Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Центр программной инженерии

ОТЧЕТ

к лабораторной работе №3 по дисциплине:

|  |
| --- |
| «Объектно-ориентированное программирование» |
| Работа с потоками данных и обработка исключений |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб-19-2 |  |  |  | Кузнецова Ю. С. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Маланова Т.В. |
|  | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2020 г.

Оглавление

[Оглавление 2](#_Toc57578098)

[1 Постановка задачи 3](#_Toc57578099)

[2 Описание структуры пользовательского меню 4](#_Toc57578100)

[3 Проектирование классов 5](#_Toc57578101)

[4 Описание методов классов и спецификация локальных переменных 2](#_Toc57578102)

[5 Таблицы тестов 6](#_Toc57578103)

[6 Результаты тестирования 11](#_Toc57578104)

[7 Код программы 17](#_Toc57578105)

[8 Список использованных источников 26](#_Toc57578106)

1 Постановка задачи

1. Применение на практике объектно-ориентированного подхода для орга-низации системы ввода/вывода.

2. Получение практического опыта описания и реализации классов, созда-ния объектов, вызова методов, а также навыков работы с потоками дан-ных и обработки исключений на базе платформы Java.

Вариант №12

а. Дан массив чисел, знак операции и операнд. Выведите на печатьмассив, полученный в результате арифметического действия, совершенного над элементами массива.

b. Подсчитайте количество гласных букв в заданном тексте.

c. Переставьте слова в предложении в обратном порядке.

2 Описание структуры пользовательского меню

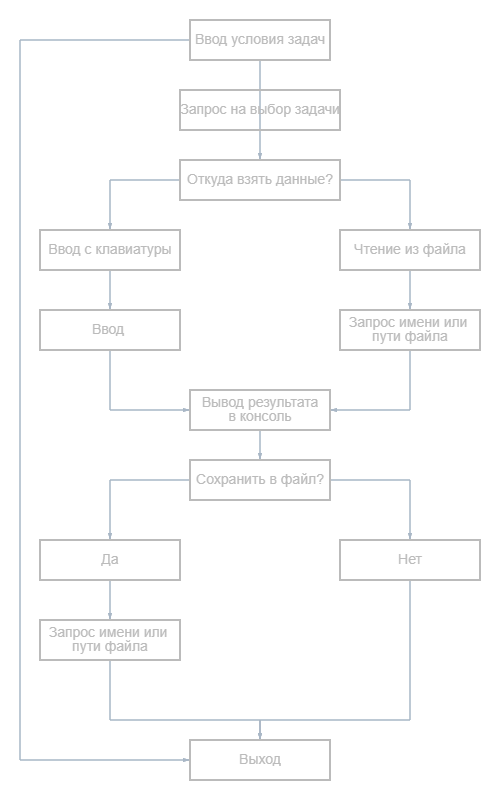


Рисунок 1 – Описание структуры пользовательского меню

3 Проектирование классов

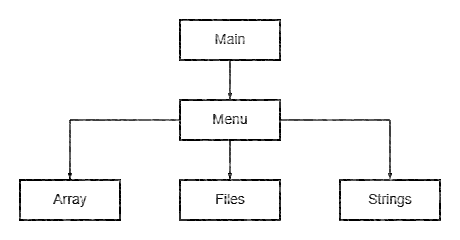


Рисунок 2 – Проектирование классов

4 Описание методов классов и спецификация локальных переменных

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Описание переменных | Тип данных |
|  |  | Class Main |  |
|  |  | Class Strings |  |
| 1 | s | Поле класса | String |
|  |  | public void Vowel() |  |
| 1 | count | Счётчик для подсчёта гласных | int |
| 2 | i | Переменная для перебора в цикле | int |
|  |  | public void opposite() |  |
| 1 | sentence | Строка s, разделённая на слова | String [] |
| 2 | change | Переменная для переворачивания массива | String |
| 3 | i | Переменная для перебора в цикле | int |
| 4 | str | Переменная для построения массива-результата | StringBuffer |
|  |  | Class Array |  |
| 1 | array | Поле класса – массив | Float [] |
| 2 | size | Поле класса – размер массива | int |
| 3 | sign | Поле класса – строка с оператором и операндом | String |
|  |  | public void Print(int size) |  |
| 1 | size | Размер массива | int |
|  |  | public void mathAct() |  |
| 2 | mark | Массив, содержащий математические знаки | Char [] |
| 3 | digits | Массив, содержащий цифры | Char [] |
| 4 | act | Переменная для определения действия над массивом | Char |
| 15 | с | Переменная для перебора в цикле | Char |
| 6 | digit | Переменная для хранения оператора и операнда | String |
| 7 | value | Переменная для перебора в цикле | Char |
| 8 | number | Переменная для поиска числа в digit | int |
| 9 | i | Переменная для перебора в цикле | int |
|  |  | @Override  Public String toString() |  |
| 1 | i | Переменная для перебора в цикле | int |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | line | Переменная для хранений значений в строковом типе | StringBuilder |
|  |  | Class Files |  |
| 1 | scan | Объект класса Scanner() для ввода данных | Scanner |
|  |  | static String strFromFile () |  |
| 1 | namefrom | Объект класса Scanner() для ввода данных | Scanner |
| 2 | rezult | Переменная – данные из файла | String |
| 3 | r | Объект класса BufferedReader для чтения из файла | BufferedReader |
| 4 | st | Переменная для хранения считанных значений | StringBuffer |
| 5 | s | Переменная для построчного считывания | String |
| 6 | i | Переменная для перебора в цикле | int |
|  |  | static void saveArray (float [] array, String filename1) |  |
| 1 | dos | Объект класса DataOutputStream dos для записи в файл | DataOutputStream dos |
| 2 | v | Переменная для перебора в цикле | var |
|  |  | static void save(int count) |  |
| 1 | nameto | Переменная для хранения имени файла | String |
| 2 | fos | Объект класса FileOutputStream для открытия файлового потока | FileOutputStream |
| 3 | pw | Объект класса PrintWriter для записи в файл | PrintWriter |
|  |  | static void saver(String str) |  |
| 1 | nameto | Переменная для хранения имени файла | String |
| 2 | br | Объект класса BufferedWriter для записи в файл | BufferedWriter |
|  |  | Static void Array arrayFromFILE (String namefrom) |  |
| 1 | arr | Объект класса Array | Array |
| 2 | br | Объект класса BufferedWriter для чтения из файла | BufferedReader |
| 3 | data | Переменная для считывания разделенного массива строк | String [] |
| 4 | size | Переменная для запоминания размера массива | int |
| 5 | temp | Числа в массиве | float [] |
| 6 | sign | Переменная для запоминания строки с оператором и операндом | String |
| 7 | i | Переменная для перебора в цикле | int |

1. Класс Array

Атрибуты класса:

private float[] array– массив чисел

private int size– размер массива

private String sign– строка для ввода оператора и операнда

Конструкторы класса:

public Array () – конструктор по умолчанию

public Array (int size) – конструктор с использованием ввода пользователем для размера массива

public Array (String sign) – конструктор с использованием ввода пользователем строки с оператором и операндом

public Array (int size, String sign) – конструктор с использованием ввода пользователем для размера массива и строки с оператором и операндом

public Array (Array input )– конструктор копирования

Методы класса:

public void Print(int size)

public void mathAct()

1. Класс Strings

Атрибуты класса:

private String s– строка

Конструкторы класса:

public Strings()– конструктор по умолчанию

public Strings(String s)– конструктор с параметром строки s

public Strings(Strings clone) – конструктор копирования

Методы класса:

public void Vowel()

public void opposite()

1. Класс Menu

Атрибуты класса:

public static Scanner scan – сканнер

Методы класса:

public static int ReadInt () – проверка на ввод числа в цикле

public static void MainMenu() – метод для работы с пользователем

public static void Print() – внутреннее меню для задания

public static void PrintOut() – метод, спрашивающий пользователя о сохранении в файл

public static void subMenu1 () – подменю для первого задания

public static void subMenu2 () – подменю для второго задания

public static void subMenu3 () – подменю для третьего задания

1. Class Files

public static void saver(String str) – метод записи строки в файл

public static void save(int count) – метод записи числа в файл

public static void saveArray (float [] array, String filename1) – метод записи массива в файл

public static String strFromFile () – метод чтения строк из файла

public static Array arrayFromFile(String filename1) – метод чтения данных для первого задания

5 Таблицы тестов

Таблица 2 – первая задача

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Выходные данные |
| 1 | 1  1  4  +98  4  -9  239  -987  0 | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 – Выход  Введите размерность массива  Введите оператор и операнд  Введите массив  Массив 132.0 89.0 337.0 -889.0  Сохранить массив в файл?  1 - да, 0 - выход  (появляется главное меню) |
| 2 | 1  2  1.txt  1  f1 | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 – Выход  Введите имя файла или местo  Массив 8.0 9.0 9.0  Сохранить массив в файл?  1 - да, 0 – выход  Введите имя файла или места, где нужно создать файл  Сохранено в файл: f1 |

Таблица 3 – вторая задача

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Выходные данные |
| 1 | 2  1  Make your dreams come true!  1  f2 | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 – Выход  Введите предложение  Количество гласных в заданном тексте: 11  Сохранить массив в файл?  1 - да, 0 – выход  Сохранено в файл: f2 |
| 2 | 2  2  2.txt  0 | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 – Выход  Введите предложение  Количество гласных в заданном тексте: 14  Сохранить массив в файл?  1 - да, 0 – выход |

Таблица 4 – третья задача

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Выходные данные |
| 1 | 3  1  Как же долго пишется отчёт  1  f3.txt | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 - Выход  Введите предложение  отчёт пишется долго же Как  Сохранить массив в файл?  1 - да, 0 - выход  Введите имя файла или места, где нужно создать файл  Строка отчёт пишется долго же Как  Сохранено в файл: f3.txt |
| 2 | 3  2  3.txt  0 | Главное меню  1 - Выполнить первое задание  -- Дан массив чисел, знак операции и операнд. Выведите на печать  массив, полученный в результате арифметического действия, совершенного над элементами массива.  2 - Выполнить второе задание  -- Подсчитайте количество гласных букв в заданном тексте.  3 - Выполнить третье задание  -- Переставьте слова в предложении в обратном порядке.  0 - Выход из программы  1 - Ввести данные с клавиатуры  2 - Считать данные из файла  0 - Выход  Введите имя файла  comes sun the rain After  Сохранить массив в файл?  1 - да, 0 - выход |

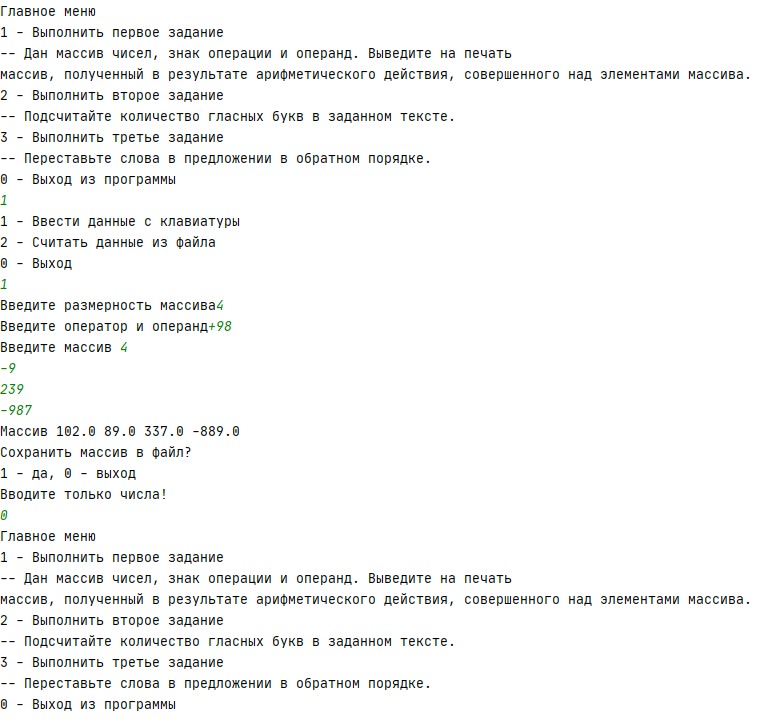
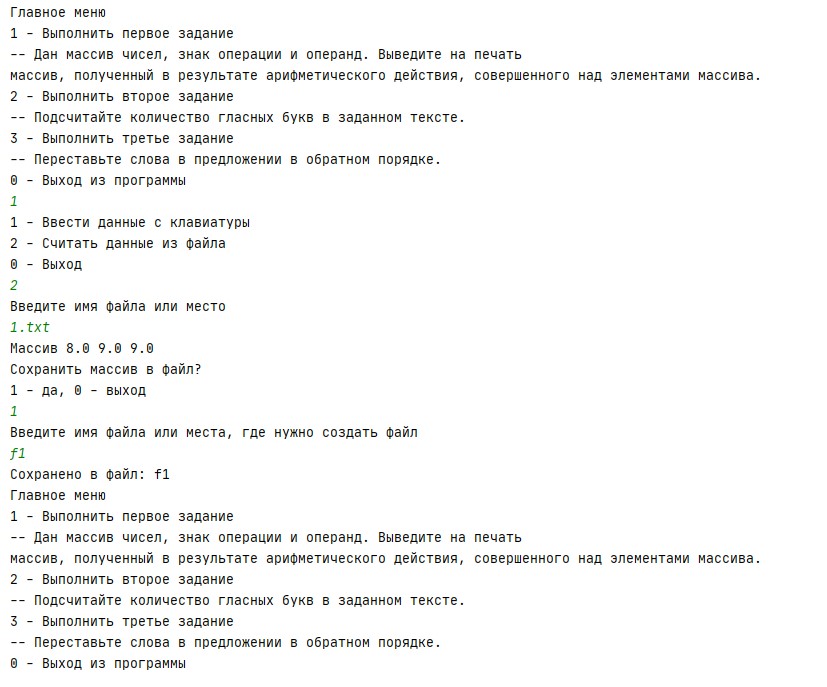
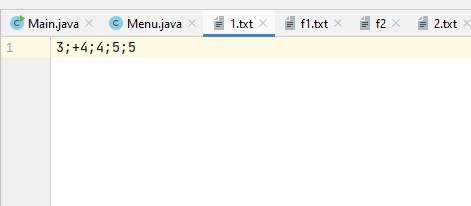
6 Результаты тестирования

Рисунок 3 – Тест 1 первой задачи





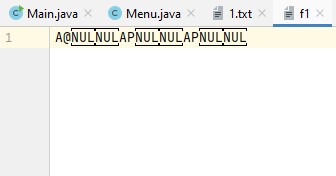
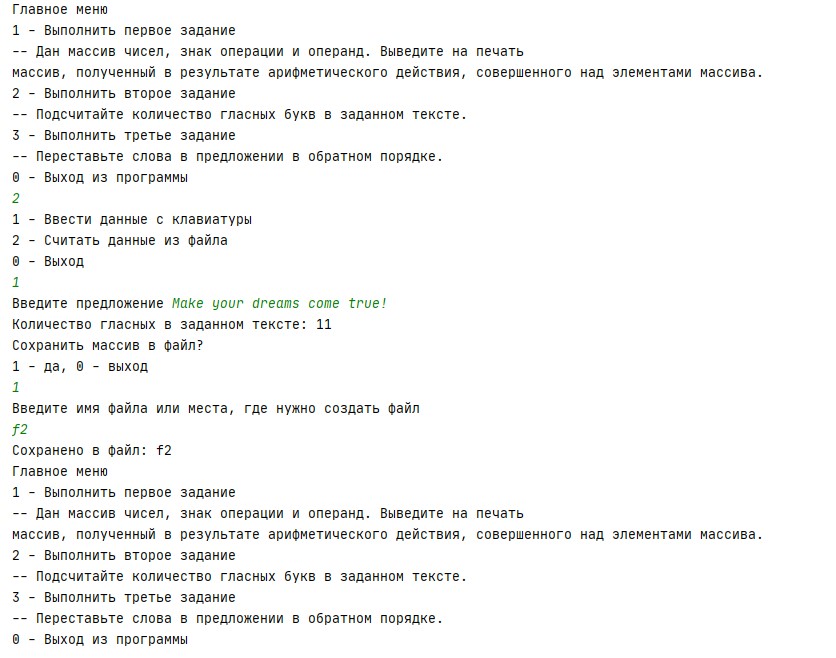


Рисунок 4, 5, 6 – тест 2 первой задачи



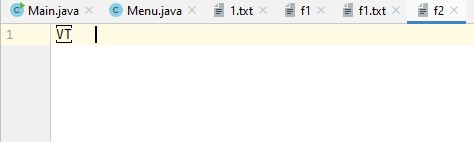
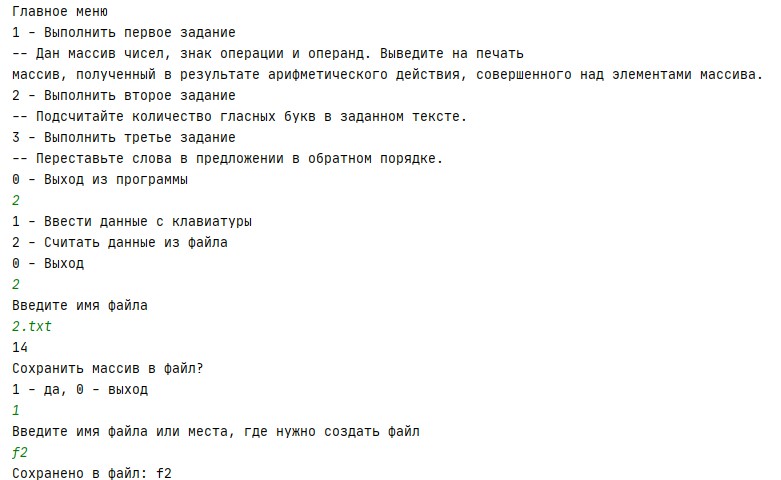
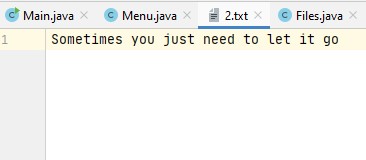


Рисунок 7, 8 – тест 1 второй задачи





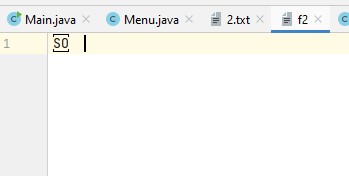
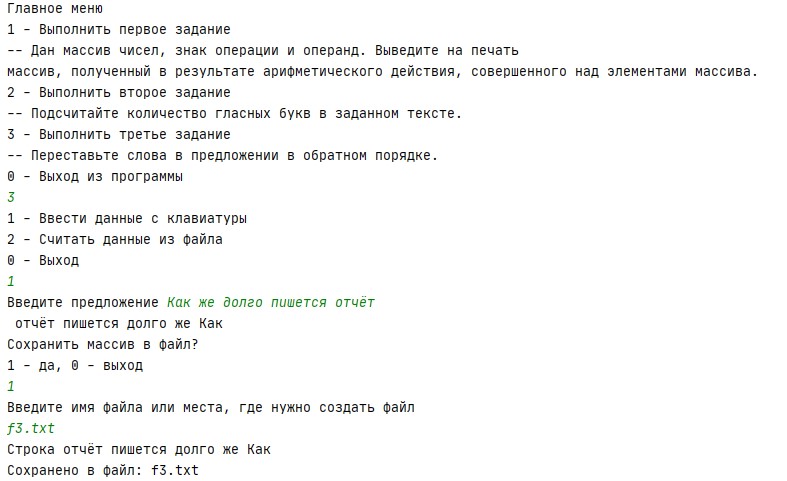


Рисунок 9, 10, 11 – тест 2 второй задачи



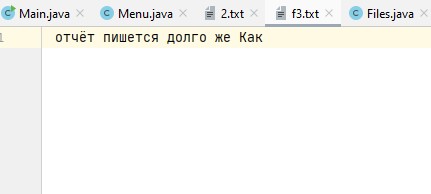
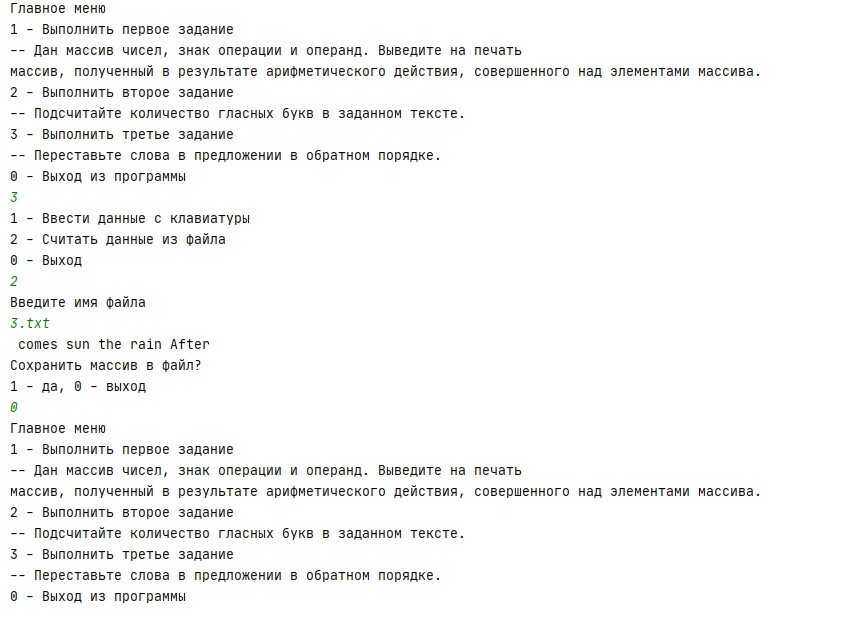


Рисунок 12, 13 – тест 1 третьей задачи



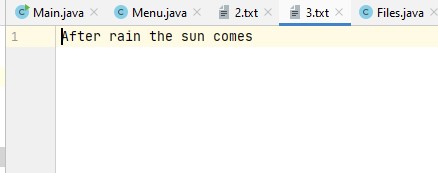


Рисунок 14, 15 – тест 2 третьей задачи

7 Код программы

Class Main

**package** com.company;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 Menu.*MainMenu*();  
 }  
}

Class Menu

**package** com.company;  
**import** java.util.Scanner;  
  
**public class** Menu {  
 **public static** Scanner *scan*;  
  
 **public** Menu() {  
 }  
  
 **public static int** ReadInt() {  
 **int** read= -1;  
  
 **try** {  
 read = Integer.*parseInt*(*scan*.nextLine());  
 } **catch** (NumberFormatException var2) {  
 System.***out***.println(**"Вводите только числа!"**);  
 *ReadInt*();  
 }  
  
 **return** read;  
 }  
  
 **public static void** MainMenu() {  
  
 **int** w;  
 **do** {  
 System.***out***.println(**"Главное меню"**);  
 System.***out***.println(**"1 - Выполнить первое задание"**);  
 System.***out***.println(**"-- Дан массив чисел, знак операции и операнд. Выведите на печать\nмассив, полученный в результате арифметического действия, совершенного над элементами массива. "**);  
 System.***out***.println(**"2 - Выполнить второе задание"**);  
 System.***out***.println(**"-- Подсчитайте количество гласных букв в заданном тексте."**);  
 System.***out***.println(**"3 - Выполнить третье задание"**);  
 System.***out***.println(**"-- Переставьте слова в предложении в обратном порядке."**);  
 System.***out***.println(**"0 - Выход из программы"**);  
 w = *ReadInt*();  
 **switch**(w) {  
 **case** 0:  
 **break**;  
 **case** 1:  
 *subMenu1*();  
 **break**;  
 **case** 2:  
 *subMenu2*();  
 **break**;  
 **case** 3:  
 *subMenu3*();  
 **break**;  
 **default**:  
 System.***out***.println(**"Ошибка, введите 1, 2, 3 или 0"**);  
 }  
 } **while**(w != 0);  
  
 }  
  
 **public static void** Print() {  
 System.***out***.println(**"1 - Ввести данные с клавиатуры"**);  
 System.***out***.println(**"2 - Считать данные из файла"**);  
 System.***out***.println(**"0 - Выход"**);  
 }  
  
 **public static void** PrintOut() {  
 System.***out***.println(**"Сохранить массив в файл?"**);  
 System.***out***.println(**"1 - да, 0 - выход"**);  
 }  
 **public static int** Positive ()  
 { **int** pos;  
 pos = *scan*.nextInt();  
 **if** ( pos <= 0)  
 {**throw new** NegativeArraySizeException(**" Вводите только положительное число"**);}  
 **return** pos;  
 }  
  
 **public static void** subMenu1() {  
 Array a = **new** Array();  
 *Print*();  
 **int** w = *ReadInt*();  
 **int** k;  
 String nameto;  
 **switch**(w) {  
 **case** 0:  
 **break**;  
 **case** 1:  
 System.***out***.print(**"Введите размерность массива"**);  
 **int** plus = *Positive*();  
 a.setSize(plus);  
 System.***out***.print(**"Введите оператор и операнд"**);  
 a.setSign(*scan*.next());  
 System.***out***.print(**"Введите массив "**);  
 **float**[] arr1 = **new float**[a.getSize()];  
  
 **for**(k = 0; k < a.getSize(); ++k) {  
 arr1[k] = (**float**)*scan*.nextInt();  
 }  
 a.setArray(arr1);  
 System.***out***.print(**"Массив "**);  
 a.mathAct();  
 System.***out***.println(a.toString());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 System.***out***.println(**"Введите имя файла или места, где нужно создать файл"**);  
 nameto = *scan*.nextLine();  
 Files.*saveArray*(a.mathAct(), nameto);  
 **return**;  
 }  
 **case** 2:  
 System.***out***.println(**"Введите имя файла или местo"**);  
 String namefrom = *scan*.nextLine();  
 Array Input = Files.*arrayFromFILE*(namefrom);  
 System.***out***.print(**"Массив "**);  
 Input.mathAct();  
 System.***out***.println(Input.toString());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 System.***out***.println(**"Введите имя файла или места, где нужно создать файл"**);  
 nameto = *scan*.nextLine();  
 Files.*saveArray*(Input.mathAct(), nameto);  
 **return**;  
 }  
 **default**:  
 System.***out***.println(**"Ошибка, введите 1, 2 или 0"**);  
 }  
  
 }  
  
 **public static void** subMenu2() {  
 Strings b = **new** Strings();  
 System.***out***.println(**"1 - Ввести данные с клавиатуры"**);  
 System.***out***.println(**"2 - Считать данные из файла"**);  
 System.***out***.println(**"0 - Выход"**);  
 **int** w = *ReadInt*();  
 **int** k;  
 **switch**(w) {  
 **case** 0:  
 **break**;  
 **case** 1:  
 Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.print(**"Введите предложение"**);  
 b.setS(sc.nextLine());  
 System.***out***.println(**"Количество гласных в заданном тексте: "** + b.Vowel());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 Files.*save*(b.Vowel());  
 **return**;  
 }  
 **case** 2:  
 b.setS(Files.*strFromFile*());  
 System.***out***.println(b.Vowel());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 Files.*save*(b.Vowel());  
 **return**;  
 }  
 **default**:  
 System.***out***.println(**"Ошибка, введите 1, 2 или 0"**);  
 }  
  
 }  
  
 **public static void** subMenu3() {  
 Strings c = **new** Strings();  
 System.***out***.println(**"1 - Ввести данные с клавиатуры"**);  
 System.***out***.println(**"2 - Считать данные из файла"**);  
 System.***out***.println(**"0 - Выход"**);  
 **int** w = *ReadInt*();  
 **int** k;  
 **switch**(w) {  
 **case** 0:  
 **break**;  
 **case** 1:  
 System.***out***.print(**"Введите предложение"**);  
 c.setS(*scan*.nextLine());  
 System.***out***.println(c.opposite());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 Files.*saver*(c.opposite());  
 **return**;  
 }  
 **case** 2:  
 c.setS(Files.*strFromFile*());  
 System.***out***.println(c.opposite());  
 *PrintOut*();  
 k = *ReadInt*();  
 **switch**(k) {  
 **case** 0:  
 **default**:  
 **return**;  
 **case** 1:  
 Files.*saver*(c.opposite());  
 **return**;  
 }  
 **default**:  
 System.***out***.println(**"Ошибка, введите 1, 2 или 0"**);  
 }  
  
 }  
  
 **static** {  
 *scan* = **new** Scanner(System.***in***);  
 }  
}

Class Files

**package** com.company;  
  
**import** java.io.\*;  
**import** java.util.Scanner;  
  
**public class** Files {  
 **public static** Scanner *scan* = **new** Scanner(System.***in***);  
 **public static** Array arrayFromFILE (String namefrom){  
 Array arr = **null**;  
 **try** (BufferedReader br = **new** BufferedReader(**new** FileReader(namefrom))){  
 String [] data = br.readLine().split(**";"**);  
 **int** size = Integer.*parseInt*(data[0]);  
 **float**[] temp = **new float**[size];  
 String sign =**""**;  
 sign+=data[1];  
 **for** (**int** i = 0; i< size; i++){  
 temp[i] = Float.*parseFloat*(data[i+2]);  
 }  
 arr = **new** Array(temp,sign);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **return** arr;  
 }  
  
 **public static** String strFromFile () {  
 System.***out***.println(**"Введите имя файла"**);  
 String namefrom = *scan*.nextLine();  
 String rezult = **null**;  
 **try** (BufferedReader r = **new** BufferedReader(**new** FileReader(namefrom))) {  
 StringBuffer st= **new** StringBuffer(**" "**);  
 String s = r.readLine();  
 **for** (**int** i = 0; i< s.length(); i ++){  
 st.append(s.charAt(i));  
 }  
 rezult= st.toString();  
 }  
 **catch** (NullPointerException e) {  
 System.***out***.println(**"Файл пустой"**); *strFromFile*();  
 }  
 **catch** (IOException e) {  
 System.***out***.println(**"Ошибка в чтение файла"**); *strFromFile*();  
 }  
 **return** rezult;  
 }  
  
  
 **public static void** saveArray (**float** [] array, String filename1) {  
 **try** (DataOutputStream dos = **new** DataOutputStream(**new** FileOutputStream(filename1)))  
 {  
 **for** (**var** v : array) {  
 dos.writeFloat(v);  
 }  
 System.***out***.println(**"Сохранено в файл: "** + filename1);  
 } **catch** (FileNotFoundException fileNotFoundException) {  
 System.***out***.println(**"Файл не найден"**);  
 } **catch** (IOException ioException) {  
 System.***out***.println(**"Ошибка записи"**);  
 }  
 }  
  
 **public static void** save(**int** count) {  
 System.***out***.println(**"Введите имя файла или места, где нужно создать файл"**);  
 String nameto = *scan*.nextLine();  
 **try** {  
 FileOutputStream fos = **new** FileOutputStream(nameto);  
 PrintWriter pw = **new** PrintWriter(fos);  
 pw.write(count);  
 System.***out***.println(**"Сохранено в файл: "** + nameto);  
 pw.close();  
 fos.close();  
 }**catch** (IOException e) {  
 System.***out***.println(**"Ошибка записи"**);  
  
 }  
 }  
  
 **public static void** saver(String str) {  
 System.***out***.println(**"Введите имя файла или места, где нужно создать файл"**);  
 String nameto = Menu.*scan*.nextLine();  
 **try** (BufferedWriter br = **new** BufferedWriter(**new** FileWriter(nameto))){  
 br.write(str);  
 System.***out***.println(**"Строка"** + str);  
 System.***out***.println(**"Сохранено в файл: "** + nameto);  
 }**catch** (IOException e) {  
 System.***out***.println(**"Ошибка записи"**);  
  
 }  
 }  
  
}

Class Array

**package** com.company;  
  
**import** java.util.Scanner;  
  
**class** Array {  
 *//Поля* **private float**[] **array**;  
 **private int size**;  
 **private** String **sign**;  
 *//Конструкторы* **public** Array(){  
 **this** (2);  
 **array** = **new float**[**size**];  
 **sign**=**"+2"**;  
 }  
 **public** Array(**int** size){  
 **this**.**size** = size;  
 **array** = **new float**[size];  
 **sign**=**"+2"**;  
 }  
 **public** Array(String sign){  
 **size**=2;  
 **array** = **new float**[**size**];  
 **this**.**sign** = sign;  
 }  
 **public** Array(**int** size, String sign){  
 **this**.**size** = size;  
 **this**.**sign** =sign;  
 **array** = **new float**[size];  
 }  
 **public** Array(**float**[] array, String sign){  
 **this**.**array** = array;  
 **this**.**sign** = sign;  
 **this**.**size** = array.**length**;  
 }  
 **public** Array(Array input){  
 **this**.**size** = input.**size**;  
 **this**.**sign** =input.**sign**;  
 **array** = **new float**[input.**size**];  
 **for**(**int** i = 0; i < input.**size**; i++){  
 **array**[i]=input.**array**[i];  
 }  
 }  
  
 **public** Array(**float**[] readable) {  
 }  
  
 *//Getters and Setters* **public void** setSign(String sign) {  
 **this**.**sign** = sign;  
 }  
 **public int** getSize() {  
 **return size**;  
 }  
 **public void** setSize(**int** size) {  
 **this**.**size** = size;  
 **this**.**array** = **new float**[**this**.**size**];  
 }  
 **public void** setArray(**float** [] array){  
 **for**(**int** i = 0; i < **this**.**size**; i++){  
 **this**.**array**[i]=array[i];  
 }  
 }  
  
 **public float**[] getArray() {  
 **return this**.mathAct();  
 }  
  
 *//Метод, выводящий размер массива* **public static void** Print(**int** size) {  
 System.***out***.print(size+**" "**);  
 }  
 *// Метод для работы с массивом* **public float** [] mathAct() {  
 **char**[] mark = **sign**.toCharArray();  
 **char**[] digits = **new char**[]{**'0'**, **'1'**, **'2'**, **'3'**, **'4'**, **'5'**, **'6'**, **'7'**, **'8'**, **'9'**};  
 **char** act = **'0'**;  
 **for** (**char** c : mark) {  
 **if** (c == **'+'**) {  
 act = **'+'**;  
 **break**;  
 }  
 **if** (c == **'-'**) {  
 act = **'-'**;  
 **break**;  
 }  
 **if** (c == **'\*'**) {  
 act = **'\*'**;  
 **break**;  
 }  
 **if** (c == **'/'**) {  
 act = **'/'**;  
 **break**;  
 }  
 }  
 String digit =**""**;  
 **for** (**char** c : mark) {  
 **for** (**char** value : digits) {  
 **if** (value == c) {  
 digit += value;  
 **break**;  
 }  
 }  
 }  
 **if** (digit.equals(**"/0"**))  
 {  
 **throw new** ArithmeticException(**"Попытка деления на 0"**);  
 }  
 **int** number = Integer.*parseInt*(digit);  
 **switch** (act) {  
 **case '+'** -> {  
 **for** (**int** i = 0; i < **this**.getSize(); i++) {  
 **this**.**array**[i] += number;  
 }  
 }  
 **case '-'** -> {  
 **for** (**int** i = 0; i < **this**.getSize(); i++) {  
 **this**.**array**[i] -= number;  
 }  
 }  
 **case '\*'** -> {  
 **for** (**int** i = 0; i < **this**.getSize(); i++) {  
 **this**.**array**[i] \*= number;  
 }  
 }  
 **case '/'** -> {  
 **for** (**int** i = 0; i < **this**.getSize(); i++) {  
 **this**.**array**[i] /= number;  
 }  
 }  
 }  
 **return array**;  
 }  
 @Override  
 **public** String toString()  
 {  
 StringBuilder line = **new** StringBuilder(**""**);  
 **for** (**int** i =0; i< **this**.**size**; i++)  
 {  
 line.append(**this**.**array**[i]).append(**" "**);  
 } **return** line.toString();  
 }

Class Strings

**package** com.company;  
  
**public class** Strings {  
 *//Field* **private** String **s**;  
 *//Конструкторы* **public** Strings(){  
 **s** =**""**;  
 }  
 **public** Strings(String s){  
 **this**.**s** = s ;  
 }  
 **public** Strings(Strings clone){  
 **this**.**s** = clone.**s**;  
 }  
 *//Getters and Setters* **public** String getS() {  
 **return s**;  
 }  
 **public void** setS(String s) {  
 **this**.**s** = s;  
 }  
 *//Метод поиска и подсчета гласных* **public int** Vowel() {  
 **int** count = 0;  
 **s**=**s**.toLowerCase();  
 **for** (**int** i = 0; i < **s**.length(); i++) {  
 **if** (**s**.charAt(i) == **'e'** || **s**.charAt(i) == **'y'** || **s**.charAt(i) == **'u'** || **s**.charAt(i) == **'i'** || **s**.charAt(i) == **'o'** || **s**.charAt(i) == **'a'**) {  
 count++;  
 }  
 } **return** count;  
 }  
 *//Метод переворачивания слов в предложении в обратном порядке* **public** String opposite() {  
 String[] sentence = **s**.split(**" "**);  
 String change;  
 **for** (**int** i = sentence.**length**-1, j = 0; i >=sentence.**length**/2 ; i--,j++) {  
 change = sentence[j];  
 sentence[j] = sentence[i];  
 sentence[i] = change;  
 }  
 StringBuffer str = **new** StringBuffer(**" "**);  
 **for** (**int** i = 0; i < sentence.**length**; i++) {  
 str.append(sentence[i]).append(**" "**);  
 }  
 **return** str.toString();  
 }  
  
}

8 Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во

ИРНИТУ, 2017. – 24 c.