МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«**Московский государственный университет
информационных технологий, радиотехники и электроники»**

МТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра Информатики и Информационных систем (ИИС)

ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ

по дисциплине «Объектно-ориентированное программирование»

Лабораторные работы № 1 – 8

Выполнил студент группы ИСБОп-01-14 Карих Д.С.

Принял Зорина Н.В.

Лабораторную работу выполнил	«__» _____ 20__г.	Карих Д.С.
«Зачтено»	«__» _____ 20__г.	Зорина Н.В.

Москва 2015

Оглавление

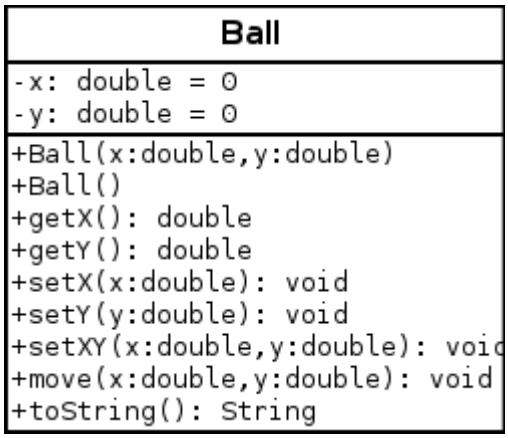
Лабораторная работа № 1

Задание № 1

По диаграмме класса UML, описывающей сущность Мяч, написать программу, которая состоит из двух классов Ball и Main.

Структура программы

Исходный код программы состоит из двух файлов: главного класса Main и класса Ball. Структура класса Ball изображена на следующей UML диаграмме:



Код программы

- Класс Ball

```
package pw.thedrhax.labs_3_sem.Task_1_1;

public class Ball {
    private double x;
    private double y;

    public Ball (double x, double y) {setXY(x,y);}
    public Ball () {this(0, 0);}

    public double getX () {return x;}
    public double getY () {return y;}

    public void setX (double x) {this.x = x;}
    public void setY (double y) {this.y = y;}
    public void setXY (double x, double y) {setX(x); setY(y);}

    public void move (double x, double y) {
        setXY(getX()+x, getY()+y);
    }
```

```
    @Override
    public String toString() {
        return "Ball @ (" + getX() + ", " + getY() + ")";
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_1_1;

public class Main {
    public static void main(String[] args) {
        // Создаем экзепляр класса Ball
        Ball ball = new Ball(1, 1);
        System.out.println(ball);

        // Передвигаем объект на 5 по x и y
        ball.move(5, 5);
        System.out.println(ball);
    }
}
```

Пример выполнения

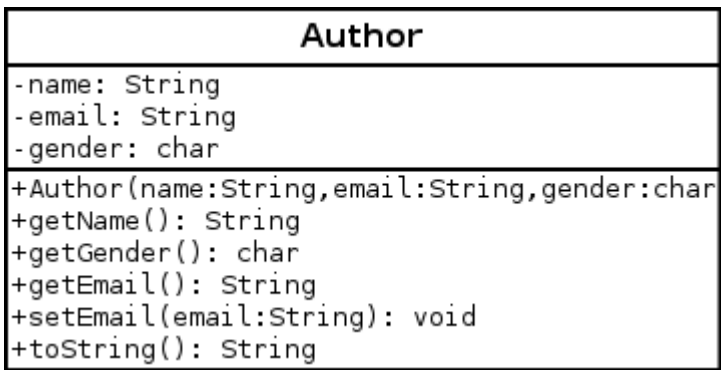
```
Ball @ (1.0, 1.0)
Ball @ (6.0, 6.0)
```

Задание № 2

По диаграмме класса UML, описывающей сущность Автор, написать программу, которая состоит из двух классов Author и Main.

Структура программы

Исходный код программы состоит из двух файлов: главного класса Main и класса Author. Структура класса Author изображена на следующей UML диаграмме:



Код программы

- Класс Author

```
package pw.thedrhax.labs_3_sem.Task_1_2;

public class Author {
    private String name;
    private String email;
    private char gender;

    public Author (String name, String email, char gender) {
        this.name = name;
        setEmail(email);
        this.gender = gender;
    }

    public String getName() {return name;}
    public char getGender() {return gender;}

    public String getEmail() {return email;}
    public void setEmail(String email) {this.email = email;}

    @Override
    public String toString() {
        return getName() + " (" + getGender() + ") at " + getEmail();
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_1_2;

public class Main {
    public static void main(String[] args) {
        // Создаем экземпляр класса Author
        Author author = new Author("Айбелив Айкенфлаев",
"email@email.com", 'м');
        System.out.println(author);

        // Меняем E-mail
        author.setEmail("new_email@email.com");
        System.out.println(author);
    }
}
```

Пример выполнения

```
Айбелив Айкенфлаев (м) at email@email.com
Айбелив Айкенфлаев (м) at new_email@email.com
```

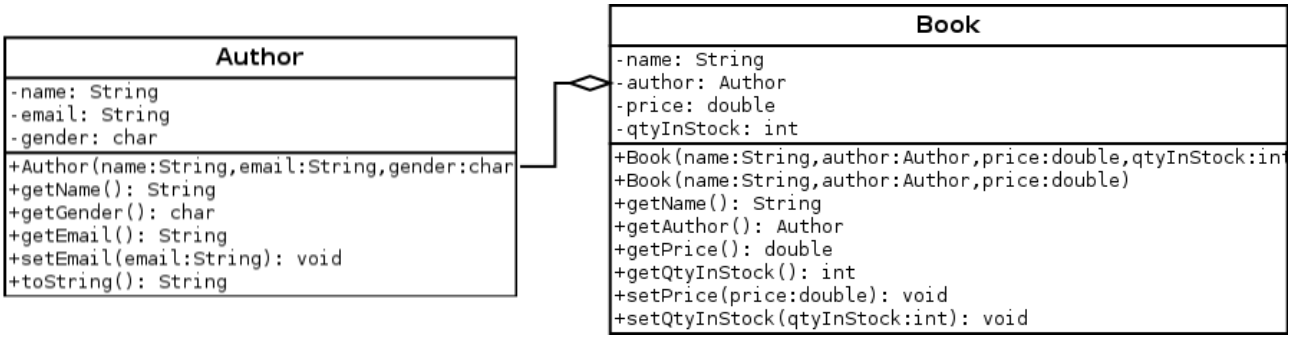
Лабораторная работа № 2

Задание № 1

Написать программу, в которой класс Книга (Book), будет находиться в отношениях композиции с классом Author.

Структура программы

Исходный код программы состоит из двух файлов: главного класса Main и класса Book, а также использует класс Author из пакета Task_1_2. Структура классов и их отношения изображены на следующей UML диаграмме:



Код программы

- Класс Book

```
package pw.thedrhax.labs_3_sem.Task_2;

// Импортируем класс Author из задания 2 ЛР 1
import pw.thedrhax.labs_3_sem.Task_1_2.Author;

public class Book {
    private String name;
    private Author author;
    private double price;
    private int qtyInStock;

    public Book (String name, Author author, double price, int
qtyInStock) {
        this.name = name;
        this.author = author;
        setPrice(price);
        setQtyInStock(qtyInStock);
    }
    public Book (String name, Author author, double price) {
```

```
        this(name, author, price, 0);
    }

    public String getName() {return name;}
    public Author getAuthor() {return author;}
    public double getPrice() {return price;}
    public int getQtyInStock() {return qtyInStock;}

    public void setPrice(double price) {this.price = price;}
    public void setQtyInStock(int qtyInStock) {this.qtyInStock =
qtyInStock;}

    @Override
    public String toString() {
        return getName() + " // " + author;
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_2;

// Импортируем класс Author из задания 2 ЛР 1
import pw.thedrhax.labs_3_sem.Task_1_2.Author;

public class Main {

    public static void main(String[] args) {
        Author author = new Author("Айбелив Айкенфлаев",
"email@email.com", 'м');
        Book book = new Book("Вера в полёты", author, 0, 1000);
        System.out.println(book);
    }
}
```

Пример выполнения

```
Вера в полёты // Айбелив Айкенфлаев (м) at email@email.com
```

Лабораторная работа № 3

Задание № 1

В массиве, элементами которого являются целые числа, произвести следующие действия:

1. Если элементы меньше заданного числа, замените их этим числом.
2. Замените все элементы с индексами в диапазоне [a, b] нулями.
3. Поменяйте местами первый и последний элементы массива.
4. Получите сумму и среднее арифметическое всех элементов массива.
5. Получите корень из суммы квадратов всех элементов массива (модуль вектора).
6. Для двух массивов получите сумму попарных произведений их членов (скалярное произведение).
7. Для двух массивов получите Евклидово расстояние между ними.
8. Найдите максимальный и минимальный элементы массива.
9. Отсортируйте элементы массива методом выбора.
10. Отсортируйте элементы массива методом пузырька.

Структура программы

Исходный код программы состоит из одного файла: главного класса Main. Программа не нуждается в документировании при помощи UML диаграмм.

Код программы

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_3;

import java.util.Arrays;
import java.util.Random;

public class Main {
    private static final Random random = new Random();
    // Генератор случайных массивов
    public static int[] generateArray (int size) {
        int[] array = new int[size];
        for (int i = 0; i < array.length; i++)
            array[i] = random.nextInt(10);
```

```
        return array;
    }

    public static void main(String[] args) {
        int[] first = generateArray(10);
        int[] second = generateArray(10);
        System.out.println("first = " + Arrays.toString(first));
        System.out.println("second = " + Arrays.toString(second));

        // Задание 1
        int min = random.nextInt(10);
        System.out.println("1) Если элементы меньше " + min + ",
заменить их на " + min);
        for (int i = 0; i < first.length; i++)
            if (first[i] < min) first[i] = min;
        System.out.println("first = " + Arrays.toString(first));

        // Задание 2
        int a = random.nextInt(5); int b = random.nextInt(5)+5;
        System.out.println("2) Заменить все элементы в диапазоне [" + a
+ ", " + b + "] нулями");
        for (int i = a; i < b; i++)
            first[i] = 0;
        System.out.println("first = " + Arrays.toString(first));

        // Задание 3
        System.out.println("3) Поменять местами первый и последний
элементы");
        int temp = first[0];
        first[0] = first[first.length-1];
        first[first.length-1] = temp;
        System.out.println("first = " + Arrays.toString(first));

        // Задание 4
        System.out.println("4) Найти сумму и среднее арифметическое");
        int sum = 0;
        for (int i : first) sum += i;
        System.out.println("Сумма: " + sum + "; Среднее: " +
sum/first.length);

        // Задание 5
        System.out.println("5) Сумма квадратов всех элементов");
        sum = 0;
        for (int i : first) sum += i * i;
        System.out.println(sum);

        // Задание 6
        System.out.println("6) Сумма попарных произведений двух
массивов");
        sum = 0;
        for (int i = 0; i < first.length; i++)
            sum += first[i] * second[i];
```

```
        System.out.println(sum);

        // Задание 7
        System.out.println("7) Евклидово расстояние между массивами");
        sum = 0;
        for (int i = 0; i < first.length; i++)
            sum += Math.pow(first[i] - second[i], 2);
        System.out.println(Math.sqrt(sum));

        // Задание 8
        System.out.println("8) Минимальный и максимальный элемент
массива");
        min = 10; int max = 0;
        for (int i : first) {
            if (i < min) min = i;
            if (i > max) max = i;
        }
        System.out.println("Минимум: " + min + ", максимум: " + max);

        // Задание 9
        System.out.println("9) Сортировка выбором");
        System.out.println(Arrays.toString(first));
        for (int i = 0; i < first.length; i++) {
            for (int j = i+1; j < first.length; j++) {
                if (first[j] < first[i]) {
                    temp = first[i];
                    first[i] = first[j];
                    first[j] = temp;
                }
            }
        }
        System.out.println(Arrays.toString(first));

        // Задание 10
        System.out.println("10) Сортировка пузырьком");
        System.out.println(Arrays.toString(second));
        for (int i = 0; i < second.length-1; i++) {
            for (int j = 0; j < second.length-i-1; j++) {
                if (second[j] > second[j+1]) {
                    temp = second[j];
                    second[j] = second[j+1];
                    second[j+1] = temp;
                }
            }
        }
        System.out.println(Arrays.toString(second));
    }
}
```

Пример выполнения

```
first = [8, 2, 0, 5, 5, 6, 0, 8, 7, 0]
```

```
second = [6, 8, 1, 2, 8, 4, 1, 8, 5, 3]
1) Если элементы меньше 3, заменить их на 3
first = [8, 3, 3, 5, 5, 6, 3, 8, 7, 3]
2) Заменить все элементы в диапазоне [4, 8] нулями
first = [8, 3, 3, 5, 0, 0, 0, 0, 7, 3]
3) Поменять местами первый и последний элементы
first = [3, 3, 3, 5, 0, 0, 0, 0, 7, 8]
4) Найти сумму и среднее арифметическое
Сумма: 29; Среднее: 2
5) Сумма квадратов всех элементов
165
6) Сумма попарных произведений двух массивов
114
7) Евклидово расстояние между массивами
14.866068747318506
8) Минимальный и максимальный элемент массива
Минимум: 0, максимум: 8
9) Сортировка выбором
[3, 3, 3, 5, 0, 0, 0, 0, 7, 8]
[0, 0, 0, 0, 3, 3, 3, 5, 7, 8]
10) Сортировка пузырьком
[6, 8, 1, 2, 8, 4, 1, 8, 5, 3]
[1, 1, 2, 3, 4, 5, 6, 8, 8, 8]
```

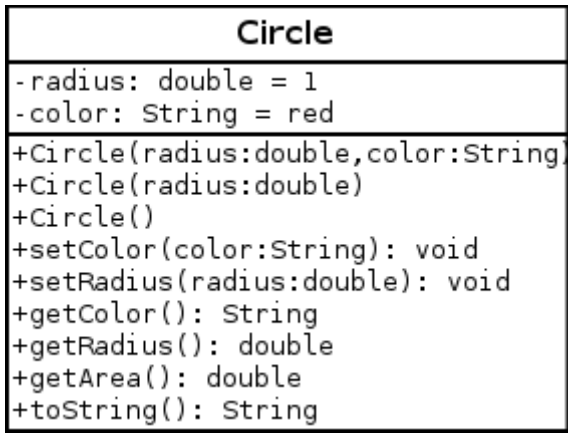
Лабораторная работа № 4

Задание № 1

По UML диаграмме класса Circle написать программу, которая состоит из двух классов Circle и Main.

Структура программы

Исходный код программы состоит из двух файлов: главного класса Main и класса Circle. Структура класса Circle изображена на следующей UML диаграмме:



Код программы

- Класс Circle

```
package pw.thedrhax.labs_3_sem.Task_4_1;

public class Circle {
    private double radius;
    private String color;

    // Конструкторы
    public Circle (double radius, String color) {
        setRadius(radius);
        setColor(color);
    }
    public Circle (double radius) {this(radius, "red");}
    public Circle () {this(1.0);}

    // Сеттеры
    public void setColor(String color) {this.color = color;}
    public void setRadius(double radius) {this.radius = radius;}

    // Геттеры
    public String getColor() {return color;}
```

```
    public double getRadius() {return radius;}
    public double getArea() {return Math.PI * Math.pow(getRadius(), 2);}

    @Override
    public String toString() {
        return getColor() + " circle with radius=" + getRadius() + " and
area=" + getArea();
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_4_1;

public class Main {
    public static void main(String[] args) {
        // Создаем экзепляр класса Circle с радиусом 2
        Circle circle = new Circle(2);
        System.out.println(circle);
        // Меняем цвет круга и его радиус
        circle.setColor("blue");
        circle.setRadius(5);
        System.out.println(circle);
    }
}
```

Пример выполнения

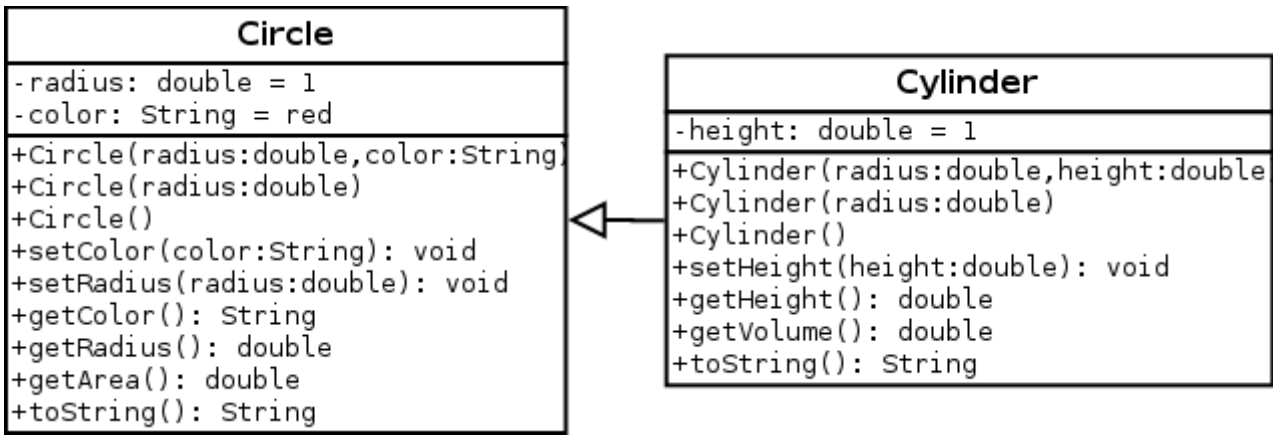
```
red circle with radius=2.0 and area=12.566370614359172
blue circle with radius=5.0 and area=78.53981633974483
```

Задание № 2

Написать программу, в которой класс Cylinder будет находиться в отношениях наследования с классом Circle.

Структура программы

Исходный код программы состоит из двух файлов: главного класса Main и класса Cylinder, а также использует класс Circle из пакета Task_4_1. Структура классов и их отношения изображены на следующей UML диаграмме:



Код программы

- Класс Cylinder

```
package pw.thedrhax.labs_3_sem.Task_4_2;

// Импортируем класс Circle из задания 1 ЛР 4
import pw.thedrhax.labs_3_sem.Task_4_1.Circle;

public class Cylinder extends Circle {
    private double height;

    public Cylinder (double radius, double height) {
        super(radius);
        this.height = height;
    }
    public Cylinder (double radius) {this(radius, 1);}
    public Cylinder () {this(1, 1);}

    public void setHeight(double height) {this.height = height;}

    public double getHeight() {return height;}
    public double getVolume() {return getArea() * getHeight();}

    @Override
    public String toString() {
        return "cylinder with base=(" + super.toString() + "), height="
+ getHeight() + " and volume=" + getVolume();
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_4_2;

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

```
        // Создаем экземпляр класса Cylinder
        Cylinder cylinder = new Cylinder(5, 5);
        System.out.println(cylinder);
    }
}
```

Пример выполнения

```
cylinder with base=(red circle with radius=5.0 and
area=78.53981633974483), height=5.0 and volume=392.69908169872417
```

Лабораторная работа № 5

Задания № 1 – 6

1. Из массива строк str {“ласточка”, “весною”, “в”, “сени”, “к”, “нам”, “летит”} создайте массив, отсортированный в лексикографическом порядке с использованием алгоритмов сортировки
    1. пузырька
    2. методом прямого выбора
2. Доработайте задание 1 таким образом, чтобы при сортировке учитывался регистр.
3. Напишите программу инвертирования строки введенной пользователем.
4. Напишите программу, которая проверяет, является ли строка, введенная пользователем палиндромом.
5. Преобразовать строку так, чтобы буквы каждого слова в ней были отсортированы по алфавиту.
6. Подсчитать сколько раз встречается в строке буква, заданная пользователем поиск по образцу.

Структура программы

Исходный код программы состоит из одного файла: главного класса Main. Программа не нуждается в документировании при помощи UML диаграмм.

Код программы

```
package pw.thedrhax.labs_3_sem.Task_5;

import java.util.Arrays;
import java.util.Scanner;

public class Main {
    // Сортировка массива строк методом пузырька
    public static String[] bubbleSort (String[] array) {
        for (int i = 0; i < array.length-1; i++) {
            for (int j = 0; j < array.length-i-1; j++) {
                if (array[j].toLowerCase().charAt(0) >
array[j+1].toLowerCase().charAt(0)) {
                    String temp = array[j];
                    array[j] = array[j+1];
                    array[j+1] = temp;
                }
```

```
            }
        }
        return array;
    }
    // Сортировка массива строк методом выбора
    public static String[] selectSort (String array[]) {
        for (int i = 0; i < array.length; i++) {
            for (int j = i+1; j < array.length; j++) {
                if (array[i].toLowerCase().charAt(0) >
array[j].toLowerCase().charAt(0)) {
                    String temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
            }
        }
        return array;
    }

    public static void main(String[] args) {
        String text = "Ласточка весною в сени к нам летит";
        String array[] = text.split(" ");

        // Задание 1а
        System.out.println("1а) Сортировка массива строк пузырьком");
        System.out.println(Arrays.toString(array));
        System.out.println(Arrays.toString(bubbleSort(array)));

        // Задание 1б
        System.out.println("1б) Сортировка массива строк выбором");
        System.out.println(Arrays.toString(array));
        System.out.println(Arrays.toString(selectSort(array)));

        // Задание 3
        System.out.println("3) Переворачивание строки");
        Scanner sc = new Scanner(System.in);
        System.out.print("Введите строку: ");
        text = sc.nextLine();
        StringBuilder buffer = new StringBuilder();
        for (int i = text.length() - 1; i >= 0; i--) {
            buffer.append(text.charAt(i));
        }
        System.out.println(buffer);

        // Задание 4
        System.out.println("4) Проверка на палиндром");
        boolean isPolyndrom = true;
        for (int i = 0; i < text.length(); i++)
            if (text.toLowerCase().charAt(text.length()-1-i) !=
text.toLowerCase().charAt(i))
                isPolyndrom = false;
        System.out.println("Введенная строка - " + (isPolyndrom ? "" :
```

```
"не ") + "палиндром");

        // Задание 5
        System.out.println("5) Сортировка букв в каждом слове");
        StringBuilder result = new StringBuilder();
        for (String word : text.split(" ")) {
            String letter_array[] = new String[word.length()];
            for (int i = 0; i < word.length(); i++)
                letter_array[i] = word.substring(i, i+1);
            letter_array = selectSort(letter_array);
            for (String s : letter_array) {
                result.append(s);
            }
            result.append(" ");
        }
        System.out.println(result);

        // Задание 6
        System.out.println("6) Посчитать появления указанной буквы");
        System.out.print("Введите букву: ");
        char letter = sc.nextLine().charAt(0);
        int count = 0;
        for (char c : text.toCharArray())
            if (c == letter) count++;
        System.out.println("Буква '" + letter + "' встречается " + count
+ " раз(а)");
    }
}
```

Пример выполнения

```
1а) Сортировка массива строк пузырьком
[Ласточка, весною, в, сени, к, нам, летит]
[весною, в, к, Ласточка, летит, нам, сени]
1б) Сортировка массива строк выбором
[весною, в, к, Ласточка, летит, нам, сени]
[весною, в, к, Ласточка, летит, нам, сени]
3) Переворачивание строки
Введите строку: Тестовая строка для переворачивания
яинавичароверепе ялд акортс яавотсеТ
4) Проверка на палиндром
Введенная строка - не палиндром
5) Сортировка букв в каждом слове
авеостТя акорст для аавввееиинопррчя
6) Посчитать появления указанной буквы
Введите букву: а
Буква 'а' встречается 4 раз(а)
```

Лабораторная работа № 6

Задание № 1

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

- Входные данные

Программа получает на вход две строки: первая строка содержит 5 карт первого игрока, вторая - 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

- Выходные данные

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 106 ходов игра не заканчивается, программа должна вывести слово botva.

Структура программы

Исходный код программы состоит из одного файла: главного класса Main. Программа не нуждается в документировании при помощи UML диаграмм.

Код программы

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_6;

import java.util.LinkedList;
import java.util.Scanner;

public class Main {
    // Условие, определяющее, чья карта сильнее
    public static boolean beats (int first, int second) {
        return (first > second && first != 9 && second != 0 || first ==
0 && second == 9);
    }
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Игроки представлены в виде связанных списков
        LinkedList<Integer> first = new LinkedList<>();
        LinkedList<Integer> second = new LinkedList<>();

        for (int i = 0; i < 5; i++)
            first.addLast(sc.nextInt());
        for (int i = 0; i < 5; i++)
            second.addLast(sc.nextInt());

        int count = 0;
        while (!first.isEmpty() && !second.isEmpty()) {
            if (beats(first.get(0), second.get(0))) {
                second.addLast(first.get(0));
                second.addLast(second.get(0));
                first.remove(0);
                second.remove(0);
            } else {
                first.addLast(first.get(0));
                first.addLast(second.get(0));
                first.remove(0);
                second.remove(0);
            }

            if (count++ == 105) {
                System.out.println("botva");
                return;
            }
        }

        System.out.println((first.isEmpty() ? "first" : "second") + " "
+ count);
    }
}
```

Пример выполнения

```
1 3 5 7 9
2 4 6 8 0
second 5
```

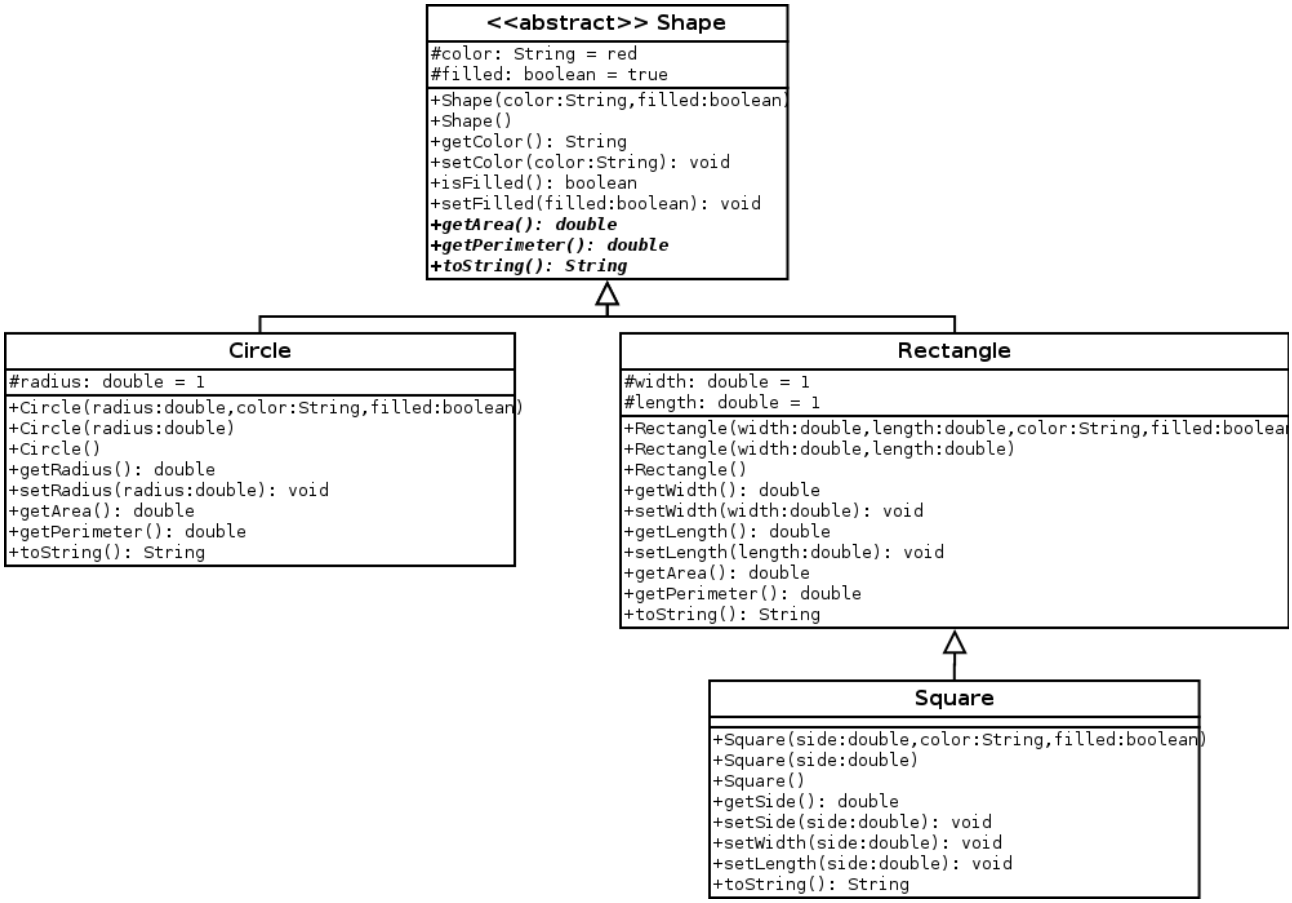
Лабораторная работа № 7

Задание № 1

Перепишите суперкласс Shape и его подклассы Circle, Rectangle и Square так, как это представленно на UML диаграмме.

Структура программы

Исходный код программы состоит из 5 файлов: главного класса Main, абстрактного класса Shape и его подклассов Circle, Rectangle и Square. Структура классов и их отношения изображены на следующей UML диаграмме:



Код программы

- Абстрактный класс Shape

```
package pw.thedrhax.labs_3_sem.Task_7;

public abstract class Shape {
    protected String color;
```

```
    protected boolean filled;

    public Shape (String color, boolean filled) {
        setColor(color);
        setFilled(filled);
    }
    public Shape () {this("red", true);}

    public String getColor () {return color;}
    public void setColor (String color) {this.color = color;}

    public boolean isFilled () {return filled;}
    public void setFilled (boolean filled) {this.filled = filled;}

    abstract public double getArea();
    abstract public double getPerimeter();
    abstract public String toString();
}
```

- Класс Circle

```
package pw.thedrhax.labs_3_sem.Task_7;

public class Circle extends Shape {
    protected double radius;

    public Circle(double radius, String color, boolean filled) {
        super(color, filled);
        setRadius(radius);
    }

    public Circle(double radius) {
        super();
        setRadius(radius);
    }

    public Circle() {
        this(1);
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return Math.PI * Math.pow(radius, 2);
    }
```

```
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }

    public String toString() {
        return "Circle with radius=" + radius +
                ", area=" + getArea() +
                ", perimeter=" + getPerimeter();
    }
}
```

- Класс Rectangle

```
package pw.thedrhax.labs_3_sem.Task_7;

public class Rectangle extends Shape {
    protected double width;
    protected double length;

    public Rectangle(double width, double length, String color, boolean
filled) {
        super(color, filled);
        setWidth(width);
        setLength(length);
    }

    public Rectangle(double width, double length) {
        super();
        setWidth(width);
        setLength(length);
    }

    public Rectangle() {
        this(1, 1);
    }

    public double getWidth() {
        return width;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getLength() {
        return length;
    }

    public void setLength(double length) {
        this.length = length;
    }
```

```
    public double getArea() {
        return width * length;
    }

    public double getPerimeter() {
        return 2 * (width + length);
    }

    public String toString() {
        return "Rectangle with width=" + width +
                ", length=" + length +
                ", area=" + getArea() +
                ", perimeter=" + getPerimeter();
    }
}
```

- Класс Square

```
package pw.thedrhax.labs_3_sem.Task_7;

public class Square extends Rectangle {
    public Square(double side, String color, boolean filled) {
        super(side, side, color, filled);
    }
    public Square(double side) {
        super(side, side);
    }
    public Square() {
        this(1);
    }

    public double getSide() {return width;}
    public void setSide(double side) {width = side; length = side;}

    public void setWidth(double side) {setSide(side);}
    public void setLength(double side) {setSide(side);}

    public String toString() {
        return "Square with side=" + getSide() +
                ", area=" + getArea() +
                ", perimeter=" + getPerimeter();
    }
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_7;

public class Main {

    public static void main(String[] args) {
        Shape s1 = new Circle(5.5, "RED", false);
```

```
        System.out.println(s1);
        System.out.println(s1.getArea());
        System.out.println(s1.getPerimeter());
        System.out.println(s1.getColor());
        System.out.println(s1.isFilled());
        // Ошибка: у класса Shape нет метода getRadius()
        //System.out.println(s1.getRadius());

        Circle c1 = (Circle)s1;
        System.out.println(c1);
        System.out.println(c1.getArea());
        System.out.println(c1.getPerimeter());
        System.out.println(c1.getColor());
        System.out.println(c1.isFilled());
        // у класса Circle есть метод getRadius()
        System.out.println(c1.getRadius());

        // Ошибка: нельзя создать экземпляр абстрактного класса
        //Shape s2 = new Shape();

        Shape s3 = new Rectangle(1.0, 2.0, "RED", false);
        System.out.println(s3);
        System.out.println(s3.getArea());
        System.out.println(s3.getPerimeter());
        System.out.println(s3.getColor());
        // Ошибка: у класса Shape не метода getLength()
        //System.out.println(s3.getLength());

        Rectangle r1 = (Rectangle)s3;
        System.out.println(r1);
        System.out.println(r1.getArea());
        System.out.println(r1.getColor());
        // у класса Rectangle есть метод getLength()
        System.out.println(r1.getLength());

        Shape s4 = new Square(6.6);
        System.out.println(s4);
        System.out.println(s4.getArea());
        System.out.println(s4.getColor());
        // Ошибка: у класса Shape нет метода getSide()
        //System.out.println(s4.getSide());

        Rectangle r2 = (Rectangle)s4;
        System.out.println(r2);
        System.out.println(r2.getArea());
        System.out.println(r2.getColor());
        // Ошибка: у класса Rectangle тоже нет метода getSide()
        //System.out.println(r2.getSide());
        System.out.println(r2.getLength());

        Square sq1 = (Square)r2;
        System.out.println(sq1);
```

```
        System.out.println(sq1.getArea());
        System.out.println(sq1.getColor());
        // у класса Square есть метод getSide()
        System.out.println(sq1.getSide());
        System.out.println(sq1.getLength());
    }
}
```

Пример выполнения

```
Circle with radius=5.5, area=95.03317777109125,
perimeter=34.55751918948772
95.03317777109125
34.55751918948772
RED
false
Circle with radius=5.5, area=95.03317777109125,
perimeter=34.55751918948772
95.03317777109125
34.55751918948772
RED
false
5.5
Rectangle with width=1.0, length=2.0, area=2.0, perimeter=6.0
2.0
6.0
RED
Rectangle with width=1.0, length=2.0, area=2.0, perimeter=6.0
2.0
RED
2.0
Square with side=6.6, area=43.559999999999995, perimeter=26.4
43.559999999999995
red
Square with side=6.6, area=43.559999999999995, perimeter=26.4
43.559999999999995
red
6.6
Square with side=6.6, area=43.559999999999995, perimeter=26.4
43.559999999999995
red
6.6
6.6
```

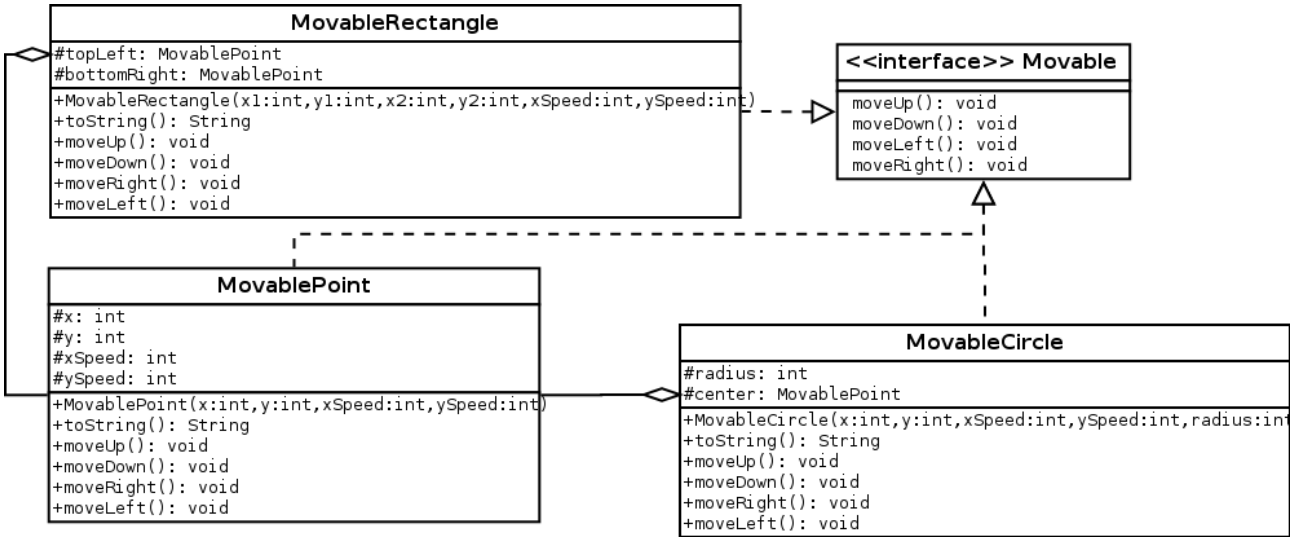
Лабораторная работа № 8

Задание № 1

Написать два класса MovablePoint и MovableCircle, которые реализуют интерфейс Movable .

Структура программы

Исходный код программы состоит из 5 файлов: главного класса Main, интерфейса Movable и его реализаций MovablePoint, MovableCircle и MovableRectangle. Структура классов и их отношения изображены на следующей UML диаграмме:



Код программы

- Интерфейс Movable

```
package pw.thedrhax.labs_3_sem.Task_8;

public interface Movable {
    void moveUp ();
    void moveDown ();
    void moveLeft ();
    void moveRight ();
}
```

- Класс MovablePoint

```
package pw.thedrhax.labs_3_sem.Task_8;
```

```
public class MovablePoint implements Movable {
    protected int x;
    protected int y;
    protected int xSpeed;
    protected int ySpeed;

    public MovablePoint (int x, int y, int xSpeed, int ySpeed) {
        this.x = x;
        this.y = y;
        this.xSpeed = xSpeed;
        this.ySpeed = ySpeed;
    }

    @Override
    public String toString() {return "Point (" + x + ", " + y + ")";}

    @Override
    public void moveUp() {y++;}
    @Override
    public void moveDown() {y--;}
    @Override
    public void moveRight() {x++;}
    @Override
    public void moveLeft() {x--;}
}
```

- Класс MovableCircle

```
package pw.thedrhax.labs_3_sem.Task_8;

public class MovableCircle implements Movable {
    protected int radius;
    protected MovablePoint center;

    public MovableCircle(int x, int y, int xSpeed, int ySpeed, int
radius) {
        this.radius = radius;
        this.center = new MovablePoint(x, y, xSpeed, ySpeed);
    }

    @Override
    public String toString() {return "Circle (" + center.x + ", " +
center.y + ") with radius=" + radius;}

    @Override
    public void moveUp() {center.moveUp();}
    @Override
    public void moveDown() {center.moveDown();}
    @Override
    public void moveRight() {center.moveRight();}
    @Override
    public void moveLeft() {center.moveLeft();}
```

```
}
```

- Класс MovableRectangle

```
package pw.thedrhax.labs_3_sem.Task_8;

public class MovableRectangle implements Movable {
    private MovablePoint topLeft;
    private MovablePoint bottomRight;

    public MovableRectangle (int x1, int y1, int x2, int y2, int xSpeed,
int ySpeed) {
        this.topLeft = new MovablePoint(x1, y1, xSpeed, ySpeed);
        this.bottomRight = new MovablePoint(x2, y2, xSpeed, ySpeed);
    }

    @Override
    public String toString() {
        return "Rectangle (" + topLeft.x + ", " + topLeft.y + ", " +
                bottomRight.x + ", " + bottomRight.y + ")";
    }

    @Override
    public void moveUp() {topLeft.moveUp(); bottomRight.moveUp();}
    @Override
    public void moveDown() {topLeft.moveDown(); bottomRight.moveDown();}
    @Override
    public void moveRight() {topLeft.moveRight();
bottomRight.moveRight();}
    @Override
    public void moveLeft() {topLeft.moveLeft(); bottomRight.moveLeft();}
}
```

- Класс Main

```
package pw.thedrhax.labs_3_sem.Task_8;

public class Main {
    public static void main(String[] args) {
        // Создаем экземпляр класса MovablePoint
        MovablePoint point = new MovablePoint(0, 0, 1, 1);
        System.out.println(point);
        // Двигаем point вверх и влево
        point.moveUp();
        point.moveLeft();
        System.out.println(point);
        // Создаем экземпляр класса MovableCircle
        MovableCircle circle = new MovableCircle(0, 0, 1, 1, 2);
        System.out.println(circle);
        // Двигаем circle вниз и вправо
        circle.moveDown();
        circle.moveRight();
```

```
        System.out.println(circle);
        // Создаем экземпляр класса MovableRectangle
        MovableRectangle rectangle = new MovableRectangle(0, 0, 1, 1, 1,
1);
        System.out.println(rectangle);
        // Двигаем rectangle вверх и влево
        rectangle.moveUp();
        rectangle.moveLeft();
        System.out.println(rectangle);
    }
}
```

Пример выполнения

```
Point (0, 0)
Point (-1, 1)
Circle (0, 0) with radius=2
Circle (1, -1) with radius=2
Rectangle (0, 0, 1, 1)
Rectangle (-1, 1, 0, 2)
```

Список использованных источников

1. Документация Java // интернет-ресурс docs.oracle.com
2. Документация IntelliJ IDEA // интернет-ресурс jetbrains.com
3. Личный опыт