ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

**ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Отчет**

по дисциплине «Разработка программных приложений на языках высокого уровня»

специальности 09.03.03 «Прикладная информатика»

**Лабораторная работа № 2**

студента очного отделения

3 курса 12002005 группы

Огородникова Льва Олеговича

Проверил:

старший преподаватель

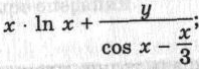
Лифиренко Максим Вячеславович

БЕЛГОРОД 2023

**Цель работы:** изучить основные понятия, синтаксис языка Java и общую структуру программ, получить практические навыки программирования на языке Java.

**Ход работы:**

**Вариант 15.**

1. Вычислить значения выражения по формуле: 

2. Дано действительное число x. Не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания, вычислить за минимальное число операций 2x^4 – Зх^3 + 4x^2 – 5х + 6.

3. Подсчитать количество целых чисел среди чисел a, b, с.

4. Даны три положительных числа. Определить, можно ли построить треугольник с длинами сторон, равным этим числам. Если можно, то ответить на вопрос, является ли он остроугольным.

5. Дано натуральное число N. Если оно делится на 4, вывести на экран ответ N = 4k (где k — соответствующее частное); если остаток от деления на 4 равен 1, N = 4k + 1; если остаток от деления на 4 равен 2, N = 4k + 2; если остаток от деления на 4 равен 3, N = 4k + 3. Например, 12 = 43, 22 = 45 + 2.

6. Натуральные числа a, b, с называются числами Пифагора, если выполняется условие а2 + b2 = с2. Напечатать все числа Пифагора, меньшие N.

7. Дано натуральное число n и действительное x. Найти сумму n членов ряда: 

8. В учебном заведении задается начало учебного дня, продолжительность «пары» или урока, продолжительность обычного и большого перерывов (и их «место» в расписании), количество пар (уроков). Получить расписание звонков на весь учебный день.

**Листинг класса Main:**

package com.company;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
import java.text.ParseException;  
  
public class Main  
{  
 */\*\* Метод main предназначен для выбора режима ввода данных,  
 \* выбора задачи и вызова соответствующего метода  
 \** ***@param*** *args Не используется.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.*

*\** ***@exception*** *ParseException При ошибке в парсинге даты.*

*\*/* public static void main(String[] args) throws FileNotFoundException, ParseException  
 {  
 System.*out*.println("Введите 1 для режима работы ввода параметров задач" +  
 " с клавиатуры или 2 для ввода параметров из файла");  
 Scanner scanner = new Scanner(System.*in*);  
 int inputMode = scanner.nextInt();  
 if (inputMode == 1 || inputMode == 2)  
 {  
 System.*out*.print("Введите номер задачи: ");  
 int nTask = scanner.nextInt();  
 Tasks tasks = new Tasks();  
 switch (nTask)  
 {  
 case 1 -> tasks.task1(inputMode);  
 case 2 -> tasks.task2(inputMode);  
 case 3 -> tasks.task3(inputMode);  
 case 4 -> tasks.task4(inputMode);  
 case 5 -> tasks.task5(inputMode);  
 case 6 -> tasks.task6(inputMode);  
 case 7 -> tasks.task7(inputMode);  
 case 8 -> tasks.task8(inputMode);  
 default -> System.*out*.println("Такой задачи нет");  
 }  
 }  
 else  
 {  
 System.*out*.println("Нет такого режима работы программы");  
 }  
 }  
}

**Листинг класса Tasks**

package com.company;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
import java.util.TimeZone;  
  
*/\*\* Класс Tasks описывает 8 задач 2-ой лабораторной работы. "  
 \** ***@author*** *Огородников Лев  
 \*/*public class Tasks  
{  
 */\*\* Метод task1 вычисляет значение выражения по формуле 15 варианта.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task1(int inputMode) throws FileNotFoundException  
 {  
 double x;  
 double y;  
 double result;  
 if (inputMode == 1)  
 {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введите x ");  
 x = scanner.nextDouble();  
 System.*out*.print("Введите y ");  
 y = scanner.nextDouble();  
 }  
 else  
 {  
 String path = "task\_1\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 x = scanner.nextDouble();  
 System.*out*.println("Ввод x из файла, x = " + x);  
 y = scanner.nextDouble();  
 System.*out*.println("Ввод y из файла, y = " + y);  
 }  
 result = x \* Math.*log*(x) + y/(Math.*cos*(x)-x/3);  
 System.*out*.println("Результат: " + result);  
 }  
  
 */\*\*  
 \* Вычисляет значение выражения 2x^4 - 3x^3 + 4x^2 - 5x + 6 для заданного значения x.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task2(int inputMode) throws FileNotFoundException  
 {  
 double x;  
 if (inputMode == 1)  
 {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите x: ");  
 x = scan.nextDouble();  
 }  
 else  
 {  
 String path = "task\_2\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 x = scanner.nextDouble();  
 System.*out*.println("Ввод из файла, x = " + x);  
 }  
 double x2 = x \* x;  
 double x4 = x2 \* x2;  
 double x3 = x2 \* x;  
 double result = (2 \* x4) - (3 \* x3) + (4 \* x2) - (5 \* x) + 6;  
 System.*out*.println("Результат выражения = " + result);  
 }  
  
 */\*\* Метод task3 считает количество целых чисел среди a, b, c  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task3(int inputMode) throws FileNotFoundException  
 {  
 double a;  
 double b;  
 double c;  
 if (inputMode == 1)  
 {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите число a: ");  
 a = scan.nextDouble();  
 System.*out*.print("Введите число b: ");  
 b = scan.nextDouble();  
 System.*out*.print("Введите число c: ");  
 c = scan.nextDouble();  
 }  
 else  
 {  
 String path = "task\_3\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 a = scanner.nextDouble();  
 System.*out*.println("Ввод a из файла, a = " + a);  
 b = scanner.nextDouble();  
 System.*out*.println("Ввод b из файла, b = " + b);  
 c = scanner.nextDouble();  
 System.*out*.println("Ввод c из файла, c = " + c);  
 }  
 int count = 0;  
 if (a == Math.*floor*(a) && !Double.*isInfinite*(a))  
 {  
 count++;  
 }  
 if (b == Math.*floor*(b) && !Double.*isInfinite*(b))  
 {  
 count++;  
 }  
 if (c == Math.*floor*(c) && !Double.*isInfinite*(c))  
 {  
 count++;  
 }  
 System.*out*.println("Количество целых чисел = " + count);  
 }  
  
 */\*\*Метод task4 определяет, можно ли построить треугольник с заданными длинами сторон,  
 \* и является ли он остроугольным.  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task4(int inputMode) throws FileNotFoundException  
 {  
 double a;  
 double b;  
 double c;  
 if (inputMode == 1)  
 {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите длину стороны a: ");  
 a = scan.nextDouble();  
 System.*out*.print("Введите длину стороны b: ");  
 b = scan.nextDouble();  
 System.*out*.print("Введите длину стороны c: ");  
 c = scan.nextDouble();  
 }  
 else  
 {  
 String path = "task\_4\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 a = scanner.nextDouble();  
 System.*out*.println("Ввод стороны a из файла, a = " + a);  
 b = scanner.nextDouble();  
 System.*out*.println("Ввод стороны b из файла, b = " + b);  
 c = scanner.nextDouble();  
 System.*out*.println("Ввод стороны c из файла, c = " + c);  
 }  
 if (a + b > c && a + c > b && b + c > a)  
 {  
 double maxSide = Math.*max*(Math.*max*(a, b), c);  
 double sumSquares = 0;  
 if (a == maxSide)  
 {  
 sumSquares = b \* b + c \* c;  
 }  
 else if (b == maxSide)  
 {  
 sumSquares = a \* a + c \* c;  
 }  
 else  
 {  
 sumSquares = a \* a + b \* b;  
 }  
 if (sumSquares > maxSide \* maxSide)  
 {  
 System.*out*.print("Треугольник остроугольный");  
 }  
 else if (sumSquares == maxSide \* maxSide)  
 {  
 System.*out*.print("Треугольник прямоугольный");  
 }  
 else  
 {  
 System.*out*.print("Треугольник тупоугольный");  
 }  
 }  
 else  
 {  
 System.*out*.print("Треугольник нельзя построить");  
 }  
 }  
  
 */\*\*Метод task5 Программа выводит значение числа N в зависимости от остатка от деления на 4  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task5(int inputMode) throws FileNotFoundException  
 {  
 int N;  
 if (inputMode == 1)  
 {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите натуральное число N: ");  
 N = scan.nextInt();  
 }  
 else  
 {  
 String path = "task\_5\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 N = scanner.nextInt();  
 System.*out*.println("Ввод натурального число N из файла, N = " + N);  
 }  
 int remainder = N % 4;  
 double k = N/4;  
 switch (remainder)  
 {  
 case 0:  
 System.*out*.println("N = " + N \* k);  
 break;  
 case 1:  
 System.*out*.println("N = " + N \* k + 1);  
 break;  
 case 2:  
 System.*out*.println("N = " + N \* k + 2);  
 break;  
 case 3:  
 System.*out*.println("N = " + N \* k + 3);  
 break;  
 default:  
 System.*out*.println("Некорректное значение числа");  
 break;  
 }  
 }  
  
 */\*\* Метод task6 выводит на экран все числа Пифагора, меньшие N..  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \* \*/* public void task6(int inputMode) throws FileNotFoundException  
 {  
 int N;  
 if (inputMode == 1)  
 {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Введите натуральное число N: ");  
 N = scan.nextInt();  
 }  
 else  
 {  
 String path = "task\_6\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 N = scanner.nextInt();  
 System.*out*.println("Ввод натурального число N из файла, N = " + N);  
 }  
 for (int a = 1; a < N; a++)  
 {  
 for (int b = a + 1; b < N; b++)  
 {  
 int c = (int) Math.*sqrt*(a \* a + b \* b);  
 if (c \* c == a \* a + b \* b && c < N)  
 {  
 System.*out*.println(a + "^2 + " + b + "^2 = " + c + "^2");  
 }  
 }  
 }  
 }  
  
 */\*\* Метод task7 находит сумму n членов ряда  
 \** ***@param*** *inputMode параметр передает режим ввода данных.  
 \** ***@exception*** *FileNotFoundException При ошибке открытия файла.  
 \*/* public void task7(int inputMode) throws FileNotFoundException  
 {  
 int n;  
 float x;  
 float sum = 0;  
 if (inputMode == 1)  
 {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число n ");  
 n = scanner.nextInt();  
 System.*out*.println("Введите число x ");  
 x = scanner.nextFloat();  
 }  
 else  
 {  
 String path = "task\_7\_data.txt";  
 Scanner scanner = new Scanner(new File(path));  
 n = scanner.nextInt();  
 System.*out*.println("Ввод из файла числa n " + n);  
 x = scanner.nextFloat();  
 System.*out*.println("Ввод из файла числa x " + x);  
 }  
 for (int i = 1; i <= n; i++)  
 {  
 long fact = 1;  
 for (int j = 2; j <= 2\*i-1; j++)  
 {  
 fact = fact \* j;  
 }  
 sum += Math.*pow*(-1,i-1)\*(Math.*pow*(x,2\*i-1)/fact);  
 }  
 System.*out*.println("Сумма n членов ряда = " + sum);  
 }  
  
 */\*\*  
 Метод task8 для получения расписания звонков на весь учебный день в учебном заведении.  
 Задаются параметры начала учебного дня, продолжительности пары, продолжительности  
 обычного и большого перерывов (и их место в расписании), количество пар.* ***@param*** *inputMode параметр передает режим ввода данных.* ***@exception*** *FileNotFoundException При ошибке открытия файла.* ***@exception*** *ParseException При ошибке в парсинге даты.  
 \*/* public void task8(int inputMode) throws FileNotFoundException, ParseException {  
 long lessonDuration, standardInterval, bigInterval, time;  
 int indexBigInterval, countLessons;  
 Date startDay;  
 SimpleDateFormat formater = new SimpleDateFormat("HH:mm");  
 formater.setTimeZone(TimeZone.*getTimeZone*("GMT"));  
 if (inputMode == 1)  
 {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Начало учебного дня: ");  
 startDay = formater.parse(scanner.nextLine());  
 time = startDay.getTime();  
 System.*out*.print("Продолжительность пары: ");  
 lessonDuration = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.print("Продолжительность обычного перерыва: ");  
 standardInterval = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.print("Продолжительность большого перерыва: ");  
 bigInterval = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.print("Его позиция: ");  
 indexBigInterval = scanner.nextInt();  
 System.*out*.print("Кол-во пар: ");  
 countLessons = scanner.nextInt();  
 System.*out*.println();  
 }  
 else  
 {  
 Scanner scanner = new Scanner(new File("task\_8\_data.txt"));  
 startDay = formater.parse(scanner.nextLine());  
 time = startDay.getTime();  
 System.*out*.println("Начало учебного дня: " + formater.format(new Date(time)));  
 lessonDuration = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.println("Продолжительность пары: " + formater.format(new Date(lessonDuration)));  
 standardInterval = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.println("Продолжительность обычного перерыва: " + formater.format(new Date(standardInterval)));  
 bigInterval = formater.parse(scanner.nextLine()).getTime();  
 System.*out*.println("Продолжительность большого перерыва: " + formater.format(new Date(bigInterval)));  
 indexBigInterval = scanner.nextInt();  
 System.*out*.println("Его позиция: " + indexBigInterval);  
 countLessons = scanner.nextInt();  
 System.*out*.println("Кол-во пар: " + countLessons);  
 System.*out*.println();  
 time = startDay.getTime();  
 }  
 for (int i = 1; i <= countLessons; i++)  
 {  
 System.*out*.println(i + " пара");  
 System.*out*.println(formater.format(new Date(time)));  
 time += lessonDuration;  
 System.*out*.println(formater.format(new Date(time)));  
 System.*out*.println();  
 if (i != countLessons)  
 {  
 System.*out*.println("Перерыв:");  
 System.*out*.println(formater.format(new Date(time)));  
 if (i != indexBigInterval)  
 {  
 time += standardInterval;  
 System.*out*.println(formater.format(new Date(time)));  
 }  
 else  
 {  
 time += bigInterval;  
 System.*out*.println(formater.format(new Date(time)));  
 }  
 System.*out*.println();  
 }  
 }  
 }  
}

Результаты работы программы представлены на рисунках 1–8.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Результат выполнения 1 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Результат выполнения 2 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Результат выполнения 3 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Результат выполнения 4 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Результат выполнения 5 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Результат выполнения 6 задания

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – Результат выполнения 7 задания

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – Результат выполнения 8 задания

**Вывод:** в ходе выполнения лабораторной работы были изучены основные понятия, синтаксис языка Java и общая структура программ, получены практические навыки программирования на языке Java.