Константы классов

Сейчас мы с вами разберем **константы** классов. Константы по сути являются свойствами, значения которых нельзя изменить.

Неизменяемые свойства нужны для того, чтобы хранить какие-то значения, которые являются постоянными и не должны быть случайно изменены.

Чтобы создать константу, ее нужно объявить через ключевое слово **const** и обязательно сразу же задать ее значение:

```php
<?php
  class Test
  {
    // Задаем константу:
    const constant = 'test';
  }
?>
```

> Обратите внимание на то, что в именах констант не пишется доллар.

Общепринято имена констант писать большими буквами, то есть не **constant**, а **CONSTANT**. Это делается для того, чтобы визуально легко было отличать константы в коде.

Давайте поправим наш класс:

```php
<?php
  class Test
  {
    // Задаем константу:
    const CONSTANT = 'test';
  }
?>
```

Давайте теперь рассмотрим, как прочитать значения константы. Здесь следует сказать то, что константы класса больше похожи не на обычные свойства, а на статические.

Это значит, что константы класса задаются один раз для всего класса, а не отдельно для каждого объекта этого класса.

Поэтому обращение к константам происходит так же, как и для статических свойств: пишем имя класса, два двоеточия и название константы:

```php
<?php
  echo Test::CONSTANT; // выведет 'test'
?>
```

> Обращение к константе отличается от обращения к статическому свойству тем, что для константы доллар не пишется, а для static свойства - пишется.

Как уже упоминалось выше, значения констант можно прочитывать, но не записывать. Попытка что-то записать в нее выдаст ошибку:

```php
<?php
  Test::CONSTANT = 'test'; // выдаст ошибку
?>
```

Обращение к константам внутри класса

Внутри класса также можно обратится к константе через **::self**, вот так:

```php
<?php
  class Test
  {
    // Задаем константу:
    const CONSTANT = 'test';

    // Сделаем метод для получения значения константы:
    function getConstant() {
      return self::CONSTANT;
    }
  }
?>
```

Воспользуемся нашим методом:

```php
<?php
  $test = new Test;
  echo $test->getConstant(); // выведет 'te
    st'
?>
```

Применение

Давайте сделаем класс **Circle** (круг), с помощью которого можно будет найти площадь круга и длину окружности.

Для работы с кругом нам понадобится число Пи, равное **3.14**. Логично будет для хранения этого числа использовать константу, чтобы случайно где-нибудь в коде наше число Пи вдруг не поменялось.

Вот частичная реализация нашего класса:

```php
<?php
  class Circle
  {
    const PI = 3.14; // запишем число ПИ в константу
    private $radius; // радиус круга

    public function __construct($radius)
    {
      $this->radius = $radius;
    }

    // Найдем площадь:
    public function getSquare()
    {
      // Пи умножить на квадрат радиуса
    }

    // Найдем длину окружности:
    public function getCircuit()
    {
      // 2 Пи умножить на радиус
    }
  }
?>
```

Задача 32.1

Реализуйте предложенный класс **Circle** самостоятельно.

Задача 32.2

С помощью написанного класса **Circle** найдите длину окружности и площадь круга с радиусом 10.