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 № 2

по дисциплине:

|  |
| --- |
| **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ Программирование** |
| **«Объекты и классы»** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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](#_Toc134136796)

[2. Описание классов: поля и методы 3](#_Toc134136797)

[3. Таблица тестов 5](#_Toc134136798)

[4. Результаты тестирования 7](#_Toc134136799)

[5. Исходный код 8](#_Toc134136800)

[6. Список использованных источников 11](#_Toc134136801)

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. Создание объекта класса, решающего задание и вызов его методов должны осуществляться из главного класса. Ввод и вывод данных осуществляется в консоль, для передачи исходных данных в экземпляр класса, решающего задачу должны быть разработаны соответствующие методы.

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 классов: поля и методы**

Класс – шаблонная структура, которая позволяет описать в программе объект, его свойства и поведение (поля и методы). Объект – экземпляр класса.

Поля – переменные, которые объявлены непосредственно в блоке класса, используются для выражения состояния объекта этого класса. А операции, реализующие поведение объекта, выражаются с помощью процедур и функций, объявленных в теле класса называются методами.

На основе выше написанного, для задач из лабораторной работы №1 были выделены такие классы, как Matrix и Sentence, поля и методы которых описаны в таблице 1.

Таблица 1 – Таблицы классов, полей и методов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Поля** | **Методы** | **Параметры** |
| lab2 | - | main(String[] args) | String[] args – аргументы переданные через консоль |
| 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() | - |

Описание используемых переменных и функции представлены в таблицах 2 и 3.

Таблица 2 – 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 | - |

Таблица 3 – Спецификации функций

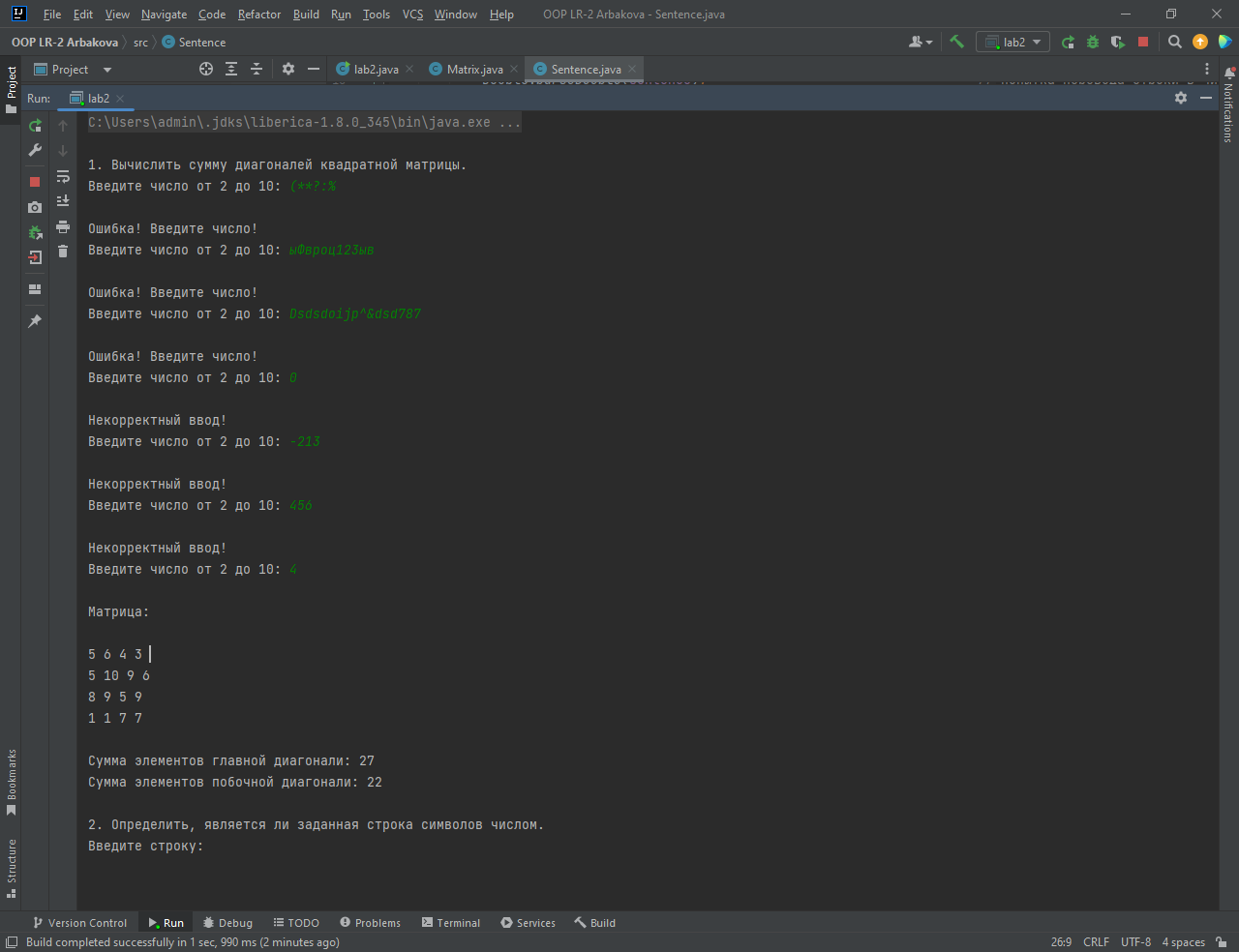
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название** | **Тип** | **Назначение** | **Параметры** |
| 1 | main | void | Главная функция приложения | String[] args |
| 2 | getRandMatrix | int[][] | Случайное заполнение матрицы значениями | int size |
| 3 | sumMatrix | void | Вычисление 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 | - |

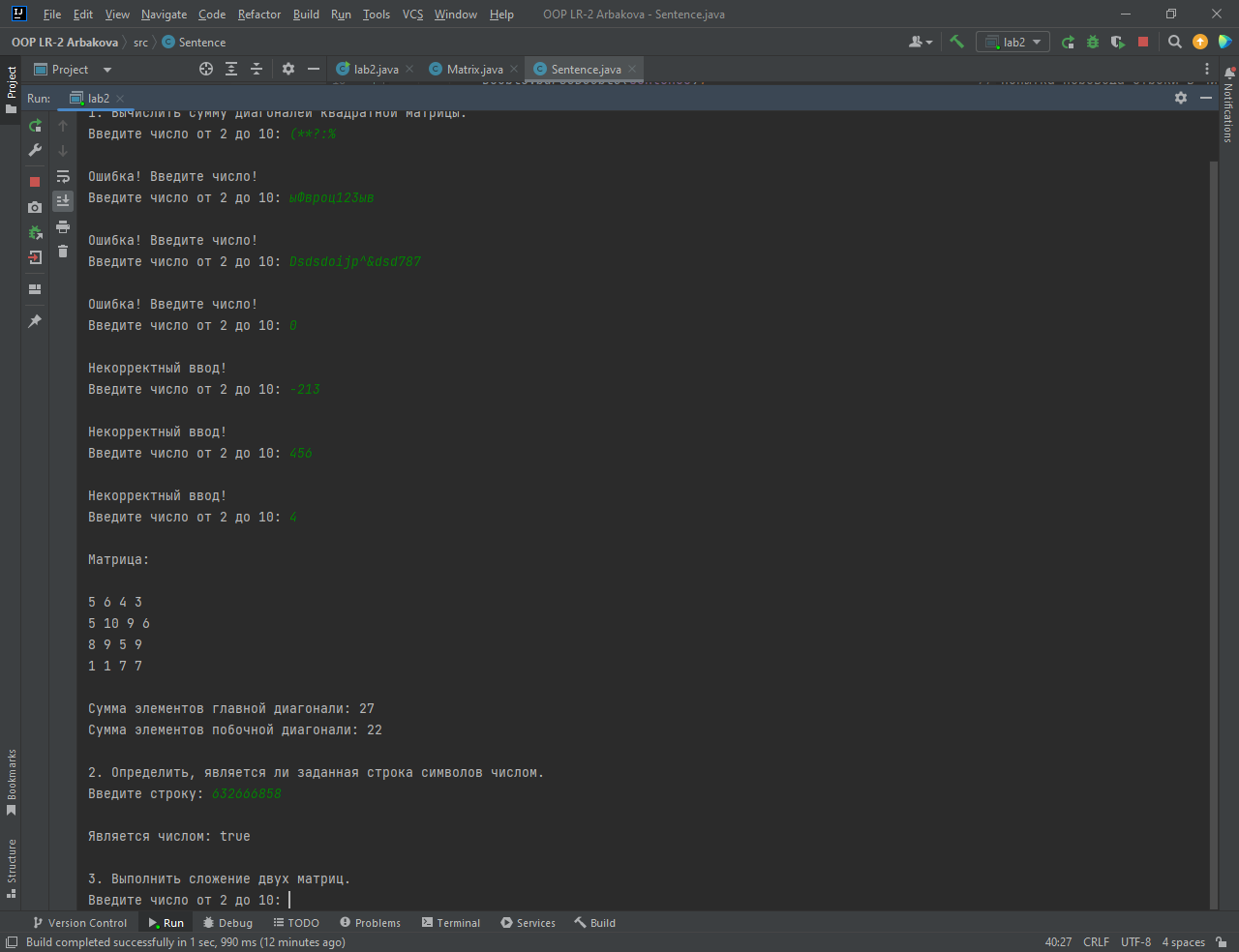
1. **Таблица тестов**

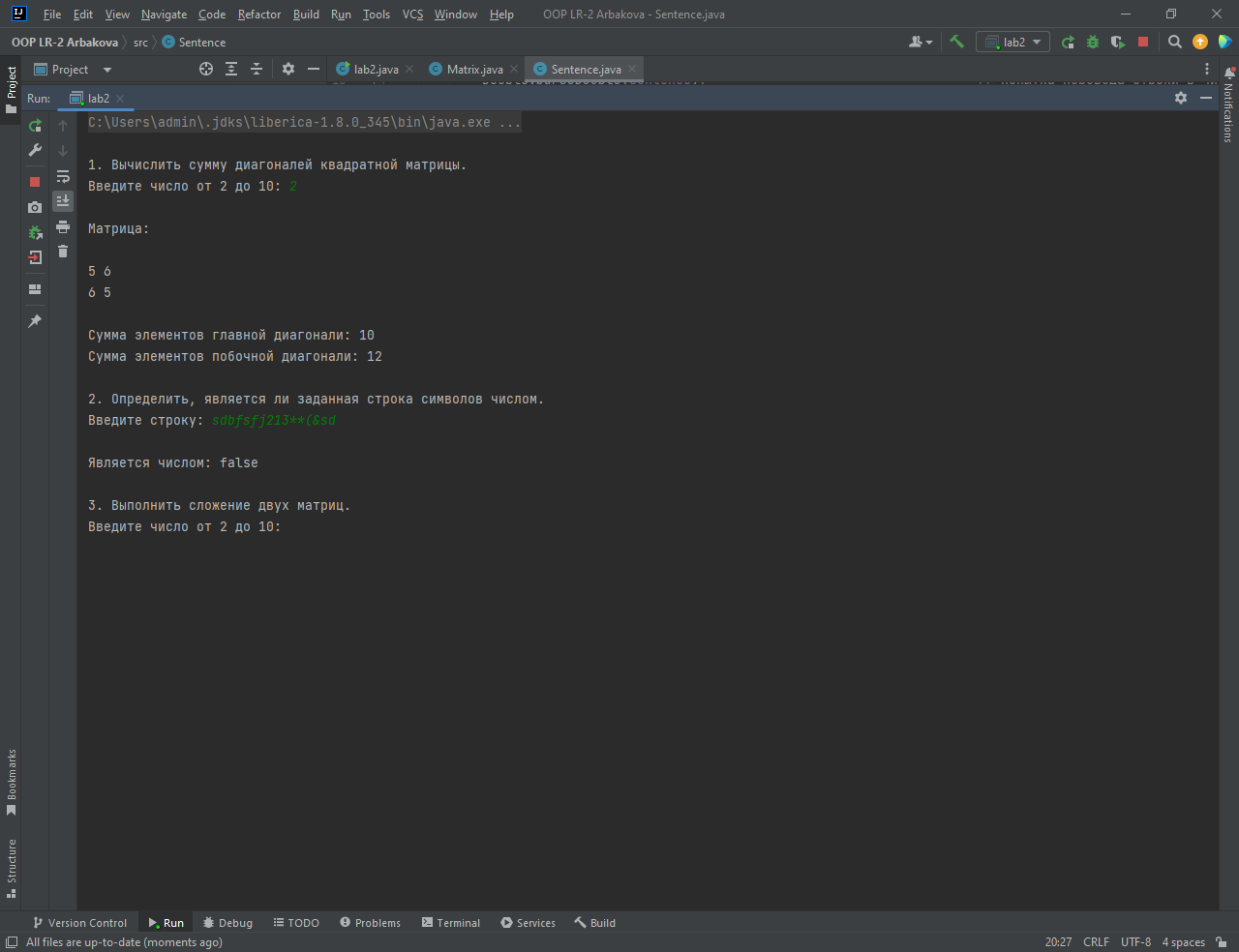
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Исходные данные** | | | | | |
| **1 задание** | | **2 задание** | **3 задание** | | |
| **size** | **A[][]** | **text** | **size** | **A[][]** | **B[][]** |
| 4 | 5 6 4 3  5 10 9 6  8 9 5 9  1 1 7 7 | 632666858  sdbfsfj213\*\*(&sd | 3 | 3 6 8  8 3 6  9 3 4 | 4 7 8  5 10 3  2 9 4 |

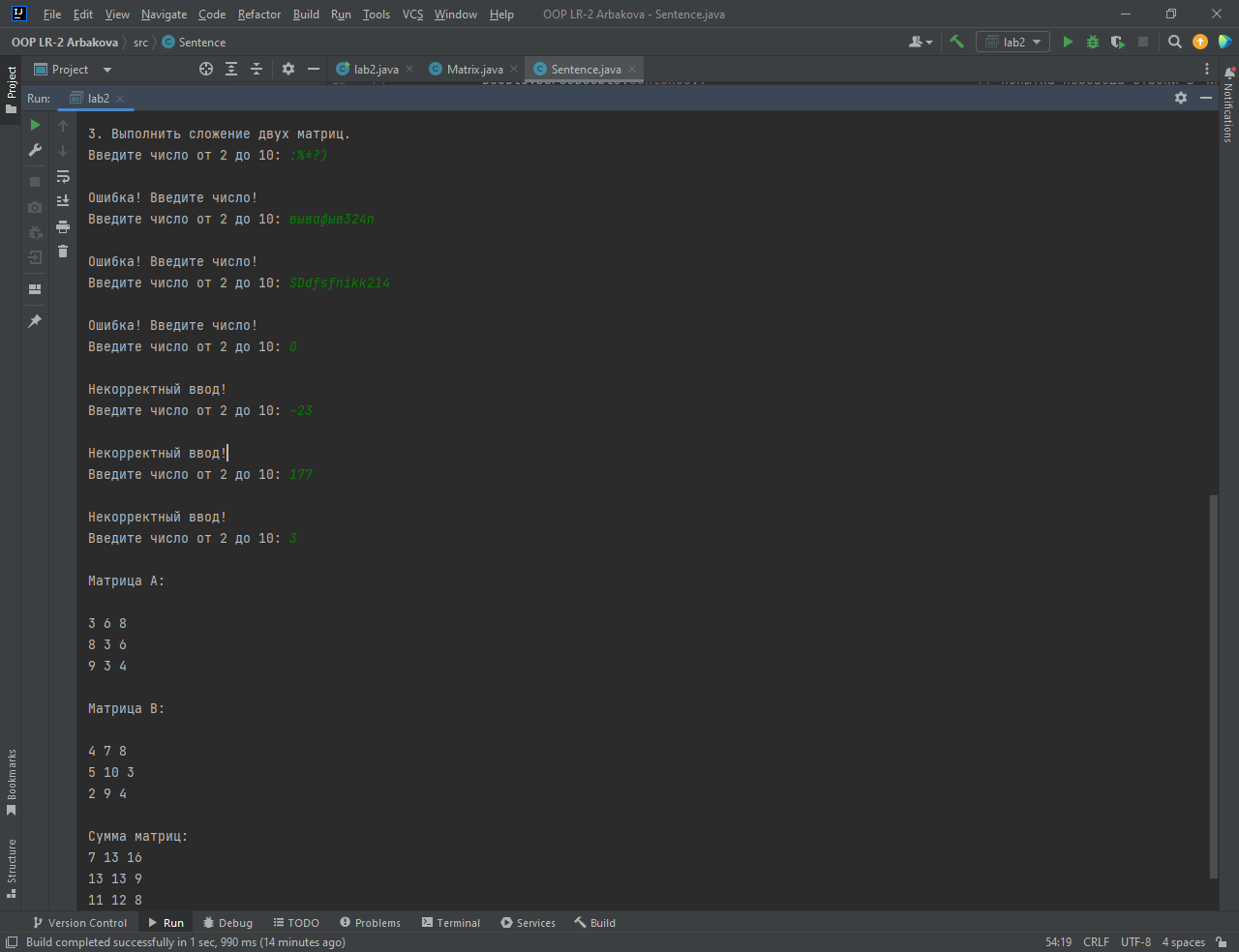
|  |  |
| --- | --- |
| **Результат** | **Примечания** |
| 1. Вычислить сумму диагоналей квадратной матрицы.  Введите число от 2 до 10: (\*\*?:%  Ошибка! Введите число!  Введите число от 2 до 10: ыФвроц123ыв  Ошибка! Введите число!  Введите число от 2 до 10: Dsdsdoijp^&dsd787  Ошибка! Введите число!  Введите число от 2 до 10: 0  Некорректный ввод!  Введите число от 2 до 10: -213  Некорректный ввод!  Введите число от 2 до 10: 456  Некорректный ввод!  Введите число от 2 до 10: 4  Матрица:  5 6 4 3  5 10 9 6  8 9 5 9  1 1 7 7  Сумма элементов главной диагонали: 27  Сумма элементов побочной диагонали: 22 | 1. Проверка на некорректный ввод:  * Символы * Буквы * Сочетание букв, цифр и символов * Числа вне заданного диапазона  1. Корректный ввод |
| 2. Определить, является ли заданная строка символов числом.  Введите строку: 632666858  Является числом: true  2. Определить, является ли заданная строка символов числом.  Введите строку: sdbfsfj213\*\*(&sd  Является числом: false | 1. Введена строка, являющаяся числом 2. Введена строка, не являющаяся числом |
| 3. Выполнить сложение двух матриц.  Введите число от 2 до 10: :%\*?)  Ошибка! Введите число!  Введите число от 2 до 10: вывафыв324п  Ошибка! Введите число!  Введите число от 2 до 10: SDdfsfnikk214  Ошибка! Введите число!  Введите число от 2 до 10: 0  Некорректный ввод!  Введите число от 2 до 10: -23  Некорректный ввод!  Введите число от 2 до 10: 177  Некорректный ввод!  Введите число от 2 до 10: 3  Матрица А:  3 6 8  8 3 6  9 3 4  Матрица В:  4 7 8  5 10 3  2 9 4  Сумма матриц:  7 13 16  13 13 9  11 12 8 | 1. Проверка на некорректный ввод:  * Символы * Буквы * Сочетание букв, цифр и символов * Числа вне заданного диапазона  1. Корректный ввод |

1. **Результаты тестирования**









1. **Исходный код**

Класс lab2:

import java.util.Scanner;  
public class lab2 {  
 public static void main(String[] args) {  
 System.*out*.print("\n1. 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  
 System.*out*.print("\n2. Определить, является ли заданная строка символов числом.\nВведите строку: "); //2 задание  
 Sentence sentence = new Sentence(new Scanner(System.*in*).nextLine()); //ввод строки  
 System.*out*.println("\nЯвляется числом: " + sentence.isNumeric()); //выводится сообщение и проверяется является ли строка числом  
 System.*out*.print("\n3. 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  
 }  
}

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.