Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 10

тема «Файлы»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Язов М.М.

Проверил: Нетбай Георгий Владимирович

Пермь, 2023

**Содержание**

# Задание 1

## 1.1. Постановка задачи

Переделать программу задания 3 лабораторной работы 9. Массив данных записан в файл и считывается в программе для обработки. Результаты обработки массива согласно вопросам задания 3 лабораторной работы 9 записываются в конец файла построчно с заголовками расшифровывающими, что за данные представлены ниже.

## 1.2. Решение задачи, код программы

import *java.util.*\*;  
import *java.io.*\*;  
  
public class *exc10\_1\_2* {  
  
 public static void writeToFile(*String data*, boolean *append*) {  
 try (*FileWriter* writer = new FileWriter("output.txt", *append*)) {  
 writer.write(*data*);  
 writer.flush();  
 } catch (*IOException e*) {  
 *System*.out.println("Error: " + *e*.getMessage());  
 }  
 }  
  
 public static *List*<*String*> readFromFile(*String filename*) {  
 *List*<*String*> lines = new ArrayList<>();  
 try (*BufferedReader* br = new BufferedReader(new FileReader(*filename*))) {  
 *String* line;  
 while ((line = br.readLine()) != null) {  
 lines.add(line);  
 }  
 } catch (*IOException e*) {  
 *System*.out.println("Error: " + *e*.getMessage());  
 }  
 return lines;  
 }  
  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
  
 *System*.out.print("Введите символ для анализа: ");  
 *String* symbol = input.nextLine();  
  
 *System*.out.print("Длина массива: ");  
 int len = input.nextInt();  
  
 *String* alphabet = "ё1234567890-=йцукенгшщзхъфывапролджэячсмитьбю.!№;%:?\*()\_+, /[]{}@#$^&~ЁЙЦУ" +  
 "КЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮqwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";  
 *Random* random = new Random();  
  
 *writeToFile*(" ", false);  
 for (int i = 0; i < len; i++) {  
 char c = alphabet.charAt(random.nextInt(alphabet.length()));  
 *writeToFile*(*String*.*valueOf*(c), true);  
 }  
  
 *List*<*String*> fileLines = *readFromFile*("output.txt");  
  
 *StringBuilder* output = new StringBuilder();  
  
 int counter = 0;  
 for (*String* line : fileLines) {  
 char[] characters = line.toCharArray();  
 for (char c : characters) {  
 if (*Character*.*isLowerCase*(c) && *Character*.*UnicodeBlock*.*of*(c) == *Character*.*UnicodeBlock*.CYRILLIC) {  
 counter++;  
 }  
 }  
 }  
 output.append("\nКоличество строчных русских букв: ").append(counter).append("\n");  
  
 int symbolCount = 0;  
 for (*String* line : fileLines) {  
 char[] characters = line.toCharArray();  
 for (char c : characters) {  
 if (*String*.*valueOf*(c).equals(symbol)) {  
 symbolCount++;  
 }  
 }  
 }  
 output.append("Количество символов '").append(symbol).append("' в массиве: ").append(symbolCount).append("\n");  
  
 if (symbol.matches("[0**-**9]+")) {  
 int digit = *Integer*.*parseInt*(symbol);  
 int digitCountInArray = 0;  
 for (*String* line : fileLines) {  
 char[] characters = line.toCharArray();  
 for (char c : characters) {  
 if (*Character*.*isDigit*(c) && *Integer*.*parseInt*(*String*.*valueOf*(c)) == digit) {  
 digitCountInArray++;  
 }  
 }  
 }  
 if (digitCountInArray >= 2) {  
 output.append("Среди символов есть 2 и более цифры, входящие в число ").append(digit).append("\n");  
 }  
 } else {  
 output.append("Введенный символ не является числом!\n");  
 }  
  
 int bracketPairsCount = 0;  
 for (int i = 0; i < fileLines.size() - 1; i++) {  
 *String* current = fileLines.get(i);  
 *String* next = fileLines.get(i + 1);  
 if ((current.equals("(") && next.equals(")")) ||  
 (current.equals("[") && next.equals("]")) ||  
 (current.equals("{") && next.equals("}"))) {  
 bracketPairsCount++;  
 }  
 }  
 if (bracketPairsCount > 0) {  
 output.append("Количество соседних закрытых скобок: ").append(bracketPairsCount).append("\n");  
 }  
  
 int spaceCounter = 0;  
 for (*String* line : fileLines) {  
 if (line.equals(" ")) {  
 spaceCounter++;  
 } else {  
 spaceCounter = 0;  
 }  
 if (spaceCounter > 2) {  
 output.append("Обнаружено более двух подряд идущих пробелов.\n");  
 break;  
 }  
 }  
  
 boolean foundSequence = false;  
 for (int i = 0; i < fileLines.size() - 2; i++) {  
 *String* num1 = fileLines.get(i);  
 *String* num2 = fileLines.get(i + 1);  
 *String* num3 = fileLines.get(i + 2);  
 if (num1.matches("\\d") && num2.matches("\\d") && num3.matches("\\d")) {  
 int n1 = *Integer*.*parseInt*(num1);  
 int n2 = *Integer*.*parseInt*(num2);  
 int n3 = *Integer*.*parseInt*(num3);  
 if (n1 > n2 && n2 > n3) {  
 foundSequence = true;  
 break;  
 }  
 }  
 }  
 if (foundSequence) {  
 output.append("Существуют такие натуральные i и j, что i < k < j < п и si, и si+1 убывающая последовательность цифр, а sj и sj+1 возрастающая последовательность цифр.\n");  
 } else {  
 output.append("Не найдено таких натуральных i и j, что i < k < j < п и si, и si+1 убывающая последовательность цифр, а sj и sj+1 возрастающая последовательность цифр.\n");  
 }  
  
 *writeToFile*(output.toString(), true);  
 *System*.out.println("Результаты записаны в файл 'output.txt'.");  
 }  
}

## 1.3. Тестирование работы программы

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

**Задача 2**

## 2.1. Постановка задачи

Записать в файл таблицу из задания 7 лабораторной работы 6. Если в таблице присутствуют дробные ответы, то вывести данные в формате 4 знака после запятой.

## 2.2. Решение задачи, код программы

import *java.io.File*;  
import *java.io.FileWriter*;  
import *java.io.IOException*;  
  
public class *exc10\_2* {  
  
 public static void Write(*String str*, *String filename*, boolean *bool*) {  
  
 try(*FileWriter* writer = new FileWriter(*filename*, *bool*)) {  
 *File* file = new File(*filename*);  
 if (!file.exists()) file.createNewFile();  
  
 writer.write(*str*);  
 writer.flush();  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 public static void main(*String*[] *args*) {  
 *String* filename = "src/test\_file.txt";  
 *Write*("", filename, false);  
  
 for (int i = 1; i <= 9; i++) {  
 for (int j = 1; j <= 9; j++) {  
 *Write*(i + " - " + j + " = " + (i - j) + "\t", filename, true);  
 }  
 *Write*("\n", filename, true);  
 }  
 }  
}

## 2.3. Тестирование работы программы

Таблица 2

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

**Задача 3**

## 3.1. Постановка задачи

Напечатать все слова из текстового файла, содержащие хотя бы одну большую латинскую букву.

## 3.2. Решение задачи, код программы

import *java.io.*\*;  
import *java.util.Scanner*;  
  
  
public class *exc10\_3* {  
  
 public static void Write(*String str*) {  
  
 try (*FileWriter* writer = new FileWriter("src/exc\_10\_4.txt")) {  
  
 *File* file = new File("src/exc\_10\_4.txt");  
 if (!file.exists()) file.createNewFile();  
 writer.write(*str*);  
 writer.flush();  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 public static void printWordsWithUpperCase(*String filename*) {  
  
 *BufferedReader* br = null;  
 try {  
 br = new BufferedReader(new FileReader(*filename*));  
 *String* line;  
  
 while ((line = br.readLine()) != null) {  
 *String*[] words = line.split("\\s+");  
 for (*String* word : words) {  
 if (*containsUppercase*(word)) {  
 *System*.out.println(word);  
 }  
 }  
 }  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
  
 private static boolean containsUppercase(*String word*) {  
 for (char c : *word*.toCharArray()) {  
 if (*Character*.*isUpperCase*(c)) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
 *System*.out.println("Введите текст: ");  
 *String* text = input.nextLine();  
 *String* filename = "src/exc\_10\_4.txt";  
 *Write*(text);  
 *printWordsWithUpperCase*(filename);  
  
 }  
}

## 3.3. Тестирование работы программы

Таблица 3

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

**Задача 4**

## 4.1. Постановка задачи

Выделить в текстовом файле все слова, разделенные символами-разделителями «\_.,;:\n\t!?», и все слова, в которых гласных латинских букв больше чем согласных букв, построчно записать в другой текстовый файл.

## 4.2. Решение задачи, код программы

import *java.io.*\*;  
import *java.util.Scanner*;  
  
public class *exc10\_4* {  
  
 public static void Write(*String str*, *String filename*){  
  
 try(*FileWriter* writer = new FileWriter(*filename*)) {  
 *File* file = new File(*filename*);  
 if (!file.exists()) file.createNewFile();  
  
 writer.write(*str*);  
 writer.flush();  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 public static void Write(*Runnable action*, *String filename*) {  
  
 try(*FileWriter* writer = new FileWriter(*filename*)) {  
 *File* file = new File(*filename*);  
 if (!file.exists()) file.createNewFile();  
  
  
  
 writer.flush();  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 private static boolean isVowels(char *ch*) {  
 *String* vowels = "aeuiu";  
 return vowels.contains(*Character*.*toString*(*ch*));  
 }

public static void hasMoreVowels(*String filename*, *FileWriter writer*) {  
 *BufferedReader* br = null;  
  
 try {  
 br = new BufferedReader(new FileReader(*filename*));  
 *String* line;  
  
 while ((line = br.readLine()) != null) {  
 *String*[] words = line.split("\\s+");  
 for (*String* word : words) {  
 int vowelsCount = 0;  
 int consonantsCount = 0;  
  
 for (char c : word.toCharArray()) {  
 if (*isVowels*(c)) {  
 vowelsCount++;  
 } else {  
 consonantsCount++;  
 }  
 }  
  
 if (vowelsCount > consonantsCount) {  
 *writer*.write(word + "\n");  
 }  
 }  
 }  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 } finally {  
 try {  
 if (br != null) {  
 br.close();  
 }  
 } catch (*IOException e*) {  
 *System*.out.println("Error closing the BufferedReader: " + *e*);  
 }  
 }  
}  
  
 private static void extractWords(*String filename*, *FileWriter writer*) {  
 *BufferedReader* br = null;  
  
 try {  
 br = new BufferedReader(new FileReader(*filename*));  
 *StringBuilder* sb = new StringBuilder();  
 int ch;  
  
 while ((ch = br.read()) != -1) {  
 char currentChar = (char) ch;  
  
 if (*Character*.*isLetterOrDigit*(currentChar)) {  
 sb.append(currentChar);  
 } else {  
 if (sb.length() > 0) {  
 *String* word = sb.toString();  
 *writer*.write(word + "\n");  
 sb.setLength(0); *// Сбросить StringBuilder для следующего слова* }  
 }  
 }  
  
 *// Проверить последнее слово в строке, если оно не заканчивается разделителем* if (sb.length() > 0) {  
 *String* word = sb.toString();  
 *writer*.write(word + "\n");  
 }  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 } finally {  
 try {  
 if (br != null) {  
 br.close();  
 }  
 } catch (*IOException e*) {  
 *System*.out.println("Error closing the BufferedReader: " + *e*);  
 }  
 }  
 }  
  
  
  
 public static void main(*String*[] *args*) {  
 *String* filename = "src/exc\_10\_5.txt";  
 *String* answer\_file = "src/exc\_10\_5ANS.txt";  
 *Scanner* input = new Scanner(*System*.in);  
 *System*.out.println("Введите текст с этими разделителями «\_.,;:\\n\\t!?»");  
 *String* text = input.nextLine();  
 *Write*(text, filename);  
  
 try (*FileWriter* writer = new FileWriter(answer\_file)) {  
 *hasMoreVowels*(filename, writer);  
 *extractWords*(filename, writer);  
 writer.flush();  
 *System*.out.println("Результаты записаны в файл exc\_10\_5res.txt");  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
}

## 4.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

**Задача 5**

## 5.1. Постановка задачи

Дано два файла с предложениями. Дописать в конец второго файла предложения из первого файла, в которых есть аббревиатуры.

## 5.2. Решение задачи, код программы

import *java.io.*\*;  
  
import *java.util.Scanner*;  
  
public class *exc10\_5\_2* {  
  
 private static boolean isAbbreviation(*String word*) {  
 return *word*.matches("^[A**-**ZА**-**Я0**-**9]+$");  
 }  
  
 public static void Write(*String str*, *String filename*) {  
 try(*PrintWriter* writer = new PrintWriter(new FileWriter(*filename*, true))) {  
 writer.println(*str*);  
 writer.flush();  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 public static void Read(*String filename*, *String filename2*) {  
 *BufferedReader* br = null;  
 try {  
 *File* file2 = new File(*filename2*);  
 if (!file2.exists()) file2.createNewFile();  
  
 br = new BufferedReader(new InputStreamReader(new FileInputStream(*filename*), "UTF-8"));  
 *String* line;  
  
 while ((line = br.readLine()) != null) {  
 *String*[] sentence = line.split("[.!?]+");  
 for (*String* words : sentence) {  
 if (*hasAbbreviation*(words)) *Write*(words, *filename2*);  
 }  
 }  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 } finally {  
 try {  
 if (br != null) br.close();  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
 }  
  
 private static boolean hasAbbreviation(*String sentence*) {  
 *String*[] words = *sentence*.split("\\s+");  
 for (*String* word : words) {  
 if (*isAbbreviation*(word)) return true;  
 }  
 return false;  
 }  
  
 public static void main(*String*[] *args*) {  
 *String* filename1 = "src/exc\_10\_5(1).txt";  
 *String* filename2 = "src/exc\_10\_5(2).txt";  
 *Scanner* input = new Scanner(*System*.in);  
 *System*.out.println("Введите текст, где есть аббревиатуры: ");  
 *String* abbr = input.nextLine();  
 *Write*(abbr, filename1);  
 *Read*(filename1, filename2);  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

**Задача 6**

## 6.1. Постановка задачи

Входной файл содержит сведения о спортсменах: ФИО, рост, вес, год рождения, цена контракта. Найти спортсменов с весом до 90 кг, годом рождения до 2007 и стоимостью контракта от 100000, подсчитать их количество и дописать ответ в конец файла.

## 6.2. Решение задачи, код программы

import *java.io.*\*;  
import *java.util.ArrayList*;  
  
public class *exc10\_6* {  
 public static *String* readFile() {  
 *StringBuilder* data = new StringBuilder();  
 try (*FileReader* reader = new FileReader("athletes.txt")) {  
 int c;  
 while ((c = reader.read()) != -1) {  
 data.append((char) c);  
 }  
 } catch (*IOException ex*) {  
 *System*.out.println(*ex*.getMessage());  
 }  
 return data.toString();  
 }  
  
 public static void writeFile(*String data*, boolean *bool*) {  
 try (*FileWriter* writer = new FileWriter("athletes.txt", *bool*)) {  
 writer.write(*data*);  
 writer.flush();  
 } catch (*IOException ex*) {  
 *System*.out.println(*ex*.getMessage());  
 }  
 }  
  
 public static void main(*String*[] *args*) {  
 *ArrayList*<*Double*> weights = new ArrayList<>();  
 *String* inputData = *readFile*();  
 int semicolonCount = 0, j = 1, count = 0;  
 boolean isWeight = true, isYear = true, isPrice = true;  
  
 for (int i = 0; i < inputData.length(); i++) {  
 if (inputData.charAt(i) == ';')  
 semicolonCount++;  
  
 if (semicolonCount == 3 && isWeight) {  
 while (true) {  
 if (inputData.charAt(i - j) == ';') {  
 weights.add(*Double*.*parseDouble*(inputData.substring(i - j + 2, i)));  
 j = 1;  
 isWeight = false;  
 break;  
 }  
 j++;  
 }  
 }  
  
 if (semicolonCount == 4 && isYear) {  
 while (true) {  
 if (inputData.charAt(i - j) == ';') {  
 weights.add(*Double*.*parseDouble*(inputData.substring(i - j + 2, i)));  
 j = 1;  
 isYear = false;  
 break;  
 }  
 j++;  
 }  
 }  
  
 if (semicolonCount == 5 && isPrice) {  
 while (true) {  
 if (inputData.charAt(i - j) == ';') {  
 weights.add(*Double*.*parseDouble*(inputData.substring(i - j + 2, i)));  
 j = 1;  
 isPrice = false;  
 break;  
 }  
 j++;  
 }  
 }  
  
 if (semicolonCount == 5) {  
 semicolonCount = 0;  
 isWeight = true;  
 isYear = true;  
 isPrice = true;  
 }  
 }  
  
 for (int i = 0; i < weights.size(); i += 3) {  
 if (weights.get(i) <= 90 && weights.get(i + 1) <= 2007 && weights.get(i + 2) > 100000) {  
 count++;  
 }  
 }  
  
 *writeFile*("\nКоличество спортсменов, удовлетворяющих критериям: " + count, true);  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

**Задача 7**

## 7.1. Постановка задачи

Записать в файл значения функции f(x) из задания 3 лабораторной работы 5 для всего диапазона переменной в виде: значение x; значение f(x). В начале файла должна быть указана постановка задачи, т.е. диапазон аргумента и вид функции (запись на языке java). Перед выводом цифровых значений должен быть сделан заголовок о том, в каком порядке идет вывод.

## 7.2. Решение задачи, код программы

import *java.io.File*;  
import *java.io.FileWriter*;  
import *java.io.IOException*;  
  
import static *java.lang.Math*.\*;  
  
public class *exc10\_7* {  
  
 public static void Write(*String str*, *String filename*, boolean *bool*) {  
  
 try (*FileWriter* writer = new FileWriter(*filename*, *bool*)) {  
 *File* file = new File(*filename*);  
 if (!file.exists()) file.createNewFile();  
  
 writer.write(*str*);  
 writer.flush();  
  
 } catch (*IOException e*) {  
 *System*.out.println("Error " + *e*);  
 }  
 }  
  
 public static void main(*String*[] *args*) {  
 *String* header = "Диапазон аргумента: -3<=x<=3\nФункция:\nif (x >= 0)\n\tf = -cos(2 \* PI \* x);\n" +  
 "else \n\tf = 3 \* PI \* x + sin(2 \* PI \* x);\nВывод в порядке возрастания x\n";  
 *String* filename = "src/test.txt";  
  
 *Write*(header, filename, false);  
 for (double x = -3; x <= 3; x += 0.5) {  
 if (x >= 0)  
 *Write*((" f("+x+") = " + (-*cos*(2 \* PI \* x)) + "\n"), filename, true);  
 else  
 *Write*(("f("+x+") = " + (3 \* PI \* x + *sin*(2 \* PI \* x)) + "\n"), filename, true);  
 }  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |