функцию "cout"
4  {
5      return 4;
6  }
7
8  int main()
9  {
10     using namespace std;      // делаем std::cout доступным по "cout"
11     cout << "Hello, world!"; // какой cout компилятор здесь должен использовать?
12
13     return 0;
14 }
```

Ошибки можно избежать, исправив 11 строчку:
`std::cout << “Hello, world!”;`

Области видимости using-объявления и using-директивы

Если объявление using или директива using используется в блоке, имена применимы только в этом блоке (они следуют обычным правилам области видимости блока). Это хорошо, поскольку снижает вероятность возникновения конфликтов имен внутри этого блока.

Если объявление using или директива using используются в глобальном пространстве имен, имена применимы ко всему остальному файлу (они имеют область видимости файла).

Отмена/замена using-стейтментов

Как только один using-стейтмент был объявлен, его невозможно отменить или заменить другим using-стейтментом в пределах области видимости, в которой он был объявлен.

Лучшее, что можно сделать — это намеренно ограничить область применения using-стейтментов с самого начала, используя правила локальной области видимости:

```
int main()
{
    {
        using namespace Boo;
        // Здесь всё относится к пространству имен Boo::
    }   // действие using namespace Boo заканчивается здесь

    {
        using namespace Foo;
        // Здесь всё относится к пространству имен Foo::
    }   // действие using namespace Foo заканчивается здесь

    return 0;
}
```

Источники:

1. https://ravesli.com/urok-54-using-statements/
2. https://radioprog.ru/post/1130