Свойства только для чтения

Сейчас мы с вами сделаем так, чтобы в объекте какое-то свойство было доступно только для чтения, но не для записи (*англ* read-only).

Это делается следующим образом: для такого свойства нужно сделать геттер, но не делать сеттер.

В этом случае свойство можно будет прочитать с помощью геттера, но нельзя будет записать, так как сеттер отсутствует.

При этом изначальное значение свойства будет задаваться в конструкторе при создании объекта.

Давайте попробуем реализовать описанное.

Пусть у нас дан вот такой класс **User**:

```php
<?php
  class User
  {
    private $name;
    private $age;
  }
?>
```

Давайте сделаем так, чтобы свойство **name** было доступно только для чтения, а свойство **age** - и для чтения и для записи. Для этого свойству **name** сделаем только геттер, а свойству **age** - и геттер и сеттер:

```php
<?php
  class User
  {
    private $name;
    private $age;

    // Геттер для имени:
    public function getName()
    {
      return $this->name;
    }

    // Геттер для возраста:
    public function getAge()
    {
      return $this->age;
    }

    // Сеттер для возраста:
    public function setAge($age)
    {
      $this->age = $age;
    }
  }
?>
```

Давайте теперь добавим конструктор объекта, в котором будем задавать начальные значения наших свойств:

```php
<?php
  class User
  {
    private $name;
    private $age;

```

```
    // Конструктор объекта:
    public function __construct($name, $age)
    {
      $this->name = $name;
      $this->age = $age;
    }

    // Геттер для имени:
    public function getName()
    {
      return $this->name;
    }

    // Геттер для возраста:
    public function getAge()
    {
      return $this->age;
    }

    // Сеттер для возраста:
    public function setAge($age)
    {
      $this->age = $age;
    }
  }
?>
```

Все - наша задача решена, убедимся в этом:

```
<?php
  $user = new User('Коля', 25); // создаем объ
    ект с начальными данными

  // Имя можно только читать, но нельзя поменять:
  echo $user->getName(); // выведет 'Коля'

  // Возраст можно и читать, и менять:
  echo $user->getAge(); // выведет 25
  echo $user->setAge(30); // установим возраст в з
    начение 30
  e cho $user->getAge(); // выведет 30
?>
```

Задача 9.1

Сделайте класс **Employee**, в котором будут следующие свойства: **name** (имя), **surname** (фамилия) и **salary** (зарплата).

Задача 9.2

Сделайте так, чтобы свойства **name** и **surname** были доступны только для чтения, а свойство **salary** - и для чтения, и для записи.