

Параметры в методах интерфейсов

При описании методов в интерфейсах необходимо указывать не только имена самих методов, но и принимаемые ими параметры.

Давайте посмотрим на примере.

Пусть у нас дан интерфейс **iMath**, описывающий класс для математических операций сложения, вычитания, умножения и деления.

Пусть этот интерфейс будет выглядеть так:

```
1  <?php
2  interface iMath
3  {
4      public function sum(); // сложить
5      public function subtract(); // вычесть
6      public function multiply(); // умножить
7      public function divide(); // поделить
8  }
9  ?>
```

Сейчас методы нашего интерфейса не принимают никаких параметров. И на самом деле методы класса, который будет реализовывать этот интерфейс, также не должны будут принимать параметров, иначе будет ошибка.

Давайте укажем параметры методов в нашем интерфейсе:

```
1  <?php
2  interface iMath
3  {
4      public function sum($a, $b); // сложить
5      public function subtract($a, $b); // вычесть
6      public function multiply($a, $b); // умножить
7      public function divide($a, $b); // поделить
8  }
9  ?>
```

Напишем теперь реализацию нашего интерфейса:

```
1  <?php
2  class Math implements iMath
3  {
4      public function sum($a, $b)
5      {
6          return $a + $b;
7      }
8
9      public function subtract($a, $b)
10     {
11         return $a - $b;
12     }
13
14     public function multiply($a, $b)
15     {
16         return $a * $b;
17     }
18
19     public function divide($a, $b)
20     {
21         return $a / $b;
```



```
22 |     }  
23 | }  
24 | ?>
```

Если попытаться в нашем классе задать другое количество параметров - у нас это просто не получится: PHP выдаст ошибку. Таким образом мы не сможем ни случайно забыть какой-то параметр, ни случайно добавить лишний.

Задача 38.1

Пусть у нас дан такой интерфейс **iUser**:

```
1 | <?php  
2 | interface iUser  
3 | {  
4 |     public function setName($name); // установить имя  
5 |     public function getName(); // получить имя  
6 |     public function setAge($age); // установить воз  
   |     раст  
7 |     public function getAge(); // получить воз  
   |     раст  
8 | }  
9 | ?>
```

Сделайте класс **User**, который будет реализовывать данный интерфейс.

