## Область видимости

Область видимости идентификатора определяет, где он доступен для использования. К идентификатору, который находится вне области видимости, доступ закрыт.

   Переменные с локальной/блочной областью видимости доступны только в пределах блока, в котором они объявлены. Это:

  \* локальные переменные;

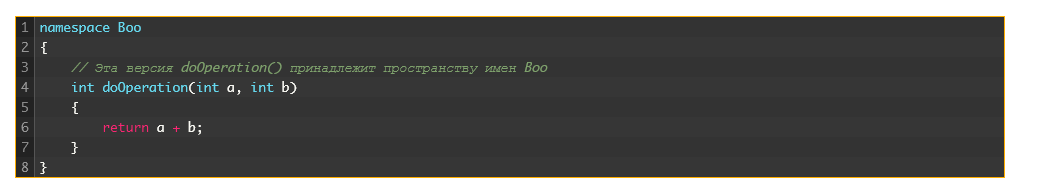
  \* параметры функции.

   Переменные с глобальной/файловой областью видимости доступны в любом месте файла. Это:

  \* глобальные переменные.

Пространство имен определяет область кода, в которой гарантируется уникальность всех идентификаторов. По умолчанию, [глобальные переменные](https://ravesli.com/urok-49-globalnye-peremennye/) и обычные функции определены в глобальном пространстве имен.

Язык C++ позволяет объявлять собственные пространства имен через ключевое слово namespace. Всё, что объявлено внутри пользовательского пространства имен, — принадлежит только этому пространству имен (а не глобальному).



## Доступ к пространству имен через оператор разрешения области видимости (::)

Способ указать компилятору искать идентификатор в определенном пространстве имен — это использовать название необходимого пространства имен вместе с оператором разрешения области видимости (::) и требуемым идентификатором.

Например, сообщим компилятору использовать версию doOperation() из пространства имен Boo:



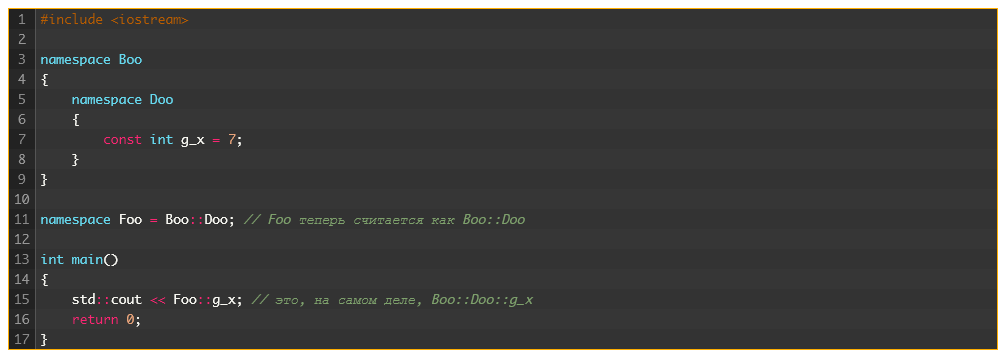
Допускается объявление пространств имен в нескольких местах (либо в нескольких файлах, либо в нескольких местах внутри одного файла). Всё, что находится внутри одного блока имен, считается частью только этого блока.



Одни пространства имен могут быть вложены в другие пространства имен.



C++ позволяет создавать псевдонимы для пространств имен:



Стоит отметить, что пространства имен в C++ не были разработаны, как способ реализации информационной иерархии — они были разработаны в качестве механизма предотвращения возникновения конфликтов имен. Как доказательство этому, вся Стандартная библиотека шаблонов находится в единственном пространстве имен std::.

Вложенность пространств имен не рекомендуется использовать, так как при неумелом использовании увеличивается вероятность возникновения ошибок и дополнительно усложняется логика программы.

**+**