МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1

Разработка библиотеки на языке программирования Java

«Технологии разработки программного обеспечения»

Выполнил студент группы ИВТм-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Орлов Д.Э./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Чистяков Г.А./

Киров 2021

1. Задание

Разработать библиотеку для работы с матрицами. Должны быть реализованы следующие методы:

* Решение СЛАУ методом LU-разложение;
* Решение СЛАУ методом Гаусса;
* Ранг исходной матрицы.

1. Листинг программы

Листинг программы приведен в приложении А.

1. Экранные формы программы

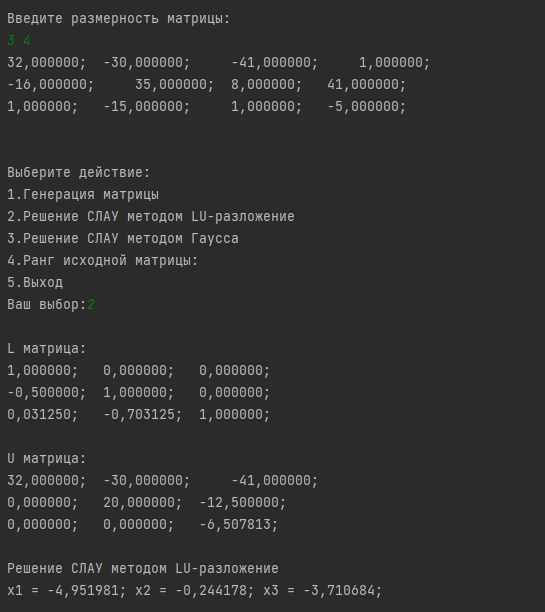


Рисунок 1 – Пример выполнения метода LU

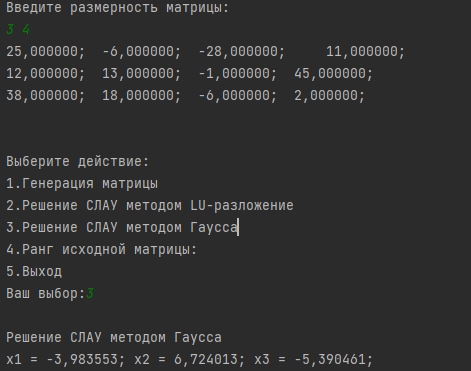


Рисунок 2 – Пример выполнения метода Gauss

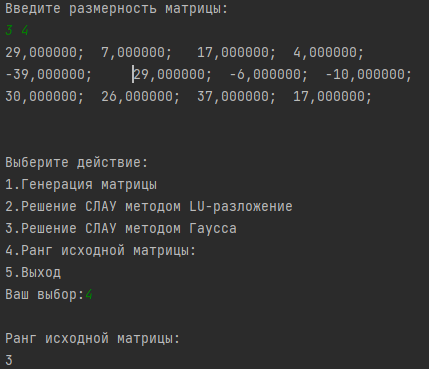


Рисунок 3 – Пример выполнения метода matrixRang

1. Вывод

В ходе выполнения лабораторной работы были изучены основные конструкции языка программирования Java, структура программы, стандартные средства ввода/вывода, работа с классами, а также создание библиотеки в виде переносимого jar архива; изучен основной функционал интегрированной среды разработки IntelliJ IDEA; написана библиотека для работы с массивами.

Приложение А

(обязательное)

Листинг программы

Main.java

import matrix.Gauss;  
import matrix.LU;  
import matrix.matrixRang;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class Main {  
  
  
 public static void main(String[] args) {  
 double[][] A = new double[0][0];  
 while (true) {  
 System.*out*.println();  
 System.*out*.print(  
 """  
 Выберите действие:  
 1.Генерация матрицы  
 2.Решение СЛАУ методом LU-разложение  
 3.Решение СЛАУ методом Гаусса  
 4.Ранг исходной матрицы:  
 5.Выход  
 Ваш выбор:"""  
 );  
 Scanner scanner = new Scanner(System.*in*);  
 int c = scanner.nextInt();  
 System.*out*.println();  
  
 switch (c) {  
 case 1:  
 try {  
 System.*out*.println("Введите размерность матрицы: ");  
 int n = scanner.nextInt();  
 int m = scanner.nextInt();  
 A = *generate*(n,m);  
 *printSystem*(A);  
 } catch (Throwable t) {  
 System.*out*.println(t);  
 }  
 break;  
 case 2:  
 if (A.length > 0 && A[0].length > 0)  
 {  
 try {  
 LU res = new LU(A);  
 System.*out*.println("L матрица:");  
 *printSystem*(res.getL());  
 System.*out*.println("U матрица:");  
 *printSystem*(res.getU());  
 System.*out*.println("Решение СЛАУ методом LU-разложение");  
 *printVector*(res.getX());  
 } catch (Throwable t) {  
 System.*out*.println(t);  
 }  
 } else System.*out*.println("Введите матрицу");  
  
 break;  
 case 3:  
 if (A.length > 0 && A[0].length > 0)  
 {  
 try{  
 Gauss gauss = new Gauss(A);  
 System.*out*.println("Решение СЛАУ методом Гаусса");  
 *printVector*(gauss.getX());  
 } catch (Throwable t) {  
 System.*out*.println(t);  
 }  
 } else System.*out*.println("Введите матрицу");  
 break;  
 case 4:  
 if (A.length > 0 && A[0].length > 0)  
 {  
 try{  
 System.*out*.println("Ранг исходной матрицы:");  
 matrixRang r = new matrixRang(A);  
 System.*out*.println(r.rang());  
 } catch (Throwable t) {  
 System.*out*.println(t);  
 }  
 } else System.*out*.println("Введите матрицу");  
 break;  
  
 case 5:  
 System.*exit*(0);  
 default:  
 System.*out*.println("Неизвестная команда");  
 break;  
 }  
  
  
 }  
 }  
  
 public static void printVector(double[] x) {  
 StringBuilder s = new StringBuilder();  
 for (int i = 0; i < x.length; i++) {  
 s.append(String.*format*("x%d = %f; ", i + 1, x[i]));  
 }  
 System.*out*.println(s);  
 }  
  
 public static double[][] generate(int size, int size1) {  
 if (size < 2 || size1 < 2)  
 throw new ArithmeticException("Некорректный размер матриц");  
 else {  
 double[][] fillerMatrix = new double[size][size1];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size1; j++) {  
 Random random = new Random();  
 fillerMatrix[i][j] = (double) (random.nextInt() % 50) + 1;  
 }  
 }  
 return fillerMatrix;  
 }  
  
 }  
  
 public static void printSystem(double[][] system){  
 for (double[] doubles : system) {  
 StringBuilder s = new StringBuilder();  
 for (int j = 0; j < system[0].length; j++) {  
 s.append(String.*format*("%f; %s", doubles[j], "\t"));  
 }  
 System.*out*.println(s);  
 }  
 System.*out*.println();  
 }  
}

Gauss.java

package matrix;  
  
public final class Gauss {  
 double[] X;  
  
 public Gauss(double[][] a) {  
 if (!*checkSystem*(a)) {  
 throw new ArithmeticException("Incorrect system for Gauss Method");  
 } else {  
 double[] x = new double[a.length];  
  
 for(int i = 0; i < x.length; ++i) {  
 x[i] = a[i][a[i].length - 1];  
 }  
  
 int k;  
 int j;  
 for(k = 1; k < a.length; ++k) {  
 for(j = k; j < a.length; ++j) {  
 double m = a[j][k - 1] / a[k - 1][k - 1];  
  
 for(int i = 0; i < a[j].length; ++i) {  
 a[j][i] -= m \* a[k - 1][i];  
 }  
  
 x[j] -= m \* x[k - 1];  
 }  
  
 this.X = x;  
 }  
  
 for(k = a.length - 1; k >= 0; --k) {  
 for(j = k + 1; j < a.length; ++j) {  
 x[k] -= a[k][j] \* x[j];  
 }  
  
 x[k] /= a[k][k];  
 }  
  
 }  
 }  
  
 public double[] getX() {  
 return this.X;  
 }  
  
 private static boolean checkSystem(double[][] system) {  
 return system.length >= 2 && system[0].length >= 2 && system.length < system[0].length;  
 }  
}

LU.java

package matrix;  
  
public final class LU {  
 private final double[][] L;  
 private final double[][] U;  
 private final double[] X;  
  
 public LU(double[][] A) {  
 if (!*checkSystem*(A)) {  
 throw new ArithmeticException("Incorrect system for LU Method");  
 } else {  
 double[][] L = new double[A.length][A.length];  
 double[][] U = new double[A.length][A.length];  
  
 int n;  
 for(n = 0; n < A.length; ++n) {  
 System.*arraycopy*(A[n], 0, U[n], 0, A.length);  
 }  
  
 n = A.length;  
  
 int k;  
 int i;  
 for(k = 0; k < n; ++k) {  
 for(i = k; i < n; ++i) {  
 L[i][k] = U[i][k] / U[k][k];  
 }  
 }  
  
 for(k = 1; k < n; ++k) {  
 int j;  
 for(i = k - 1; i < n; ++i) {  
 for(j = i; j < n; ++j) {  
 L[j][i] = U[j][i] / U[i][i];  
 }  
 }  
  
 for(i = k; i < n; ++i) {  
 for(j = k - 1; j < n; ++j) {  
 U[i][j] -= L[i][k - 1] \* U[k - 1][j];  
 }  
 }  
 }  
  
 double[] y = new double[n];  
 double[] x = new double[n];  
 double[] B = new double[n];  
  
 int i;  
 for(i = 0; i < A.length; ++i) {  
 B[i] = A[i][A[0].length - 1];  
 }  
  
 double sum;  
 int j;  
 for(i = 0; i < n; ++i) {  
 sum = 0.0D;  
  
 for(j = 0; j < i; ++j) {  
 sum += L[i][j] \* y[j];  
 }  
  
 y[j] = B[i] - sum;  
 }  
  
 for(i = n - 1; i >= 0; --i) {  
 sum = 0.0D;  
  
 for(j = i + 1; j < n; ++j) {  
 sum += U[i][j] \* x[j];  
 }  
  
 x[i] = (y[i] - sum) / U[i][i];  
 }  
  
 this.L = L;  
 this.U = U;  
 this.X = x;  
 }  
 }  
  
 public double[][] getU() {  
 return this.U;  
 }  
  
 public double[] getX() {  
 return this.X;  
 }  
  
 public double[][] getL() {  
 return this.L;  
 }  
  
 private static boolean checkSystem(double[][] system) {  
 if (system.length < 2) {  
 return false;  
 } else {  
 for(int i = 0; i < system.length; ++i) {  
 if (system[0].length != system.length + 1) {  
 return false;  
 }  
 }  
  
 return true;  
 }  
 }  
}

matrixRang.java

package matrix;  
  
public final class matrixRang {  
 int rang;  
  
 public matrixRang(double[][] A) {  
 int r = 0;  
  
 for(int q = 1; q <= Math.*min*(A.length, A[0].length); ++q) {  
 double[][] B = new double[q][q];  
  
 for(int a = 0; a < A.length - (q - 1); ++a) {  
 for(int b = 0; b < A[0].length - (q - 1); ++b) {  
 for(int c = 0; c < q; ++c) {  
 System.*arraycopy*(A[a + c], b, B[c], 0, q);  
 }  
  
 Determine dd = new Determine(B);  
 if (dd.getValue() != 0.0D) {  
 r = q;  
 }  
 }  
 }  
 }  
  
 this.rang = r;  
 }  
  
 public int rang() {  
 return this.rang;  
 }  
}