МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

ОБОСОБЛЕННОЕ ПОДРАЗДЕЛЕНИЕ

«КОЛЛЕДЖ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ»

Отчет по лабораторной работе № 3

“Средства для ввода данных”

Выполнил учащийся гр. ТО-011

Ясюкевич Полина

Проверили

Янович Н.И.

Рогалевич П.И.

Минск, 2022

**Цель:** научить использовать средства ввода данных языка Java Scaner и DialogBox, а также организовывать файловый ввод/вывод данных.

**Практические задания и методические указания**

1. Изучить теоретические сведения по теме «Ввод данных с клавиатуры с

использованием класса Scanner», выполнить приведенные примеры.

2. Модернизировать код индивидуального задания 1 из лабораторной

работы №1, добавив в них форматированный ввод данных с клавиатуры.

3. Изучить теоретические сведения по теме «Ввод данных из текстового

файла через класс Scanner» и «Форматированный вывод данных в текстовый

файл», выполнить приведенные примеры.

4. Выполнить индивидуальные задания согласно варианту.

5. Модернизировать код индивидуального задания 1 из лабораторной

работы №1, добавив в них ввод данных через диалоговое окно.

6. Изучить теоретические сведения по теме «Ввод данных через аргументы

метода main()», выполнить приведенные примеры.

7. Модернизировать программный код, разработанный по индивидуальному

заданию: имя файла передавать в качестве аргумента метода main().

8. Создать отчет, включив в него описание работы по пунктам 1-7, ответы

на контрольные вопросы.

**Задание 1, 3, 6**

**Пример 1**

import java.util.Scanner;

public class Primer1 {

public static void main(String[] args) {

int num1;

double num2;

String str;

Scanner in = new Scanner(System.in);

System.out.print("Enter an integer: ");

num1 = in.nextInt();

System.out.print("Enter a floating point: ");

num2 = in.nextDouble();

System.out.print("Enter a string: ");

str = in.next();

System.out.printf("%s, Sum of %d & %.2f is %.2f%n", str, num1, num2, num1 + num2);

}

}

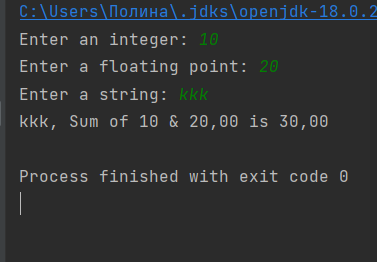


Рисунок 1 – результат выполнения программы

**Пример 2**

import java.util.Scanner;

public class Primer2 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Enter a string (with space): ");

String str = in.nextLine(); //чтение строки с пробелами

System.out.printf("%s%n", str);

in.close();

}}

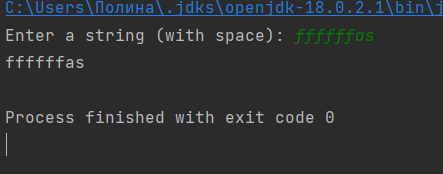
****

Рисунок 2 – результат выполнения программы

**Пример 3**

import java.io.FileNotFoundException;

import java.io.File;

import java.util.Scanner;

public class Primer3 {

public static void main(String[] args) throws FileNotFoundException {

// Construct a Scanner to scan a text file

Scanner in = new Scanner(new File("in.txt"));

int anInt = in.nextInt(); // next String

System.out.println(anInt);

double aDouble = in.nextDouble();// next double String str = in.next(); // next int

System.out.println(aDouble);

String str = in.next(); // next int

System.out.println(str);

String line = in.nextLine(); // entire line

System.out.println(line);

}

}

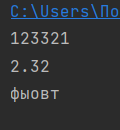
****

Рисунок 3 – Пример ввода данных из файла

**Пример 4**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class Primer4 {

public static void main(String[] args) throws FileNotFoundException {

class ScannerFromFile {

public static void main(String[] args) throws FileNotFoundException {

Scanner in = new Scanner(new File("ЛР3.in")); }

}

}

}

**Пример 5**

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

class Primer5 {

public static void main(String[] args) {

try {

Scanner in = new Scanner(new File("test.in"));}

catch (FileNotFoundException ex) {// если возникло исключение

ex.printStackTrace();} //распечатать трассировку стека

}

}

**Пример 6**

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

class Primer6 {

public static void main(String[] args)

throws FileNotFoundException

{ Scanner in = new Scanner(new File("test.in"));

}

}

**Пример 7**

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

public class Primer7 {

public static void main(String[] args) {

try { // основной алгоритм

System.out.println("Start of the main logic");

System.out.println("Try opening a file ...");

Scanner in = new Scanner(new File("test.in"));

System.out.println("File Found, processing the file ...");

System.out.println("End of the main logic");

}

catch (FileNotFoundException ex){// обработка ошибок

System.out.println("File Not Found caught ..."); }

finally {// всегда выполняется независимо от статуса исключения

System.out.println("finally-block runs regardless of the state of exception");

}

// после try-catch-finally

System.out.println("After try-catch-finally, life goes on...");

}

}

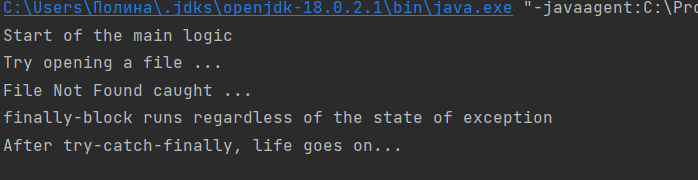
****

Рисунок 7 – результат выполнения программы

**Пример 8**

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.NoSuchElementException;

public class Primer8 {

public static void main(String[] args) { int num1;

double num2;

String name;

try {

Scanner in = new Scanner(new File("ЛР3.txt"));

while(in.hasNext()){//пока есть символы в потоке

num1=in.nextInt();

num2=in.nextDouble();

name=in.next();

System.out.println(num1+" "+num2+" "+name+" ");

}

in.close();

} //обработка возможных ошибок

catch (FileNotFoundException ex) { // если файл не найден

ex.printStackTrace(); //печать трассировки стека

}

catch (NoSuchElementException ex) {

/\*нет значения для считывания или значение не соответствуют ожидаемому формату (InputMismatchException является подклассом NoSuchElementException) \*/

System.out.println("Input File is incorrect ...");

}

}

}

**Пример 9**

import java.io.File;

import java.util.Formatter;

import java.io.FileNotFoundException;

public class Primer9 {

public static void main(String[] args) {

try {

/\*используем класс Formatter для записи форматированного вывода в текстовый файл\*/

Formatter out = new Formatter(new File("out.txt"));

//записываем в файл метода format() (похож на printf())

int n1 = 1234;

double n2 = 55.66;

String name = "Tanya";

out.format("Hi %s,%n", name);

out.format("The sum of %d and %.2f is %.2f%n", n1, n2, n1 + n2); out.close();//закрыть файл

System.out.println("Done"); //вывод в консоль

}

catch (FileNotFoundException ex) { //ловим исключение

ex.printStackTrace(); //печать трассировки стека

}

}

}

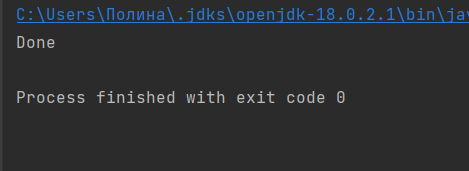
****

Рисунок 9 – результат выполнения программы

**Пример 10**

import javax.swing.JOptionPane; // импорт класса JOptionPane

public class Primer10 {

public static void main(String[] args) {

String radiusStr;

double radius, area;

//чтение данных из диалогового окна ввода в переменную radiusStr

radiusStr = JOptionPane.showInputDialog("Введите радиус круга");

radius = Double.parseDouble(radiusStr);//преобразование String в double

area = radius\*radius\*Math.PI;

System.out.println("Площадь = " + area); }}

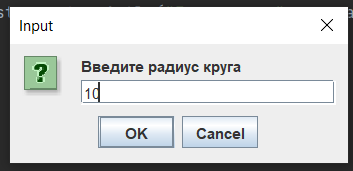
****

Рисунок 10.1– результат выполнения программы

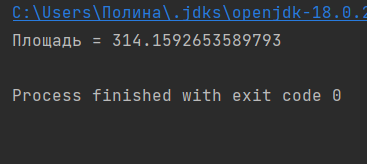
****

Рисунок 10.2 – результат выполнения программы

**Пример 11**

public class Primer11 {

public static void main(String[] args) {

class MainParamTest{

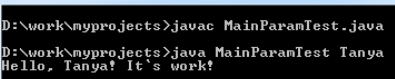
public static void main (String[] args){

System.out.println("Hello, " + args[0]+ "! It`s work! "); }

}

}

}



**Задание 2**

Модернизировать код индивидуального задания 1 из лабораторной

работы №1, добавив в них форматированный ввод данных с клавиатуры.

import java.util.Scanner;

class Zadanie2

{

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Введите цену в рублях: ");

int rub = in.nextInt();

double usd = (rub/2.53);

System.out.println("USD = "+usd);

double euro = (rub/2.51);

System.out.println("EURO = "+usd);

}

}

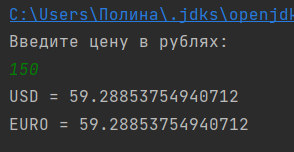


Рисунок 12 – Пример выполнения программы задания 2

**Задание 4**

Создать файл данных, компонентами которого натуральные числа.

Определить количество компонент, которые делятся на 3 без остатка.

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

public class Zadanie4 {

public static void main(String[]args) throws FileNotFoundException {

int b = 0;

System.out.println("Данные которые храняться в файле: ");

Scanner in = new Scanner(new File("Zad2.txt"));

while (in.hasNextInt()){

int a = in.nextInt();

if(a%3==0){

b = b+1;

}

System.out.println(a);

}

System.out.println("Количество без остатка: "+b);

}

}

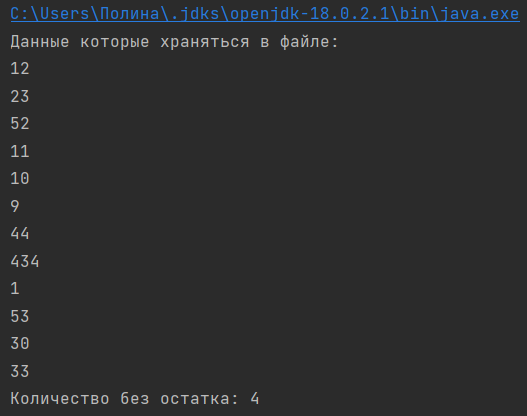
****

Рисунок 13 – Пример выполнения программы задания 4

**Задание 5**

Модернизировать код индивидуального задания 1 из лабораторной

работы №1, добавив в них ввод данных через диалоговое окно.

import javax.swing.JOptionPane; // импорт класса JOptionPane

public class Zadanie5 {

public static void main(String[] args) {

String rub;

double usdt, euro, rubl;

rub = JOptionPane.showInputDialog("Введите цену в рублях");

rubl = Double.parseDouble(rub);//п\реобразование String в double

usdt =rubl \* 2.53;

euro = rubl / 2.51;

System.out.println("Доллар = " + usdt);

System.out.println("Евро = " + euro);

}

}

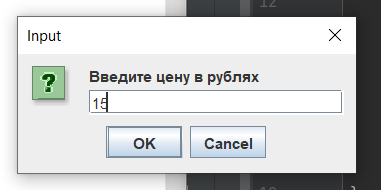


Рисунок 14.1 – Пример выполнения программы задания 5

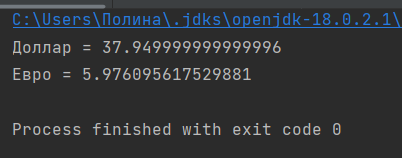


Рисунок 14.2 – Пример выполнения программы задания 5

**Задание 7**

Модернизировать программный код, разработанный по индивидуальному

заданию: имя файла передавать в качестве аргумента метода main().

import java.util.Scanner;

import java.io.File;

import java.io.FileNotFoundException;

public class laba3 {

public static void main(String[]args) throws FileNotFoundException {

int b = 0;

System.out.println("Данные которые храняться в файле: ");

Scanner in = new Scanner(new File(""+args[0]));

while (in.hasNextInt()){

int a = in.nextInt();

if(a%3==0){

b = b+1;

}

System.out.println(a);

}

System.out.println("Количество без остатка: "+b);

}

}

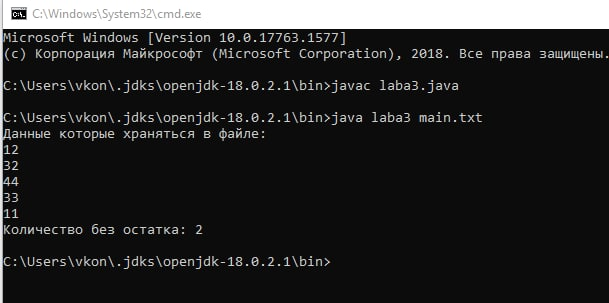


Рисунок 15 – Пример выполнения программы задания 7

**Контрольные вопросы**

1. Какие стандартные потоки поддерживаются в языке Java?

InPutStream – поток ввода используется для считывания данных с источника.

OutPutStream – поток вывода используется для записи данных по месту назначения.

1. С помощью каких средств осуществляется ввод данных с клавиатуры в программах на Java.

import java.util.Scanner;

Scanner in = new Scanner(System.in);

in.nextInt(); / in.nextDouble(); / in.next();

1. **Изобразите в виде блок-схемы алгоритм ввода данных.**

Подключение пакета java.util.Scanner

Написание конструкции

Scanner in = new Scanner(System.in)

Ввод данных через консоль

Считывание с помощью конструкции in.nextInt() данных в переменную

1. **С помощью каких средств осуществляется ввод/вывод данных из текстовых файлов в программах на Java.**

InPutStream – поток ввода используется для считывания данных с источника.

OutPutStream – поток вывода используется для записи данных по месту назначения.

1. **Что такое исключения в Java? Как реализуется их обработка?**

В мире программирования возникновение ошибок и непредвиденных ситуаций при выполнении программы называют **исключением**. В программе исключения могут возникать в результате неправильных действий пользователя, отсутствии необходимого ресурса на диске, или потери соединения с сервером по сети.

* try – определяет блок кода, в котором может произойти исключение;
* catch – определяет блок кода, в котором происходит обработка исключения;
* finally – определяет блок кода, который является необязательным, но при его наличии выполняется в любом случае независимо от результатов выполнения блока try.
* throw – используется для возбуждения исключения;
* throws – используется в сигнатуре методов для предупреждения, о том что метод может выбросить исключение.

1. **Является ли обработка исключений в программном коде обязательной?**

Нет, не является.

1. **Изобразите в виде блок-схемы алгоритмы файлового ввода и вывода данных.**

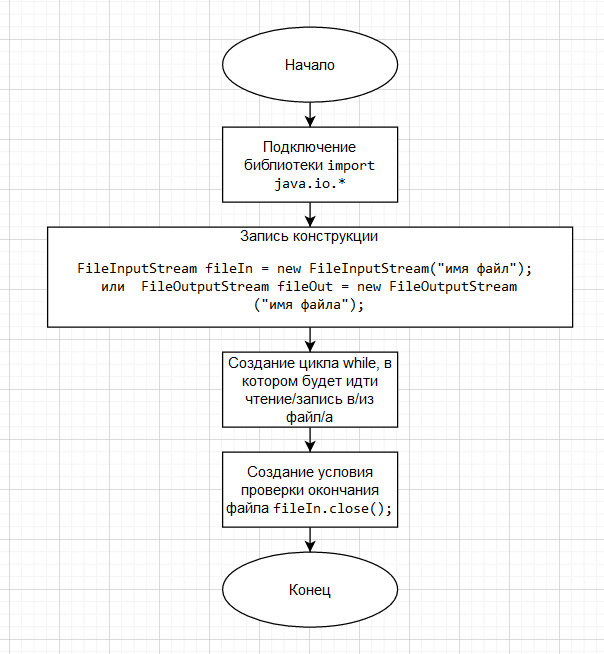


Рисунок 13 – Блок-схема алгоритма

1. **Объясните назначение класса JOptionPane.**

JOptionPane унаследован от базового класса JComponent библиотеки Swing, так что можно работать с ним напрямую, т.е. создавать экземпляры класса *JOptionPane* и настраивать их свойства. Использование стандартных диалоговых окон существенно упрощает разработку приложения и позволяет ускорить процесс освоения пользователем интерфейса.

1. **Изобразите в виде блок-схемы алгоритм ввода данных через диалоговое окно в программах на языке Java.**

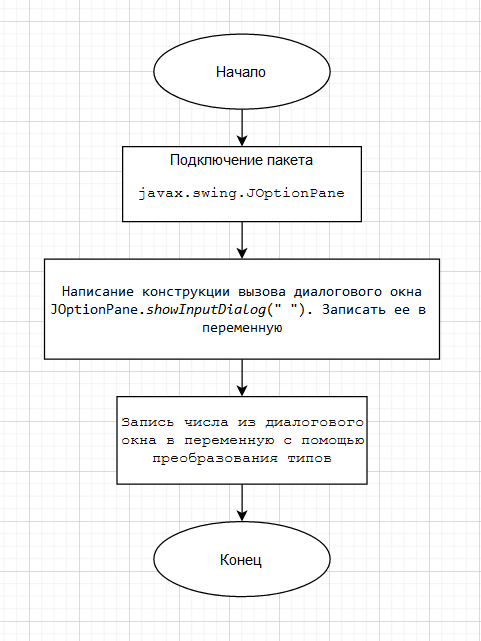


Рисунок 14 – Блок схема алгоритма

1. **Где хранятся в программе данные, переданные в качестве аргументов метода main ()? Каким образом передать параметры в программу?**

Как и в языке С, в языке Java метод main() может принимать параметры,  
которые можно передать через командную строку. Принятые параметры хранятся в массиве args[] в виде строковых литералов.