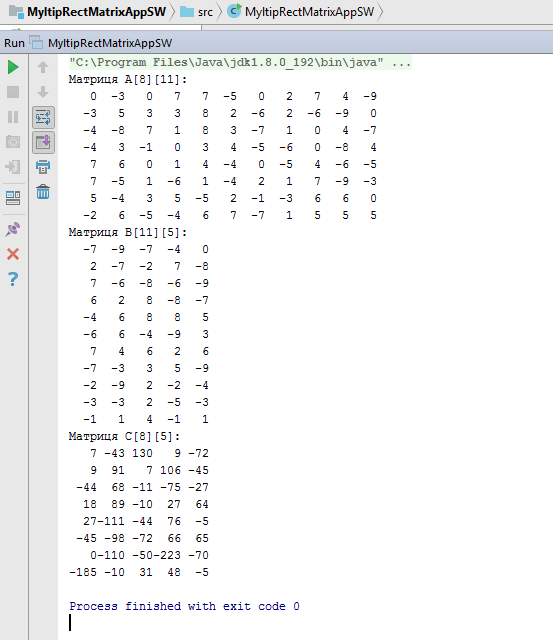
Програма множення прямокутних матриць

Підготував Моренко Олексій

Учень 10-Мі класу

Умова. A[8][11]xB[11][5]=C[8][5]

Результат:



Матриця А[8][11]:

0 -3 0 7 7 -5 0 2 7 4 -9

-3 5 3 3 8 2 -6 2 -6 -9 0

-4 -8 7 1 8 3 -7 1 0 4 -7

-4 3 -1 0 3 4 -5 -6 0 -8 4

7 6 0 1 4 -4 0 -5 4 -6 -5

7 -5 1 -6 1 -4 2 1 7 -9 -3

5 -4 3 5 -5 2 -1 -3 6 6 0

-2 6 -5 -4 6 7 -7 1 5 5 5

Матриця B[11][5]:

-7 -9 -7 -4 0

2 -7 -2 7 -8

7 -6 -8 -6 -9

6 2 8 -8 -7

-4 6 8 8 5

-6 6 -4 -9 3

7 4 6 2 6

-7 -3 3 5 -9

-2 -9 2 -2 -4

-3 -3 2 -5 -3

-1 1 4 -1 1

Матриця C[8][5]:

7 -43 130 9 -72

9 91 7 106 -45

-44 68 -11 -75 -27

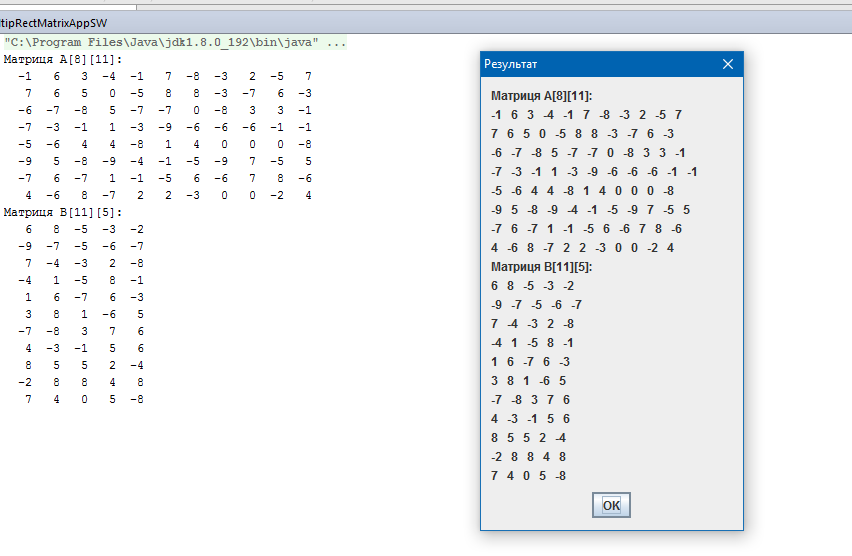
18 89 -10 27 64

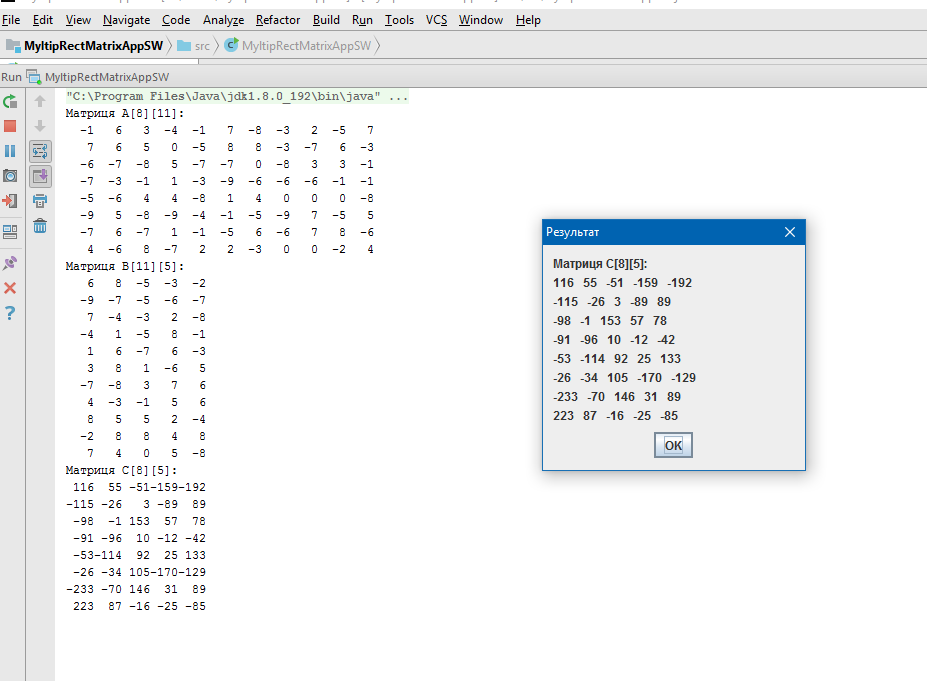
27-111 -44 76 -5

-45 -98 -72 66 65

0-110 -50-223 -70

-185 -10 31 48 -5



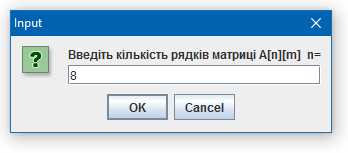


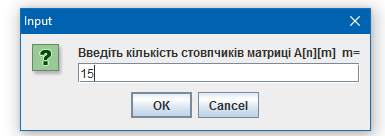
Лістинг

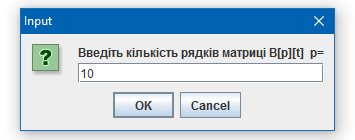
**import** javax.swing.\*;  
**import** java.util.Random;  
**public class** MyltipRectMatrixAppSW {  
 **public static void** main(String[] args) {  
 MyltipRectMatrixApp app = **new** MyltipRectMatrixApp();  
 app.setVisible(**false**);  
 }  
  
 **public static class** MyltipRectMatrixApp **extends** JFrame {  
  
 **public int n** = 8;  
 **public int m** = 11;  
 **public int p** = 11;  
 **public int t** = 5;  
 **public int**[][] **A** = **new int**[**n**][**m**];  
 **public int**[][] **B** = **new int**[**p**][**t**];  
 **public int**[][] **C** = **new int**[**n**][**t**];  
  
 **public** String **ms** = **""**;  
 **public** String **tmp** = **""**;  
  
 MyltipRectMatrixApp() {  
  
  
 System.***out***.println(**"Матриця А["** + **n** + **"]["** + **m** + **"]:"**);  
 **ms** = **"Матриця А["** + **n** + **"]["** + **m** + **"]:"**;  
 **for** (**int** i = 0; i < **n**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **m**; j++) {  
 **A**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **A**[i][j]);  
 **ms** += **A**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
  
  
  
 **ms**+=**"\n"**;  
 System.***out***.println(**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**);  
 **ms**+=**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**;  
  
 **for** (**int** i = 0; i < **p**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **t**; j++) {  
 **B**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **B**[i][j]);  
 **ms** += **B**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
  
JOptionPane.*showMessageDialog*(**null**, **ms**,**"Результат"**,JOptionPane.***PLAIN\_MESSAGE***);  
  
 System.***out***.println(**"Матриця C["** + **n** + **"]["** + **t** + **"]:"**);  
 **ms**=**"Матриця C["** + **n** + **"]["** + **t** + **"]:"**;  
 **for** (**int** i = 0; i <**n** ; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j <**t** ; j++) {  
 **C**[i][j]=0;  
 **for** (**int** k = 0; k <**m** ; k++) {  
 **C**[i][j]+=**A**[i][k]\***B**[k][j];  
 }  
 System.***out***.printf(**"%4d"**, **C**[i][j]);  
 **ms** += **C**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
JOptionPane.*showMessageDialog*(**null**, **ms**,**"Результат"**,JOptionPane.***PLAIN\_MESSAGE***);  
  
  
  
  
  
  
 }  
  
 **private static int** rnd(**int** min, **int** max) {  
 **return** (**new** Random()).ints(min, max).iterator().nextInt();  
 }  
 }  
}

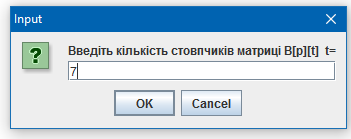
Нова версія програми – ведення діалогів з користувачем.

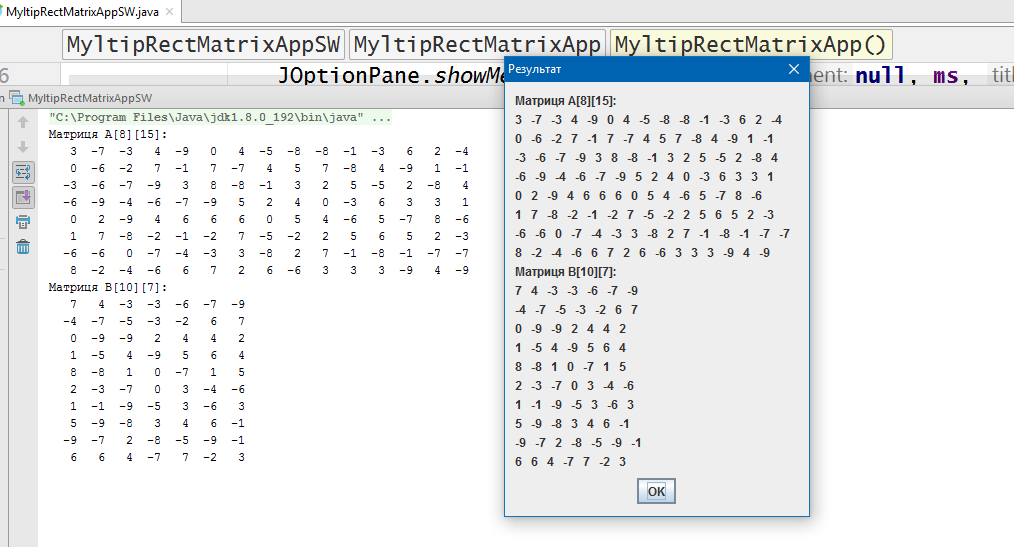
Тест 1. Невірний розмір матриць

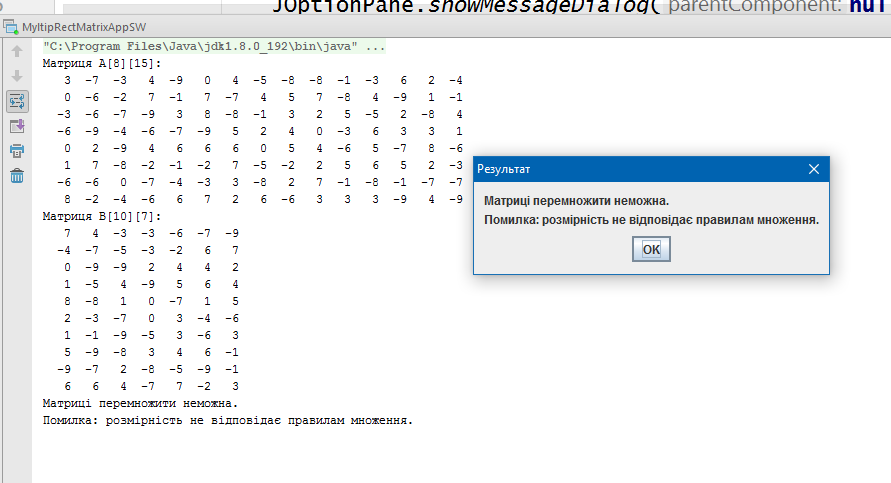




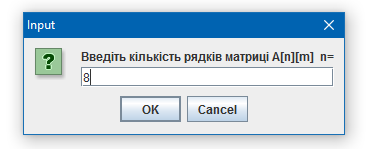


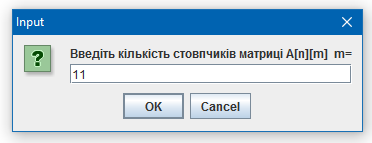


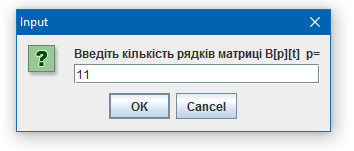


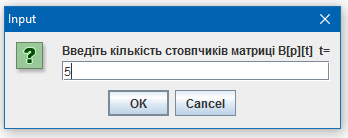


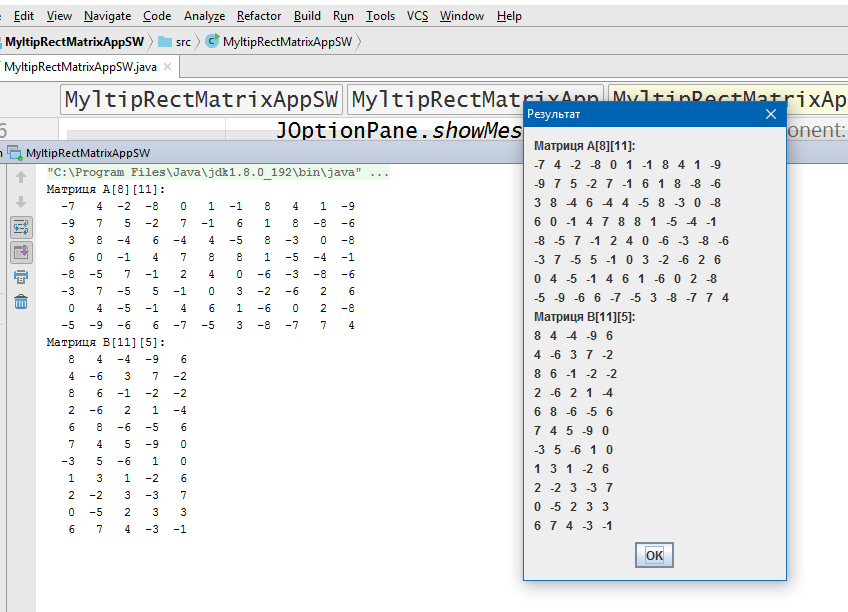
Тест 2. Вірне виконання

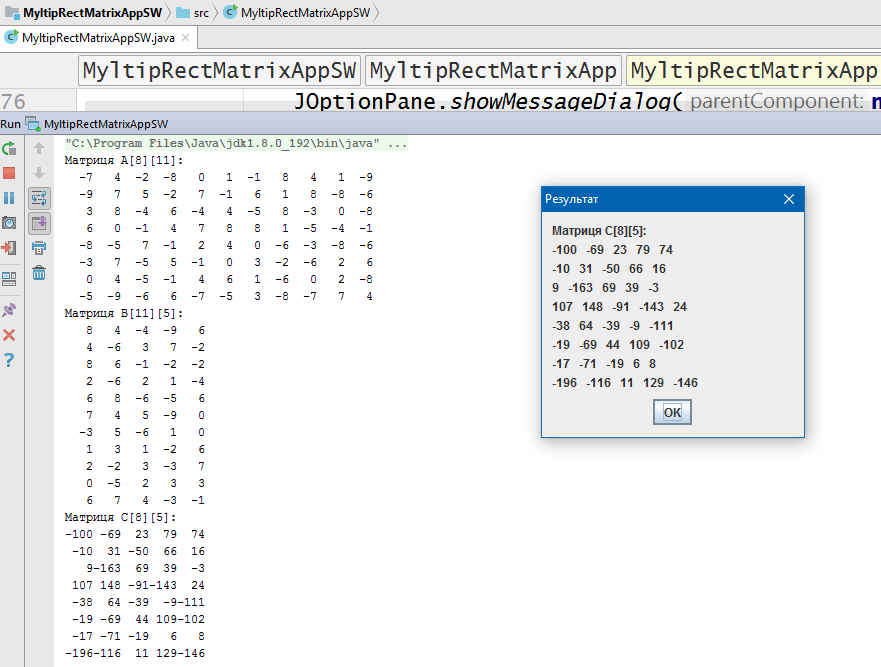








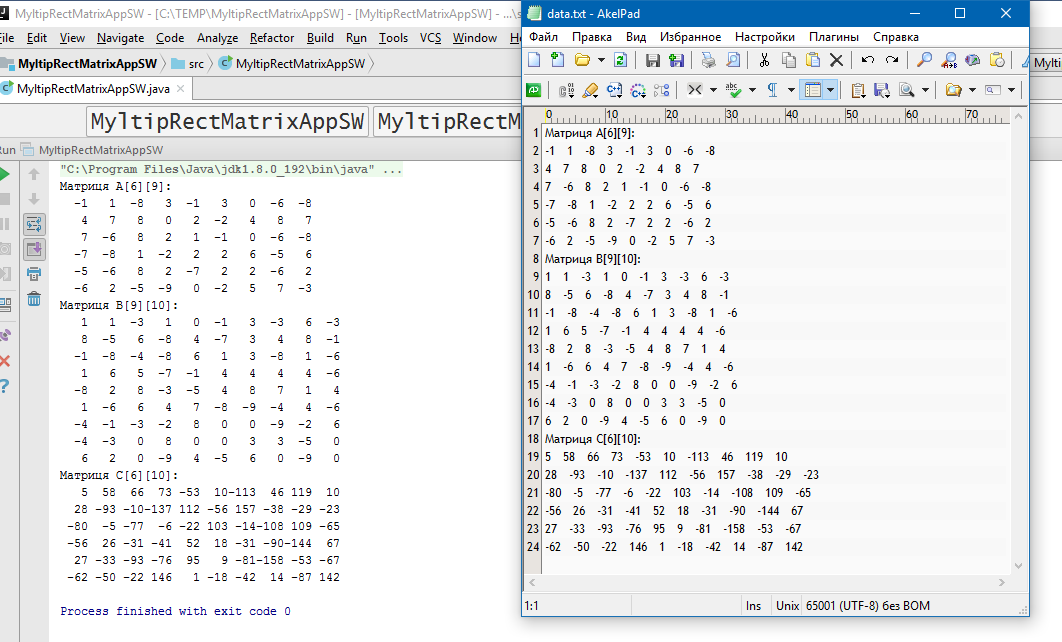


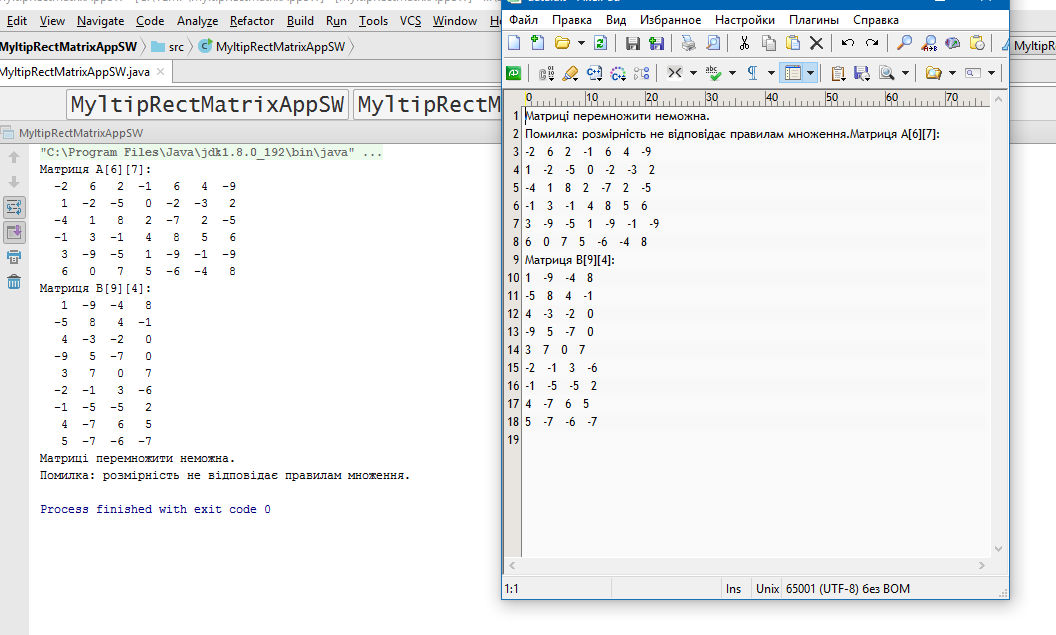


Лістинг програми з діалогом

**import** javax.swing.\*;  
**import** java.util.Random;  
**public class** MyltipRectMatrixAppSW {  
 **public static void** main(String[] args) {  
 MyltipRectMatrixApp app = **new** MyltipRectMatrixApp();  
 app.setVisible(**false**);  
 }  
  
 **public static class** MyltipRectMatrixApp **extends** JFrame {  
  
 **public int n** = 100;  
 **public int m** = 100;  
 **public int p** = 100;  
 **public int t** = 100;  
 **public int**[][] **A** = **new int**[**n**][**m**];  
 **public int**[][] **B** = **new int**[**p**][**t**];  
 **public int**[][] **C** = **new int**[**n**][**t**];  
  
 **public** String **ms** = **""**;  
 **public** String **tmp** = **""**;  
  
 MyltipRectMatrixApp() {  
  
  
**n**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість рядків матриці A[n][m] n="**));  
**m**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість стовпчиків матриці A[n][m] m="**));  
  
**p**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість рядків матриці B[p][t] p="**));  
**t**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість стовпчиків матриці B[p][t] t="**));  
  
 System.***out***.println(**"Матриця А["** + **n** + **"]["** + **m** + **"]:"**);  
 **ms** = **"Матриця А["** + **n** + **"]["** + **m** + **"]:"**;  
 **for** (**int** i = 0; i < **n**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **m**; j++) {  
 **A**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **A**[i][j]);  
 **ms** += **A**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
  
  
  
 **ms**+=**"\n"**;  
 System.***out***.println(**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**);  
 **ms**+=**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**;  
  
 **for** (**int** i = 0; i < **p**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **t**; j++) {  
 **B**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **B**[i][j]);  
 **ms** += **B**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
  
 JOptionPane.*showMessageDialog*(**null**, **ms**,**"Результат"**,JOptionPane.***PLAIN\_MESSAGE***);  
  
 **if**(**m**==**p**) {  
 System.***out***.println(**"Матриця C["** + **n** + **"]["** + **t** + **"]:"**);  
 **ms** = **"Матриця C["** + **n** + **"]["** + **t** + **"]:"**;  
 **for** (**int** i = 0; i < **n**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **t**; j++) {  
 **C**[i][j] = 0;  
 **for** (**int** k = 0; k < **m**; k++) {  
 **C**[i][j] += **A**[i][k] \* **B**[k][j];  
 }  
 System.***out***.printf(**"%4d"**, **C**[i][j]);  
 **ms** += **C**[i][j] + **" "**;  
 }  
 System.***out***.println();  
 }  
 JOptionPane.*showMessageDialog*(**null**, **ms**, **"Результат"**, JOptionPane.***PLAIN\_MESSAGE***);  
 }  
 **else** {  
 System.***out***.println(**"Матриці перемножити неможна.\nПомилка: розмірність не відповідає правилам множення."**);  
  
 **tmp**=**"Матриці перемножити неможна.\nПомилка: розмірність не відповідає правилам множення."**;  
 JOptionPane.*showMessageDialog*(**null**, **tmp**, **"Результат"**, JOptionPane.***PLAIN\_MESSAGE***);  
 }  
  
  
  
  
  
 }  
  
 **private static int** rnd(**int** min, **int** max) {  
 **return** (**new** Random()).ints(min, max).iterator().nextInt();  
 }  
 }  
}

Доробити виведення даних у окремий файл результатів data.txt





Лістинг

**import** javax.swing.\*;  
**import** java.io.File;  
**import** java.io.FileWriter;  
**import** java.io.IOException;  
**import** java.util.Random;  
**public class** MyltipRectMatrixAppSW {  
 **public static void** main(String[] args) {  
 MyltipRectMatrixApp app = **new** MyltipRectMatrixApp();  
 app.setVisible(**false**);  
 }  
  
 **public static class** MyltipRectMatrixApp **extends** JFrame {  
  
 **public int n** = 100;  
 **public int m** = 100;  
 **public int p** = 100;  
 **public int t** = 100;  
 **public int**[][] **A** = **new int**[**n**][**m**];  
 **public int**[][] **B** = **new int**[**p**][**t**];  
 **public int**[][] **C** = **new int**[**n**][**t**];  
  
 **public** String **ms** = **""**;  
 **public** String **tmp** = **""**;  
  
 **public** File **otf**=**new** File(**"data.txt"**);  
  
 MyltipRectMatrixApp() {  
  
  
 **n**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість рядків матриці A[n][m] n="**));  
 **m**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість стовпчиків матриці A[n][m] m="**));  
  
 **p**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість рядків матриці B[p][t] p="**));  
 **t**=Integer.*parseInt*(JOptionPane.*showInputDialog*(**"Введіть кількість стовпчиків матриці B[p][t] t="**));  
  
 System.***out***.println(**"Матриця А["** + **n** + **"]["** + **m** + **"]:"**);  
 **ms** = **"Матриця А["** + **n** + **"]["** + **m** + **"]:"**;  
 **for** (**int** i = 0; i < **n**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **m**; j++) {  
 **A**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **A**[i][j]);  
 **ms** += **A**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
  
  
  
 **ms**+=**"\n"**;  
 System.***out***.println(**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**);  
 **ms**+=**"Матриця B["** + **p** + **"]["** + **t** + **"]:"**;  
  
 **for** (**int** i = 0; i < **p**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **t**; j++) {  
 **B**[i][j] = *rnd*(-9, 9);  
 System.***out***.printf(**"%4d"**, **B**[i][j]);  
 **ms** += **B**[i][j]+**" "**;  
 }  
 System.***out***.println();  
 }  
 **tmp**=**"\n"**;  
 **tmp**+=**ms**;  
 **tmp**+=**"\n"**;  
 JOptionPane.*showMessageDialog*(**null**, **ms**,**"Результат"**,JOptionPane.***PLAIN\_MESSAGE***);  
  
 **if**(**m**==**p**) {  
 System.***out***.println(**"Матриця C["** + **n** + **"]["** + **t** + **"]:"**);  
 **ms** = **"Матриця C["** + **n** + **"]["** + **t** + **"]:"**;  
 **for** (**int** i = 0; i < **n**; i++) {  
 **ms** += **"\n"**;  
 **for** (**int** j = 0; j < **t**; j++) {  
 **C**[i][j] = 0;  
 **for** (**int** k = 0; k < **m**; k++) {  
 **C**[i][j] += **A**[i][k] \* **B**[k][j];  
 }  
 System.***out***.printf(**"%4d"**, **C**[i][j]);  
 **ms** += **C**[i][j] + **" "**;  
 }  
 System.***out***.println();  
 }  
 JOptionPane.*showMessageDialog*(**null**, **ms**, **"Результат"**, JOptionPane.***PLAIN\_MESSAGE***);  
 **tmp**+=**ms**;  
 **tmp**+=**"\n"**;  
 }  
 **else** {  
 System.***out***.println(**"Матриці перемножити неможна.\nПомилка: розмірність не відповідає правилам множення."**);  
  
 **tmp**=**"Матриці перемножити неможна.\nПомилка: розмірність не відповідає правилам множення."**;  
 JOptionPane.*showMessageDialog*(**null**, **tmp**, **"Результат"**, JOptionPane.***PLAIN\_MESSAGE***);  
 **tmp**+=**ms**;  
 **tmp**+=**"\n"**;  
 }  
  
  
  
  
 **try** {  
 **otf**.createNewFile();  
 FileWriter writer=**new** FileWriter(**otf**);  
 writer.write(**tmp**);  
 writer.flush();  
 writer.close();  
Runtime.*getRuntime*().exec(**"C:\\BORLAND\\AKELPAD\\AKELPAD.EXE data.txt"**);  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
 **private static int** rnd(**int** min, **int** max) {  
 **return** (**new** Random()).ints(min, max).iterator().nextInt();  
 }  
 }  
}

Програма пошуку в квадратних матрицях

Підготував Моренко Олексій

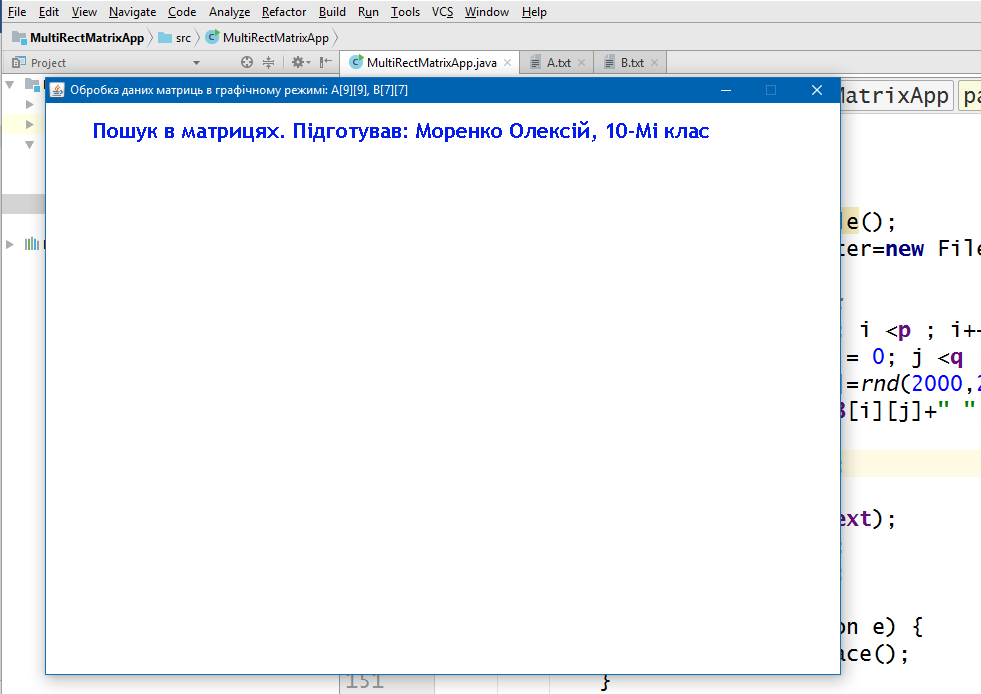
Учень 10-Мі класу

Умова. A[9][9], B[7][7]

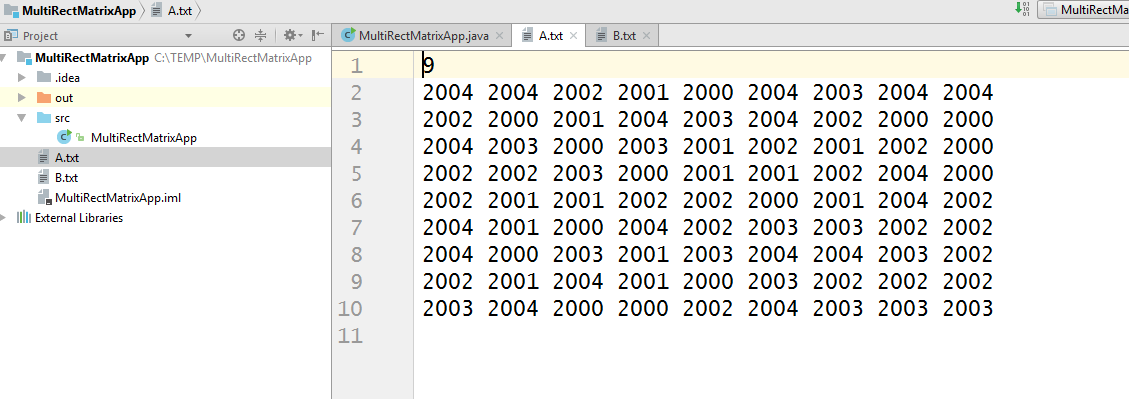
Пошук максимального/мінімального

Перший етап.

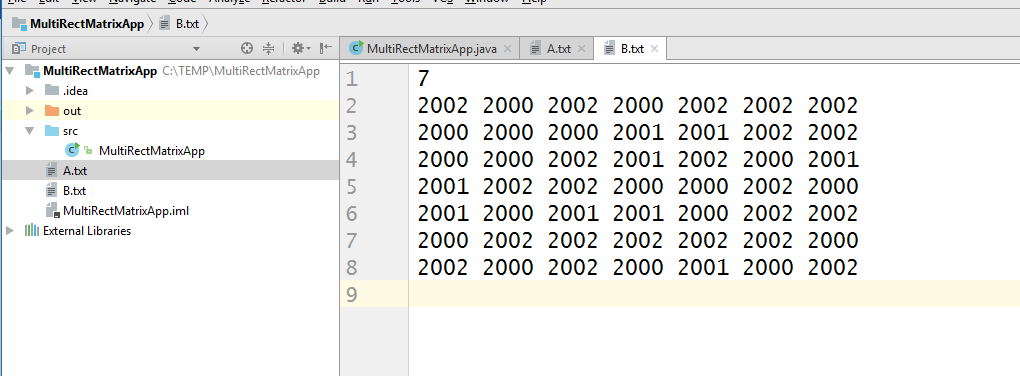
Заповнення матриць A[9][9] і B[7][7] випадковими цілими числами з проміжку [2000;2004]. Збереження інформації в файлах A.txt, B.txt



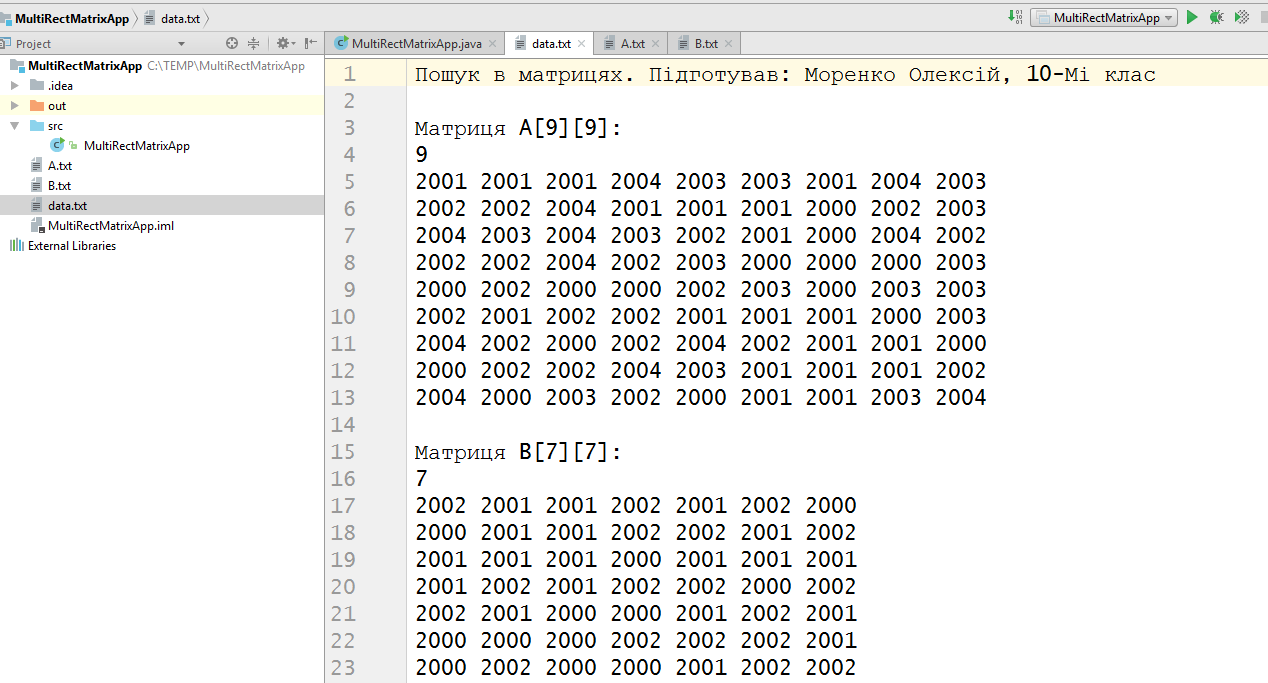
Файл A.txt



Файл B.txt

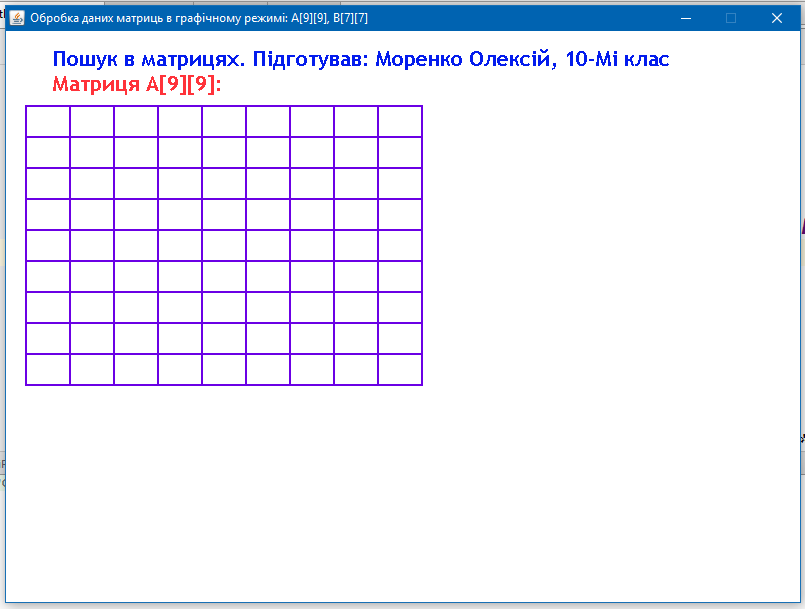


Формуємо загальний звіт (файл data.txt)

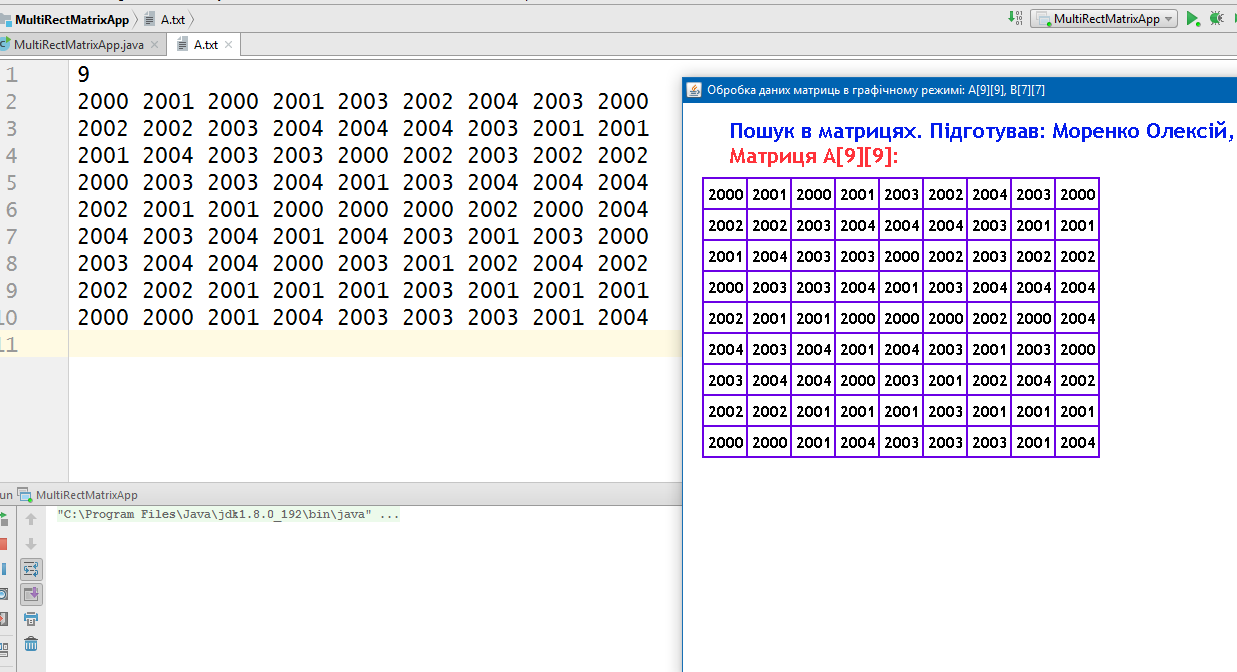


Другий етап. Креслення границь матриць

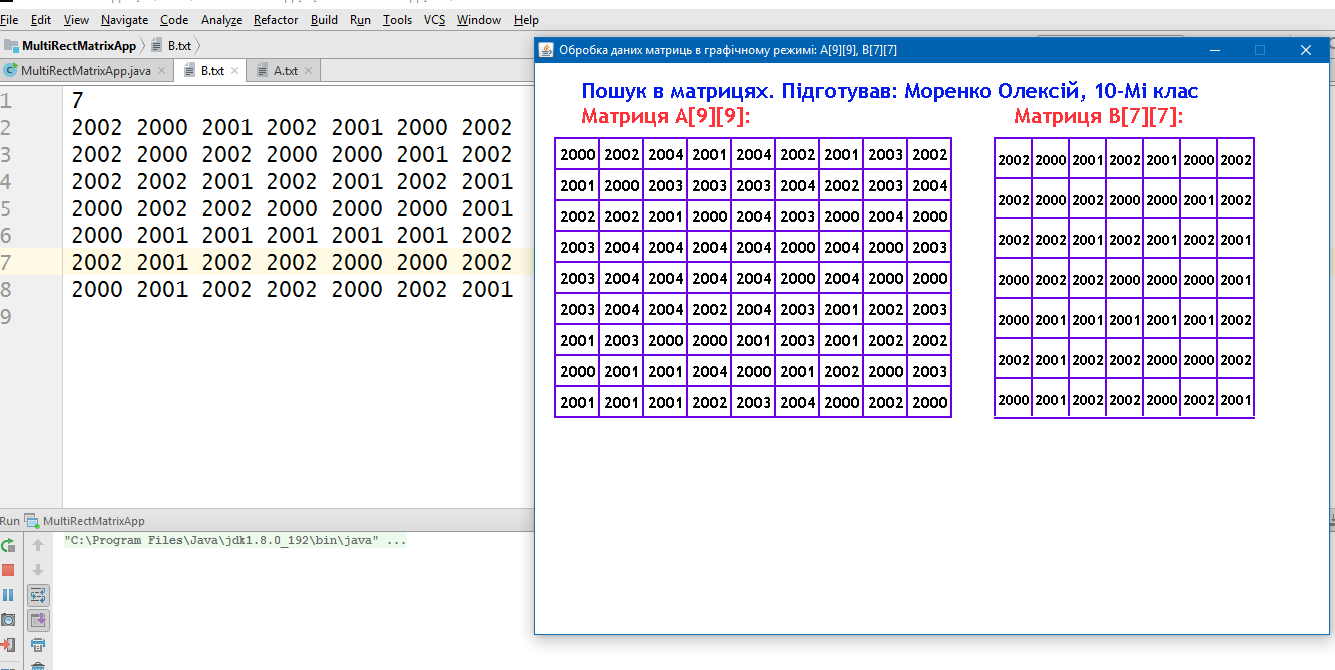
(графічна бібліотека java.awt.\*)



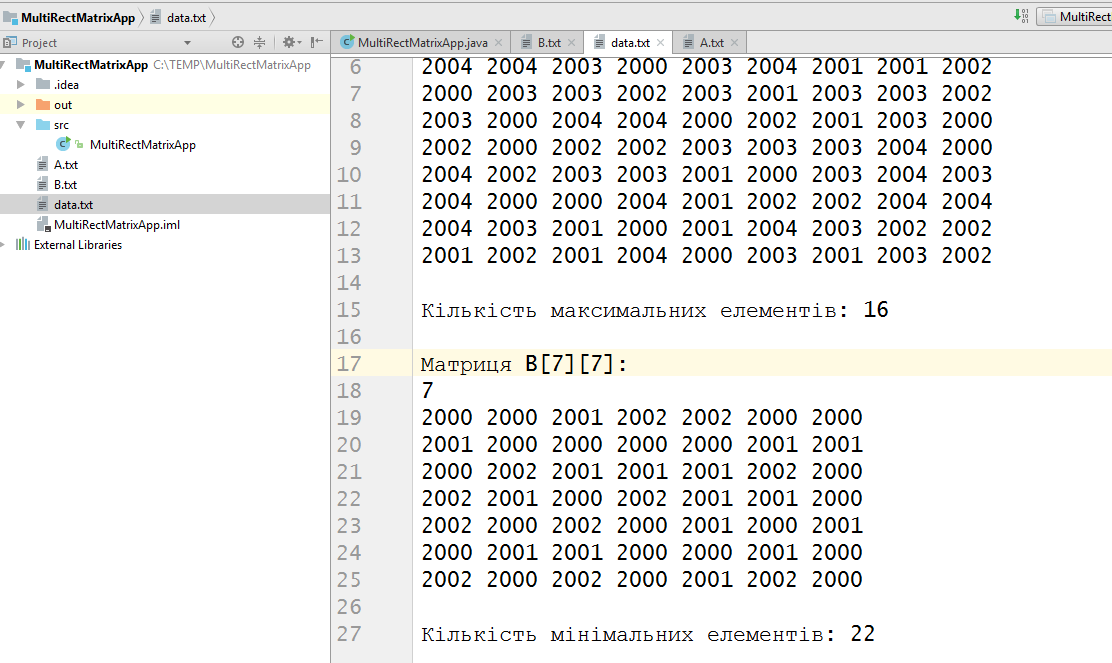
Заповнюємо матрицю A[9][9]:



Заповнені матриці A[9][9] і B[7][7]:



Розробляю алгоритм підрахування максимальних в A[9][9] та мінімальних в B[7][7] елементів.



Формую графічне зображення результатів пошуку



Лістинг

**import** org.w3c.dom.css.RGBColor;  
  
**import** java.awt.\*;  
**import** java.awt.event.WindowEvent;  
**import** java.awt.event.WindowListener;  
**import** java.io.File;  
**import** java.io.FileNotFoundException;  
**import** java.io.FileWriter;  
**import** java.io.IOException;  
**import** java.util.Random;  
**import** java.util.Scanner;  
  
**public class** MultiRectMatrixApp {  
  
 **public static void** main(String[] args) **throws** FileNotFoundException {  
  
 **new** GrMultiRectMatrixApp();  
  
 }  
  
 **public static int** rnd(**int** min,**int** max){  
 **return new** Random().ints(min,max).iterator().nextInt();  
 }  
  
  
 **public static class** GrMultiRectMatrixApp **extends** Frame **implements** WindowListener {  
  
 @Override  
 **public void** windowOpened(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 dispose();  
 System.*exit*(0);  
 }  
  
 @Override  
 **public void** windowClosed(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowIconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeiconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowActivated(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeactivated(WindowEvent e) {  
  
 }  
  
 **public int n**=9;  
 **public int m**=9;  
 **public int p**=7;  
 **public int q**=7;  
 **public int**[][] **A**=**new int**[**n**][**m**];  
 **public int**[][] **B**=**new int**[**p**][**q**];  
 **public int**[][] **C**=**new int**[**n**][**q**];  
  
 **public** String **ms**,**t**,**z**, **text**;  
  
 **public** File **af**=**new** File(**"A.txt"**);  
 **public** File **bf**=**new** File(**"B.txt"**);  
 *//public File cf=new File("C.txt");* **public** File **otf**=**new** File(**"data.txt"**);  
  
  
 **public int RH**,**RW**,**minx**,**miny**,**maxx**,**maxy**,**cx**,**cy**,**dx**,**dy**;  
 **public int x**,**y**,**x1**,**y1**,**x2**,**y2**,**x3**,**y3**,**x4**,**y4**;  
  
 **public float wt**=1.5f;  
  
  
 **public int acounter**=0;  
 **public int bcounter**=0;  
  
  
*//public Scanner scana=new Scanner(af);* **public void** paint(Graphics g){  
  
 Dimension rc=getSize();  
 Insets in=getInsets();  
  
 **RH**=rc.**width**;  
 **RW**=rc.**height**;  
  
 **minx**=in.**left**;  
 **maxx**=**RH**-in.**right**;  
  
 **miny**=in.**top**;  
 **maxy**=**RW**-in.**bottom**;  
  
 **cx**=Math.*abs*(**maxx**-**minx**)/2;  
 **cy**=Math.*abs*(**maxy**-**miny**)/2;  
  
 **dx**=**dy**=25;  
  
 g.setColor(**new** Color(4, 28, 237));  
 **text**=**"Пошук в матрицях. Підготував: Моренко Олексій, 10-Мі клас"**;  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,21));  
 g.drawString(**text**,50,**miny**+35);  
  
 **ms**=**text**;  
 **ms**+=**"\n\nМатриця A[9][9]:\n"**;  
 **try** {  
 **af**.createNewFile();  
 FileWriter writer=**new** FileWriter(**af**);  
 **text**=**n**+**"\n"**;  
 *//text+=m+"\n";* **acounter**=0;  
 **for** (**int** i = 0; i <**n** ; i++) {  
 **for** (**int** j = 0; j <**m** ; j++) {  
 **A**[i][j]=*rnd*(2000,2005);  
 **if**(**A**[i][j]==2004) **acounter**++;  
 **text**+=**A**[i][j]+**" "**;  
 }  
 **text**+=**"\n"**;  
 }  
 writer.write(**text**);  
 writer.flush();  
 writer.close();  
 **ms**+=**text**;  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 **t**=**"\nКількість максимальних елементів: "**+**acounter**;  
 **ms**+=**t**;  
  
  
 **ms**+=**"\n\nМатриця B[7][7]:\n"**;  
 **try** {  
 **bf**.createNewFile();  
 FileWriter writer=**new** FileWriter(**bf**);  
 **text**=**p**+**"\n"**;  
 *//text+=q+"\n";* **bcounter**=0;  
 **for** (**int** i = 0; i <**p** ; i++) {  
 **for** (**int** j = 0; j <**q** ; j++) {  
 **B**[i][j]=*rnd*(2000,2003);  
 **if**(**B**[i][j]==2000) **bcounter**++;  
 **text**+=**B**[i][j]+**" "**;  
 }  
 **text**+=**"\n"**;  
 }  
 writer.write(**text**);  
 writer.flush();  
 writer.close();  
 **ms**+=**text**;  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 **z**=**"\nКількість мінімальних елементів: "**+**bcounter**;  
 **ms**+=**z**;  
  
 **try** {  
 **otf**.createNewFile();  
 FileWriter writer=**new** FileWriter(**otf**);  
 writer.write(**ms**);  
 writer.flush();  
 writer.close();  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 Graphics2D g2=(Graphics2D)g;  
 g2.setStroke(**new** BasicStroke(**wt**));  
  
 g.setColor(**new** Color(255, 51, 58));  
 **text**=**"Матриця A[9][9]:"**;  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,21));  
 g.drawString(**text**,50,**miny**+60);  
  
 g.setColor(**new** Color(107, 0, 229));  
 **dx**=(**maxx**-**minx**)/(2\***m**);  
 **dy**=(**maxy**-**miny**)/(2\***n**);  
 **x**=**minx**+20; **y**=**miny**+75;  
 **for** (**int** i = 0; i <=**n** ; i++) {  
 g.drawLine(**x**,**y**+i\***dy**,**x**+**maxx**-401,**y**+i\***dy**);  
 }  
 **for** (**int** i = 0; i <=**m** ; i++) {  
 g.drawLine(**x**+i\***dx**,**y**,**x**+i\***dx**,**y**+**maxy**-320);  
 }  
  
 g.setColor(**new** Color(0, 0, 0));  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,16));  
 **x**=**x**+**dx**/2-17; **y**=**y**+**dy**/2+7;  
 **for** (**int** i = 0; i <**n** ; i++) {  
 **for** (**int** j = 0; j <**m** ; j++) {  
 **text**= String.*valueOf*(**A**[i][j]);  
 g.drawString(**text**,**x**+j\***dx**,**y**+i\***dy**);  
 }  
 }  
  
 *////////////////////////////////////////////////////////////* g.setColor(**new** Color(255, 51, 58));  
 **text**=**"Матриця B[7][7]:"**;  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,21));  
 g.drawString(**text**,**minx**+480,**miny**+60);  
  
 g.setColor(**new** Color(107, 0, 229));  
 **dx**=(**maxx**-**minx**)/(3\***q**);  
 **dy**=(**maxy**-**miny**)/(2\***p**);  
 **x**=**minx**+460; **y**=**miny**+75;  
 **for** (**int** i = 0; i <=**p** ; i++) {  
 g.drawLine(**x**,**y**+i\***dy**,**x**+**maxx**-538,**y**+i\***dy**);  
 }  
 **for** (**int** i = 0; i <=**q** ; i++) {  
 g.drawLine(**x**+i\***dx**,**y**,**x**+i\***dx**,**y**+**maxy**-320);  
 }  
  
 g.setColor(**new** Color(0, 0, 0));  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,14));  
 **x**=**x**+**dx**/2-15; **y**=**y**+**dy**/2+7;  
 **for** (**int** i = 0; i <**p** ; i++) {  
 **for** (**int** j = 0; j <**q** ; j++) {  
 **text**= String.*valueOf*(**B**[i][j]);  
 g.drawString(**text**,**x**+j\***dx**,**y**+i\***dy**);  
 }  
 }  
  
 */////////////////////////////////////////////////* g.setColor(**new** Color(255, 51, 58));  
 **text**=**"Кількість максимальних елементів"**;  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,21));  
 g.drawString(**text**,**minx**+20,**miny**+380);  
 **text**=**"в матриці A[9][9] дорівнює "**+**acounter**;  
 g.drawString(**text**,**minx**+20,**miny**+405);  
  
 *///////////////////////////////////////////////////* g.setColor(**new** Color(255, 51, 58));  
 **text**=**"Кількість мінімальних елементів"**;  
 g.setFont(**new** Font(**"Trebuchet Ms"**,Font.***BOLD***,21));  
 g.drawString(**text**,**minx**+440,**miny**+380);  
 **text**=**"в матриці B[7][7] дорівнює "**+**bcounter**;  
 g.drawString(**text**,**minx**+440,**miny**+405);  
  
 }  
  
  
 **public** GrMultiRectMatrixApp() **throws** FileNotFoundException {  
 **super**(**"Обробка даних матриць в графічному режимі: A[9][9], B[7][7]"**);  
 addWindowListener(**this**);  
 setLayout(**null**);  
 setLocation(80,80);  
 setSize(800,600);  
 **this**.setResizable(**false**);  
 show();  
 }  
 }  
}