Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

|  |
| --- |
| Институт информационных технологий и анализа данных |

наименование института

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине:

|  |
| --- |
| **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ Программирование** |
| **«Работа с потоками данных и обработка исключений»** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил | АСУб-20-2 |  |  |  | Арбакова А.В. |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил |  |  |  |  | Бакшеева Е.Н. |
|  | должность |  | подпись |  | Фамилия И.О. |

Иркутск 2021 г.

**Содержание**

[1. Постановка задачи 3](#_Toc134187016)

[2. Описание структуры пользовательского меню 3](#_Toc134187017)

[3. Проектирование классов 3](#_Toc134187018)

[4. Описание методов классов и спецификация локальных переменных 3](#_Toc134187019)

[5. Таблица тестов 3](#_Toc134187020)

[6. Результат тестирования 3](#_Toc134187021)

[7. Исходный код 3](#_Toc134187022)

[8. Список использованных источников 3](#_Toc134187023)

1. **Постановка задачи**

Провести объектную декомпозицию задачи из индивидуального задания и разработать класс, содержащий соответствующие поля для хранения необходимых данных и методы, обеспечивающие достаточную для решения задачи функциональность класса. Разработать класс, реализующий текстовое меню, позволяющее осуществлять выбор вариантов выполнения индивидуального задания. Создание объекта класса, решающего задание и вызов его методов должны осуществляться из класса, реализующего меню. При выполнении задания предусмотреть методы для организации ввода и выводы данных как на консоль, так и в текстовые файлы. Работа с файлами может быть организована методами класса, реализующего пользовательское меню, либо в рамках специально разработанного класса. Для всех проверяемых исключений должны быть реализованы обработчики, в которых выводится сообщение о возникновении исключительной ситуации и предпринимаются меры для продолжения работы программы в штатном режиме.

Задачи из 1 лабораторной работы по 1 варианту:

1. Вычислите сумму диагоналей квадратной матрицы.

2. Определите, является ли заданная строка символов числом.

3. Выполнить сложение двух матриц.

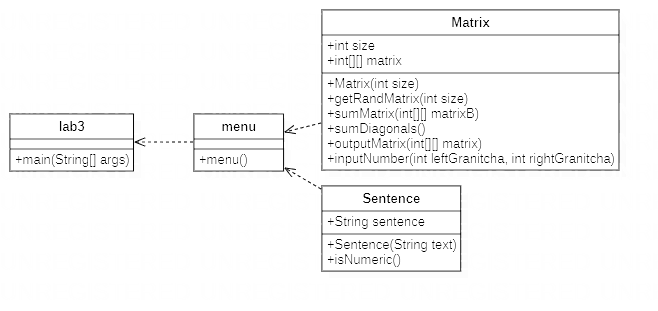
1. **Описание структуры пользовательского меню**

Главное меню программы, состоит из следующих предложенных пользователю вариантов:

1. Вычислить сумму диагоналей квадратной матрицы. – Осуществляет выполнение 1-го задания.
2. Определить, является ли заданная строка символов числом. – Осуществляет выполнение 2-го задания.
3. Выполнить сложение двух матриц. – Осуществляет выполнение 3-го задания.
4. Выход. – Выполняется выход из программы.
5. **Проектирование классов**

Таблица 1 – Классы.

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| lab3 | Главный класс, из которого вызываются остальные классы |
| menu | Класс-меню |
| Sentence | Класс, использующийся во 2-м задании для проверки строки |
| Matrix | Класс, описывающий свойства матрицы и её функции и использующийся для заданий 1 и 3 |

****

1. **Описание методов классов и спецификация локальных переменных**

Таблица 2 – Таблицы классов, полей и методов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Поля** | **Методы** | **Параметры** |
| lab2 | - | main(String[] args) | String[] args – аргументы переданные через консоль |
| menu | - | menu() | - |
| Matrix | int size  int[][] matrix | Matrix(int size) | int size – размер матрицы |
| getRandMatrix(int size) | int size – размер матрицы |
| sumMatrix(int[][] matrixB) | int[][] matrixB – значения матрицы В |
| sumDiagonals() | - |
| outputMatrix(int[][] matrix) | int[][] matrix – значение матрицы |
| inputNumber(int leftGranitcha, int rightGranitcha) | int leftGranitcha – «левая» граница значений для вводимого числа размерности матрицы int rightGranitcha – «правая» граница значений для вводимого числа размерности матрицы |
| Sentence | String sentence | Sentence(String text) | String text – вводимая строка, которая проверяется на число |
| isNumeric() | - |

Таблица 3 – Спецификации переменных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название** | **Тип** | **Назначение** | **ОДЗ** |
| 1 | input | int | Ввод размерности матриц в классе lab2 | [2;10] |
| 2 | size | int | Размерность матрицы | [2;10] |
| 3 | matrix | int[][] | Двумерный массив, хранящий значения матрицы | [0;2147483647] |
| 4 | i | int | Счетчик | [0;2147483647] |
| 5 | j | int | Счетчик | [0;2147483647] |
| 6 | k | int | Переменная для преобразования типа данных | [0;2147483647] |
| 7 | sumMatrix | int[][] | Двумерный массив, хранящий значения суммы матриц А и В | [0;2147483647] |
| 8 | sumMain | int | Сумма главной диагонали матрицы | [0;2147483647] |
| 9 | sumSide | int | Сумма побочной диагонали матрицы | [0;2147483647] |
| 10 | input | String | Ввод размерности матриц в классе Matrix | - |
| 11 | x | int | Выводимое значение размерности матриц в классе Matrix | [2;10] |
| 12 | isCorrect | boolean | Правильность введенных значений | [True / False] |
| 13 | console | Scanner | Поток ввода | - |
| 14 | builder | StringBuilder | Выводит на экран оследовательность символов | - |
| 15 | isWork | boolean | Состояние программы | [True / False] |
| 16 | scanner | Scanner | Поток ввода | - |
| 17 | i | String | Ввод пункта меню | [1;4] |

Таблица 4 – Спецификации функций.

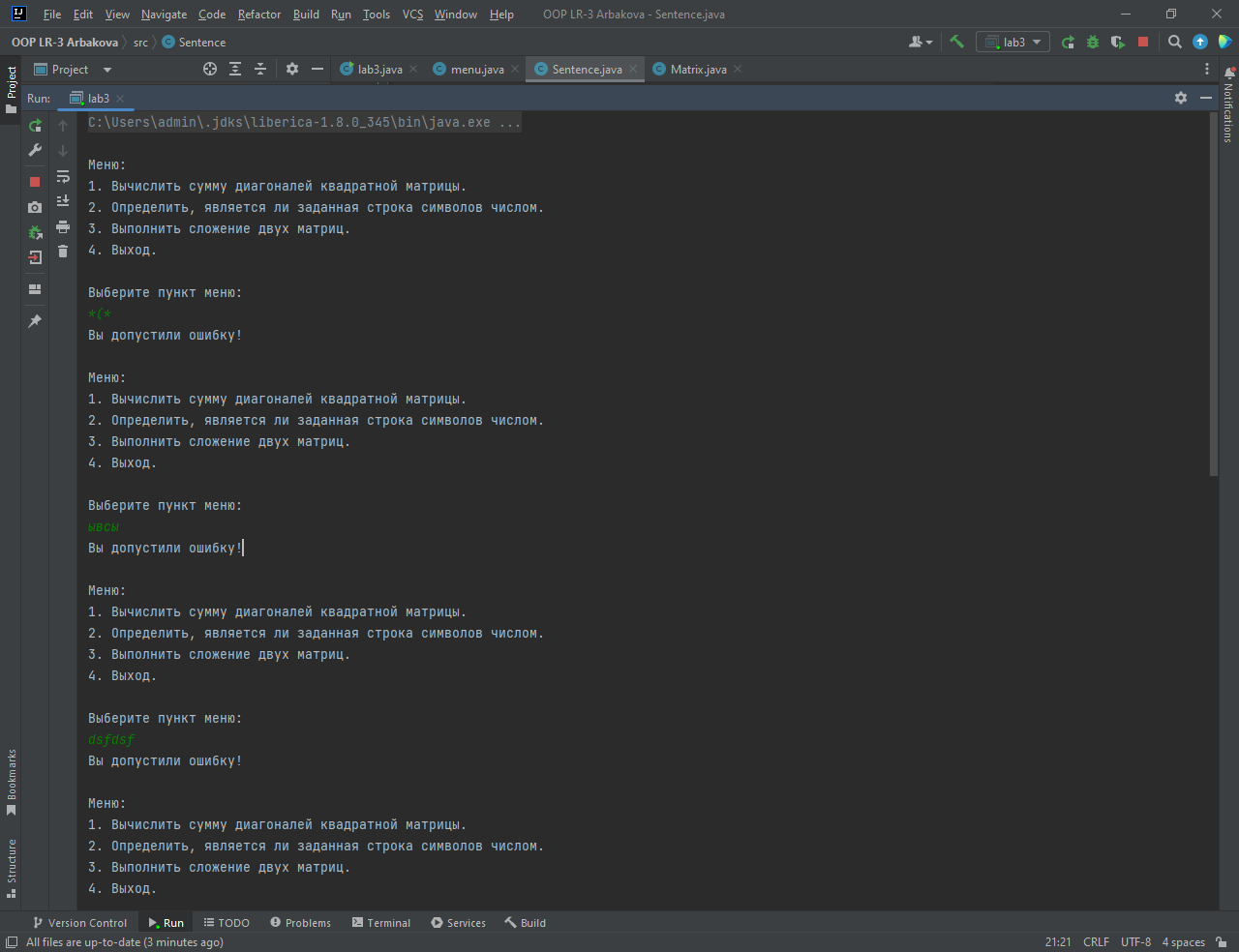
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название** | **Тип** | **Назначение** | **Параметры** |
| 1 | main | void | Главная функция приложения | String[] args |
| 2 | getRandMatrix | int[][] | Случайное заполнение матрицы значениями | int size |
| 3 | sumMatrix | void | Вычисление суммы матриц А и В | int[][] matrixB |
| 4 | sumDiagonals | void | Вычисление сумм диагоналей – главной и побочной | - |
| 5 | outputMatrix | void | Вывод матрицы на экран | int[][] matrix |
| 6 | inputNumber | int | Ввод размерности матрицы | int leftGranitcha, int rightGranitcha |
| 7 | isNumeric | boolean | Является ли введенная строка числом | - |
| 8 | menu | void | Вывод меню | - |

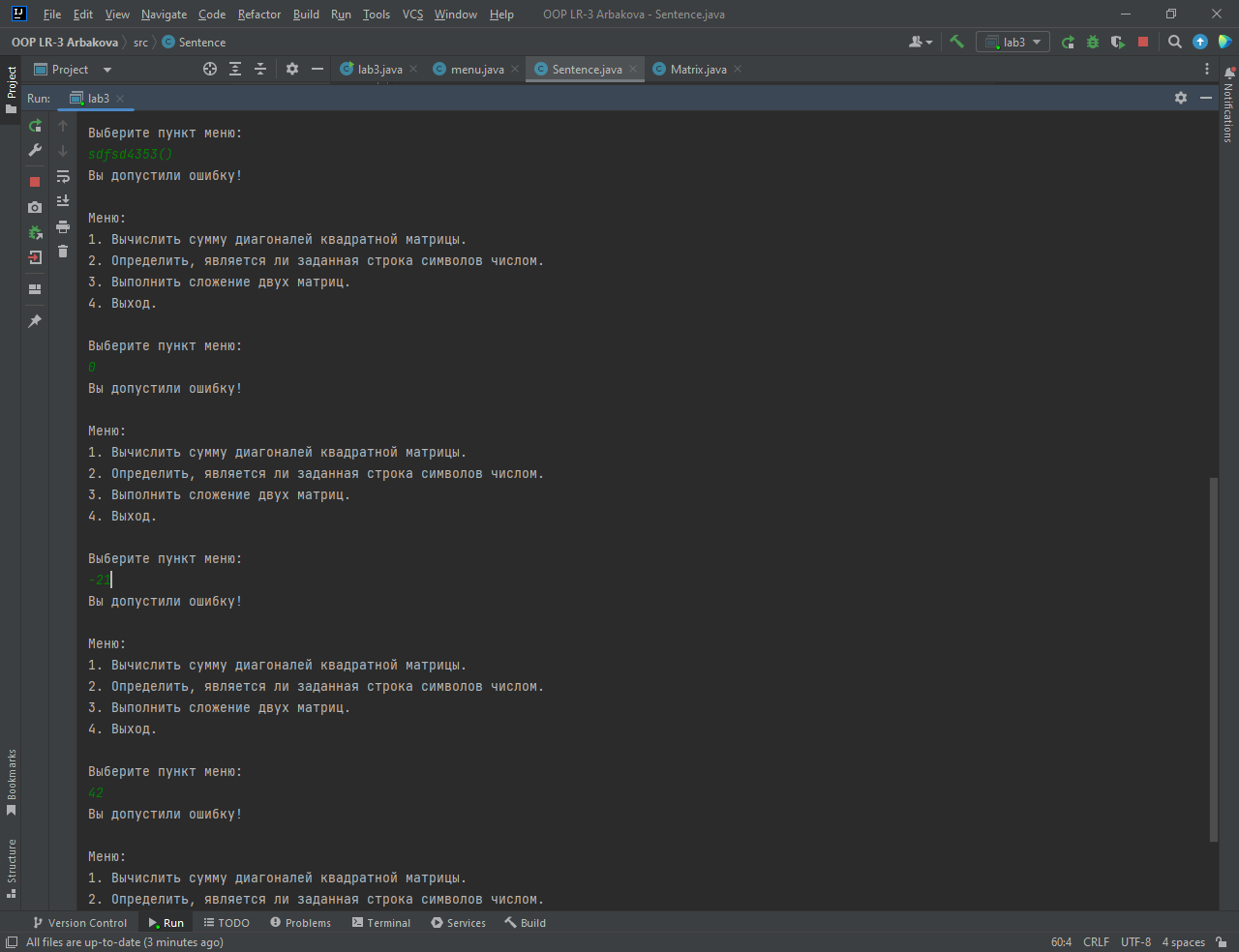
1. **Таблица тестов**

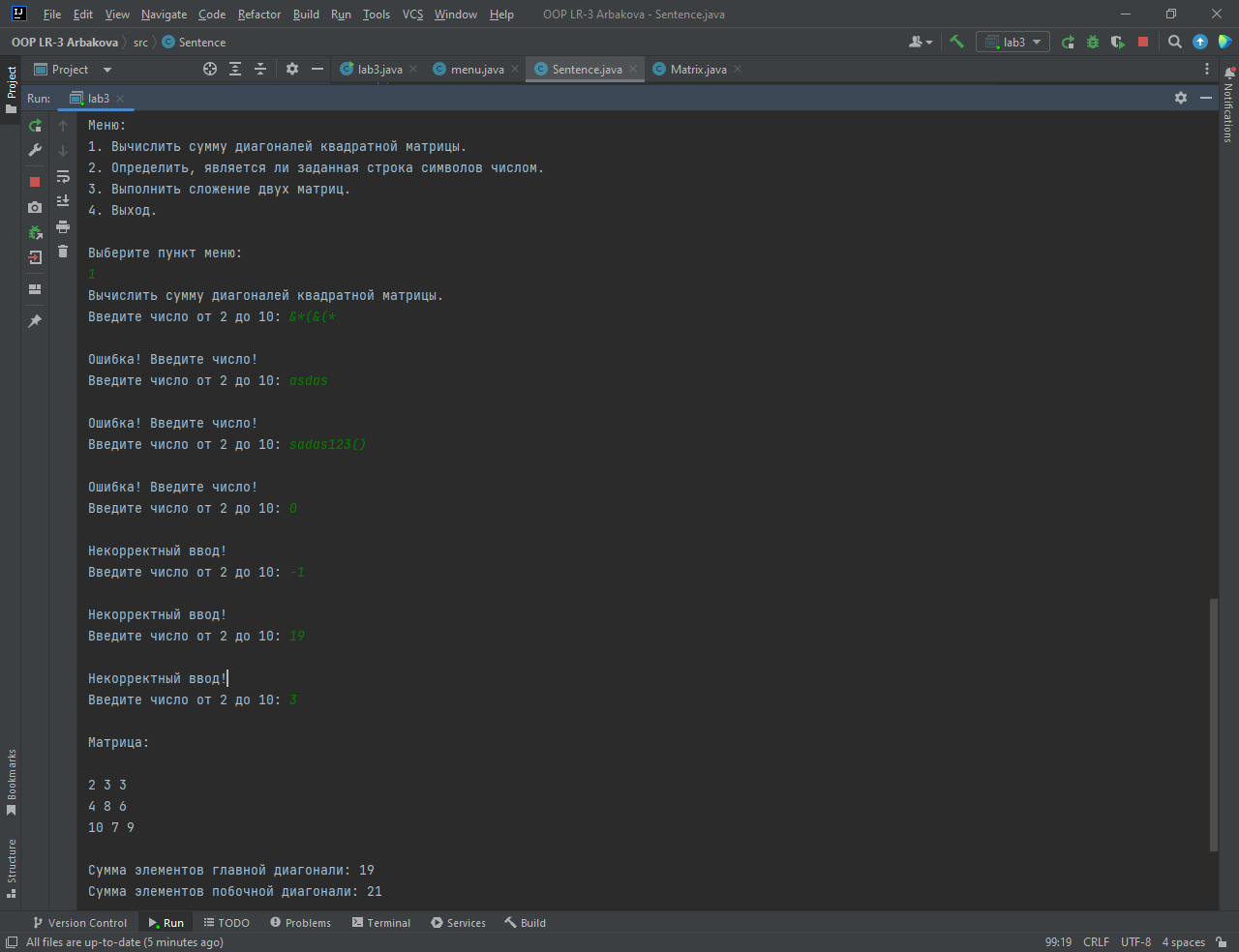
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Исходные данные** | | | | | |
| **1 задание** | | **2 задание** | **3 задание** | | |
| **size** | **A[][]** | **text** | **size** | **A[][]** | **B[][]** |
| 3 | 2 3 3  4 8 6  10 7 9 | 23647863  sfjkhj124 | 4 | 9 8 6 9  2 4 3 7  6 5 1 6  4 3 6 10 | 10 9 9 5  1 2 7 9  8 2 6 8  2 1 5 8 |

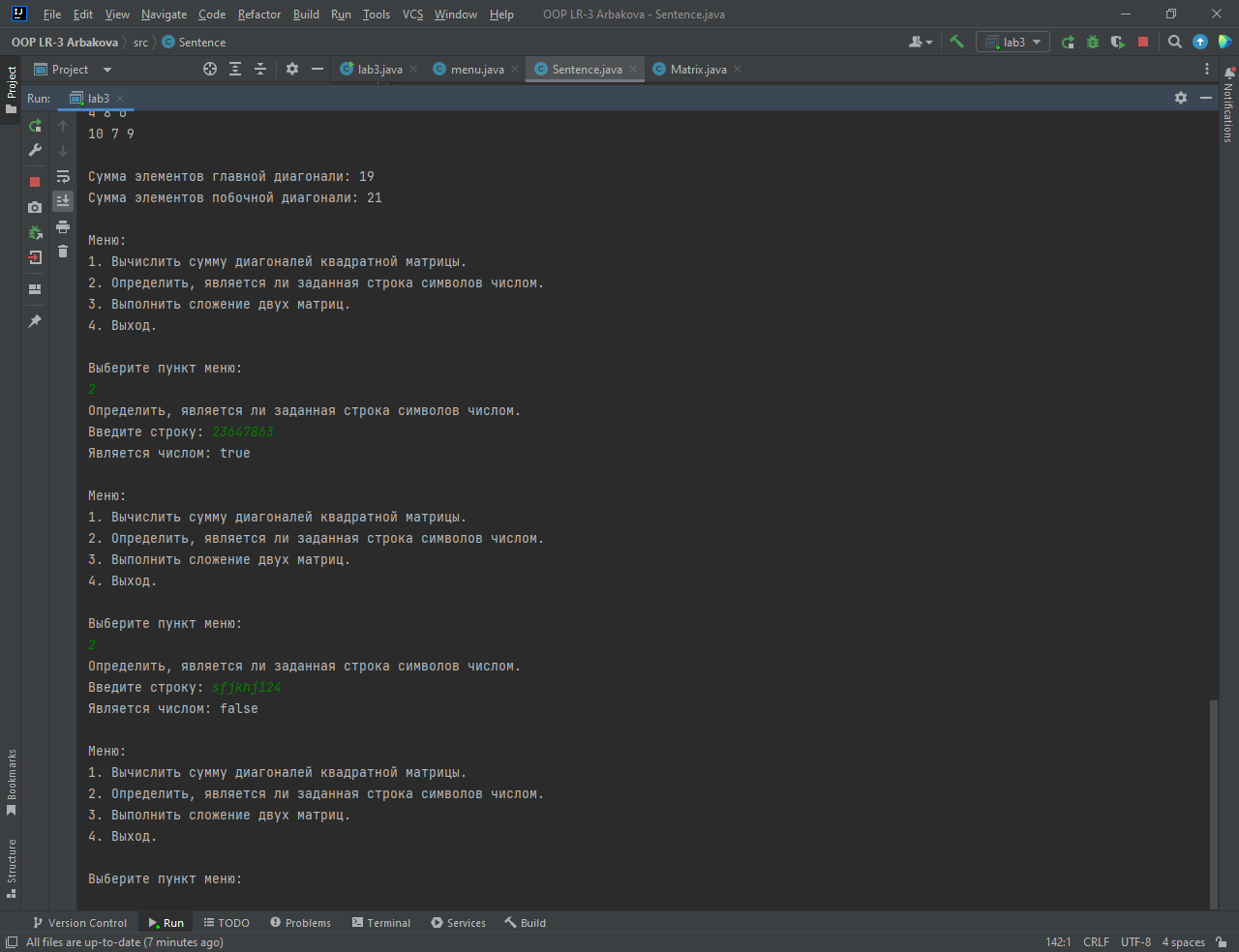
|  |  |
| --- | --- |
| **Результат** | **Примечания** |
| Меню:  1. Вычислить сумму диагоналей квадратной матрицы.  2. Определить, является ли заданная строка символов числом.  3. Выполнить сложение двух матриц.  4. Выход.  Выберите пункт меню:  \*(\*  Вы допустили ошибку!  Выберите пункт меню:  ывсы  Вы допустили ошибку!  Выберите пункт меню:  dsfdsf  Вы допустили ошибку!  Выберите пункт меню:  sdfsd4353()  Вы допустили ошибку!  Выберите пункт меню:  0  Вы допустили ошибку!  Выберите пункт меню:  -21  Вы допустили ошибку!  Выберите пункт меню:  42  Вы допустили ошибку!  Меню:  1. Вычислить сумму диагоналей квадратной матрицы.  2. Определить, является ли заданная строка символов числом.  3. Выполнить сложение двух матриц.  4. Выход.  Выберите пункт меню:  1 | 1. Проверка на некорректный ввод:  * Символы * Буквы * Сочетание букв, цифр и символов * Числа вне заданного диапазона  1. Корректный ввод |
| Вычислить сумму диагоналей квадратной матрицы.  Введите число от 2 до 10: &\*(&(\*  Ошибка! Введите число!  Введите число от 2 до 10: asdas  Ошибка! Введите число!  Введите число от 2 до 10: sadas123()  Ошибка! Введите число!  Введите число от 2 до 10: 0  Некорректный ввод!  Введите число от 2 до 10: -1  Некорректный ввод!  Введите число от 2 до 10: 19  Некорректный ввод!  Введите число от 2 до 10: 3  Матрица:  2 3 3  4 8 6  10 7 9  Сумма элементов главной диагонали: 19  Сумма элементов побочной диагонали: 21  Меню:  1. Вычислить сумму диагоналей квадратной матрицы.  2. Определить, является ли заданная строка символов числом.  3. Выполнить сложение двух матриц.  4. Выход. | 1. Проверка на некорректный ввод:  * Символы * Буквы * Сочетание букв, цифр и символов * Числа вне заданного диапазона  1. Корректный ввод |
| Выберите пункт меню:  2  Определить, является ли заданная строка символов числом.  Введите строку: 23647863  Является числом: true  Выберите пункт меню:  2  Определить, является ли заданная строка символов числом.  Введите строку: sfjkhj124  Является числом: false | 1. Введена строка, являющаяся числом 2. Введена строка, не являющаяся числом |
| Выберите пункт меню:  3  Выполнить сложение двух матриц.  Введите число от 2 до 10: \*(\*)  Ошибка! Введите число!  Введите число от 2 до 10: sdas  Ошибка! Введите число!  Введите число от 2 до 10: sdfsf324\*(  Ошибка! Введите число!  Введите число от 2 до 10: 0  Некорректный ввод!  Введите число от 2 до 10: -12  Некорректный ввод!  Введите число от 2 до 10: 12  Некорректный ввод!  Введите число от 2 до 10: 4  Матрица А:  9 8 6 9  2 4 3 7  6 5 1 6  4 3 6 10  Матрица В:  10 9 9 5  1 2 7 9  8 2 6 8  2 1 5 8  Сумма матриц:  19 17 15 14  3 6 10 16  14 7 7 14  6 4 11 18 | 1. Проверка на некорректный ввод:  * Символы * Буквы * Сочетание букв, цифр и символов * Числа вне заданного диапазона  1. Корректный ввод |
| Меню:  1. Вычислить сумму диагоналей квадратной матрицы.  2. Определить, является ли заданная строка символов числом.  3. Выполнить сложение двух матриц.  4. Выход.  Выберите пункт меню:  4  Выполнен выход из программы. | 1. Выход из программы |

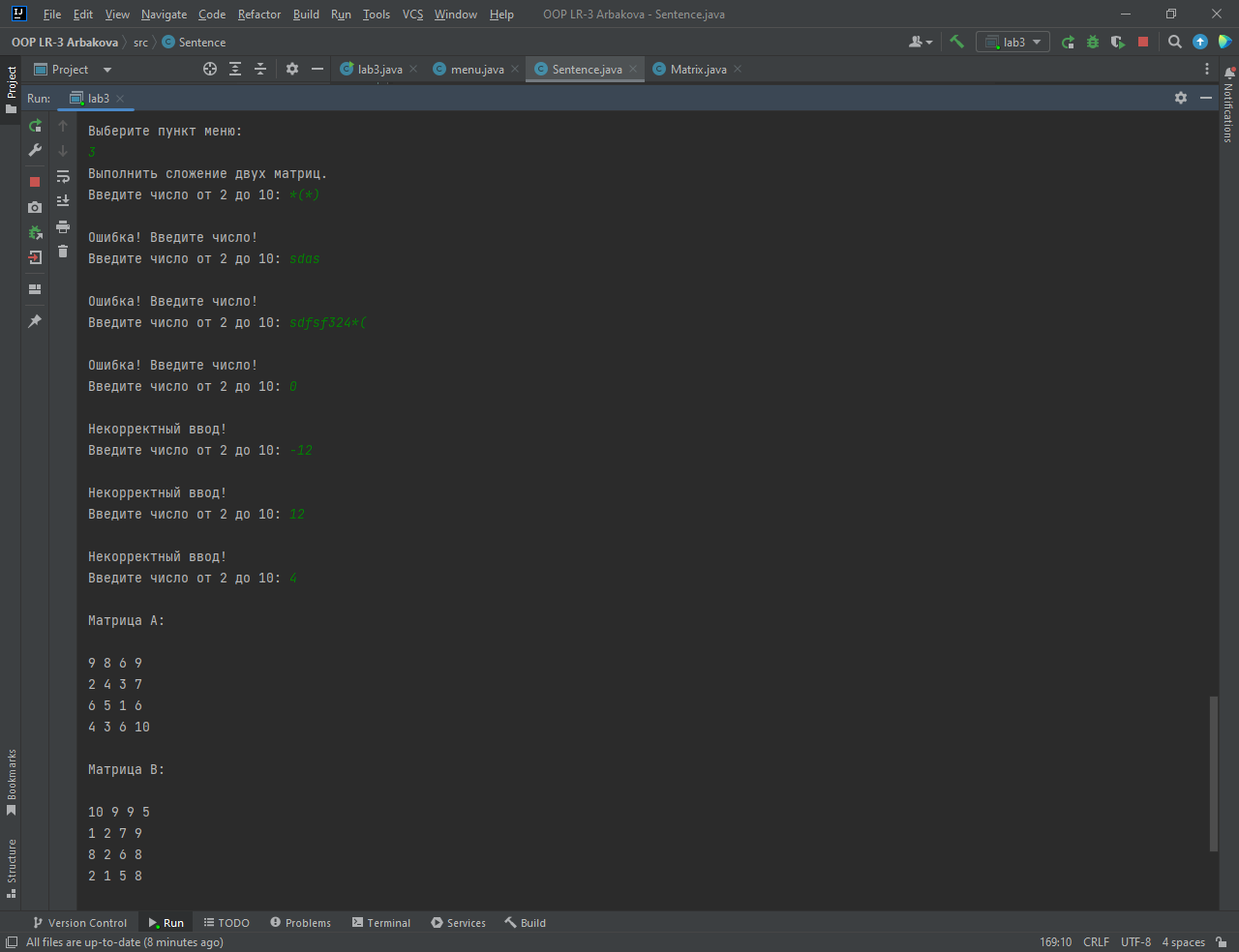
1. **Результат тестирования**

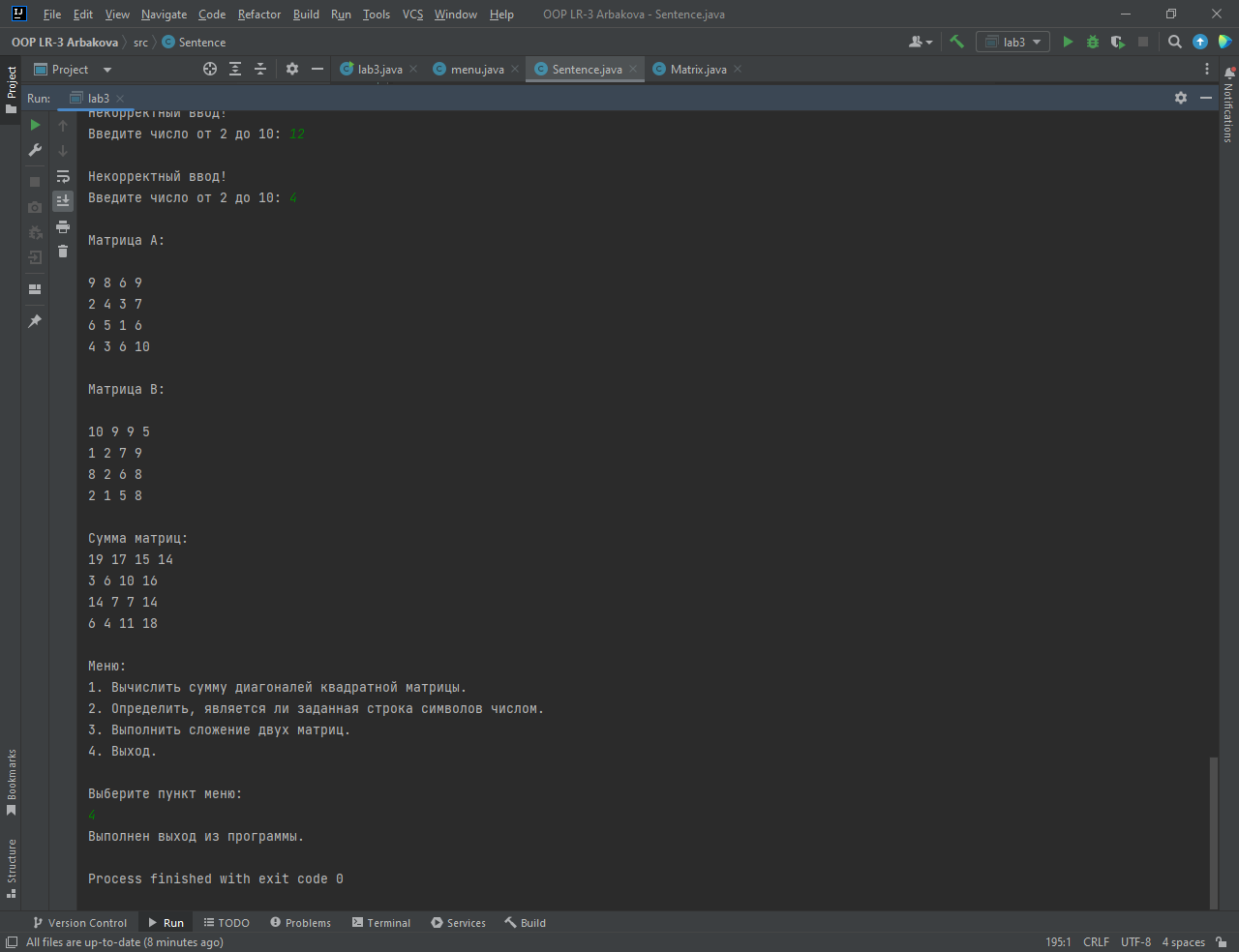












1. **Исходный код**

Класс lab3:

public class lab3 {  
 public static void main(String[] args) {  
 menu.*menu*(); //вызов меню  
 }  
}

Класс menu:

import java.util.Scanner;  
public class menu {  
 public static void menu() {  
 StringBuilder builder = new StringBuilder();  
 builder.append("\nМеню:\n") //пункты меню  
 .append("1. Вычислить сумму диагоналей квадратной матрицы.\n")  
 .append("2. Определить, является ли заданная строка символов числом.\n")  
 .append("3. Выполнить сложение двух матриц.\n")  
 .append("4. Выход.\n");  
 boolean isWork = true;  
  
 while (isWork) {  
 System.*out*.println(builder.toString()); //вывод меню на экран  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Выберите пункт меню: \n"); String i = scanner.nextLine();  
 switch (i) { //выбор пункта меню из диапазона 1-4  
 case "1":  
 System.*out*.print("Вычислить сумму диагоналей квадратной матрицы.\n"); //1 задание  
 int input = Matrix.*inputNumber*( 2, 10); //ввод числа - размер квадратной матрицы  
 Matrix matrix = new Matrix(input); //создание объекта - матрица для задания 1  
 matrix.sumDiagonals(); //вычисление сумм диагноалей  
 break;  
 case "2":  
 System.*out*.print("Определить, является ли заданная строка символов числом.\nВведите строку: "); //2 задание  
 Sentence sentence = new Sentence(scanner.nextLine()); //ввод строки  
 System.*out*.println("Является числом: " + sentence.isNumeric()); //выводится сообщение и проверяется является ли строка числом  
 break;  
 case "3":  
 System.*out*.print("Выполнить сложение двух матриц.\n"); //3 задание  
 input = Matrix.*inputNumber*( 2, 10); //ввод числа - размер квадратных матриц  
 Matrix A = new Matrix(input); //создание объекта - матрица А для задания 3  
 Matrix B = new Matrix(input); //создание объекта - матрица B для задания 3  
 A.sumMatrix(B.matrix); //вывод суммы матриц А и В  
 break;  
 case "4":  
 System.*out*.print("Выполнен выход из программы.\n"); //выход из программы  
 isWork = false;  
 break;  
 default:  
 System.*out*.println("Вы допустили ошибку!");  
 }  
 }  
 }  
}

Класс Matrix:

import java.util.Scanner;  
public class Matrix {  
 int size; //размер матрицы  
 int[][] matrix; //переменная matrix  
 public Matrix(int size) { //конструктор класса  
 this.size = size;  
 this.matrix = getRandMatrix(size); //получение рандомной матрицы с указанным размером  
 }  
 public int[][] getRandMatrix(int size) { //функция - получения рандомной матрицы  
 int[][] matrix = new int[size][size];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 double k = Math.*random*() \* 10 + 1;  
 matrix[i][j] = (int) k;//изменение типов данных  
 }  
 } return matrix;  
 }  
 public void sumMatrix(int[][] matrixB) { //функция - сумма матриц - для 3 задания  
 int[][] sumMatrix = new int[size][size];  
 System.*out*.println("\nМатрица А:\n");  
 outputMatrix(this.matrix); //вызов функции - вывод матрицы А  
 System.*out*.println("\nМатрица В:\n");  
 outputMatrix(matrixB); //вызов функции - вывод матрицы В  
 System.*out*.print("\nСумма матриц:\n");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 sumMatrix[i][j] = matrix[i][j] + matrixB[i][j];  
 System.*out*.print(sumMatrix[i][j] + " ");  
 } System.*out*.print("\n");  
 }  
 }  
 public void sumDiagonals() { //функция - сумма диагоналей матрицы - для 1 задания  
 int sumMain = 0;  
 System.*out*.println("\nМатрица:\n");  
 outputMatrix(this.matrix); //вызов функции - вывод матрицы  
 System.*out*.print("\nСумма элементов главной диагонали: "); //сумма главной диагонали  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 if (i == j) {  
 sumMain = sumMain + this.matrix[i][j];  
 }  
 }  
 }  
 System.*out*.print(sumMain); //вывод суммы главной диагонали  
 int sumSide = 0;  
 System.*out*.print("\nСумма элементов побочной диагонали: "); //сумма побочной диагонали  
 for (int i = 0; i < size; i++) {  
 for (int j = size - 1; j >= 0; j--) {  
 if (i + j == size - 1) {  
 sumSide = sumSide + matrix[i][j];  
 }  
 }  
 }  
 System.*out*.print(sumSide + "\n"); //вывод суммы побочной диагонали  
 }  
 private void outputMatrix(int[][] matrix) { //функция вывода матрицы на экран  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 } System.*out*.print("\n");  
 }  
 }  
 public static int inputNumber(int leftGranitcha, int rightGranitcha) { //функция ввода размерности матрицы  
 String input = "";  
 int x = 0;  
 boolean isCorrect = true;  
 Scanner console = new Scanner(System.*in*);  
 do {  
 try {  
 System.*out*.print("Введите число от " + leftGranitcha + " до " + rightGranitcha + ": ");  
 input = console.nextLine();  
 x = Integer.*parseInt*(input);  
 if ((x>=leftGranitcha) && (x <= rightGranitcha)){  
 break;  
 } else { System.*out*.print("\nНекорректный ввод!\n");}  
 }catch(NumberFormatException e) {  
 isCorrect = false;  
 System.*out*.print("\nОшибка! Введите число!\n");  
 }  
 } while (true);  
 return x;  
 }  
}

Класс Sentence:

import java.util.Scanner;  
public class Sentence {  
 String sentence; //переменная типа строки класса Sentence  
 public Sentence(String text){ //начало конструктора класса Sentence на вход строка-текст  
 this.sentence = text; //присвоение переменной string sentence класса - значение text  
 }  
 Scanner console = new Scanner(System.*in*);  
 public boolean isNumeric() { // функция правда-ложь является ли числом  
 try {  
 Double.*parseDouble*(sentence); // попытка перевода строки в число  
 return true; // если получилось - правда  
 } catch (NumberFormatException e) {  
 return false; // не получилось  
 }  
 }  
}

1. **Список использованных источников**
2. Шилдт, Герберт Java 8. Руководство для начинающих / Герберт Шилдт. - М.: Вильямс, 2015. - 720 c.
3. Берд, Барри Java для чайников / Барри Берд. - М.: Диалектика / Вильямс, 2013. - 521 c.
4. Савитч, Уолтер Язык Java. Курс программирования / Уолтер Савитч. - М.: Вильямс, 2015. - 928 c.