**Домашнее задание 6**

**Задача 1**  
Сделайте класс Employee (работник), в котором будут следующие свойства - name (имя), age(возраст), salary (зарплата).

Создайте объект класса Employee, затем установите его свойства в следующие значения - имя 'Иван', возраст 25, зарплата 1000.

Создайте второй объект класса Employee, установите его свойства в следующие значения - имя 'Вася', возраст 26, зарплата 2000.

Выведите на экран сумму зарплат Ивана и Васи.

Выведите на экран сумму возрастов Ивана и Васи.

**Задача 2**

Сделайте класс Employee, в котором будут следующие private свойства: name (имя), age (возраст) и salary (зарплата).

Сделайте геттеры и сеттеры для всех свойств класса Employee.

Дополните класс Employee приватным методом isAgeCorrect, который будет проверять возраст на корректность (от 1 до 100 лет). Этот метод должен использоваться в сеттере setAge перед установкой нового возраста (если возраст не корректный - он не должен меняться).

Пусть зарплата наших работников хранится в долларах. Сделайте так, чтобы геттер getSalary добавлял в конец числа с зарплатой значок доллара.

То есть еще раз для полной ясности условия задачи: в свойстве salary зарплата будет хранится просто числом, но геттер будет возвращать ее с долларом на конце.

Задача 3

Сделайте класс User, в котором будут следующие свойства только для чтения: name (имя), surname(фамилия), Начальные значения этих свойств должны устанавливаться в конструкторе. Сделайте также геттеры этих свойств.

Сделайте так, чтобы третьим параметром в конструктор передавалась дата рождения работника в формате *год-месяц-день* Запишите ее в свойство birthday. Сделайте геттер для этого свойства.

Сделайте приватный метод calculateAge, который параметром будет принимать дату рождения, а возвращать возраст с учетом того, был ли уже день рождения в этом году, или нет.

Сделайте так, чтобы метод calculateAge вызывался в конструкторе объекта, рассчитывал возраст пользователя и записывал его в приватное свойство age. Сделайте геттер для этого свойства.

Сделайте класс Employee, который будет наследовать от класса User. Пусть новый класс имеет свойство salary, в котором будет хранится зарплата работника. Зарплата должна передаваться четвертым параметром в конструктор объекта. Сделайте также геттер для этого свойства.