Передача объектов параметрами

Пусть у нас дан вот такой класс **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;
    }
  }
?>
```

Давайте сделаем еще и класс **EmployeesCollection**, который будет хранить массив работников, то есть массив объектов класса **Employee**.

Пусть работники будут храниться в свойстве **employees**, а для добавления работников будет существовать метод **add**. Этот метод параметром будет принимать объект класса **Employee** и записывать его в конец массива **$this->employees**:

```php
<?php
  class EmployeesCollection
  {
    private $employees = []; // массив работников, по
      умолчанию пустой

    // Добавляем нового работника:
    public function add($employee)
    {
      $this->employees[] = $employee; // $employee - о
        бъект класса Employee
     }

  }
?>
```

Давайте также добавим в наш класс метод **getTotalSalary**, который будет находить суммарную зарплату всех хранящихся работников:

```php
<?php
  class EmployeesCollection
  {
    private $employees = [];
```

```
5
6       public function add($employee)
7       {
8         $this->employees[] = $employee;
9       }
10
11      // Находим суммарную зарплату:
12      public function getTotalSalary()
13      {
14        $sum = 0;
15
16        // Перебираем работников циклом:
17        foreach ($this->employees as $employee) {
18          $sum += $employee->getSalary(); // пол
              учаем з/п работника через метод getSalary()
19        }
20
21        return $sum;
22      }
23    }
24  ?>
```

Давайте проверим работу класса **EmployeesCollection**:

```
1  <?php
2    $employeesCollection = new EmployeesCollection;
3
4    $employeesCollection->add(new Employee('Ко
       ля', 100));
5    $ employeesCollection->add(new Employee('Ва
       ся', 200));
6    $ employeesCollection->add(new Employee('Пе
       тя', 300));
7
8    echo $employeesCollection->getTotalSalary();
       // выведет 600
9  ?>
```

Итак, как вы видите, объекты одного класса можно параметрами передавать в другой класс.

Давайте сделаем наш класс **EmployeesCollection** более жизненным и добавим метод получения всех работников и метод подсчета:

```
1  <?php
2    class EmployeesCollection
3    {
4      private $employees = [];
5
6      // Получаем всех работников в виде массива:
7      public function get()
8      {
9        return $this->employees;
10     }
11
12     // Подсчитываем количество хранимых работников:
13     public function count()
14     {
15       return count($this->employees);
16     }
17
18     public function add($employee)
19     {
20       $this->employees[] = $employee;
21     }
22
23     public function getTotalSalary()
24     {
25       $sum = 0;
```

```
26
27          foreach ($this->employees as $employee) {
28            $sum += $employee->getSalary();
29          }
30
31          return $sum;
32        }
33    }
34  ?>
```

Задача 25.1

Сделайте класс **Product** (товар), в котором будут приватные свойства **name** (название товара), **price** (цена за штуку) и **quantity**. Пусть все эти свойства будут доступны только для чтения.

Задача 25.2

Добавьте в класс **Product** метод **getCost**, который будет находить полную стоимость продукта (сумма умножить на количество).

Задача 25.3

Сделайте класс **Cart** (корзина). Данный класс будет хранить список продуктов (объектов класса **Product**) в виде массива. Пусть продукты хранятся в свойстве **products**.

Задача 25.4

Реализуйте в классе **Cart** метод **add** для добавления продуктов.

Задача 25.5

Реализуйте в классе **Cart** метод **remove** для удаления продуктов. Метод должен принимать параметром название удаляемого продукта.

Задача 25.6

Реализуйте в классе **Cart** метод **getTotalCost**, который будет находить суммарную стоимость продуктов.

Задача 25.7

Реализуйте в классе **Cart** метод **getTotalQuantity**, который будет находить суммарное количество продуктов (то есть сумму свойств **quantity** всех продуктов).

Задача 25.8

Реализуйте в классе **Cart** метод **getAvgPrice**, который будет находить среднюю стоимость продуктов (суммарная стоимость делить на количество всех продуктов).