Метод может возвращать значение, какой-либо результат. В примере выше были определены два метода, которые имели тип **void**. Методы с таким типом не возвращают никакого значения. Они просто выполняют некоторые действия.

Но методы также могут возвращать некоторое значение. Для этого применяется оператор **return**, после которого идет возвращаемое значение:

|  |  |
| --- | --- |
| 1 | return возвращаемое значение; |

Например, определим метод, который возвращает значение типа string:

|  |  |
| --- | --- |
| 1  2  3  4 | string GetMessage()  {  return "Hello";  } |

Метод GetMessage имеет тип string, следовательно, он должен возвратить строку. Поэтому в теле метода используется оператор **return**, после которого указана возвращаемая строка.

При этом методы, которые в качестве возвращаемого типа имеют любой тип, кроме **void**, обязательно должны использовать оператор return для возвращения значения. Например, следующее определение метода некорректно:

|  |  |
| --- | --- |
| 1  2  3  4 | string GetMessage()  {  Console.WriteLine("Hello");  } |

Также между возвращаемым типом метода и возвращаемым значением после оператора return должно быть соответствие. Например, в следующем случае возвращаемый тип - string, но метод возвращает число (тип int), поэтому такое определение метода некорректно:

|  |  |
| --- | --- |
| 1  2  3  4 | string GetMessage()  {  return 3; // Ошибка! Метод должен возвращать строку, а не число  } |

Результат методов, который возвращают значение, мы можем присвоить переменным или использовать иным образом в программе:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | string GetMessage()  {  return "Hello";  }    string message = GetMessage(); // получаем результат метода в переменную message  Console.WriteLine(message); // Hello |

Метод GetMessage() возвращает значение типа string. Поэтому мы можем присвоить это значение какой-нибудь переменной типа string: string message = GetMessage();

Либо даже передать в качестве значения параметру другого метода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | string GetMessage()  {  return "Hello";  }  void PrintMessage(string message)  {  Console.WriteLine(message);  }  PrintMessage(GetMessage()); |

В вызове PrintMessage(GetMessage()) сначада вызывается метод GetMessage() и его результат передается параметру message метода PrintMessage

После оператора return также можно указывать сложные выражения или вызовы других методов, которые возвращают определенный результат. Например, определим метод, который возвращает сумму чисел:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | int Sum(int x, int y)  {  return x + y;  }    int result = Sum(10, 15); // 25  Console.WriteLine(result); // 25    Console.WriteLine(Sum(5, 6)); // 11 |

Метод Sum() имеет тип int, следовательно, он должен возвратить значение типа int - целое число. Поэтому в теле метода используется оператор **return**, после которого указано возвращаемое число (в данном случае результат суммы переменных x и y).

### Сокращенная версия методов с результатом

Также мы можем сокращать методы, которые возвращают значение:

|  |  |
| --- | --- |
| 1  2  3  4 | string GetMessage()  {  return "hello";  } |

аналогичен следующему методу:

|  |  |
| --- | --- |
| 1 | string GetMessage() => "hello"; |

А метод

|  |  |
| --- | --- |
| 1  2  3  4 | int Sum(int x, int y)  {  return x + y;  } |

аналогичен следующему методу:

|  |  |
| --- | --- |
| 1 | int Sum(int x, int y) => x + y; |

### Выход из метода

Оператор return не только возвращает значение, но и производит выход из метода. Поэтому он должен определяться после остальных инструкций. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5 | string GetHello()  {  return "Hello";  Console.WriteLine("After return");  } |

С точки зрения синтаксиса данный метод корректен, однако его инструкция Console.WriteLine("After return") не имеет смысла - она никогда не выполнится, так как до ее выполнения оператор **return** возвратит значение и произведет выход из метода.

Однако мы можем использовать оператор **return** и в методах с типом void. В этом случае после оператора return не ставится никакого возвращаемого значения (ведь метод ничего не возвращает). Типичная ситуация - в зависимости от опеределенных условий произвести выход из метода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | void PrintPerson(string name, int age)  {  if(age > 120 || age < 1)  {  Console.WriteLine("Недопустимый возраст");  return;  }  Console.WriteLine($"Имя: {name} Возраст: {age}");  }    PrintPerson("Tom", 37); // Имя: Tom Возраст: 37  PrintPerson("Dunkan", 1234); // Недопустимый возраст |

Здесь метод PrintPerson() в качестве параметров принимает имя и возраст пользователя. Однако в методе вначале мы проверяем, соответствует ли возраст некоторому диапазону (меньше 120 и больше 0). Если возраст находится вне этого диапазона, то выводим сообщение о недопустимом возрасте и с помощью оператора **return** выходим из метода. После этого метод заканчивает свою работу.

Однако если возраст корректен, то выводим информацию о пользователе на консоль. Консольный вывод:

Имя: Tom Возраст: 37  
Недопустимый возраст