

Чаще всего, лексика замыкания используется в функциональных языках программирования. Замыкание – это специальный тип функции, с помощью которого она ссылается на свободные переменные. Это позволяет замкнутым функциям использовать переменные из *внешнего окружения*, несмотря на то что они не входят в границы. Когда функция создана, внешние переменные, которыми мы пользуемся, «захватываются», иными словами, они связаны с замкнутой функцией, так что они становятся доступными. Часто это обозначает то, что делаются копии значений переменных, когда инициализируется замыкание.

Пример: мы создаем переменную «nonLocal» типа integer. Во второй строчке создаем экземпляр делегата «Action», что выводит в сообщение значение переменной типа integer. В конце мы запускаем функцию-делегат, чтобы увидеть сообщения.

```
int nonLocal = 1;

Action closure = delegate
{
    Console.WriteLine("{0} + 1 = {1}", nonLocal, nonLocal + 1);
};

closure(); // 1 + 1 = 2
```