

Методы

Методы могут создаваться самим программистом, а могут быть стандартными, созданными авторами языка.

- Каждый метод имеет свое собственное имя.
- Имена методов задаются по тем же правилам, что и имена переменных.
- Чтобы метод выполнил свои действия, его надо вызвать. Для этого следует написать его имя и круглые скобки. Круглые скобки, которые идут после имени, означают, что мы имеем дело именно с методом, а не с переменной.

Чаще всего метод выполняет действия не просто так, а используя некоторые исходные данные.

Например, метод `sin()` - синус, вычисляет значение синуса угла, используя в качестве исходных данных значение угла:

```
x = Math.sin(5);
```

Исходные данные, их также называют входными параметрами, пишутся внутри круглых скобок.

В данном случае метод вычисляет синус угла равного пяти.

Общий вид объявления метода:

```
модификатор_доступа другие_модификаторы тип_результата  
имя_метода(список_параметров){  
    тело метода;  
}
```

Модификаторы доступа (`public`, `protected`, `default`, `private`) рассматриваются позже.

Другие модификаторы, к примеру **`static`** – необходим для того, чтобы вызвать метод из метода `main`.

Тип результата - если в результате работы метода должно получиться число, здесь следует указать тип числа, например, `float` или `long`.

Если метод просто проделывает какие-то действия и не возвращает никакого значения, в качестве типа результата следует указать пустой тип **`void`**.

Имя метода задается программистом по собственному усмотрению, но следует учитывать ограничения, аналогичные правилам, используемым при создании переменных.

После имени метода ставятся круглые скобки со списком входных переменных (параметров). Таким образом, задается, какие данные метод будет получать при своем вызове. Входных переменных может быть любое количество, если входные параметры не нужны, то следует поставить просто пустые скобки.

Пример метода получающего на вход два числа `double` и возвращающего число `double`:

```
double max(double a, double b)
```

Если метод возвращает значение, то его работа должна завершаться оператором **return**, после которого пишется возвращаемое значение.

Пример:

```
double cube(double x) {  
    double y = x * x * x;  
    return y;  
}
```

Метод получает исходное значение, которое попадает в переменную *x*. Затем создается переменная *y* и в нее заносится результат перемножения *x*. В конце значение *y* возвращается с помощью **return**.

И затем мы можем использовать метод в выражениях:

```
z = cube(15) / (cube(3) + cube(x + 2));
```

Каждый раз, когда метод встречается в выражении, происходит его вызов, он получает значение, указанное в скобках. Если это выражение, то сначала вычисляется результат выражения, а затем передается в метод. Когда метод выполнил свою работу, его результат подставляется в выражение, откуда его вызвали.

Метод **main**

Вернемся к простейшей программе на Java.

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

Метод **main** запускается, когда вы запускаете вашу программу. Поэтому его часто называют "точка входа".