**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота № 2**

з дисципліни  
«Розробка програмного забезпечення на платформі Java»

Виконав: Перевірив:

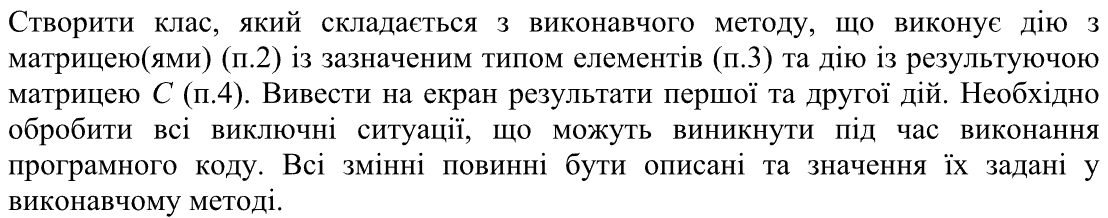
студент групи ІМ-11 Олександр Миронович Ковальчук  
Царик Микола Миколайович  
варіант відповідно до списку: 22

**Київ 2023**

**Тема:** Масиви в мові програмування Java.

**Мета:** Ознайомлення з масивами та використання основних методів їх обробки в мові програмування Java. Здобуття навичок у використанні масивів в мові програмування Java.

**Завдання:**



Я не маю доступу до заліковки тому рахую свій номер по принципу

“Номер групи” + “Номер у списку” (1122)

C5 = 1122 mod 5 = 2 => C = A + B

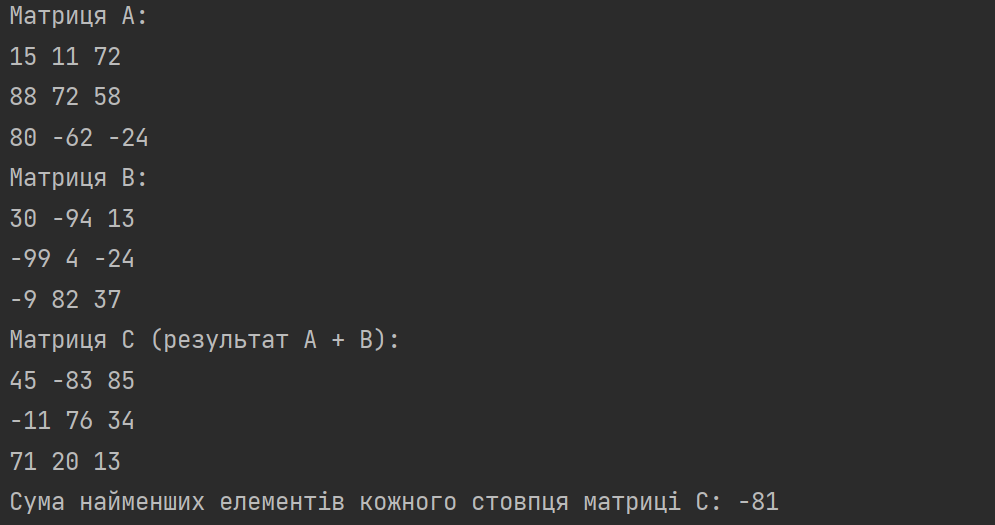
C7 = 1122 mod 7 = 2 => short

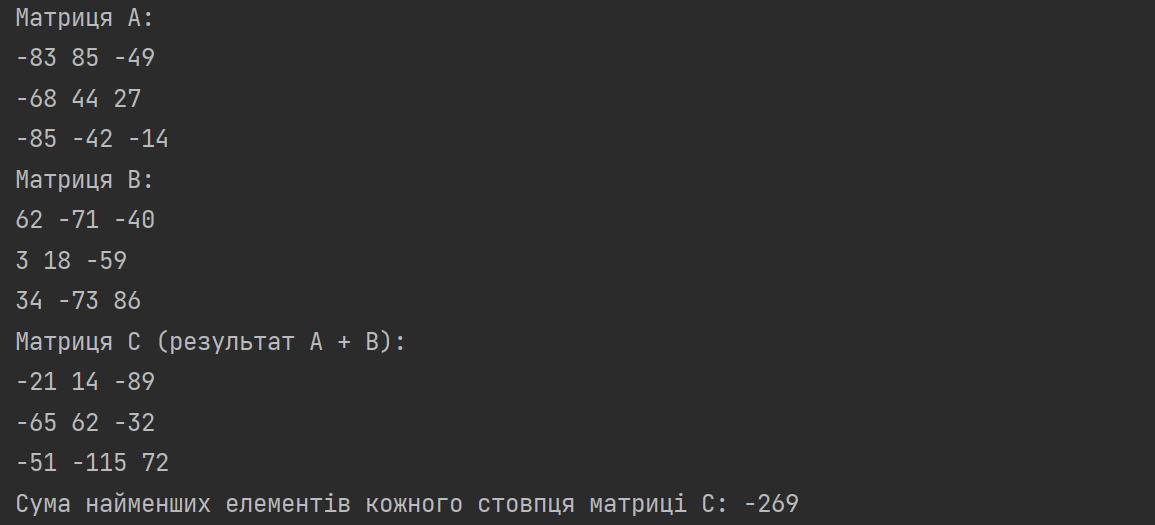
C11 = 1122 mod 11 = 0 => Обчислити суму найменших елементів кожного стовпця матриці

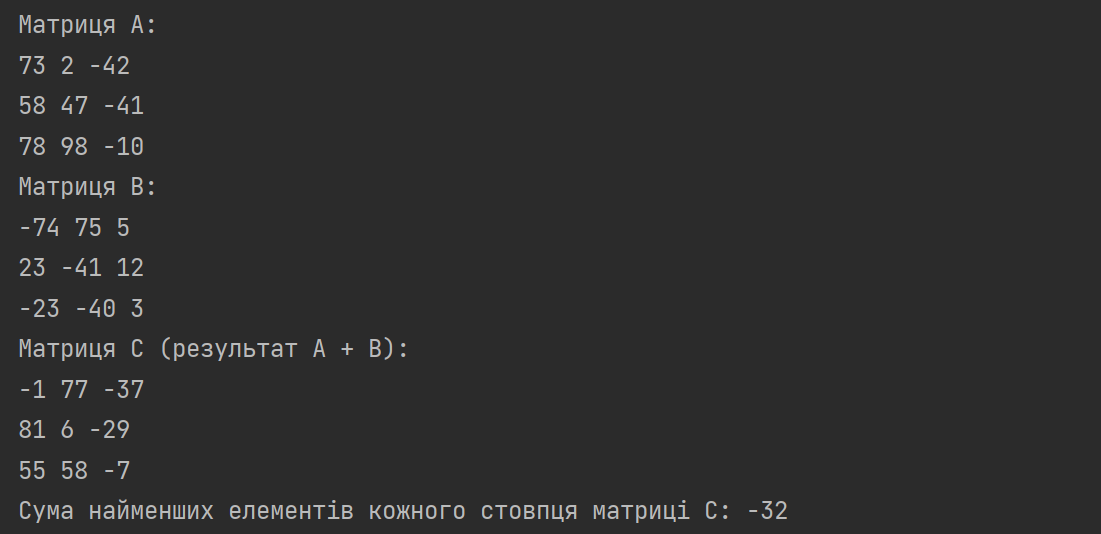
**Лістинг програми:**

public class MatrixOperations {  
 public static void main(String[] args) {  
 try{  
 int rows = 3;  
 int cols = 3;  
 short[][] A = new short[rows][cols];  
 short[][] B = new short[rows][cols];  
  
 // Заповнюємо матриці A та B деякими значеннями від -100 до 100  
 int min = -100;  
 int max = 100;  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 A[i][j] = (short) (Math.*random*() \* (max - min) + min);  
 B[i][j] = (short) (Math.*random*() \* (max - min) + min);  
 }  
 }  
  
 short[][] C = *addMatrices*(A, B);  
  
 System.*out*.println("Матриця A:");  
 *printMatrix*(A);  
 System.*out*.println("Матриця B:");  
 *printMatrix*(B);  
 System.*out*.println("Матриця C (результат A + B):");  
 *printMatrix*(C);  
  
 short columnSum = *sumOfMinimumsInColumns*(C);  
 System.*out*.println("Сума найменших елементів кожного стовпця матриці C: " + columnSum);  
 } catch (Exception e) {  
 System.*err*.println("Помилка: " + e.getMessage());  
 }  
 }  
  
 public static short[][] addMatrices(short[][] A, short[][] B) {  
 int rows = A.length;  
 int cols = A[0].length;  
 short[][] C = new short[rows][cols];  
  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 C[i][j] = (short) (A[i][j] + B[i][j]);  
 }  
 }  
  
 return C;  
 }  
  
 public static short sumOfMinimumsInColumns(short[][] matrix) {  
 int rows = matrix.length;  
 int cols = matrix[0].length;  
 short[] minimals = new short[cols];  
  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 if (i == 0 || minimals[j] > matrix[i][j]) {  
 minimals[j] = matrix[i][j];  
 }  
 }  
 }  
  
 short sum = 0;  
 for (int j = 0; j < cols; j++) {  
 sum += minimals[j];  
 }  
  
 return sum;  
 }  
  
 public static void printMatrix(short[][] matrix) {  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[i].length; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
}

**Запустимо програму кілька разів** **(щоб спробувати різні значення в матрицях)**







**Висновок**

Під час лабораторної роботи я працював з масивами в Java, навчився створювати, заповнювати та обробляти масиви, а також розробив клас з виконавчим методом для обробки матриць. Обробка виключень є важливим етапом розробки, але в моєму випадку не було особливиї кейсів , яких треба було запобігти. Ця лабораторна робота надала мені практичні навички і розуміння важливості обробки виключень при розробці програм на Java.