Государственное бюджетное образовательное учреждение высшего

профессионального образования

«Поволжский государственный университет телекоммуникаций

и информатики»

Кафедра ПрИ

Лабораторная работа №1 на тему

**«Решение систем линейных алгебраических уравнений. Точные методы»**

по дисциплине «Численные методы»

Вариант №6

**Выполнил**: студент группы ПРИ-22

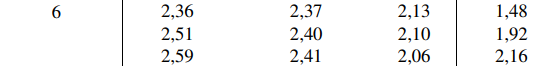
Дубинкин Данила Юрьевич

Самара 2024

**1. Цель работы:**

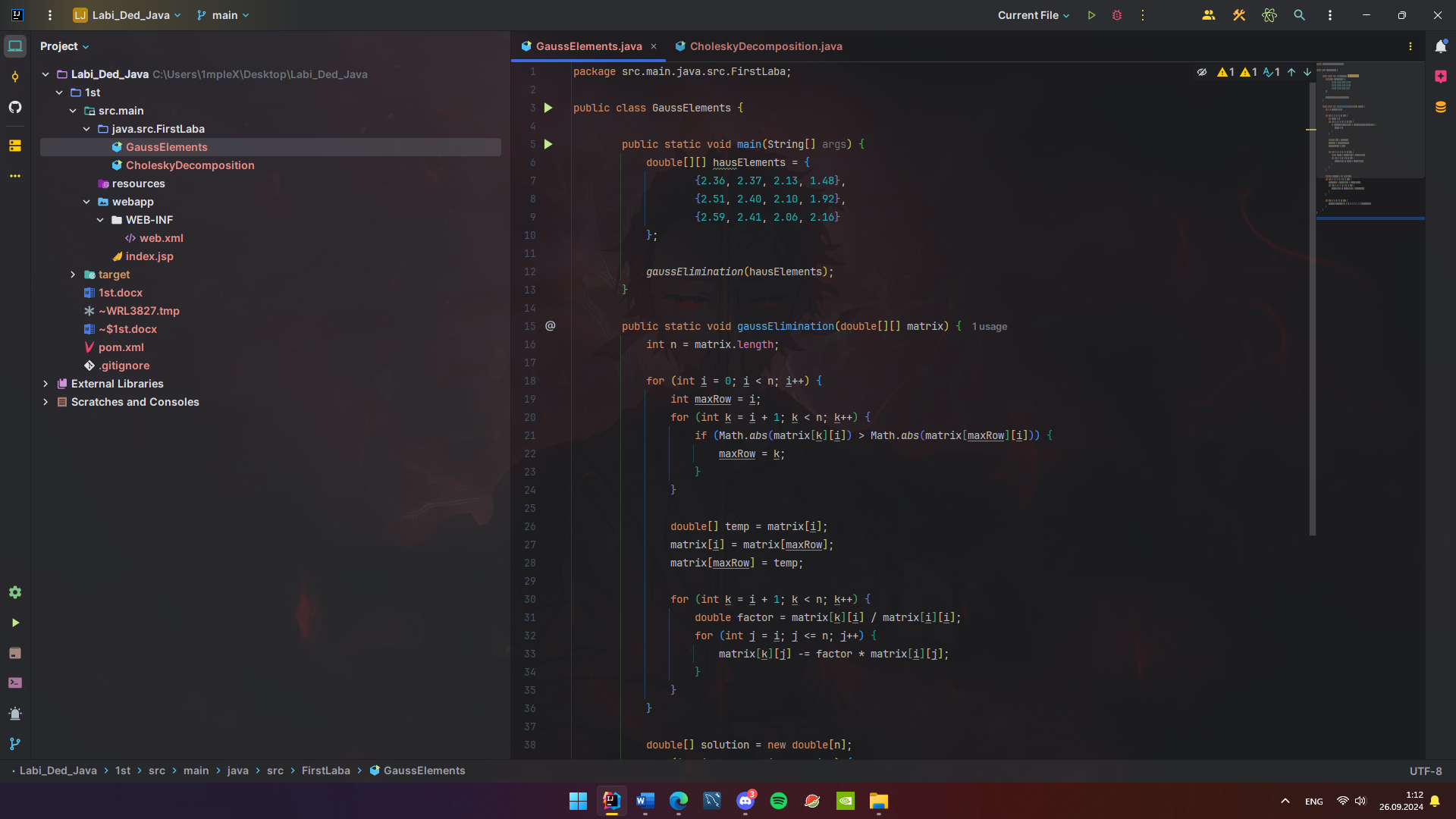
научится решить систему уравнений, методом Гаусса и Холецкого. Разобраться в их алгоритме.

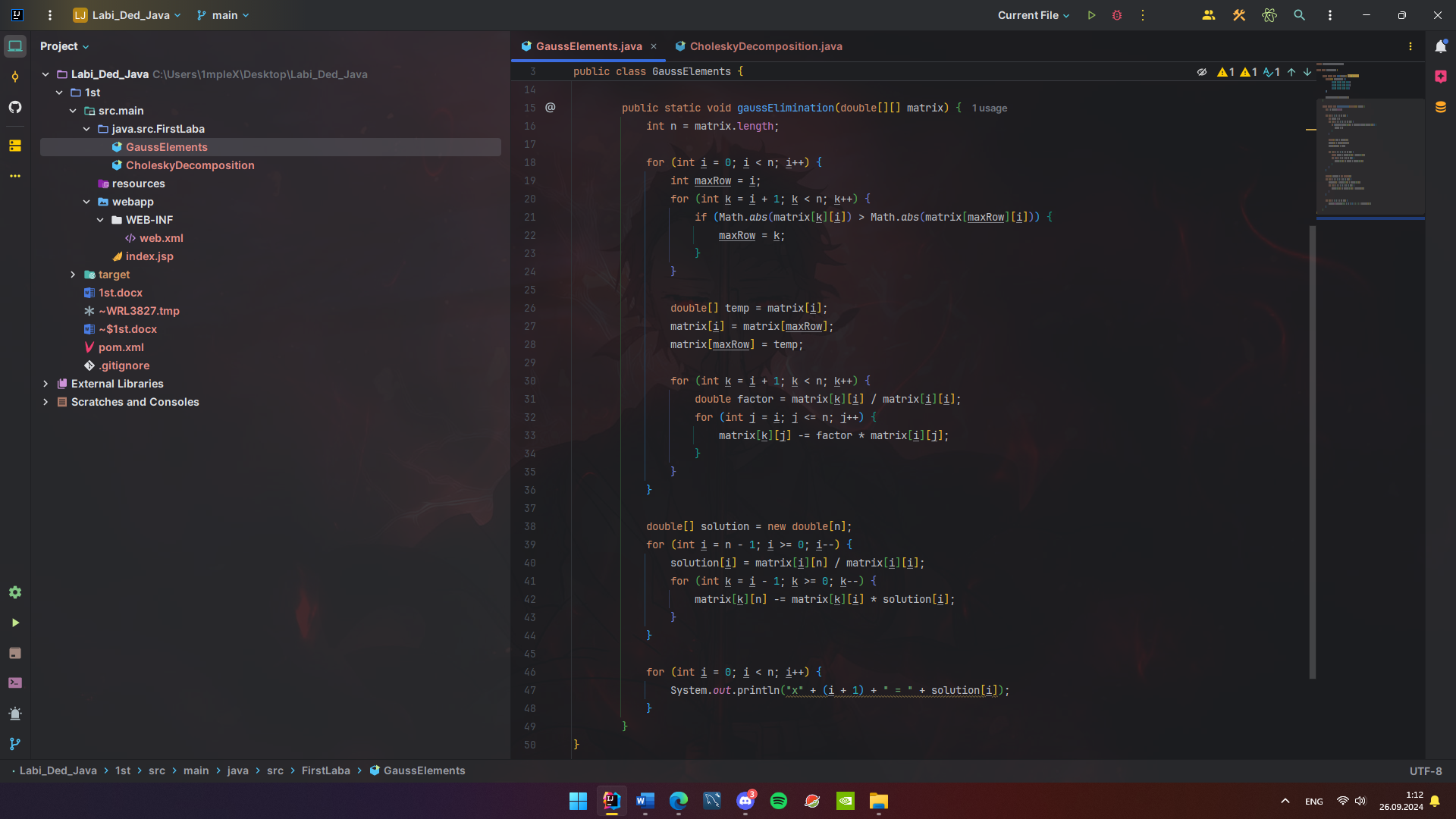
**2. Исходные Данные:**

****

**3. Листинг программы:**

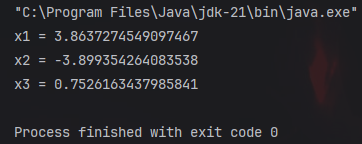
**3.1.1 Метод Гаусса:**





package src.main.java.src.FirstLaba;  
  
public class GaussElements {  
  
 public static void main(String[] args) {  
 double[][] hausElements = {  
 {2.36, 2.37, 2.13, 1.48},  
 {2.51, 2.40, 2.10, 1.92},  
 {2.59, 2.41, 2.06, 2.16}  
 };  
  
 *gaussElimination*(hausElements);  
 }  
  
 public static void gaussElimination(double[][] matrix) {  
 int n = matrix.length;  
  
 for (int i = 0; i < n; i++) {  
 int maxRow = i;  
 for (int k = i + 1; k < n; k++) {  
 if (Math.*abs*(matrix[k][i]) > Math.*abs*(matrix[maxRow][i])) {  
 maxRow = k;  
 }  
 }  
  
 double[] temp = matrix[i];  
 matrix[i] = matrix[maxRow];  
 matrix[maxRow] = temp;  
  
 for (int k = i + 1; k < n; k++) {  
 double factor = matrix[k][i] / matrix[i][i];  
 for (int j = i; j <= n; j++) {  
 matrix[k][j] -= factor \* matrix[i][j];  
 }  
 }  
 }  
 double[] solution = new double[n];  
 for (int i = n - 1; i >= 0; i--) {  
 solution[i] = matrix[i][n] / matrix[i][i];  
 for (int k = i - 1; k >= 0; k--) {  
 matrix[k][n] -= matrix[k][i] \* solution[i];  
 }  
 }  
  
 for (int i = 0; i < n; i++) {  
 System.*out*.println("x" + (i + 1) + " = " + solution[i]);  
 }  
 }  
}

**3.1.2 Результат:**

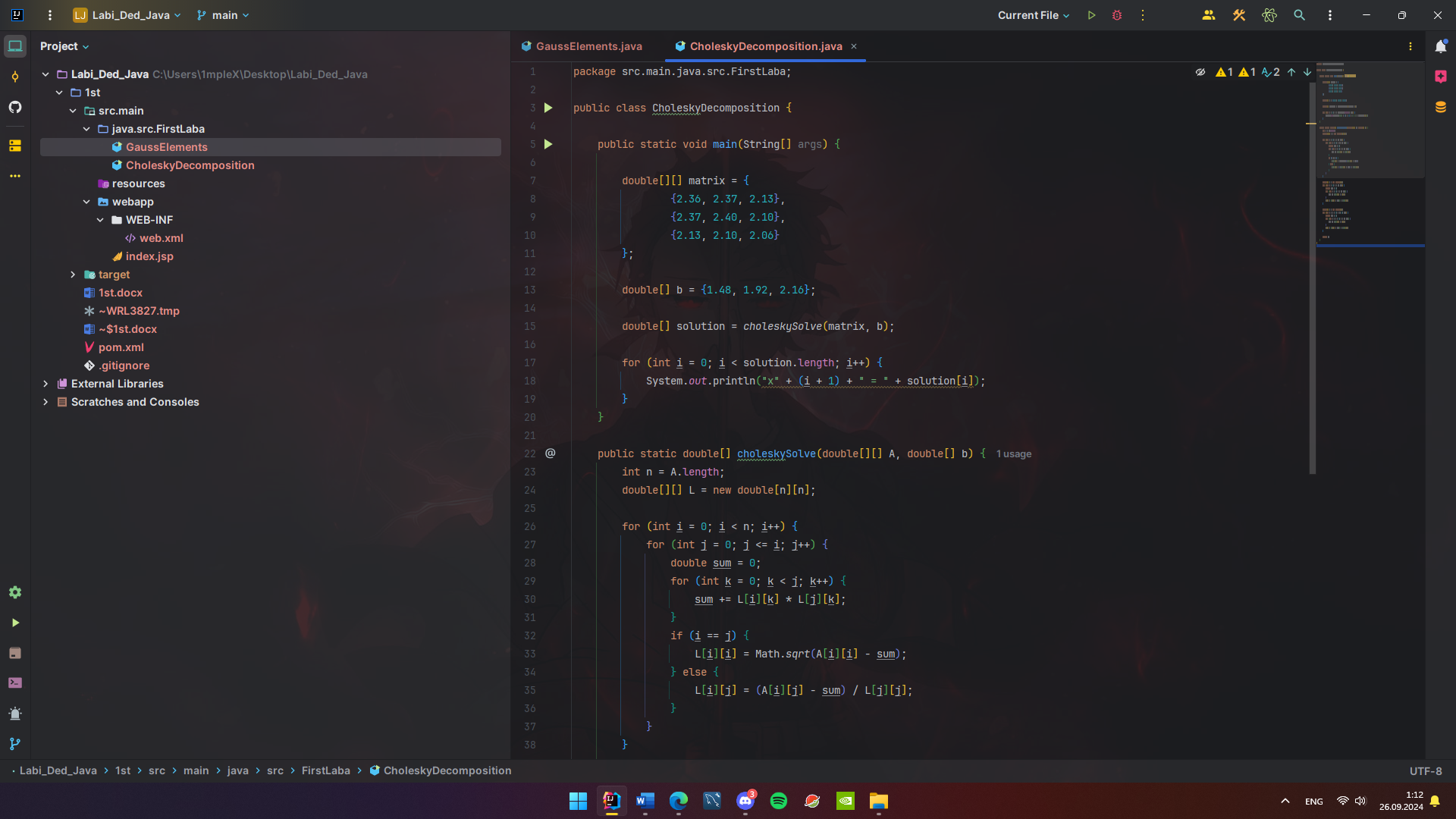
****

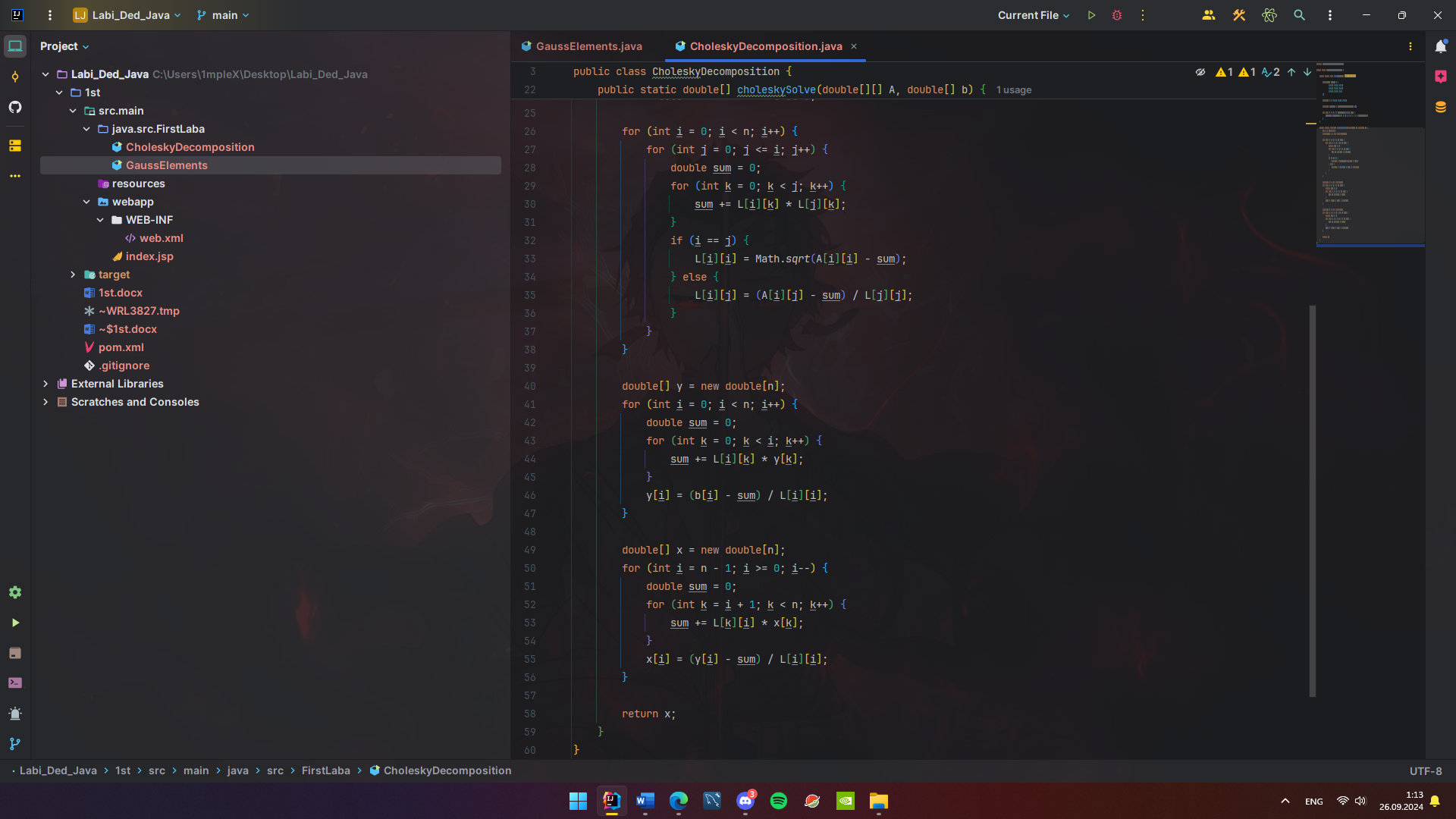
x1 = 3.8637274549097467

x2 = -3.899354264083538

x3 = 0.7526163437985841

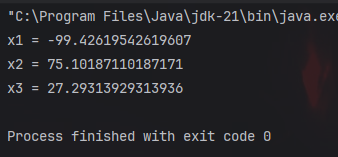
**3.2.1 Метод Холецкого:**





package src.main.java.src.FirstLaba;  
public class CholeskyDecomposition {  
 public static void main(String[] args) {  
 double[][] matrix = {  
 {2.36, 2.37, 2.13},  
 {2.37, 2.40, 2.10},  
 {2.13, 2.10, 2.06}  
 };  
 double[] b = {1.48, 1.92, 2.16};  
 double[] solution = *choleskySolve*(matrix, b);  
 for (int i = 0; i < solution.length; i++) {  
 System.*out*.println("x" + (i + 1) + " = " + solution[i]);  
 }  
 }  
 public static double[] choleskySolve(double[][] A, double[] b) {  
 int n = A.length;  
 double[][] L = new double[n][n];  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j <= i; j++) {  
 double sum = 0;  
 for (int k = 0; k < j; k++) {  
 sum += L[i][k] \* L[j][k];  
 }  
 if (i == j) {  
 L[i][i] = Math.*sqrt*(A[i][i] - sum);  
 } else {  
 L[i][j] = (A[i][j] - sum) / L[j][j];  
 }  
 }  
 }  
 double[] y = new double[n];  
 for (int i = 0; i < n; i++) {  
 double sum = 0;  
 for (int k = 0; k < i; k++) {  
 sum += L[i][k] \* y[k];  
 }  
 y[i] = (b[i] - sum) / L[i][i];  
 }  
 double[] x = new double[n];  
 for (int i = n - 1; i >= 0; i--) {  
 double sum = 0;  
 for (int k = i + 1; k < n; k++) {  
 sum += L[k][i] \* x[k];  
 }  
 x[i] = (y[i] - sum) / L[i][i];  
 }  
 return x;  
 }  
}

**3.2.2 Результат:**

****

x1 = -99.42619542619607

x2 = 75.10187110187171

x3 = 27.29313929313936