МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине СПП

Тема: «Базовые навыки работы с файловой системой в Java»

Выполнил

студент группы ПО-5

Романюк В.А

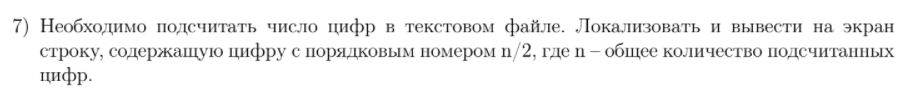
Проверил

Крощенко А. А.

Брест, 2021

Цель работы: приобрести базовые навыки работы с файловой системой в Java.

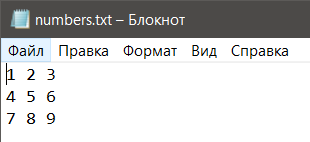


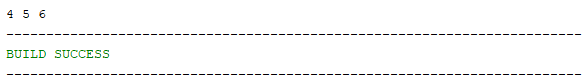


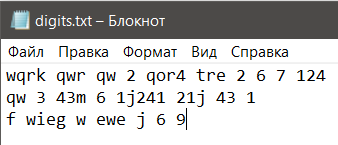
Код программы

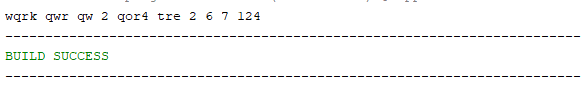
package com.company;  
  
import java.io.\*;  
  
class Lab2Task1 {  
  
 public static void main(String[] args) {  
 try {  
 if (args.length != 1) {  
 if (args.length == 0) {  
 System.*out*.println("Не указан аргумент.");  
 System.*exit*(0);  
 }  
 else {  
 System.*out*.println("Указано больше 1 аргумента.");  
 System.*exit*(0);  
 }  
 }  
  
 BufferedReader bfr = new BufferedReader(new FileReader(args[0]));  
 String str = "";  
 int digitCount = 0;  
  
 while((str = bfr.readLine()) != null) {  
 char[] strChar;  
 strChar = str.toCharArray();  
 for (int i = 0; i < strChar.length; i++) {  
 if (strChar[i] == '0' ||  
 strChar[i] == '1' ||  
 strChar[i] == '2' ||  
 strChar[i] == '3' ||  
 strChar[i] == '4' ||  
 strChar[i] == '5' ||  
 strChar[i] == '6' ||  
 strChar[i] == '7' ||  
 strChar[i] == '8' ||  
 strChar[i] == '9')  
 {  
 digitCount++;  
 }  
 }  
 }  
  
 bfr.close();  
 int index = 0;  
  
 if (digitCount != 0) {  
 index = digitCount / 2;  
 }  
 else {  
 System.*out*.println("Цифры не найдены.");  
 System.*exit*(0);  
 }  
  
 digitCount = 0;  
 bfr = new BufferedReader(new FileReader(args[0]));  
  
 while((str = bfr.readLine()) != null) {  
 char[] strChar;  
 strChar = str.toCharArray();  
 for (int i = 0; i < strChar.length; i++) {  
 if (strChar[i] == '0' ||  
 strChar[i] == '1' ||  
 strChar[i] == '2' ||  
 strChar[i] == '3' ||  
 strChar[i] == '4' ||  
 strChar[i] == '5' ||  
 strChar[i] == '6' ||  
 strChar[i] == '7' ||  
 strChar[i] == '8' ||  
 strChar[i] == '9')  
 {  
 digitCount++;  
 if (digitCount == index) {  
 System.*out*.println(str);  
 bfr.close();  
 System.*exit*(0);  
 }  
 }  
 }  
 }  
 }  
 catch(IOException e) {  
 System.*out*.println("Ошибка!\n" + e.getMessage());  
 e.printStackTrace();  
 }  
 }  
}

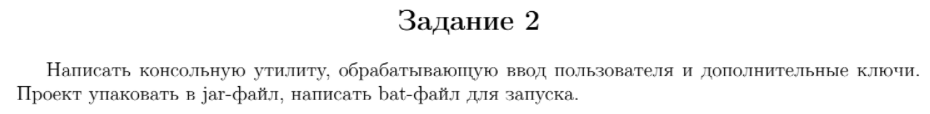
Тестирование

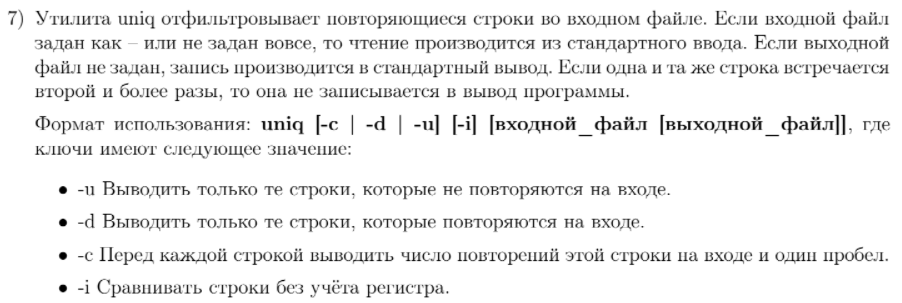








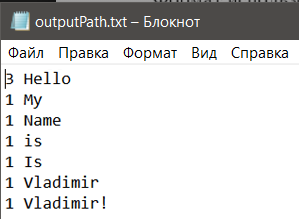
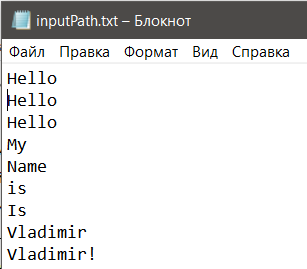


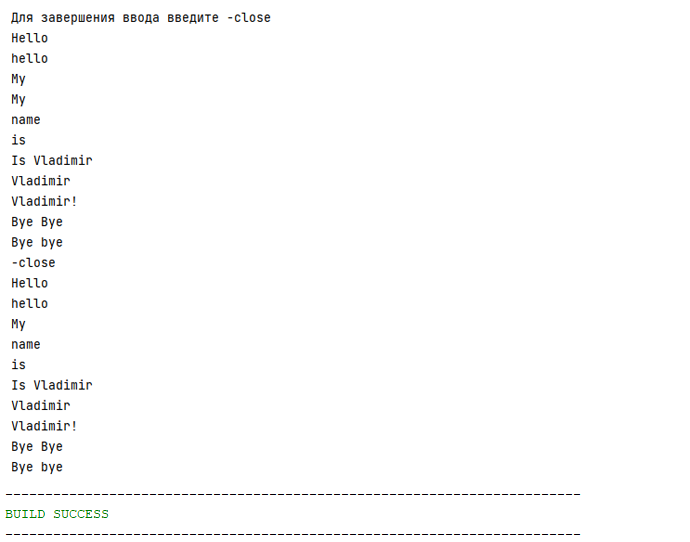


Код программы

package com.company;  
  
import java.io.\*;  
  
class Lab2Task2 {  
 public static void main(String[] args) {  
 try {  
 byte[] arguments = {0, 0, 0, 0};  
  
 for(String elem : args) {  
 switch(elem) {  
 case "-u":  
 arguments[0]++;  
 break;  
 case "-d":  
 arguments[1]++;  
 break;  
 case "-c":  
 arguments[2]++;  
 break;  
 case "-i":  
 arguments[3]++;  
 break;  
 }  
 }  
  
 if(arguments[0] > 1 || arguments[1] > 1 || arguments[2] > 1 || arguments[3] > 1) {  
 System.out.println("Вводить один и тот же ключ разрешено только один раз.");  
 System.exit(0);  
 }  
  
 if(arguments[0] == 1 && arguments[1] == 0 && arguments[2] == 0 ||  
 arguments[0] == 0 && arguments[1] == 1 && arguments[2] == 0 ||  
 arguments[0] == 0 && arguments[1] == 0 && arguments[2] == 1 ||  
 arguments[0] == 0 && arguments[1] == 0 && arguments[2] == 0)  
 {}  
 else {  
 System.out.println("Разрешено указать только один из ключей [-u, -c, -d]");  
 System.exit(0);  
 }  
  
 int pathsCount = 0;  
 int keysCount = 0;  
  
 for(int elem : arguments) {  
 keysCount += elem;  
 }  
  
 pathsCount = args.length - keysCount;  
  
 if(pathsCount > 2) {  
 System.out.println("Разрешено указывать только по одному пути для выходного и входного файлов.");  
 System.exit(0);  
 }  
  
 BufferedReader reader = null;  
 BufferedWriter writer = null;  
  
 switch(pathsCount) {  
 case 0:  
 reader = new BufferedReader(new InputStreamReader(System.in));  
 writer = new BufferedWriter(new OutputStreamWriter(System.out));  
 System.out.println("Для завершения ввода введите \"-close\"");  
 break;  
 case 1:  
 if(args[args.length - 1].equals("-")) {  
 reader = new BufferedReader(new InputStreamReader(System.in));  
 System.out.println("Для завершения ввода введите \"-close\"");  
 }  
 else {  
 reader = new BufferedReader(new FileReader(args[args.length - 1]));  
 }  
 writer = new BufferedWriter(new OutputStreamWriter(System.out));  
 break;  
 case 2:  
 if(args[args.length - 2].equals("-")) {  
 reader = new BufferedReader(new InputStreamReader(System.in));  
 System.out.println("Для завершения ввода введите \"-close\"");  
 }  
 else {  
 reader = new BufferedReader(new FileReader(args[args.length - 2]));  
 }  
 if(args[args.length - 1].equals("-")) {  
 writer = new BufferedWriter(new OutputStreamWriter(System.out));  
 }  
 else {  
 writer = new BufferedWriter(new FileWriter(args[args.length - 1]));  
 }  
 break;  
 }  
  
 String text = "";  
 String newLine = "";  
  
 while((newLine = reader.readLine()) != null) {  
 if(newLine.equals("-close")) {  
 break;  
 }  
 text += newLine + "\n";  
 }  
  
 String[] textLines = text.split("\n");  
 boolean keyI = false;  
 if(arguments[3] == 1) {  
 keyI = true;  
 }  
  
 if(arguments[0] == 1) {  
 boolean doRepeat = false;  
  
 for(int i = 0; i < textLines.length; i++) {  
 for(int j = 0; j < textLines.length; j++) {  
 if(isEqual(textLines[i], textLines[j], keyI) && i != j) {  
 doRepeat = true;  
 break;  
 }  
 }  
 if(!doRepeat) {  
 writer.write(textLines[i] + "\n");  
 }  
 doRepeat = false;  
 }  
 }  
 else if(arguments[1] == 1) {  
 for(int i = 0; i < textLines.length; i++) {  
 for(int j = 0; j < textLines.length; j++) {  
 if(isEqual(textLines[i], textLines[j], keyI) && i != j) {  
 writer.write(textLines[i] + "\n");  
 break;  
 }  
 }  
 }  
 }  
 else if(arguments[2] == 1) {  
 boolean[] boolTextLines = new boolean[textLines.length];  
 int strRepeatCount = 0;  
  
 for(int i = 0; i < textLines.length; i++) {  
 if(!boolTextLines[i]) {  
 for(int j = 0; j < textLines.length; j++) {  
 if(isEqual(textLines[i], textLines[j], keyI)) {  
 boolTextLines[j] = true;  
 strRepeatCount++;  
 }  
 }  
 writer.write(strRepeatCount + " " + textLines[i] + "\n");  
 strRepeatCount = 0;  
 }  
 }  
 }  
 else {  
 for(int i = 0; i < textLines.length; i++) {  
 writer.write(textLines[i] + "\n");  
 }  
 }  
  
 reader.close();  
 writer.close();  
 }  
 catch(FileNotFoundException e) {  
 System.out.println("Ошибка!\n" + e.getMessage());  
 }  
 catch(IOException e) {  
 System.out.println("Ошибка!\n" + e.getMessage());  
 }  
 }  
  
 static boolean isEqual(String first, String second, boolean lowerMode) {  
 if(lowerMode == true) {  
 return first.toLowerCase().equals(second.toLowerCase());  
 }  
 else {  
 return first.equals(second);  
 }  
 }  
}

Тестирование





Вывод: приобрел базовые навыки работы с файловой системой в Java.