|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  "**МИРЭА** - **Российский** **технологический университет**"  **РТУ** **МИРЭА** | |
| Институт информационных технологий (ИИТ)  Кафедра информационного и прикладного программного обеспечения (ИППО) | |
|  | |

|  |  |
| --- | --- |
| **Отчет по лабораторным работам** | |
| **По дисциплине «Объектно-ориентированное программирование»** | |
| Выполнил студент группы ИКБО-13-17 | Хитров Н.О. |
| |  |  |  | | --- | --- | --- | | Отчет выполнен | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  | |  |  |  | | «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |   Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Хлебникова В.Л. |

Москва 2018

# Таблица успеваемости

**Лабораторные занятия**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №/р. | Страница в отчете. | Студент выполнил:  (подпись) | Преподаватель принял:  (подпись) | Балл  (максимально возможный) | Балл (Фактический) |
| 1. | *4-9* |  |  | 2 |  |
| 2. | 10-13 |  |  | 2 |  |
| 3. | 14-17 |  |  | 4 |  |
| 4. | 18-20 |  |  | 6 |  |
| 5. | 21-23 |  |  | 6 |  |
| 6. | 24-29 |  |  | 8 |  |
| 7. | 30-32 |  |  | 8 |  |
| 8. | 33-35 |  |  | 8 |  |
|  |  |  | Итоговый балл: | 44 |  |

Оглавление

[Таблица успеваемости 2](#_Toc531626919)

[Лабораторная работа №1 4](#_Toc531626920)

[Лабораторная работа №2 8](#_Toc531626921)

[Лабораторная работа №3 10](#_Toc531626922)

[Лабораторная работа №4 13](#_Toc531626923)

[Лабораторная работа №6 18](#_Toc531626924)

[Лабораторная работа №7 20](#_Toc531626925)

[Лабораторная работа №8 21](#_Toc531626926)

# Лабораторная работа №1

**Цель лабораторной работы**: получение практических навыков разработки программ, изучение синтаксиса языка Java, его конструкций, а также изучение стандартного ввода и вывода данных.

**Задания:**

1. Вывести на экран сумму чисел массива с помощью циклов for, while, do while.

2. Вывести на экран аргументы командной строки в цикле for.

3. Вывести на экран первые 10 чисел гармонического ряда.

4. Сгенерировать массив целых чисел случайным образом, вывести его на экран, отсортировать его, и снова вывести на экран.

5. Создать метод, вычисляющую факториал числа с помощью цикла, проверить работу метода.

**Ход выполнения работы:**

**Выполнение задания 1.1:**

Во время выполнения программы пользователь вводит массив чисел:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab01.task01;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("elements count: ");  int size = Math.abs(scanner.nextInt());  int[] array = new int[size];  for (int i = 0; i < array.length; i++) {  array[i] = scanner.nextInt();  }  int sum = 0;  for (int anArray : array) {  sum += anArray;  }  System.out.println("for: " + sum);  int i = sum = 0;  while (i < array.length) {sum += array[i++];}  System.out.println("while: " + sum);  i = sum = 0;  do {  sum += array[i++];  } while (i < array.length);  System.out.println("do while: " + sum);  }  } |

Вывод результатов выполнения данной программы:

|  |
| --- |
| 1 3 6 0 6 8 9 6 3 9  for: 51  while: 51  do while: 51 |

**Выполнение задания 1.2:**

Данная функция представляет собой цикл for, который выводит аргументы командной строки:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab01.task02;  public class Main {  public static void main(String[] args) {  if (args.length > 0) {  for (String arg : args) {  System.out.print(arg + " ");  }  System.out.println();  } else {  System.out.println("no arguments");  }  }  } |

Вывод результата работы данной функции будет следующим:

|  |
| --- |
| nick@Aspire-V3-772G:~/Projects/Labs$ java Lab arg01 arg002 arg003  arg01  arg002  arg003  nick@Aspire-V3-772G:~/Projects/Labs$ |

**Выполнение задания 1.3:**

Для выполнения задания используется цикл for со счетчиком и форматирование строки вывода при помощи функции print:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab01.task03;  public class Main {  public static void main(String[] args) {  System.out.print("numbers: ");  for (int i = 1; i <= 10; i++)  System.out.printf("%.5g ", (1. / i));  System.out.println();  }  } |

Вывод работы программы:

|  |
| --- |
| 0.5  0.6666667  0.75  0.8  0.8333333  0.85714287  0.875  0.8888889  0.9 |

Данный вывод представляет собой гармонический ряд.

**Выполнение задания 1.4:**

Данное задание выполнено с помощью пользовательского класса Sort, который содержит метод для автоматической генерации и сортировки массива, заполненного случайными значениями.

|  |
| --- |
| package ru.mirea.khitrov.labs.lab01.task04;  import java.util.Arrays;  import java.util.Random;  public class Main {  public static void main(String[] args) {  Random random = new Random();  int length = random.nextInt(20) + 1;  int[] a = new int[length];  System.out.print("Default array: ");  for (int i = 0; i < a.length; i++) {  a[i] = random.nextInt() % 1000;  System.out.print(a[i] + " ");  }  Arrays.sort(a);  System.out.print("Sorted array: ");  for (int i: a) {  System.out.print(i + " ");  }  System.out.println();  }  } |

В результате вызова данной функции будет предоставлен следующий вывод:

|  |
| --- |
| Default array: 201 645 121 333 489  Sorted array: 121 201 333 489 645 |

**Выполнение задания 1.5:**

Метод, описанный в задании, реализован с помощью отдельного метода классе Main:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab01.task05;  public class Main {  private static int calculateFactorial(int number) {  if (number <= 0) {  return 1;  } else {  int result = 1;  for (int i = 1; i <= number; i++) {result \*= i;}  return result;  }  }  public static void main(String[] args) {  System.out.println("factorial: " + calculateFactorial(5));  System.out.println("factorial: " + calculateFactorial(10));  }  } |

Вывод работы данного метода выглядит следующим образом:

|  |
| --- |
| factorial: 120  factorial: 3628800 |

**Вывод:**

В результате выполнения данной лабораторной работы были получены практические навыки работы с языком Java, изучены основы синтаксиса и конструкций данного языка.

# Лабораторная работа №2

**Цель лабораторной работы**: Изучить основные концепции объектно-ориентированного программирования, изучить понятие класса и научиться создавать классы.

**Задание**: Создать класс, описывающий модель окружности (Circle). В классе должны быть описаны нужные свойства окружности и методы для получения, изменения этих свойств. Протестировать работу класса в классе CircleTest, содержащим метод статический main(String[] args).

**Ход выполнения работы:**

Для выполнения данной лабораторной работы было создано несколько классов, со своими методами конструкторами.

Класс Circle, содержащий переменные вещественного типа для хранения координат окружности и радиуса:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab02.task01;  public class Circle {  private double x, y, r;  public Circle() {  this(0, 0, 0);  }  public Circle(double x, double y, double r) {  this.x = x;  this.y = y;  this.r = r;  }  public double getX() {  return x;  }  public void setX(double x) {  this.x = x;  }  public double getY() {  return y;  }  public void setY(double y) {  this.y = y;  }  public double getR() {  return r;  }  public void setR(double r) {  this.r = r;  }  @Override  public String toString() {  return "Circle {" +  "x=" + x +  ", y=" + y +  ", r=" + r +  '}';  }  } |

Класс CircleTest для проверки корректной работы класса Circle:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab02.task01;  public class CircleTest {  public static void main(String[] args) {  System.out.println(new Circle());  System.out.println(new Circle(1, 2, 3));  Circle c = new Circle(3, 4, 5);  System.out.printf("Circle: x=%f y=%f r=%f\n", c.getX(), c.getY(), c.getR());  c.setX(30.15);  c.setR(11);  System.out.println(c);  }  } |

Вывод работы программы выглядит следующим образом:

|  |
| --- |
| Circle {x=0, y=0, r=0}  Circle {x=1, y=2, r=3}  Circle: x=3 y=4 r=5  Circle {x=30.15, y=4, r=11} |

**Вывод:** в ходе выполнения данной лабораторной работы были изучены основные концепции объектно-ориентированного программирования, понятие класса и был получен навык создания классов.

# Лабораторная работа №3

**Цель лабораторной работы**: Изучить понятие наследования, и научиться реализовывать наследование в Java.

**Задание**: Создать абстрактный класс, описывающий посуду (Dish). С помощью наследования реализовать различные виды посуды, имеющие свои свойства и методы. Протестировать работу классов.

**Ход выполнения работы:**

В ходе выполнения данной лабораторной было создано три класса, не включая класс Main.

Абстрактный класс Dash, c первоначальным комплектом методов:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab03.task01;  public abstract class Dash {  protected double cost;  public Dash() {  cost = 0;  }  public Dash(double cost) {  this.cost = cost;  }  public double getCost() {  return cost;  }  public void setCost(double cost) {  this.cost = cost;  }  public abstract void PrintInformation();  } |

Класс Dish, наследованный от абстрактного класса Dash:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab03.task01;  public class Dish extends Dash {  protected double volume;  public Dish() {  super();  volume = 0;  }  public Dish(double cost, double volume) {  super(cost);  this.volume = volume;  }  public double getVolume() {  return volume;  }  public void setVolume(double volume) {  this.volume = volume;  }  @Override  public void PrintInformation() {  System.out.println("Dish: volume="+volume+", cost="+cost);  }  } |

Класс Pan, наследованный от абстрактного класса Dash:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab03.task01;  public class Pan extends Dash {  protected double volume;  Pan() {  super();  volume = 0;  }  Pan(double cost, double volume) {  super(cost);  this.volume = volume;  }  public double getVolume() {  return volume;  }  public void setVolume(double volume) {  this.volume = volume;  }  @Override public void PrintInformation() {  System.out.println("Pan: volume="+volume+", cost="+cost);  }  } |

Чтобы вызвать процесс взаимодействия данных классов, используется код Main класса лабораторной работы:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab03.task01;  public class Main {  public static void main(String[] args) {  Dash dash = new Dish();  dash.PrintInformation();  dash = new Dish(16, 3);  dash.PrintInformation();  dash = new Pan();  dash.PrintInformation();  dash = new Pan(18, 8);  dash.PrintInformation();  }  } |

Вывод программы выглядит следующим образом:

|  |
| --- |
| Dish: volume=0, cost=0  Dish: volume=3, cost=16  Pan: volume=0, cost=0  Pan: volume=8, cost=18 |

**Вывод:** В ходе выполнения данной лабораторной работы были изучено понятие наследования, а также получен навык реализовывать наследование в Java.

# Лабораторная работа №4

**Цель данной лабораторной работы**: Изучить понятие интерфейса, научиться создавать интерфейсы в Java и применять их в программах.

**Задание**: Создать интерфейс Nameable, с методом getName(), возвращающим имя объекта, реализующего интерфейс. Проверить работу для различных объектов (например, можно создать классы, описывающие разные сущности, которые могут иметь имя: планеты, машины, животные и т. д.).

**Ход выполнения работы:**

Для того, чтобы в полной мере выполнить задание лабораторной работы, был создан интерфейс Nameable, а также три сопутствующих класса, имплементирующих приведенный интерфейс:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab04.task01;  public interface Nameable {  String getName();  } |

Три класса, который используют приведенный интерфейс: Car, Planet, Animal.

Класс Car:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab04.task01;  public class Car implements Nameable {  private String carType;  public Car(String carType) {  this.carType= carType;  }  public String getCarType() {  return carType;  }  public void setCarType(String carType) {  this.carType = carType;  }  @Override  public String getName() {  return carType;  }  } |

Класс Planet:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab04.task01;  public class Planet implements Nameable {  private String planetName;  public Planet(String planetName) {  this.planetName = planetName;  }  public String getPlanetName() {  return planetName;  }  public void setPlanetName(String planetName) {  this.planetName = planetName;  }  @Override  public String getName() {  return planetName;  }  } |

Класс Animal:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab04.task01;  public class Animal implements Nameable {  private String animalName;  public Animal(String animalName) {  this.animalName = animalName;  }  public String getAnimalName() {  return animalName;  }  public void setAnimalName(String animalName) {  this.animalName = animalName;  }  @Override  public String getName() {  return animalName;  }  } |

Чтобы вызвать процесс взаимодействия данных классов, используется код основного класса лабораторной работы.

|  |
| --- |
| package ru.mirea.khitrov.labs.lab04.task01;  public class Main {  public static void main(String[] args) {  Nameable nameable = new Planet("Earth");  System.out.println(nameable.getName());  nameable = new Car("Subaru");  System.out.println(nameable.getName());  nameable = new Animal("Barsic");  System.out.println(nameable.getName());  }  } |

Вывод программы во время выполнения:

|  |
| --- |
| Earth  Subaru  Barsic |

**Вывод:** в ходе выполнения данной лабораторной работы были изучены интерфейсы в языке Java, были получены навыки их создания и применять в программах.

**Лабораторная работа №5**

**Цель данной лабораторной работы:** Научиться создавать графический интерфейс пользователя, освоить на практике работу с различными объектами для создания ГИП, менеджерами размещения компонентов.

**Задание:** Создать окно, отобразить в нем картинку, путь к которой указан в аргументах командной строки.

**Ход выполнения работы:**

В контексте выполнения работы был создан класс окна приложения:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab05.task02;  import javax.imageio.ImageIO;  import javax.swing.\*;  import java.awt.\*;  import java.awt.image.BufferedImage;  import java.io.File;  import java.io.IOException;  public class Main {  public static void main(String[] args) {  if (args.length != 1) {  System.err.println("usage: imageviewer [path-to-image]");  System.exit(1);  }  try {  BufferedImage image = ImageIO.read(new File(args[0]));  ImageIcon icon = new ImageIcon(image);  JFrame frame = new JFrame("Image Viewer");  frame.setLayout(new FlowLayout());  frame.setSize(image.getWidth(), image.getHeight());  JLabel label = new JLabel();  label.setIcon(icon);  frame.add(label);  frame.setVisible(true);  frame.setResizable(false);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  } catch (IOException e) {  System.err.println(e.getMessage());  System.exit(2);  }  }  } |

В качестве аргумента командной строки программе передается путь до файла example-image.jpg, который содержит графическое изображение. Выводом работы этой программы является окно приложения с изображением.



Рисунок 1 – Окно приложения

**Вывод:** в ходе выполнения данной лабораторной работы были получены навыки создания графического интерфейса приложения.

# Лабораторная работа №6

**Цель данной лабораторной работы:** Научиться обрабатывать различные события для разных компонентов (кнопок, меню и т. д.).

**Задание:** Реализовать программу на Java с JTextArea и двумя меню: Цвет: который имеет возможность выбора из три возможных: синий, красный и черный. Шрифт: три вида: “Times New Roman”, “MS Sans Serif”, “Courier New”. Вы должны написать программу, которая с помощью меню, может изменять шрифт и цвет текста, написанного в JTextArea.

**Ход выполнения работы:**

Для выполнения данной лабораторной работы был создан класс, который создаёт окно, содержащее несколько зон, реагирующих на поднесение курсора.

|  |
| --- |
| package ru.mirea.khitrov.labs.lab06.task03;  import javax.swing.\*;  import java.awt.\*;  public class Main {  private static final int fontSize = 15;  public static void main(String[] args) {  JTextArea textArea = new JTextArea(10, 25);  JFrame frame = new JFrame("Text Changer");  JMenuBar menuBar = new JMenuBar();  JMenu colorMenu = new JMenu("Color");  JMenu fontMenu = new JMenu("Font");  colorMenu.add(createColorChangingItem(textArea, "Blue", Color.BLUE));  colorMenu.add(createColorChangingItem(textArea, "Red", Color.RED));  colorMenu.add(createColorChangingItem(textArea, "Black", Color.BLACK));  fontMenu.add(createFontChangingItem(textArea, "Times New Roman"));  fontMenu.add(createFontChangingItem(textArea, "MS Sans Serif"));  fontMenu.add(createFontChangingItem(textArea, "Courier New"));  menuBar.add(colorMenu);  menuBar.add(fontMenu);  frame.add(textArea);  frame.setSize(500, 300);  frame.setResizable(false);  frame.setJMenuBar(menuBar);  frame.setVisible(true);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  }  private static JMenuItem createFontChangingItem(JTextArea area, String fontName) {  JMenuItem menuItem = new JMenuItem(fontName);  menuItem.addActionListener(e -> area.setFont(new Font(fontName, Font.PLAIN, fontSize)));  return menuItem;  }  private static JMenuItem createColorChangingItem(JTextArea area, String cname, Color c) {  JMenuItem menuItem = new JMenuItem(cname);  menuItem.addActionListener(e -> area.setForeground(c));  return menuItem;  }  } |

Результат работы программы выглядит следующим образом:

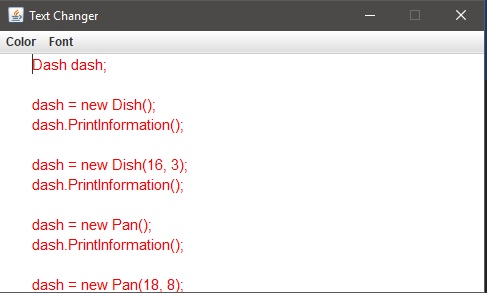


Рисунок 1 – Окно приложения.

**Вывод:** в ходе выполнения данной лабораторной работы были изучены способы обработки графических элементов на основе пользовательского ввода.

# Лабораторная работа №7

**Цель данной лабораторной работы**: Изучение работы с различными коллекциями в Java.

**Задание**: Протестировать работу коллекции LinkedList.

**Ход выполнения работы:**

Вся работа происходит в основном файле программы. Программа принимает на вход значения, пока не будет введена пустая строка:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab07.task02;  import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStreamReader;  import java.util.LinkedList;  public class Main {  public static void main(String[] args) throws IOException {  BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  LinkedList<String> list = new LinkedList<>();  String s;  System.out.println("Enter empty line to stop adding items:");  while (!(s = reader.readLine()).isEmpty()) {  list.add(s);  }  System.out.println(list);  System.out.println("First: " + list.getFirst());  System.out.println("Last: " + list.getLast());  list.clear();  System.out.println("List after clearing: " + list.toString());  }  } |

Работа программы протекает следующим образом:

|  |
| --- |
| Enter empty line to stop adding items:  1  2  3  [1, 2, 3]  First: 1  Last: 3  List after clearing: [] |

**Вывод:** в ходе выполнения данной лабораторной работы были получены навыки работы с коллекциями в Java.

# Лабораторная работа №8

**Цель данной лабораторной работы**: Освоить на практике работу с файлами на языке Java. Получить практические навыки по чтению и записи данных в файл.

**Задание**: Реализовать вывод информации из файла на экран

**Ход выполнения работы:**

Программа создает объект класса BufferedReader на основе класса FileReader, который получает имя файла через аргумент командной строки, после чего содержимое файла построчно выводится:

|  |
| --- |
| package ru.mirea.khitrov.labs.lab08.task02;  import java.io.\*;  public class Main {  public static void main(String[] args) {  if (args.length != 1) {  System.err.println("usage: fileprinter filename");  System.exit(1);  }  try (BufferedReader reader = new BufferedReader(new FileReader(args[0]))) {  String line;  while ((line = reader.readLine()) != null) {  System.out.println(line);  }  } catch (IOException e) {  System.err.println(e.getMessage());  System.exit(2);  }  }  } |

Работа программы протекает следующим образом:

|  |
| --- |
| nick@Aspire-V3-772G:~/Projects/Labs$ java Lab input\_file.txt  строка1  строка2  строка3  nick@Aspire-V3-772G:~/Projects/Labs$ |

**Вывод:** в ходе выполнения данной лабораторной работы были получены навыки работы с файлами на языке Java.

**Библиографический список**

1. Роберт Лафоре, Структуры данных и алгоритмы в Java
2. [В. Белов](http://www.ozon.ru/person/4530084/), [В. Чистякова](http://www.ozon.ru/person/4530088/) Алгоритмы и структуры данных. Учебник[Инфра-м](http://www.chaconne.ru/izdat.php?id=14253) 2016  240 стр. ISBN: 5906818256
3. Богачев К.Ю. Основы параллельного программирования– М.: Бином. Лаборатория знаний, 2003. -.342 с.
4. Зыль С.Н. Операционная система реального времени QNX Neutrino: от теории кпрактике – Изд. 2-е - СПб.: БХВ-Петербург, 2004. – 192 с.