Государственное бюджетное образовательное учреждение высшего

профессионального образования

«Поволжский государственный университет телекоммуникаций

и информатики»

Кафедра ПрИ

Лабораторная работа №3 на тему

**«Решение плохо обусловленных систем линейных алгебраических уравнений»**

по дисциплине «Численные методы»

Вариант №6

**Выполнил**: студент группы ПРИ-22

Дубинкин Данила Юрьевич

Самара 2024

**1. Цель работы:**

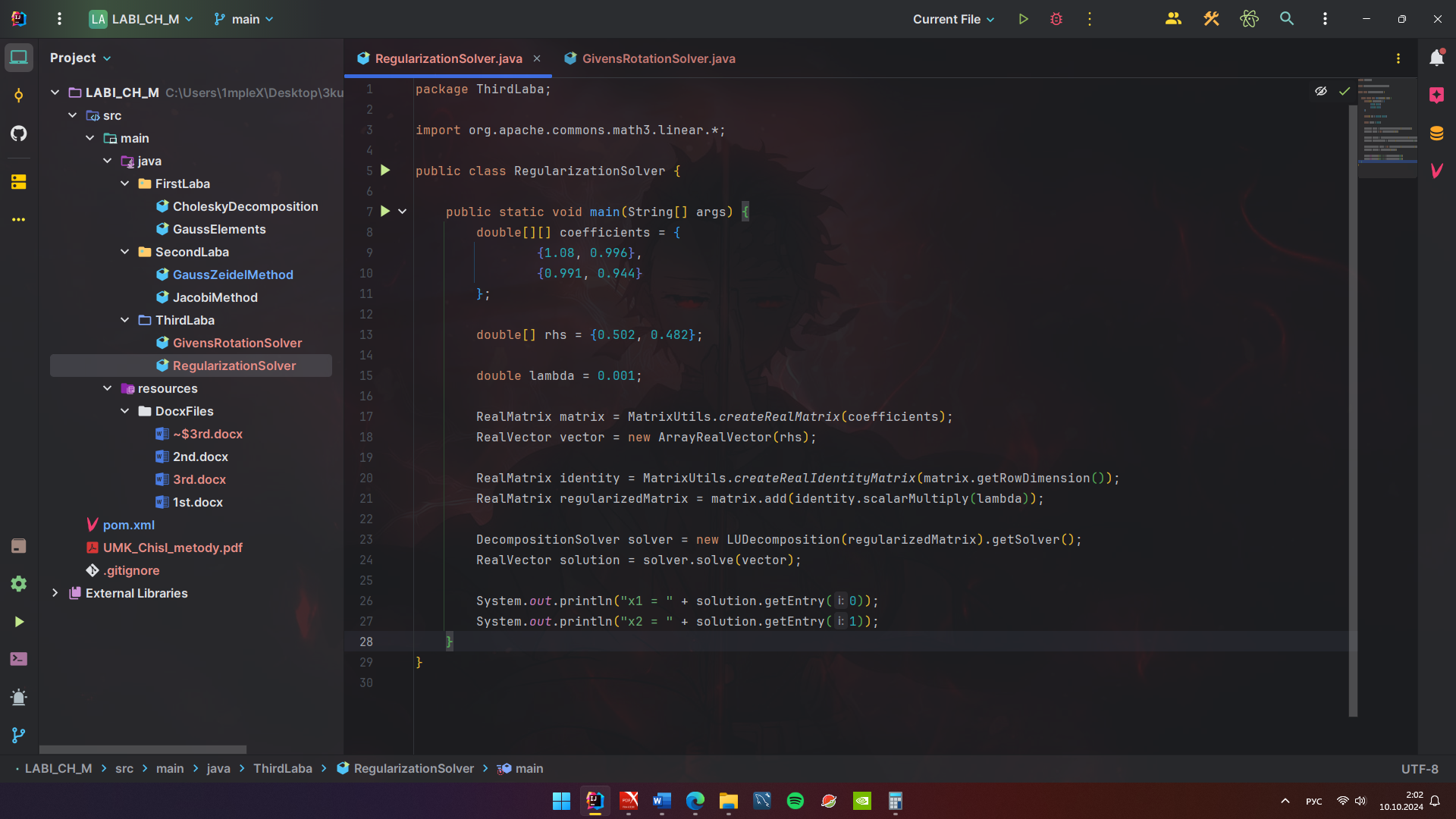
научится решить систему уравнений, методом Регуляризации Тихонова и вращением Гвиноса. Разобраться в их алгоритме.

**2. Исходные Данные:**

****

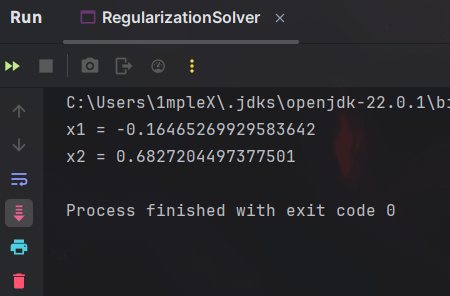
**3. Листинг программы:**

**3.1.1 Метод Регуляризации Тихонова:**



**package ThirdLaba;  
  
import org.apache.commons.math3.linear.\*;  
  
public class RegularizationSolver {  
  
 public static void main(String[] args) {  
 double[][] coefficients = {  
 {1.08, 0.996},  
 {0.991, 0.944}  
 };  
  
 double[] rhs = {0.502, 0.482};  
  
 double lambda = 0.001;  
  
 RealMatrix matrix = MatrixUtils.createRealMatrix(coefficients);  
 RealVector vector = new ArrayRealVector(rhs);  
  
 RealMatrix identity = MatrixUtils.createRealIdentityMatrix(matrix.getRowDimension());  
 RealMatrix regularizedMatrix = matrix.add(identity.scalarMultiply(lambda));  
  
 DecompositionSolver solver = new LUDecomposition(regularizedMatrix).getSolver();  
 RealVector solution = solver.solve(vector);  
  
 System.out.println("x1 = " + solution.getEntry(0));  
 System.out.println("x2 = " + solution.getEntry(1));  
 }  
}**

**3.1.2 Результат:**

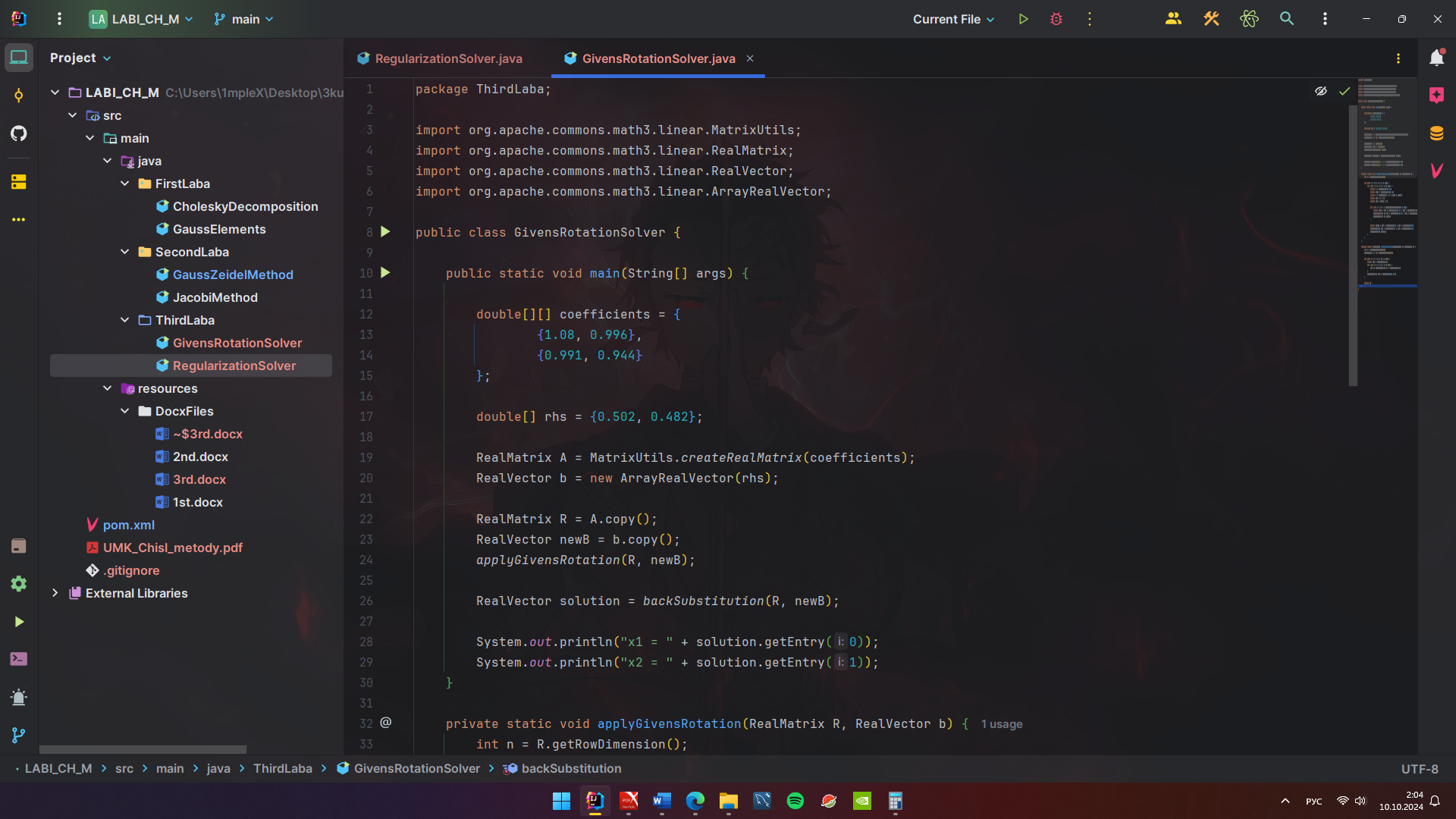


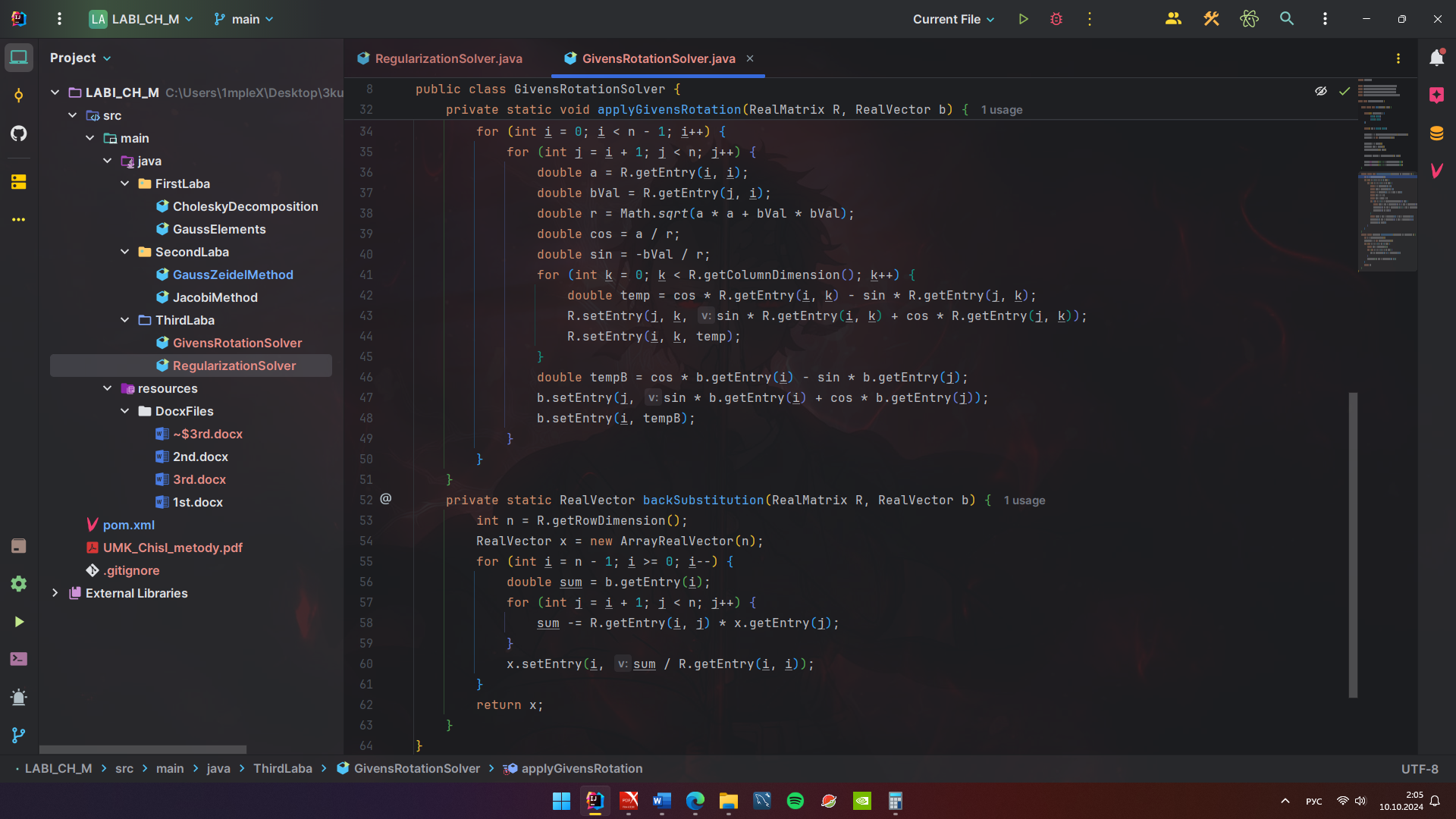
Результат:

x1 = -0.16465269929583642

x2 = 0.6827204497377501

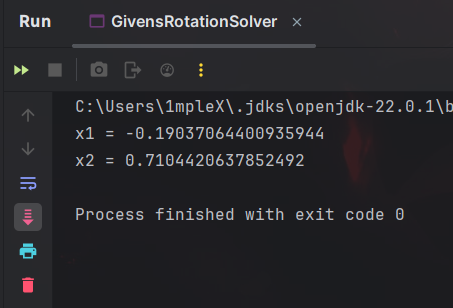
**3.2.1 Метод Вращения Гвиноса:**





**package ThirdLaba;  
  
import org.apache.commons.math3.linear.MatrixUtils;  
import org.apache.commons.math3.linear.RealMatrix;  
import org.apache.commons.math3.linear.RealVector;  
import org.apache.commons.math3.linear.ArrayRealVector;  
  
public class GivensRotationSolver {  
  
 public static void main(String[] args) {  
  
 double[][] coefficients = {  
 {1.08, 0.996},  
 {0.991, 0.944}  
 };  
  
 double[] rhs = {0.502, 0.482};  
  
 RealMatrix A = MatrixUtils.createRealMatrix(coefficients);  
 RealVector b = new ArrayRealVector(rhs);  
  
 RealMatrix R = A.copy();  
 RealVector newB = b.copy();  
 applyGivensRotation(R, newB);  
  
 RealVector solution = backSubstitution(R, newB);  
  
 System.out.println("x1 = " + solution.getEntry(0));  
 System.out.println("x2 = " + solution.getEntry(1));  
 }  
  
 private static void applyGivensRotation(RealMatrix R, RealVector b) {  
 int n = R.getRowDimension();  
 for (int i = 0; i < n - 1; i++) {  
 for (int j = i + 1; j < n; j++) {  
 double a = R.getEntry(i, i);  
 double bVal = R.getEntry(j, i);  
 double r = Math.sqrt(a \* a + bVal \* bVal);  
 double cos = a / r;  
 double sin = -bVal / r;  
 for (int k = 0; k < R.getColumnDimension(); k++) {  
 double temp = cos \* R.getEntry(i, k) - sin \* R.getEntry(j, k);  
 R.setEntry(j, k, sin \* R.getEntry(i, k) + cos \* R.getEntry(j, k));  
 R.setEntry(i, k, temp);  
 }  
 double tempB = cos \* b.getEntry(i) - sin \* b.getEntry(j);  
 b.setEntry(j, sin \* b.getEntry(i) + cos \* b.getEntry(j));  
 b.setEntry(i, tempB);  
 }  
 }  
 }  
 private static RealVector backSubstitution(RealMatrix R, RealVector b) {  
 int n = R.getRowDimension();  
 RealVector x = new ArrayRealVector(n);  
 for (int i = n - 1; i >= 0; i--) {  
 double sum = b.getEntry(i);  
 for (int j = i + 1; j < n; j++) {  
 sum -= R.getEntry(i, j) \* x.getEntry(j);  
 }  
 x.setEntry(i, sum / R.getEntry(i, i));  
 }  
 return x;  
 }  
}**

**3.2.2 Результат:**

****

**Результат:**

x1 = -0.19037064400935944

x2 = 0.7104420637852492