

# Java. Лабораторная работа №2

## №1. Класс `Invoice`

### Условие

Создайте класс с именем `Invoice`, который может использоваться хозяйственным магазином для представления счёта за товар, проданный в магазине. Класс `Invoice` должен включать четыре элемента информации в виде переменных экземпляра:

- Номер детали (тип `String`).
- Описание детали (тип `String`).
- Количество приобретаемого товара (тип `int`).
- Цену за единицу товара (тип `double`).

Ваш класс должен иметь конструктор, инициализирующий эти четыре переменные экземпляра.

Предоставьте методы `set` и `get` для каждой переменной экземпляра.

Кроме того, создайте метод с именем `getInvoiceAmount`, который вычисляет сумму счёта (т.е. умножает количество на цену за единицу), а затем возвращает эту сумму как значение типа `double`. Если количество неположительное, его следует установить в 0. Если цена за единицу неположительная, её следует установить в 0.0.

Напишите тестовое приложение с именем `InvoiceTest`, которое демонстрирует возможности класса `Invoice`.

```
class Invoice {  
    private String part_number;  
    private String description;  
    private int quantity;  
    private double price_per_item;  
  
    // Конструктор  
    public Invoice(String part_number, String description, int quantity,  
double price_per_item) {  
        this.part_number = part_number;  
        this.description = description;  
    }  
}
```

```
    set_quantity(quantity);
    set_price_per_item(price_per_item);
}

// Getter и Setter для part_number
public String get_part_number() {
    return part_number;
}

public void set_part_number(String part_number) {
    this.part_number = part_number;
}

// Getter и Setter для description
public String get_description() {
    return description;
}

public void set_description(String description) {
    this.description = description;
}

// Getter и Setter для quantity
public int get_quantity() {
    return quantity;
}

public void set_quantity(int quantity) {
    if (quantity < 0) {
        this.quantity = 0;
    } else {
        this.quantity = quantity;
    }
}

// Getter и Setter для price_per_item
public double get_price_per_item() {
    return price_per_item;
}

public void set_price_per_item(double price_per_item) {
    if (price_per_item < 0) {
        this.price_per_item = 0.0;
    } else {
        this.price_per_item = price_per_item;
    }
}
```

```
}

// Метод для вычисления суммы счета
public double get_invoice_amount() {
    return quantity * price_per_item;
}
}

// Тестовое приложение InvoiceTest
class InvoiceTest {
    public static void main(String[] args) {
        System.out.println("Тестирование класса Invoice:");

        Invoice invoice1 = new Invoice("121212", "Кебаб", 10, 5.99);
        System.out.println("Номер детали: " + invoice1.get_part_number());
        System.out.println("Описание: " + invoice1.get_description());
        System.out.println("Количество: " + invoice1.get_quantity());
        System.out.println("Цена за единицу: " +
invoice1.get_price_per_item());
        System.out.println("Сумма счета: " +
invoice1.get_invoice_amount());

        System.out.println();

        Invoice invoice2 = new Invoice("31890", "Дом", -5, 2.50);
        System.out.println("Номер детали: " + invoice2.get_part_number());
        System.out.println("Описание: " + invoice2.get_description());
        System.out.println("Количество: " + invoice2.get_quantity()); // 0
        System.out.println("Цена за единицу: " +
invoice2.get_price_per_item());
        System.out.println("Сумма счета: " +
invoice2.get_invoice_amount()); // 0.0

        System.out.println();

        Invoice invoice3 = new Invoice("11111", "Шар для боулинга", 3,
-1.0);
        System.out.println("Номер детали: " + invoice3.get_part_number());
        System.out.println("Описание: " + invoice3.get_description());
        System.out.println("Количество: " + invoice3.get_quantity());
        System.out.println("Цена за единицу: " +
invoice3.get_price_per_item()); // 0.0
        System.out.println("Сумма счета: " +
invoice3.get_invoice_amount()); // 0.0
    }
}
```

## №2. Класс Employee

### Условие

Создайте класс с именем `Employee`, который включает три переменные экземпляра:

- Имя (тип `String`).
- Фамилия (тип `String`).
- Ежемесячная зарплата (тип `double`).

Предоставьте конструктор, инициализирующий эти три переменные экземпляра.

Предоставьте методы `set` и `get` для каждой переменной экземпляра.

Если ежемесячная зарплата неположительная, не присваивайте ей значение.

Напишите тестовое приложение с именем `EmployeeTest`, которое демонстрирует возможности класса `Employee`. Создайте два объекта `Employee` и выведите годовую зарплату каждого объекта. Затем увеличьте зарплату каждого сотрудника на 10% и снова выведите годовую зарплату.

```
class Employee {  
    private String first_name;  
    private String last_name;  
    private double monthly_salary;  
  
    // Конструктор  
    public Employee(String first_name, String last_name, double monthly_salary) {  
        this.first_name = first_name;  
        this.last_name = last_name;  
        set_monthly_salary(monthly_salary);  
    }  
  
    // Getter и Setter для first_name  
    public String get_first_name() {  
        return first_name;  
    }  
  
    public void set_first_name(String first_name) {
```

```
    this.first_name = first_name;
}

// Getter и Setter для last_name
public String get_last_name() {
    return last_name;
}

public void set_last_name(String last_name) {
    this.last_name = last_name;
}

// Getter и Setter для monthly_salary
public double get_monthly_salary() {
    return monthly_salary;
}

public void set_monthly_salary(double monthly_salary) {
    if (monthly_salary > 0) {
        this.monthly_salary = monthly_salary;
    }
}

// Метод для получения годовой зарплаты
public double get_annual_salary() {
    return monthly_salary * 12;
}

}

// Тестовое приложение EmployeeTest
public class EmployeeTest {
    public static void main(String[] args) {
        System.out.println("Тестирование класса Employee:");

        // Создание двух объектов
        Employee employee1 = new Employee("Джон", "Доу", 5000.0);
        Employee employee2 = new Employee("Джейн", "Доу", 6000.0);

        // Вывод годовой зарплаты
        System.out.println(employee1.get_first_name() + " " +
employee1.get_last_name() +
                " - Годовая зарплата: " +
employee1.get_annual_salary());
        System.out.println(employee2.get_first_name() + " " +
employee2.get_last_name() +
                " - Годовая зарплата: " +
```

```

employee2.get_annual_salary());

        // Увеличение зарплаты на 10%
        employee1.set_monthly_salary(employee1.get_monthly_salary() *
1.10);
        employee2.set_monthly_salary(employee2.get_monthly_salary() *
1.10);

        System.out.println();

        // Вывод обновленной годовой зарплаты
        System.out.println(employee1.get_first_name() + " " +
employee1.get_last_name() +
                " - Новая годовая зарплата: " +
employee1.get_annual_salary());
        System.out.println(employee2.get_first_name() + " " +
employee2.get_last_name() +
                " - Новая годовая зарплата: " +
employee2.get_annual_salary());

        // Тест с отрицательной зарплатой (не присваивается)
        employee1.set_monthly_salary(-1000.0);
        System.out.println(employee1.get_first_name() + " " +
employee1.get_last_name() +
                " - Зарплата после установки отрицательного
значения: " +
                employee1.get_monthly_salary()); // Остается
прежней
    }
}

```

## №3. Класс Date

### Условие

Создайте класс с именем `Date`, который включает три переменные экземпляра:

- Месяц (тип `int`).
- День (тип `int`).
- Год (тип `int`).

Предоставьте конструктор, который инициализирует эти три переменные экземпляра и предполагает, что переданные значения корректны.

Предоставьте методы `set` и `get` для каждой переменной экземпляра.

Создайте метод `displayDate`, который отображает месяц, день и год, разделённые косой чертой (/).

Напишите тестовое приложение с именем `DateTest`, которое демонстрирует возможности класса `Date`.

```
class Date {  
    private int month;  
    private int day;  
    private int year;  
  
    // Конструктор (предполагает корректные значения)  
    public Date(int month, int day, int year) {  
        this.month = month;  
        this.day = day;  
        this.year = year;  
    }  
  
    // Getter и Setter для month  
    public int get_month() {  
        return month;  
    }  
  
    public void set_month(int month) {  
        this.month = month;  
    }  
  
    // Getter и Setter для day  
    public int get_day() {  
        return day;  
    }  
  
    public void set_day(int day) {  
        this.day = day;  
    }  
  
    // Getter и Setter для year  
    public int get_year() {  
        return year;  
    }  
  
    public void set_year(int year) {  
        this.year = year;  
    }  
}
```

```
}

// Метод для отображения даты
public void display_date() {
    System.out.println(month + "/" + day + "/" + year);
}
}

// Тестовое приложение DateTest
public class DateTest {
    public static void main(String[] args) {
        System.out.println("Тестирование класса Date:");

        // Создание объекта
        Date date = new Date(10, 19, 2025);

        // Вывод с помощью display_date
        System.out.print("Дата: ");
        date.display_date();

        System.out.println();

        // Демонстрация getter/setter
        System.out.println("Месяц: " + date.get_month());
        System.out.println("День: " + date.get_day());
        System.out.println("Год: " + date.get_year());

        // Изменение значений
        date.set_month(12);
        date.set_day(31);
        date.set_year(2025);

        System.out.println();
        System.out.print("Обновленная дата: ");
        date.display_date();
    }
}
```