МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

кафедра Информационные системы

09.03.02 Информационные системы и технологии (уровень бакалавриата)

Лабораторная работа №1

Основы программирования на языке Java в среде Eclipse

по дисциплине: «Платформа Java»

Вариант – 8

Выполнил

студент 3 курса группы ИС/б-33-о

Генералов Николай Николаевич

Отметка о зачете\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Проверил

ст. пр. \_\_\_\_\_\_\_\_\_\_\_ Кузнецов С. А.

(должность) (подпись) (фамилия, инициалы)

г. Севастополь

2018 г.

# ЦЕЛЬ РАБОТЫ

В ходе выполнения лабораторной работы необходимо ознакомиться с функциональными возможностями среды разработки Eclipse, основами языка Java, приобрести практические навыки создания консольных приложений на языке Java, с возможностью доступа к файлам.

# ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу на языке Java, выполняющую вычисление длины окружности и площади круга по заданному радиусу. Предусмотреть:

* Ввод входных данных с клавиатуры(по умолчанию)и из файла(при запуске с параметром *–i<filename>).*
* Вывод результатов на экран(по умолчанию) и в файл(при запуске с параметром *–o<filename*>).
* Возможность запуска с двумя параметрами: *–i<filename1> –o<filename2>.* Предусмотреть обработку ошибок с использованием операторов языка Java: *try, catch, finally*.
* Ознакомиться со средствами отладки программ в среде Eclipse.
* Выполнить отладку разработанной программы.
* Проверить правильность работы программы на тестовых примерах.

# ТЕКСТ ПРОГРАММЫ

Файл “calcCircleProp.java”

import java.util.Scanner;

public class calcCircleProp {

public static void main(String[] args) {

// TODO Auto-generated method stub

StreamChanger streamChanger = new StreamChanger(args);

Scanner dataScanner = new Scanner(System.in);

if (!streamChanger.finIsOpen()) {

streamChanger.getStdOutputStream().println("Enter circle radius: ");

}

double radius = 0.0;

try {

if (!dataScanner.hasNextDouble()) {

throw new typeMissmatchException();

}

radius = dataScanner.nextDouble();

} catch (typeMissmatchException exception) {

streamChanger.getStdOutputStream().println(exception);

}

CircleFigure circle = new CircleFigure(radius);

CirclePropLogger logger = new CirclePropLogger(circle);

logger.saveProperties();

}

}

Файл “streamChanger.java”

import java.io.\*;

public class StreamChanger{

static private PrintStream stdOutStream = System.out;

static private InputStream stdInStream = System.in;

private FileInputStream \_fin = null;

private FileOutputStream \_fout = null;

public StreamChanger(String args[]){

try {

for (int i = 0; (i < args.length - 1); ++i) {

switch (args[i]) {

case "-i":{

\_fin = new FileInputStream (args[i + 1]);

System.setIn(\_fin);

break;

}

case "-o":{

\_fout = new FileOutputStream(args[i + 1]);

System.setOut(new PrintStream(\_fout));

break;

}

}

}

}

catch (FileNotFoundException exOb) {

stdOutStream.println(exOb.getMessage());

System.exit(1);

}

}

public boolean finIsOpen() {

return (\_fin == null ? false : true);

}

public boolean foutIsOpen() {

return (\_fout == null ? false : true);

}

public static PrintStream getStdOutputStream() {

return stdOutStream;

}

public static InputStream getStdInputStream() {

return stdInStream;

}

public static void ResetStreams() {

System.setIn(stdInStream);

System.setOut(stdOutStream);

}

@Override

protected void finalize() throws Throwable {

// TODO Auto-generated method stub

if (\_fin != null) {

\_fin.close();

}

if (\_fout != null) {

\_fout.close();

}

ResetStreams();

super.finalize();

}

}

Файл “CircleFigure.java”

public class CircleFigure {

double radius;

public CircleFigure(double \_rad){

radius = \_rad;

}

public double getSquare() {

return (Math.PI \* radius \* radius);

}

public double getLength() {

return (2 \* Math.PI \* radius);

}

}

Файл “PropLogger.java”

public interface PropLogger {

public void saveProperties();

}

Файл “CirclePropLogger.java”

public interface PropLogger {

public void saveProperties();

}

“TypeMissmatchException.java”

import javax.xml.ws.handler.MessageContext;

public class typeMissmatchException extends Exception {

public typeMissmatchException() {

super("Type missmatch exception. Variable waiting another type.");

}

public typeMissmatchException(String str) {

super(str);

}

}

# ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для проверки правильности работы программы было осуществлено несколько тестовых запусков с разным количеством параметров, а так же при их отсутствии. Исходные данные, а так же результаты выполнения изображены на рисунках 1-10.

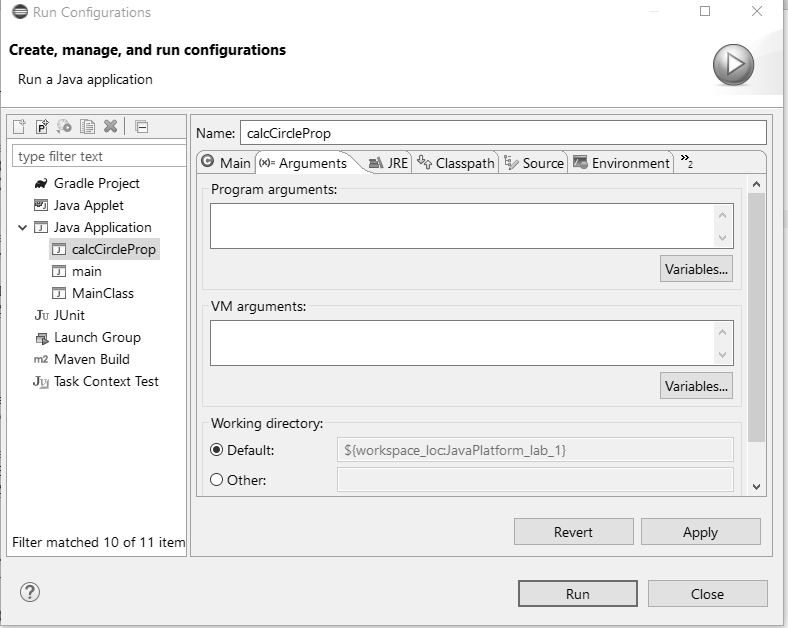


Рисунок 1 ­– Запуск программы без параметров, при помощи интегрированной среды разработки Eclipse.

При вводе целочисленного значения, исходные данные автоматически приводятся к типу double, что не вызывает ошибки при выполнении.

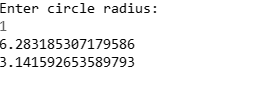


Рисунок 2 – Результат выполнения программы, при вводе целочисленного значения

Далее, в качестве исходных данных была введена текстовая строка:

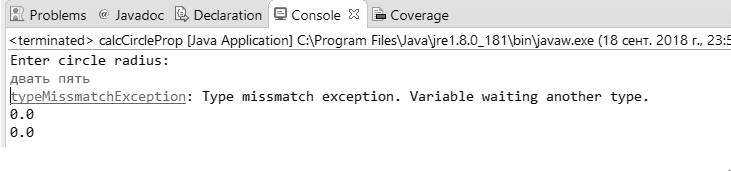


Рисунок 3 – Результат выполнения программы, при вводе текстовой строки, в качестве исходных данных

В результате было сгенерировано пользовательское исключение, гласящее о не совместимости типов.

При следующем запуске, программе были переданы параметры, позволяющие считать исходные данные из текстового файла. Текстовый файл «file\_1.txt» содержит число: 1,1213.

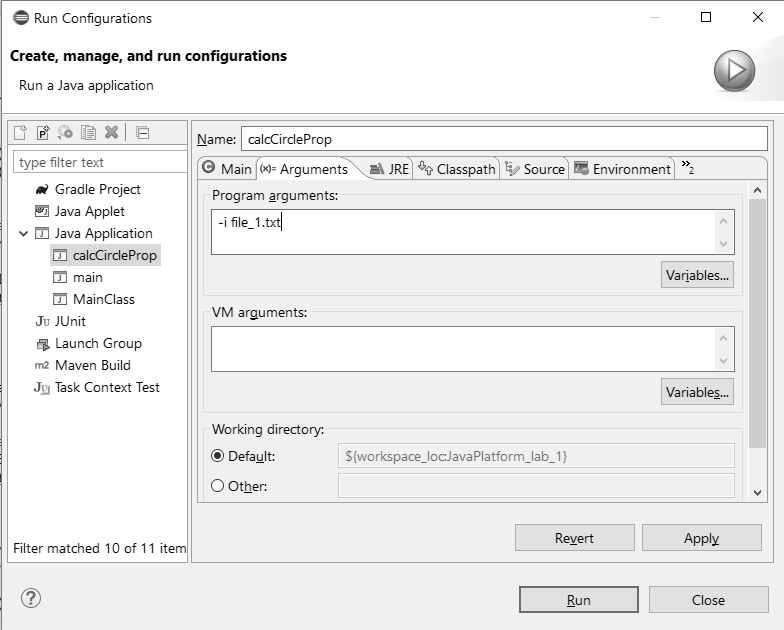


Рисунок 4 – Запуск программы с параметрами «-i file\_1.txt»

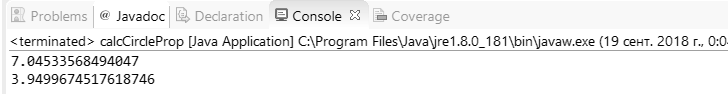


Рисунок 5 – Результат выполнения программы с параметрами «-i file\_1.txt»

В случае если программе не удается открыть указанный файл, генерируется соответствующее исключение и программа завершает работу:

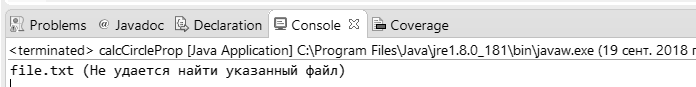


Рисунок 6 – Генерация исключения, при ошибке открытия файла

При запуске программы с параметрами «-o file\_2.txt», программа требует на ввод число типа double, и сохраняет результат в файл с указанным именем. Если файл отсутствовал, то он будет создан:

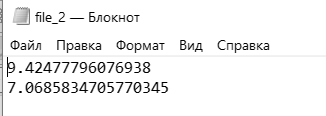


Рисунок 7 – Содержимое файла file\_2.txt, при использовании в качестве исходных данных число 1,5

# ВЫВОД

В ходе выполнения лабораторной работы были изучены основные типы данных, синтаксические конструкции циклов, ветвления языка Java. Был исследован встроенный механизм генерации и обработки ошибок.

Используя возможности интегрированной среды разработки Eclipse Photon, была написана и отлажена программа, позволяющая по заданному радиусу, вычислить длину окружности и площадь круга. Программа поддерживает возможность считывания и записи данных в файлы, имена которых можно указать при передаче соответствующих параметров командной строки.

По результатам тестирования можно сделать вывод, что программа работоспособна и поставленная цель достигнута.