Начальные значения свойств в конструкторе

Пусть у нас есть какой-то класс с двумя свойствами:

```php
<?php
  class Test
  {
    public $prop1;
    public $prop2;
  }
?>
```

Давайте сделаем так, чтобы при создании объекта класса эти свойства имели какие-либо значения.

Как вы уже знаете, в момент создания объекта вызывается метод **__construct**. Давайте зададим начальные значения свойства в этом методе:

```php
<?php
  class Test
  {
    public $prop1;
    public $prop2;

    public function __construct()
    {
      $this->prop1 = 'value1'; // начальное зна
        чение свойства
      $this->prop2 = 'value2'; // начальное зна
        чение свойства
     }

  }

  $test = new Test;
  echo $test->prop1; // выведет 'value1'
  echo $test->prop2; // выведет 'value2'
?>
```

Применение

Пусть у нас есть класс **Student** с двумя свойствами - **name** и **course** (курс студента).

Сделаем так, чтобы имя студента приходило параметром при создании объекта, а курс автоматически принимал значение **1**:

```php
<?php
  class Student
  {
    private $name;
    private $course;

    public function __construct($name)
    {
      $this->name = $name;
      $this->course = 1; // курс изначально рав
        ен 1
     }

```

```
    }
?>
```

Сделаем геттеры для наших свойств:

```php
<?php
  class Student
  {
    private $name;
    private $course;

    public function __construct($name)
    {
      $this->name = $name;
      $this->course = 1;
    }

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

    // Геттер курса:
    public function getCourse()
    {
      return $this->course;
    }
  }
?>
```

Пусть имя созданного студента будет неизменяемым и доступным только для чтения, а вот для курса мы сделаем метод, который будет переводить нашего студента на следующий курс:

```php
<?php
  class Student
  {
    private $name;
    private $course;

    public function __construct($name)
    {
      $this->name = $name;
      $this->course = 1;
    }

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

    // Геттер курса:
    public function getCourse()
    {
      return $this->course;
    }

    // Перевод студента на новый курс:
    public function transferToNextCourse()
    {
      $this->course++;
    }
  }
?>
```

Проверим работу нашего класса:

```php
<?php
  $student = new Student('Коля'); // создаем объ
```

```
    ект класса
3
4   echo $student->getCourse(); // выведет 1 -
        начальное значение
5   $ student->transferToNextCourse(); // переведем сту
    дента на следующий курс
6   e cho $student->getCourse(); // выведет 2
7   ?>
```

Задача 12.1

Не подсматривая в мой код реализуйте такой же класс **Student**.

Задача 12.2

Модифицируйте метод **transferToNextCourse** так, чтобы при переводе на новый курс выполнялась проверка того, что новый курс не будет больше **5**.