Наследование от класса и реализация интерфейса

Класс может наследовать от другого класса и при этом реализовывать какой-то интерфейс. Рассмотрим на практическом примере.

Пусть мы хотим сделать класс **Programmer** (программист), у которого будет имя, зарплата и список языков, которые знает программист.

Пока наше описание класса весьма расплывчато: да, в классе будет имя, зарплата, языки - но какие методы будут в нашем классе?

Давайте более точно опишем наш класс с помощью интерфейса **iProgrammer**:

```php
<?php
  interface iProgrammer
  {
    public function __construct($name, $salary); // задаем имя и зарплату
     public function getName(); // получить имя
    public function getSalary(); // получить зарплату
     public function getLangs(); // получить массив языков, которые знает программист
     public function addLang($lang); // добавить язык в массив языков
  }
?>
```

Пусть мы покопались в уже реализованных нами классах и, оказывается, у нас уже есть похожий класс **Employee**. Он реализует не все методы класса **Programmer**, но часть.

Вот код уже существующего у нас класса **Employee**:

```php
<?php
  class Employee
  {
    private $name;
    private $salary;

    public function __construct($name, $salary)
    {
      $this->name = $name;
      $this->salary = $salary;
    }

    public function getName()
    {
      return $this->name;
    }

    public function getSalary()
    {
      return $this->salary;
    }
  }
?>
```

Логично в нашем случае сделать так, чтобы наш новый класс **Programmer** унаследовал часть необходимых себе методов от класса **Employee** (а часть мы потом реализуем уже в самом новом классе):

```php
<?php
  class Programmer extends Employee
  {

  }
?>
```

При этом нам известно, что класс **Pogrammer** должен реализовывать интерфейс **iProgrammer**:

```php
<?php
  class Programmer implements iProgrammer
  {

  }
?>
```

Давайте совместим наследование от класса **Employee** и реализацию интерфейса **iProgrammer**:

```php
<?php
  class Programmer extends Employee implements iPr
    ogrammer
  {

  }
?>
```

Получится, что наш класс **Pogrammer** унаследует от класса **Employee** методы **__construct**, **getName()** и **getSalary()**, а методы **addLang** и **getLangs** нам придется реализовать:

```php
<?php
  class Programmer extends Employee implements iPr
    ogrammer
  {
    public function addLang($lang)
    {
      // реализация
    }

    public function getLangs()
    {
      // реализация
    }
  }
?>
```

Интерфейсу **iProgrammer** все равно, родные методы у класса или унаследованные - главное, чтобы все описанные методы были реализованы.

Задача 43.1

Скопируйте код моего класса **Employee** и код интерфейса **iProgrammer**. Не копируйте мою заготовку класса **iProgrammer** - не подсматривая в мой код реализуйте этот класс самостоятельно.