|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |  |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |  |

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕТНАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

Практические РАБОТы

по дисциплине «Программирование на языка 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

[Практическая работа №1 4](#_Toc145510389)

[Цель работы: 4](#_Toc145510390)

[Задача №3 и №4 5](#_Toc145510391)

[Условие задачи: 5](#_Toc145510392)

[Решение: 6](#_Toc145510393)

[Результат выполнения программы: 8](#_Toc145510394)

[Задача №5 9](#_Toc145510395)

[Условие задачи: 9](#_Toc145510396)

[Решение: 10](#_Toc145510397)

[Результат выполнения программы: 11](#_Toc145510398)

[Задача №6 12](#_Toc145510399)

[Условие задачи 12](#_Toc145510400)

[Решение: 13](#_Toc145510401)

[Результат выполнения программы: 14](#_Toc145510402)

[Задача №7 15](#_Toc145510403)

[Условие задачи: 15](#_Toc145510404)

[Решение: 16](#_Toc145510405)

[Результат выполнения программы: 17](#_Toc145510406)

[Вывод: 18](#_Toc145510407)

[Практическая работа №2 19](#_Toc145510408)

[Цель работы: 19](#_Toc145510409)

[Задача №1 20](#_Toc145510410)

[Условие: 20](#_Toc145510411)

[Решение: 21](#_Toc145510412)

[Класс Author 21](#_Toc145510413)

[Класс TestAuthor 23](#_Toc145510414)

[Результат выполнения программы: 24](#_Toc145510415)

[Задача №2 25](#_Toc145510416)

[Условие задачи: 25](#_Toc145510417)

[Решение: 26](#_Toc145510418)

[Результат выполнения программы: 27](#_Toc145510419)

[Задача №5 28](#_Toc145510420)

[Условие задачи: 28](#_Toc145510421)

[Решение: 29](#_Toc145510422)

[Результат выполнения программы: 32](#_Toc145510423)

[Вывод: 33](#_Toc145510424)

[Практическая работа №3 34](#_Toc145510425)

[Цель работы: 34](#_Toc145510426)

[Задача №1 35](#_Toc145510427)

[Условие задачи: 35](#_Toc145510428)

[Решение: 36](#_Toc145510429)

[Результат выполнения программы: 37](#_Toc145510430)

[Вывод 38](#_Toc145510431)

[Практическая работа №4 39](#_Toc145510432)

[Цель работы: 39](#_Toc145510433)

[Задача №1 40](#_Toc145510434)

[Условие задачи: 40](#_Toc145510435)

[Решение: 41](#_Toc145510436)

[Результат выполнения программы: 42](#_Toc145510437)

[Задача №1 Блок 2 43](#_Toc145510438)

[Условие задачи: 43](#_Toc145510439)

[Решение: 44](#_Toc145510440)

[Результат выполнения программы: 49](#_Toc145510441)

[Вывод 50](#_Toc145510442)

[Используемая литература 51](#_Toc145510443)

# **Практическая работа №1**

## **Цель работы:**

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

## **Задача №3 и №4**

### **Условие задачи:**

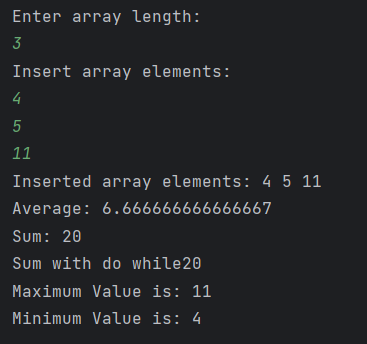
Написать программу, в результате которой массив чисел создается с помощью инициализации (как в Си) вводится и считается в цикле сумма элементов целочисленного массива, а также среднее арифметическое его элементов результат выводится на экран. Использовать цикл for.

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:**

****

## **Задача №5**

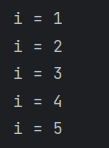
### **Условие задачи:**

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:**

****

## **Задача №6**

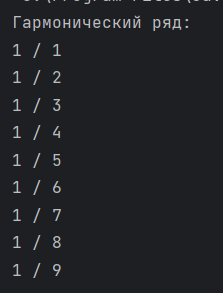
### **Условие задачи**

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:**

****

## **Задача №7**

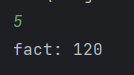
### **Условие задачи:**

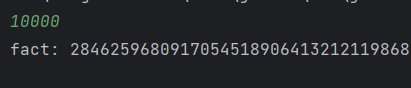
Написать программу, которая с помощью метода класса, вычисляет факториал числа (использовать управляющую конструкцию цикла), проверить работу метода.

### **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:**

****

****

## **Вывод:**

Познакомились со средой разработки Intelij IDE, научиились 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:**

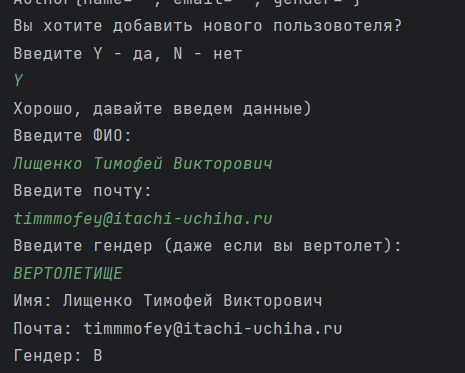
### **Класс Author**

|  |
| --- |
| 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 пользовотеля?\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:**

****

## **Задача №2**

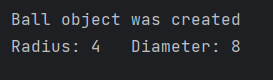
### **Условие задачи:**

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:**

****

## **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:**

****

## **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**

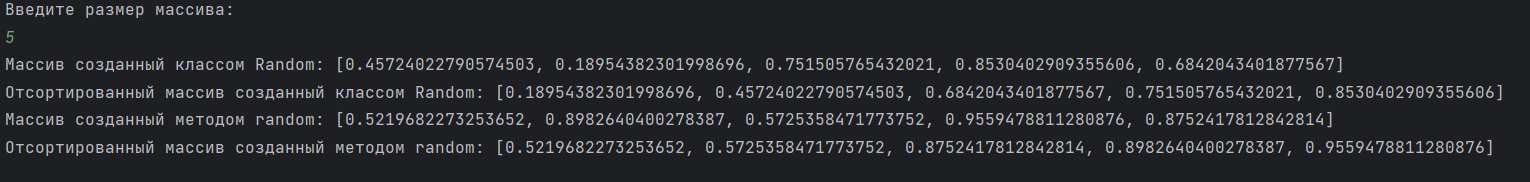
### **Условие задачи:**

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:**



## **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 оболочками 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.

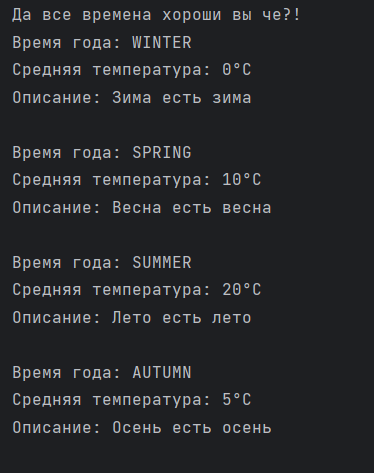
5) Создать метод getDescription, возвращающий строку “Холодное время года”. Переопределить метод getDescription - для константы Лето метод должен возвращать “Теплое время года”.

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:**



## **Задача №1 Блок 2**

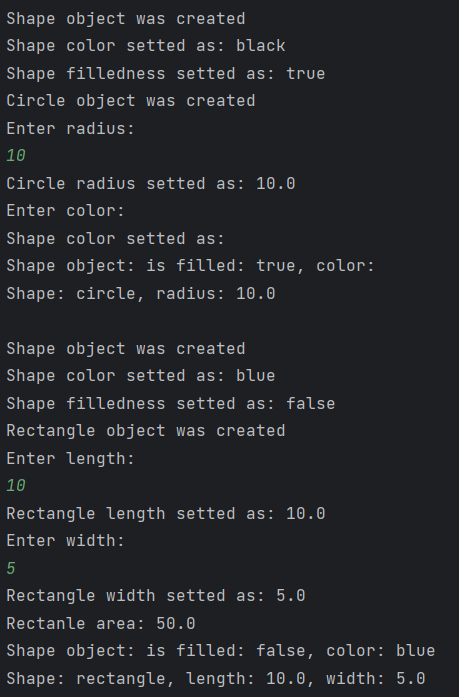
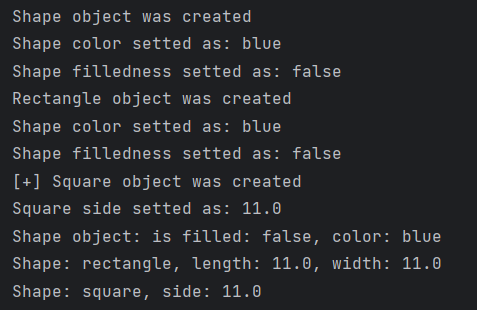
### **Условие задачи:**

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() взвращает периметр фигуры). Переопределите в дочерних класс методы класса родителя toString(), getArea(), getPerimeter() и getType(). Создать класс-тестер для вывода информации об объекте и продемонстирировать вызов методов использую 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("Rectanle 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:**

## **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.