FileManager.java

```java
/**
 * Класс для файла
 */
public class FileManager {
    /**
     * Конструктор класса
     * @param args Путь к файлу
     */
    public FileManager(String[] args){
        // Инициализировать путь к файлу
    }

    /**
     * Чтение матрицу из файла.
     * @return Матрица
     */
    public Float[][] readFile(){
        //Читать числа из файла, и сохранить в массив, возвращать тот массив.
    }

    /**
     * Запись матрицы в файл
     * @param matrix, который нужно сохранить в файл.
     */
    public void writeFile(Float[][] matrix){
        // Записывать результат в файл.
    }
}
```

Matrix.java

```java
/**
 * Класс матрицы
 */
public class Matrix {
    /**
     * Храниние матрицы
     */
    private final Float[][] matrix;
    /**
     * Размер матрицы
     */
    private final int size;
    /**
     * Конструктор класса
     * @param matrix, которые получить из файла
     */
    public Matrix(Float [][] matrix){
        // Инициализировать матрицу и ее размер
    }
    /**
     * Хронение результат
     */
    private final Float [] x = new Float[size];
```

```java
    /**
     * Вычислить результат
     */
    public void solve(){
        // Нужно использовать вложенний цикл. Цикл продалжается от оканчание
матрицы до начало.
            // Найти последный элемент, т.е X_n. А потом его умножить на
соответствующее число вернной строки и сохранить.
            // Вычетить последный элемент строки от предыдущего полученного
результата и делить на текущий результат
            // Найти результат и сохранить его в отдельном массиве.
    }
    /**
     * Приведение матрицы к трегулольному виду
     */
    public void transform(){
        // Нужно использовать вложенний цикл. Цикл продалжается до приведения
матрицы к трегольному виду
            // Вычислить коэффициент с помощью числа, которые расположены на одной
столбце.
            // Сложить две строки.
            // Распечатать результат после каждого шага.
    }
    /**
     * Сложение две строки.
     * @param row_1 Первое слагаемое
     * @param row_2 Второе слагаемое
     */
    private void addEquation(int row_1, int row_2){
        // Сложить две строки и сохранить в месте 2 строки.
    }
    /**
     * Умжонить строку на коэффициент
     * @param coefficient коэффициент
     * @param row Строка
     */
    private void multiple(Float coefficient, int row){
        // Умжонить все число в строке на коеффициент и сохранить в своем месте.
    }
    /**
     * Вычисление коэффициент
     * @param a Число, на которое должен умножить коэффициент.
     * @param b Число, который должен превратиться в нуль.
     * @return коэффициент, на которой нужно умжножить строку.
     */
    private Float findCoefficient(Float a, Float b){
        // Если число a равно 0, то возвраащает 0. возвращать значение -b/2.
    }
    /**
     * Метод, который провреят размер матрицу
     * @return Если размер матрицы не удовлетворил требование, возрващает false.
     */
    private boolean checkSystem(){
        // Если размер меньше 2 или разность между размерами строки и столбци
больше чем 1 возвращает false
    }
    /**
     * Вычисление размер матрицы
```

```
     * @return Размер матрицы
     */
    private int size(){
    }
    /**
     * Распечатать матрицу
     */
    public void printSystem(){
    }
    /**
     * Распечатать результат
     */
    public void printResult(){
    }
}
```

App.java

```
/**
 * Main class
 */
public class App {
    /**
     * main метод
     * @param args путь к файлу.
     */
    public static void main(String[] args) {
        FileManager fileManager = new FileManager(args);
        Matrix matrix = new Matrix(fileManager.readFile());
        matrix.printSystem();
        matrix.transform();
        matrix.solve();
        matrix.printResult();
    }
}
```