



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий

КАФЕДРА ИНСТРУМЕНТАЛЬНОГО И ПРИКЛАДНОГО
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

ПРАКТИЧЕСКИЕ РАБОТЫ
ПО ДИСЦИПЛИНЕ «Программирование на языке Джава»

Выполнил студент группы ИМБО-02-22

Лищенко Т. В.

Принял старший преподаватель

Рачков А.В.

Практические работы работа выполнены «__»_____2023г.

«Зачтено» «__»_____2023г.

Москва 2023

Оглавление

<i>Практическая работа №1</i>	4
Цель работы:	4
Задача №3 и №4	5
Условие задачи:	5
Решение:	6
Результат выполнения программы:	8
Задача №5	9
Условие задачи:	9
Решение:	10
Результат выполнения программы:	11
Задача №6	12
Условие задачи	12
Решение:	13
Результат выполнения программы:	14
Задача №7	15
Условие задачи:	15
Решение:	16
Результат выполнения программы:	17
Вывод:	18
<i>Практическая работа №2</i>	19
Цель работы:	19
Задача №1	20
Условие:	20
Решение:	21
Класс Author	21
Класс TestAuthor	23
Результат выполнения программы:	24
Задача №2	25
Условие задачи:	25
Решение:	26
Результат выполнения программы:	27
Задача №5	28
Условие задачи:	28
Решение:	29
Результат выполнения программы:	32
Вывод:	33
<i>Практическая работа №3</i>	34
Цель работы:	34

Задача №1	35
Условие задачи:	35
Решение:	36
Результат выполнения программы:	37
Вывод	38
<i>Практическая работа №4</i>	39
Цель работы:	39
Задача №1	40
Условие задачи:	40
Решение:	41
Результат выполнения программы:	42
Задача №1 Блок 2	43
Условие задачи:	43
Решение:	44
Результат выполнения программы:	49
Вывод	50
<i>Используемая литература</i>	51

Практическая работа №1

Цель работы:

Познакомиться со средой разработки IntelliJ IDE, реализовать простейшие задачи на языке JAVA

Задача №3 и №4

Условие задачи:

Написать программу, в результате которой массив чисел создается с помощью инициализации (как в Си) вводится и считается в цикле сумма элементов целочисленного массива, а также среднее арифметическое его элементов результат выводится на экран. Использовать цикл for.

Написать программу, в результате которой массив чисел вводится пользователем с клавиатуры считается сумма элементов целочисленного массива с помощью циклов do while, while, также необходимо найти максимальный и минимальный элемент в массиве, результат выводится на экран.

Решение:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in); // Объявляем Scanner
        System.out.println("Enter array length: ");

        int size = input.nextInt(); // Читаем с клавиатуры размер массива и
записываем в size
        int array[] = new int[size]; // Создаём массив int размером в size

        System.out.println("Insert array elements:");

        /*Пройдёмся по всему массиву, заполняя его*/
        for (int i = 0; i < size; i++)
            array[i] = input.nextInt(); // Заполняем массив элементами,
введёнными с клавиатуры

        System.out.print ("Inserted array elements:");

        for (int i = 0; i < size; i++) {
            System.out.print (" " + array[i]); // Выводим на экран, полученный
массив
        }

        System.out.println();

        int sum = 0;
        for (int i = 0; i < size; i++){
            sum += array[i];
        }

        double average;
        average = (double) sum / array.length;

        System.out.println("Average: " + average);
        System.out.println("Sum: " + sum);

        int j = 0;
        int sum_while = 0;

        do{
```

```

        sum_while += array[j];
        j++;
    } while(j!=size);

    System.out.println("Sum with do while" + sum_while);

    int max = getMax(array);
    System.out.println("Maximum Value is: " + max);

    int min = getMin(array);
    System.out.println("Minimum Value is: " + min);
}
// Здесь находим максимум
public static int getMax(int[] inputArray){
    int maxValue = inputArray[0];
    for(int i=1; i < inputArray.length; i++){
        if(inputArray[i] > maxValue){
            maxValue = inputArray[i];
        }
    }

    return maxValue;
}
// здесь находим минимум
public static int getMin(int[] inputArray){
    int minValue = inputArray[0];
    for(int i=1;i<inputArray.length;i++){
        if(inputArray[i] < minValue){
            minValue = inputArray[i];
        }
    }
    return minValue;
}
}

```

Результат выполнения программы:

```
Enter array length:
3
Insert array elements:
4
5
11
Inserted array elements: 4 5 11
Average: 6.666666666666667
Sum: 20
Sum with do while20
Maximum Value is: 11
Minimum Value is: 4
```


Задача №5

Условие задачи:

Написать программу, в результате которой выводятся на экран аргументы командной строки в цикле for.

Решение:

```
public class Main {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 5; i++) {  
            System.out.println("i = " + i);  
        }  
    }  
}
```

Результат выполнения программы:

```
i = 1  
i = 2  
i = 3  
i = 4  
i = 5
```

Задача №6

Условие задачи

Написать программу, в результате работы, которой выводятся на экран первые 10 чисел гармонического ряда (форматировать вывод).

Решение:

```
public class Main {  
    public static void main(String[] args) {  
        int num = 10; // number of values we want in a series  
        System.out.println("Гармонический ряд: ");  
        for (int i = 1; i < num; i++){  
            System.out.println("1 / " + i);  
        }  
    }  
}
```

Результат выполнения программы:

Гармонический ряд:

1 / 1

1 / 2

1 / 3

1 / 4

1 / 5

1 / 6

1 / 7

1 / 8

1 / 9

Задача №7

Условие задачи:

Написать программу, которая с помощью метода класса, вычисляет факториал числа (использовать управляющую конструкцию цикла), проверить работу метода.

Решение:

```
import java.math.BigInteger;
import java.util.Scanner;
class factoriall{
    public static BigInteger factorialmath(int n) {
        BigInteger result = BigInteger.ONE;
        for (int i = 2; i <= n; i++)
            result = result.multiply(BigInteger.valueOf(i));
        return result;
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number = input.nextInt();
        System.out.println("fact: " + factoriall.factorialmath(number));
    }
}
```


Результат выполнения программы:

```
5  
fact: 120
```

```
10000  
fact: 2846259680917054518906413212119868
```

Вывод:

Познакомились со средой разработки IntelliJ IDE, научились решать простейшие задачи на языке программирования JAVA

Практическая работа №2

Цель работы:

Реализовать классы объектов, их сеттеры, геттеры.

Задача №1

Условие:

По диаграмме класса UML, описывающей сущность Автор. Необходимо написать программу, которая состоит из двух классов Author и TestAuthor. Класс Author должен содержать реализацию методов, представленных на диаграмме класса на рисунке 2.4.

Решение:

Класс Author

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

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

    public String getName(){
        return name;
    }

    public String getEmail(){
        return email;
    }

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

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

    public void setGender(char gender){
        this.gender = gender;
    }

    public char getGender() {
        return gender;
    }

    @Override
    public String toString() {
        return "Author{ " +
            "name=" + name + "\" +
            ", email=" + email + "\" +
            ", gender=" + gender +
            '"';
    }
}
```

$\}$
$\}$

Класс TestAuthor

```
import java.util.Objects;
import java.util.Scanner;

public class TestAuthor {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        Author a1 = new Author("", "", '-');
        System.out.println(a1.toString());

        System.out.println("Вы хотите добавить нового пользователя?\nВведите
Y - да, N - нет");
        String check = in.nextLine();

        if (!Objects.equals(check, "Y")) {
            System.out.println("Ладно, тогда до скорой встречи");
        } else {
            System.out.println("Хорошо, давайте введем данные");

            System.out.println("Введите ФИО:");
            String name = in.nextLine();
            a1.setName(name);

            System.out.println("Введите почту:");
            String email = in.nextLine();
            a1.setEmail(email);

            System.out.println("Введите гендер (даже если вы вертолет:");
            char gender = in.next().charAt(0);
            a1.setGender(gender);
        }
        System.out.println("Имя: "+a1.getName());
        System.out.println("Почта: "+a1.getEmail());
        System.out.println("Гендер: "+a1.getGender());
    }
}
```

Результат выполнения программы:

```
Коспот {name=" ", email=" ", gender=" "}  
Вы хотите добавить нового пользователя?  
Введите Y - да, N - нет  
Y  
Хорошо, давайте введем данные)  
Введите ФИО:  
Лищенко Тимофей Викторович  
Введите почту:  
timmmofey@itachi-uchiha.ru  
Введите гендер (даже если вы вертолет):  
ВЕРТОЛЕТИЩЕ  
Имя: Лищенко Тимофей Викторович  
Почта: timmmofey@itachi-uchiha.ru  
Гендер: В
```


Задача №2

Условие задачи:

По UML диаграмме класса, представленной на рис. 2.5 написать программу, которая состоит из двух классов. Один из них Ball должен реализовывать сущность мяч, а другой с названием TestBall тестировать работу созданного класса. Класс Ball должен содержать реализацию методов, представленных на UML. Диаграмма на рисунке описывает сущность Мяч написать программу. Класс Ball моделирует движущийся мяч.

Решение:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // Ball and book classes test  
  
        Ball ball = new Ball(4);  
  
        ball.display_info();  
  
    }  
  
}  
  
class Ball {  
  
    int radius;  
  
    int diameter;  
  
  
    Ball(int radius) {  
  
        this.radius = radius;  
  
        this.diameter = this.radius * 2;  
  
        System.out.println("\nBall object was created");  
  
    }  
  
  
    void display_info() {  
  
        System.out.printf("Radius: %s \tDiameter: %d\n", this.radius, this.diameter);  
  
    }  
  
}
```

Результат выполнения программы:

```
Ball object was created  
Radius: 4   Diameter: 8
```

Задача №5

Условие задачи:

Разработайте и реализуйте класс Dog (Собака), поля класса описывают кличку и возраст собаки. Необходимо выполнить следующие действия: определить конструктор собаки, чтобы принять и инициализировать данные экземпляра., включить стандартные методы (аксессоры) для получения и установки для имени и возраста, включить метод для перевода возраста собаки в “человеческий” возраст (возраст семь раз собаки), включите метод ToString, который возвращает описание экземпляра собаки в виде строки. Создание класса тестера под названием ПитомникСобак, реализует массив собак и основной метод этого класса позволяет добавить в него несколько объектов собаки.

Решение:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        // Dog class test
        Dog dog_1 = new Dog("Dog_1", 5);
        dog_1.set_age(6);
        System.out.println(dog_1);

        Dog dogs[] = {new Dog("Dog_2", 9), new Dog("Dog_3", 10), new
Dog("Dog_4", 11)};

        // Dog kennel test
        Dog_kennel dog_kennel = new Dog_kennel();
        System.out.println(dog_kennel);

        dog_kennel.add_dog(dog_1);
        System.out.println(dog_kennel);

        dog_kennel.add_dogs(dogs);
        System.out.println(dog_kennel);
    }
}

class Dog {
    private String name;
    private int age;

    // Init
    Dog(String name, Integer age) {
        this.name = name;
        this.age = age;
        System.out.println("\nDog object with name: " + this.name + " was
created");
    }

    // Age setter
    public void set_age(int age) {
        if (age > 0 && age < 100) {
            this.age = age;
            System.out.println("Dog age setted as: " + this.age);
        }
    }
}
```

```

    }
}

// Age getter
public int get_age() {
    return this.age;
}

// Name setter
public void set_name(String name) {
    this.name = name;
    System.out.println("Dog name setted as: " + this.name);
}

// Name getter
public String get_name() {
    return this.name;
}

// Dog age as human age getter
public int get_human_age() {
    return this.age * 7;
}

public String toString() {
    return "Dog " + this.name + " with age " + this.age;
}
}

class Dog_kennel {
    private List<Dog> dog_kennel_array = new ArrayList<Dog>();

    {
        System.out.println("\nDog kennel object was created");
    }

    // Add one dog
    public void add_dog(Dog dog) {
        this.dog_kennel_array.add(dog);
        System.out.println(dog + ". This dog added into the dog kennel
successfully");
    }

    // Add many dogs

```

```
public void add_dogs(Dog dogs[]) {  
    this.dog_kennel_array.addAll(new ArrayList<Dog>(Arrays.asList(dogs)));  
    System.out.println("Dogs: " + Arrays.toString(dogs) + " added into the dog  
kennel successfully");  
}  
  
public String toString() {  
    if (!this.dog_kennel_array.isEmpty()) return "Dog kennel: " +  
this.dog_kennel_array;  
    else return "Dog kennel empty!";  
}  
}
```

Результат выполнения программы:

```
Dog object with name: Dog_1 was created
Dog age setted as: 6
Dog Dog_1 with age 6

Dog object with name: Dog_2 was created

Dog object with name: Dog_3 was created

Dog object with name: Dog_4 was created

Dog kennel object was created
Dog kennel empty!
Dog Dog_1 with age 6. This dog added into the dog kennel successfully
Dog kennel: [Dog Dog_1 with age 6]
Dogs: [Dog Dog_2 with age 9, Dog Dog_3 with age 10, Dog Dog_4 with age 11] added into the dog kennel successfully
Dog kennel: [Dog Dog_1 with age 6, Dog Dog_2 with age 9, Dog Dog_3 with age 10, Dog Dog_4 with age 11]
```


Вывод:

Научились работать с классами в Java.

Практическая работа №3

Цель работы:

Данной практической работы - изучить работу с классами Math и Random основные концепции объектно-ориентированного программирования, научиться программировать математические вычисления с использованием этих классов, а также познакомиться с классами оболочками и их использованием в Джава программах и научиться форматировать вывод строк.

Задача №1

Условие задачи:

Создать массив вещественных чисел случайным образом, вывести его на экран, отсортировать его, и снова вывести на экран (использовать два подхода к генерации случайных чисел – метод `random()` класса `Math` и класс `Random`).

Решение:

```
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        Random rand = new Random();
        System.out.println("Введите размер массива: ");
        int size = in.nextInt();
        while (size <= 0) {
            System.out.println("Размер массива не может быть <= 0. Попробуйте снова: ");
            size = in.nextInt();
        }
        double[] arr = new double[size];
        for (int i = 0; i < size; i++) {
            arr[i] = rand.nextDouble();
        }
        System.out.println("Массив созданный классом Random: " + Arrays.toString(arr));
        Arrays.sort(arr);
        System.out.println("Отсортированный массив созданный классом Random: " + Arrays.toString(arr));

        for (int i = 0; i < size; i++) {
            arr[i] = Math.random();
        }
        System.out.println("Массив созданный методом random: " + Arrays.toString(arr));
        Arrays.sort(arr);
        System.out.println("Отсортированный массив созданный методом random: " + Arrays.toString(arr));
    }
}
```

Результат выполнения программы:

```
Введите размер массива:  
5  
Массив созданный классом Random: [0.45724022790574503, 0.18954382301998696, 0.751505765432021, 0.8530402909355606, 0.6842043401877567]  
Отсортированный массив созданный классом Random: [0.18954382301998696, 0.45724022790574503, 0.6842043401877567, 0.751505765432021, 0.8530402909355606]  
Массив созданный методом random: [0.5219682273253652, 0.8982640400278387, 0.5725358471773752, 0.9559478811280876, 0.8752417812842814]  
Отсортированный массив созданный методом random: [0.5219682273253652, 0.5725358471773752, 0.8752417812842814, 0.8982640400278387, 0.9559478811280876]
```

Вывод

Изучили работу с классами Math и Random основные концепции объектно-ориентированного программирования, научиться программировать математические вычисления с использованием этих классов, а также познакомиться с классами оболочки и их использованием в Джава программах и научиться форматировать вывод строк.

Практическая работа №4

Цель работы:

Познакомиться с новым ссылочным типом данных перечислением, научиться разрабатывать перечисления и использовать их в своих программах.

Задача №1

Условие задачи:

Создать перечисление, содержащее названия времен года.

- 1) Создать переменную, содержащую ваше любимое время года и распечатать всю информацию о нем.
- 2) Создать метод, который принимает на вход переменную созданного вами enum типа. Если значение равно Лето, выводим на консоль “Я люблю лето” и так далее. Используем оператор switch.
- 3) Перечисление должно содержать переменную, содержащую среднюю температуру в каждом времени года.
- 4) Добавить конструктор, принимающий на вход среднюю температуру.
- 5) Создать метод getDescription, возвращающий строку “Холодное время года”. Переопределить метод getDescription - для константы Лето метод должен возвращать “Теплое время года”.
- 6) В цикле распечатать все времена года, среднюю температуру и описание времени года.

Решение:

```
enum Season {
    WINTER(0),
    SPRING(10),
    SUMMER(20),
    AUTUMN(5);
    private int avgtemp;

    Season(int avgtemp) {
        this.avgtemp = avgtemp;
    }

    public String getInfo() {
        switch (this) {
            case AUTUMN:
                return "Осень есть осень";
            case WINTER:
                return "Зима есть зима";
            case SUMMER:
                return "Лето есть лето";
            case SPRING:
                return "Весна есть весна";
            default:
                return "ты дурак?";
        }
    }

    public int getAvgtemp() {
        return this.avgtemp;
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println("Да все времена хороши вы че?!");

        for (Season season : Season.values()) {
            System.out.println("Время года: " + season);
            System.out.println("Средняя температура: " + season.getAvgtemp() + "°C");
            System.out.println("Описание: " + season.getInfo());
            System.out.println();
        }
    }
}
```

Результат выполнения программы:

Да все времена хороши вы че?!

Время года: WINTER

Средняя температура: 0°C

Описание: Зима есть зима

Время года: SPRING

Средняя температура: 10°C

Описание: Весна есть весна

Время года: SUMMER

Средняя температура: 20°C

Описание: Лето есть лето

Время года: AUTUMN

Средняя температура: 5°C

Описание: Осень есть осень

Задача №1 Блок 2

Условие задачи:

Необходимо реализовать простейший класс Shape (Фигура). Добавьте метод класса `getType()` (тип фигуры, возвращает строку тип String название фигуры). С помощью наследования создайте дочерние классы Circle, Rectangle и Square. (из предыдущей практической работы). Также реализуйте во всех классах методы `getArea()` (возвращает площадь фигуры), `getPerimeter()` (возвращает периметр фигуры). Переопределите в дочерних классах методы класса родителя `toString()`, `getArea()`, `getPerimeter()` и `getType()`. Создать класс-тестер для вывода информации об объекте и продемонстрировать вызов методов используя родительскую ссылку. Объяснить работу программы.

Решение:

/* Задания на практическую работу № 4.1
Задания на абстрактные классы
Перепишите суперкласс Shape из задания 1, сделайте его
абстрактным и наследуйте подклассы, так как это представлено на UML
диаграмме на рис. 4.1.1 Circle, Rectangle и Square. */

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // Circle class test
        Circle circle = new Circle(0, "black", true);
        System.out.println("Enter radius: ");
        int radius_circle = in.nextInt();
        circle.set_radius(radius_circle);
        System.out.println("Enter color: ");
        String color_circle = in.nextLine();
        circle.set_color(color_circle);
        System.out.println(circle);

        // Rectangle class test
        Rectangle rectangle = new Rectangle();
        System.out.println("Enter length: ");
        int length_rect = in.nextInt();
        rectangle.set_length(length_rect);
        System.out.println("Enter width: ");
        int width_rect = in.nextInt();
        rectangle.set_width(width_rect);
        System.out.println("Rectangle area: " + rectangle.get_area());
        System.out.println(rectangle);

        // Square class test
        Square square = new Square(10, "white", false);
        square.set_side(11);
        System.out.println(square);
    }
}

abstract class Shape {
    private String color;
    private boolean filled;

    Shape() {
        System.out.println("\nShape object was created");
    }

    // Color getter
```

```

String get_color() {
    return this.color;
}

// Color setter
void set_color(String color) {
    this.color = color;
    System.out.println("Shape color setted as: " + this.color);
}

// Filled getter
boolean is_filled() {
    return this.filled;
}

// Filled getter
void set_filled(boolean filled) {
    this.filled = filled;
    System.out.println("Shape filledness setted as: " + this.filled);
}

// Area getter
double get_area() {
    return 0.0;
}

// Perimetr getter
double get_perimeter() {
    return 0.0;
}

public String toString() {
    return "Shape object: is filled: " + this.filled + ", color: " + this.color;
}
}

class Circle extends Shape {
    protected double radius;

    Circle(double radius, String color, boolean filled) {
        super.set_color(color);
        super.set_filled(filled);
        this.radius = radius;
        System.out.println("Circle object was created");
    }

    // Radius getter
    double get_radius() {
        return this.radius;
    }

    // Radius setter

```

```

void set_radius(double radius) {
    if (radius > 0) {
        this.radius = radius;
        System.out.println("Circle radius setted as: " + this.radius);
    } else {
        System.out.println("Circle radius must be > 0");
    }
}

@Override
double get_area() {
    return Math.PI * this.radius * this.radius;
}

@Override
double get_perimeter() {
    return 2 * Math.PI * radius;
}

@Override
public String toString() {
    System.out.println(super.toString());
    return "Shape: circle, radius: " + this.radius;
}
}

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

    Rectangle() {
        super.set_color("blue");
        super.set_filled(false);
        this.width = get_width();
        this.length = get_length();
        System.out.println("Rectangle object was created");
    }

    // Width getter
    double get_width() {
        return this.width;
    }

    // Width setter
    void set_width(double width) {
        if (width > 0) {
            this.width = width;
            System.out.println("Rectangle width setted as: " + this.width);
        } else {
            System.out.println("Rectangle width must be > 0");
        }
    }
}

```

```

    }

    // Length getter
    double get_length() {
        return this.length;
    }

    // Width setter
    void set_length(double length) {
        if (length > 0) {
            this.length = length;
            System.out.println("Rectangle length setted as: " + this.length);
        } else {
            System.out.println("Rectangle length must be > 0");
        }
    }

    @Override
    double get_area() {
        return this.length * this.width;
    }

    @Override
    double get_perimeter() {
        return 2 * (this.length + this.width);
    }

    @Override
    public String toString() {
        System.out.println(super.toString());
        return "Shape: rectangle, length: " + this.length + ", width: " + this.width;
    }
}

class Square extends Rectangle {

    Square(double side, String color, boolean filled) {
        super.set_color("blue");
        super.set_filled(false);
        this.width = side;
        this.length = side;
        System.out.println("[+] Square object was created");
    }

    // Side getter
    double get_side() {
        return this.width;
    }

    // Side setter
    void set_side(double side) {

```

```
    if (side > 0) {
        this.width = side;
        this.length = side;
        System.out.println("Square side setted as: " + this.width);
    } else {
        System.out.println("Square side must be > 0");
    }
}

@Override
public String toString() {
    System.out.println(super.toString());
    return "Shape: square, side: " + this.width;
}
}
```


Результат выполнения программы:

```
Shape object was created
Shape color setted as: black
Shape filledness setted as: true
Circle object was created
Enter radius:
10
Circle radius setted as: 10.0
Enter color:
Shape color setted as:
Shape object: is filled: true, color:
Shape: circle, radius: 10.0

Shape object was created
Shape color setted as: blue
Shape filledness setted as: false
Rectangle object was created
Enter length:
10
Rectangle length setted as: 10.0
Enter width:
5
Rectangle width setted as: 5.0
Rectanle area: 50.0
Shape object: is filled: false, color: blue
Shape: rectangle, length: 10.0, width: 5.0

Shape object was created
Shape color setted as: blue
Shape filledness setted as: false
Rectangle object was created
Shape color setted as: blue
Shape filledness setted as: false
[+] Square object was created
Square side setted as: 11.0
Shape object: is filled: false, color: blue
Shape: rectangle, length: 11.0, width: 11.0
Shape: square, side: 11.0
```

Вывод

Познакомились с новым ссылочным типом данных перечислением, научиться разрабатывать перечисления и использовать их в своих программах, а также с абстрактными классами.

Используемая литература

Конспект лекций по дисциплине «Программирование на языке Джава»,
РТУ МИРЭА, лектор – старший преподаватель Зорина Н.В.