Лабораторная работа 10. Основы объектно-ориентированного программирования. Классы и объекты

1. Создать класс для работы с точкой на плоскости (в декартовой системе координат).

Класс должен включать вещественные поля для координат точки x и y.
Класс должен включать строковое поле для хранения текстовой надписи для точки.

Все поля класса должны быть закрытыми (private)

Включить в класс геттеры и сеттеры для всех полей. Значения, записываемые в строковое поле не должны быть длиннее чем 30 символов; лишние символы обрезать.

Включить в класс два конструктора: с полным и с пустым списком параметров.

Реализовать в классе метод вычисления расстояния от точки до начала координат.

Добавить в класс метод для ввода всех данных с клавиатуры.

Добавить в класс реализацию метода toString().

В основной программе
a) Создать две точки на плоскости. Для первой точки задать координаты (3; 4) и надпись “A”. Данные второй точки ввести с клавиатуры.
Вывести расстояние между этими точками.
Увеличить значения координат первой точки в 2 раза и снова вывести расстояние между точками.

b) Создать массив из 5 точек, ввести все данные с клавиатуры.
Вывести расстояние от каждой точки до начала координат. Указать, какая из точек дальше всех от начала координат.
Считать, что точки являются вершинами многоугольника, вывести его периметр.

2. Создать класс Person, соответствующий личным данным человека:
- фамилия (String),
- имя (String),
- дата рождения (LocalDate),
- пол (один символ char)

Определить все методы, необходимые для правильной работы с объектами такого класса:
- геттеры, сеттеры, причем для поля **пол** в сеттере проверять корректность устанавливаемых значений
- конструкторы
- функции ввода-вывода,

Создать метод, возвращающий возраст человека – количество полных лет.

Протестировать все возможности класса на подходящих примерах

Написать программу, вводящую данные об указанном пользователем количестве людей в массив и выводящую для каждого из них фамилию, инициалы имени, возраст и пол.

Примеры

```
public class Product {


   // поля класса
    private String name;
    private double price;
    private double discount;


   // метод для расчета стоимости
    public double cost(double amount) {
        return price * amount * (1.0 - discount / 100.0);
    }
// конструктор (по умолчанию)
    public Product() {
    }
// конструктор с полным списком параметров
    public Product(String name, double price, double discount) {
        this.name = name;
        this.price = price;
        this.discount = discount;
    }
// геттеры и сеттеры
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        if (price > 0) {
            this.price = price;
        }
    }

    public double getDiscount() {
        return discount;
    }

    public void setDiscount(double discount) {
        this.discount = discount;
    }

    @Override
    public String toString() {
        return  name + " : price = " + price +
                ", discount = " + discount;
    }
}
```

Пример использования объектов

```
public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Product p = new Product("coffee", 5, 50);

        System.out.println("Сколько " + p.getName() + " Вы хотите?");
        double a = sc.nextDouble();
        System.out.println("Это стоит " + p.cost(a));

        int numberOfGoods = 4;
        Product[] goods = {
                new Product("chips", 2.0, 25.0),
                new Product("phone", 300.0, 0.0),
                new Product("boeing 777", 1_000_000.0, 0.0),
                new Product("coffee", 5.0, 50.0),
        };

        System.out.println("Что Вы хотите купить? Введите количество товара...");
        double sum = 0.0;
        for (int i = 0; i < numberOfGoods; i++) {
            System.out.print(   goods[i].getName()  + " : "  );
            double amount = sc.nextDouble();

            double c = goods[i].cost(amount);
            System.out.println("\t стоимость = " + c);

            sum += c;
        }
        System.out.println("---------------------------------");
        System.out.println("Все вместе стоит: " + sum);
    }
```

Примеры с LocalDate

```
        LocalDate n = LocalDate.now(); // сегодня
        System.out.println(n);

        ZoneId timezone = ZoneId.of("America/Sao_Paulo");
        LocalDate today = LocalDate.now(timezone);
        System.out.println("Сейчас в Бразилии = " + today);

        // создание новой даты с известными данными
       // LocalDate d = new LocalDate(2022, 11, 15); // error
        LocalDate d = LocalDate.of(2022, 11, 15); // год-месяц-день

        // ввод, вывод, форматирование дат
        Scanner scanner = new Scanner(System.in);
        LocalDate t = LocalDate.parse(scanner.nextLine()); // ввод

        System.out.println(t);  // вывод и форматирование
        System.out.println(t.format(DateTimeFormatter.BASIC_ISO_DATE));
        System.out.println(t.format(DateTimeFormatter.ISO_WEEK_DATE));

        System.out.println(t.format(DateTimeFormatter.ofPattern("dd MM yyyy")));
        System.out.println(t.format(DateTimeFormatter.ofPattern("dd LLLL yyyy")));


// отдельные компоненты даты
        System.out.println( d.getDayOfMonth() ); // день
        System.out.println( d.getDayOfWeek() ); // день недели

        System.out.println( d.getMonthValue() ); // номер месяца
        System.out.println( d.getMonth()); // название месяца

        System.out.println( d.getYear() ); // год

        System.out.println(d);  // вся дата в String

// операции с датами -
        LocalDate p = n.plusDays(90);
        p = n.minusDays(10); // сдвиги +/-, ...
        p = n.plusMonths(9); // сдвиги +/-, ...
        p = n.minusMonths(2); //  сдвиги +/-, ...
        p = n.plusYears(4); // сдвиги +/-, ...
        p = n.plusWeeks(4); //  сдвиги +/-, ...

        System.out.println(p);

        System.out.println( Period.between(n, LocalDate.of(2022, 4, 1)).getDays());
//-----------------------------------------------------------------------
```