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

**Задание 1. Напишите программу, которая читает текст построчно, а затем разбивает каждую строку на лексемы и выводит их в обратном порядке.**

**Выполнение:**

**Код программы**

package lab2\_1;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.List;

public class myProgram {

public static void main(String[] args){

String result = "";

List<String> lines;

try {

lines = Files.readAllLines(Paths.get("file.txt"));

for(int i = 0;i<lines.size();i++){

String newLine ="";

String[] words = lines.get(i).split(" ");

for(int j = words.length-1;j>=0;j--){

newLine +=words[j]+" ";

}

result += newLine + "\n";

}

} catch (IOException e) { e.printStackTrace();}

try { Files.write(Paths.get("file2.txt"),result.getBytes());}

catch (IOException e) { e.printStackTrace();}

}

}

**Спецификация ввода**

>java myProgram.java

\*Лексемы считываются из файла с названием file.txt

**Пример**

>java myProgram.java

\*Содержимое файла file.txt:

Hello my name is Maksim

How are you

Go to the shop

**Спецификация вывода**

\*Результат выводится в файл с названием file2.txt

**Пример**

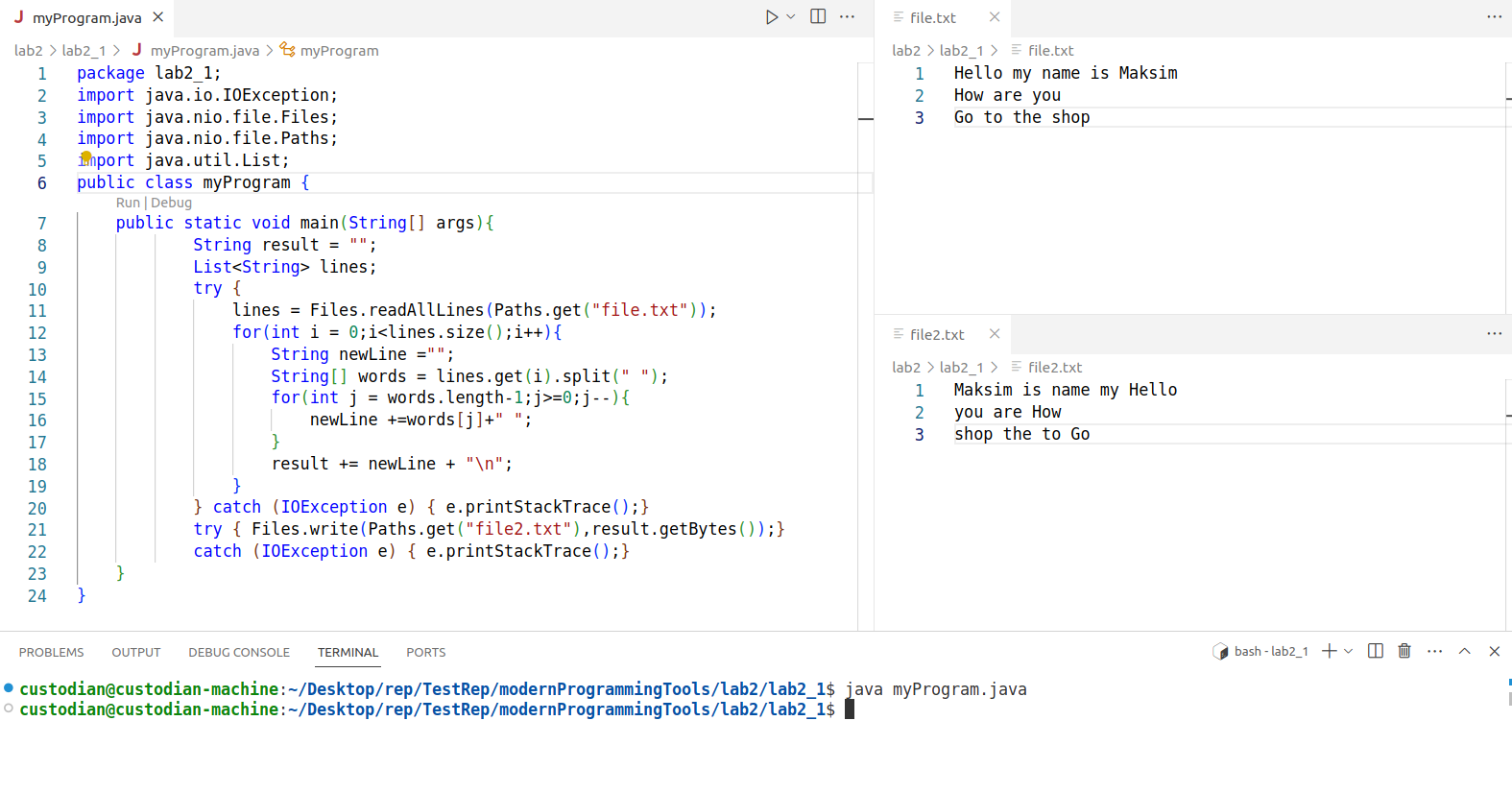
\*Содержимое файла file2.txt:

Maksim is name my Hello

you are How

shop the to Go

**Рисунки с результатами работы программы**

****

**Задание 2. Утилита uniq отфильтровывает повторяющиеся строки во входном файле. Если входной файл задан как – или не задан вовсе, то чтение производится из стандартного ввода. Если выходной файл не задан, запись производится в стандартный вывод. Если одна и та же строка встречается второй и более разы, то она не записывается в вывод программы. Формат использования: uniq [-c | -d | -u] [-i] [входной\_файл [выходной\_файл]], где ключи имеют следующее значение:**

**• -u Выводить только те строки, которые не повторяются на входе.**

**• -d Выводить только те строки, которые повторяются на входе.**

**• -c Перед каждой строкой выводить число повторений этой строки на входе и один пробел.**

**• -i Сравнивать строки без учёта регистра.**

**Выполнение:**

**Код программы**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.List;

public class myProgram {

public static void main(String[] args){

boolean haveC = false;

boolean haveD = false;

boolean haveU = false;

boolean haveI = false;

boolean haveIFile = false;

String iFile = "";

boolean haveOFile = false;

String oFile = "";

if(args.length>0){

if(args[0].equals("-c"))haveC = true;

else if(args[0].equals("-d"))haveD = true;

else if(args[0].equals("-u"))haveU = true;

else {haveIFile = true; iFile = args[0];}

}

if(args.length>1){

if(args[1].equals("-i")){

haveI = true;

if(args.length>3) {haveOFile = true; oFile = args[3];}

if(args.length>2) {haveIFile = true; iFile = args[2];}

}

else{

if((args.length>2)||(haveIFile)) {haveOFile = true; oFile = args[2];}

haveIFile = true;

iFile = args[1];

}

}

List<String> allLines = new ArrayList<String>();

try{

if(haveIFile) allLines = Files.readAllLines(Paths.get(iFile));

else allLines = Files.readAllLines(Paths.get("file.txt"));

} catch(IOException e){ e.printStackTrace();}

String[] tempLines = new String[allLines.size()];

if(haveI){

for(int i = 0;i<allLines.size();i++) tempLines[i] = allLines.get(i).toLowerCase();

} else for(int i = 0;i<allLines.size();i++) tempLines[i] = allLines.get(i);

int[] linesNum = new int[allLines.size()];

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

for(int j = i+1; j<tempLines.length;j++){

if(linesNum[i]>=0){

if(tempLines[i].equals(tempLines[j])){

linesNum[i]++;

linesNum[j]--;

}

}

}

}

String result = "";

if(haveC){

for(int i = 0;i<linesNum.length;i++) if(linesNum[i] >-1) result += Integer.toString(linesNum[i]+1)+" "+tempLines[i]+"\n";

} else if(haveD){

for(int i = 0;i<linesNum.length;i++) if(linesNum[i] > 0) result += tempLines[i]+"\n";

}else if(haveU){

for(int i = 0;i<linesNum.length;i++) if(linesNum[i] == 0) result += tempLines[i]+"\n";

}

try{

if(haveOFile) Files.write(Paths.get(oFile),result.toString().getBytes());

Files.write(Paths.get("file2.txt"),result.toString().getBytes());

} catch(IOException e){ e.printStackTrace();}

}

}

**Спецификация ввода**

>uniq [-c | -d | -u] [-i] [входной\_файл [выходной\_файл]]

**Пример**

>uniq -c file.txt file2.txt

\*Пример входного файла:

one

two

two

three

three

three

four

four

four

four

**Спецификация вывода**

Обработанные строки выводятся в выходной файл, по умолчанию file2.txt

**Пример**

\*Пример выходного файла:

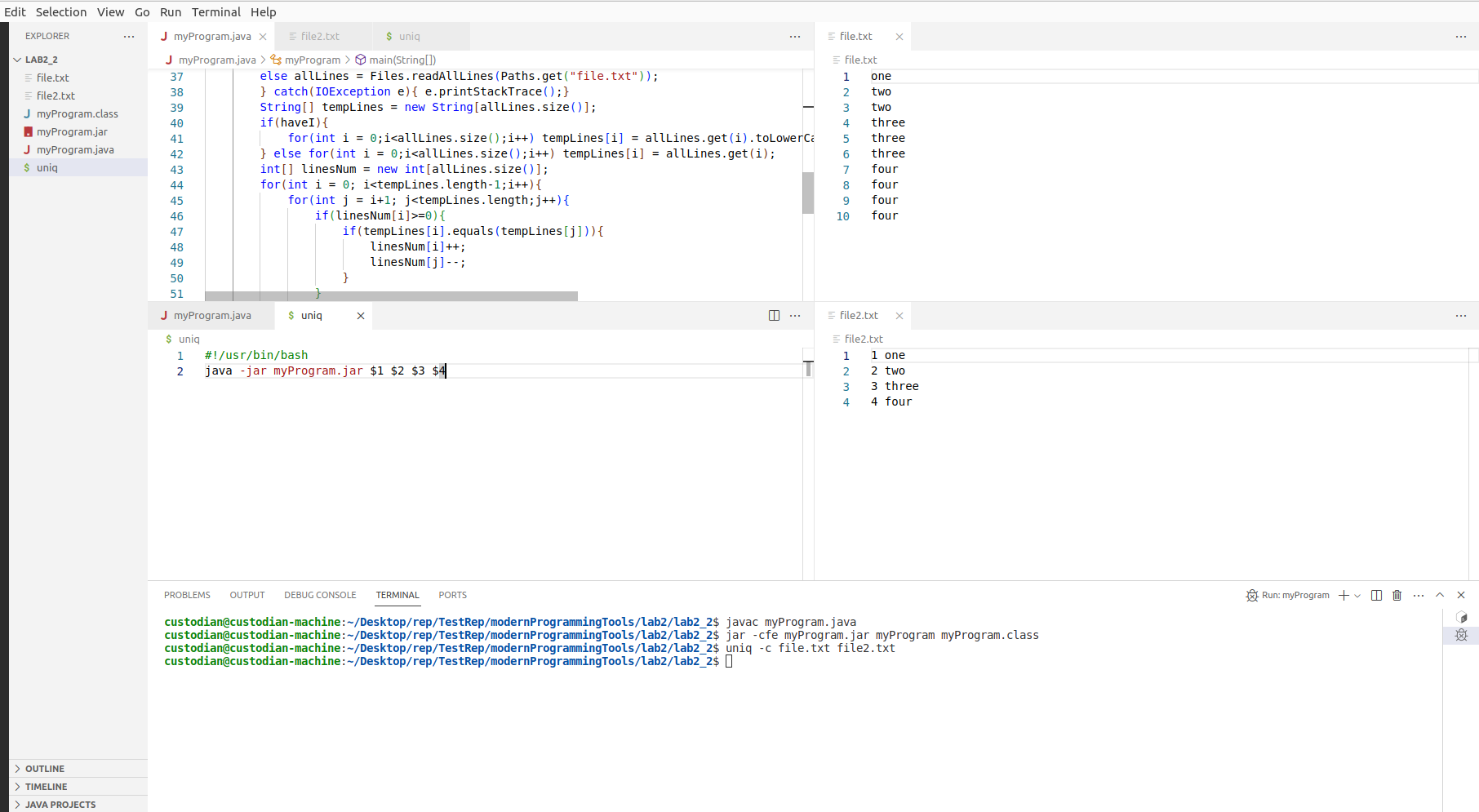
1 one

2 two

3 three

4 four

**Рисунки с результатами работы программы**

****

**Вывод:** научился работать с файловой системой Java, получил практический опыт разработки простейших программ на ЯП JAVA.