

Програма множення прямокутних матриць

Підготував Моренко Олексій

Учень 11-Мі класу

Умова. $A[8][11] \times B[11][5] = C[8][5]$

Результат:

```
"C:\Program Files\Java\jdk1.8.0_192\bin\java" ...
Матриця A[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

Process finished with exit code 0
|
```

Матриця A[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

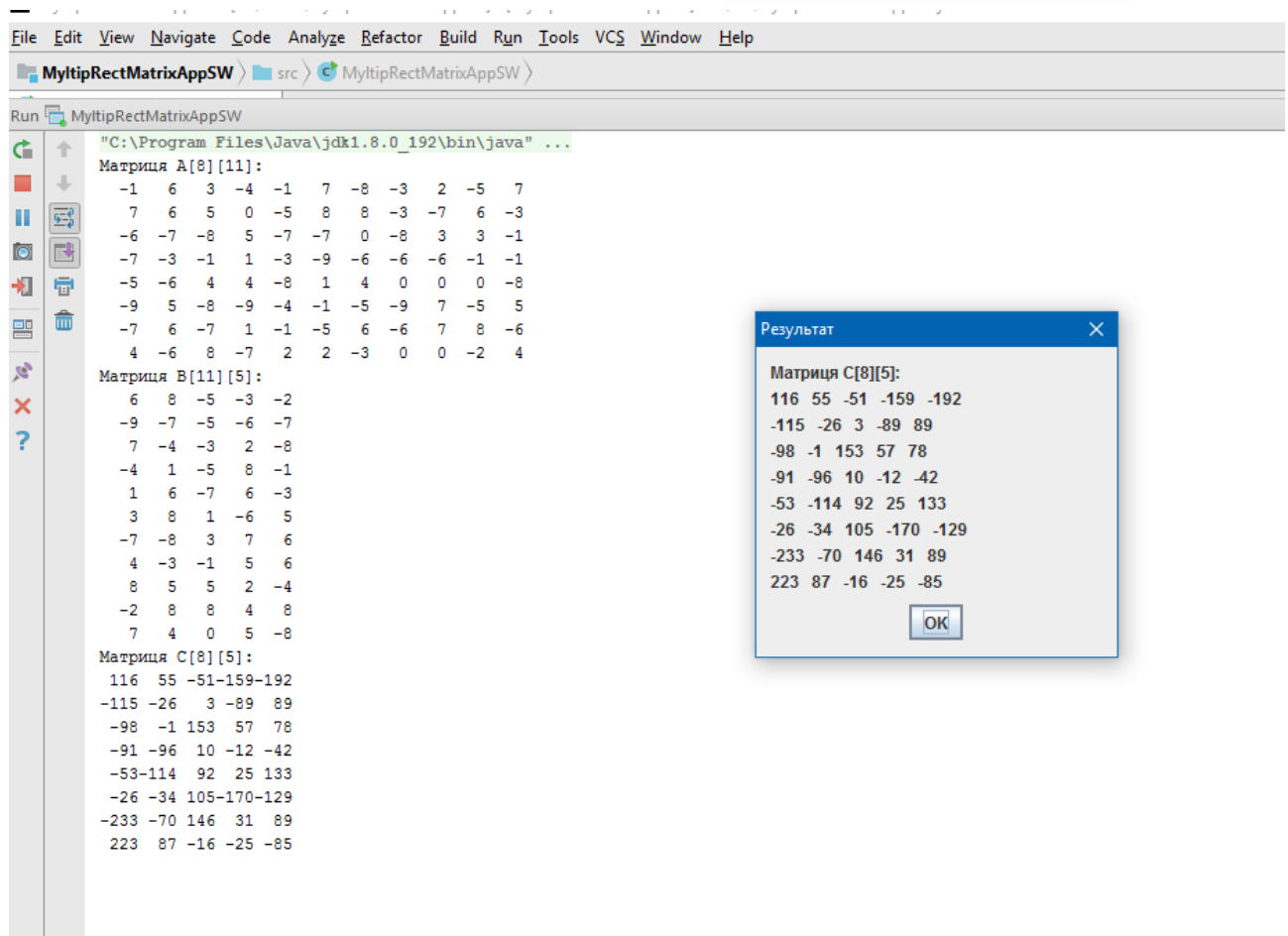
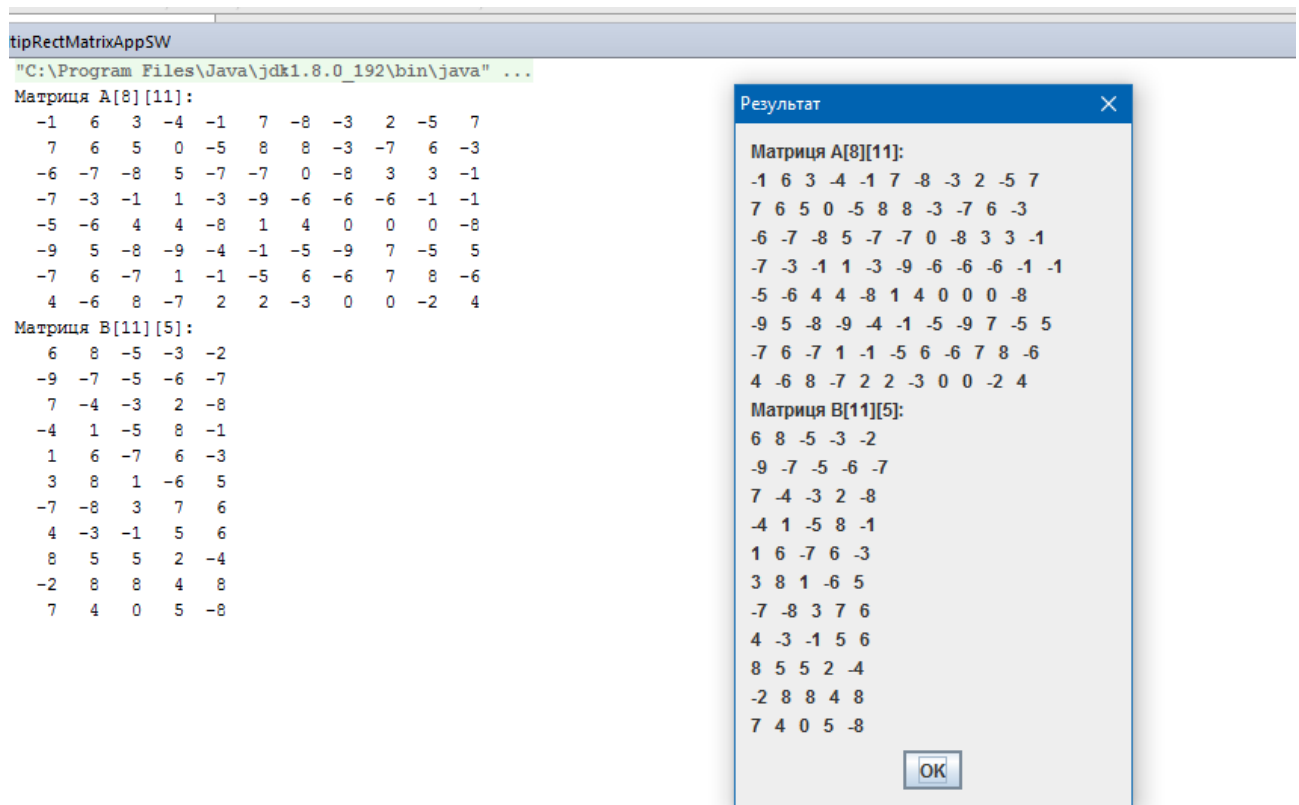
Матриця B[11][5]:

-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

Використання бібліотеки javax.swing.*



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 A[" + n + "][" + m + "]:");
            ms = "Матриця A[" + 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);
}

```

```

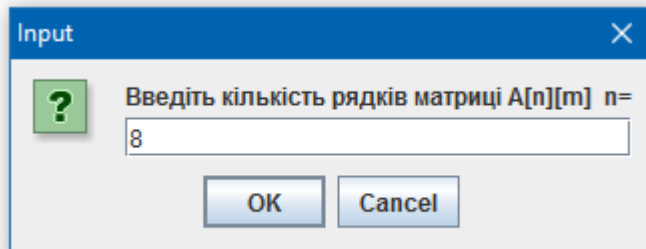
    }

    private static int rnd(int min, int max) {
        return (new Random()).ints(min, max).iterator().nextInt();
    }
}

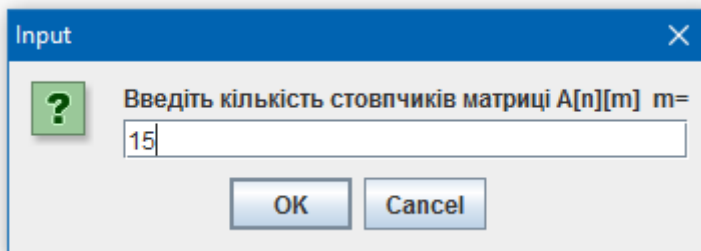
```

Нова версія програми - ведення діалогів з користувачем.

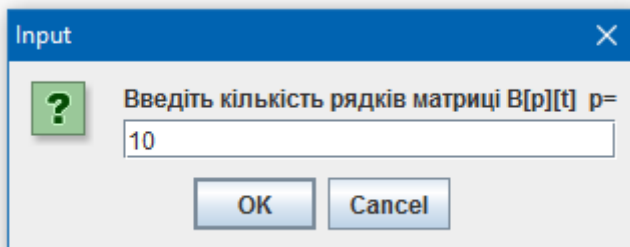
Тест 1. Невірний розмір матриць



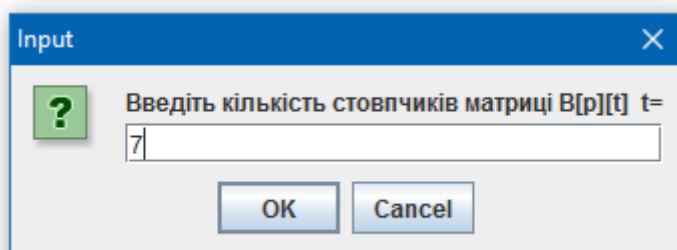
Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Введіть кількість рядків матриці A[n][m] n=", a text input field containing the value "8", and "OK" and "Cancel" buttons.



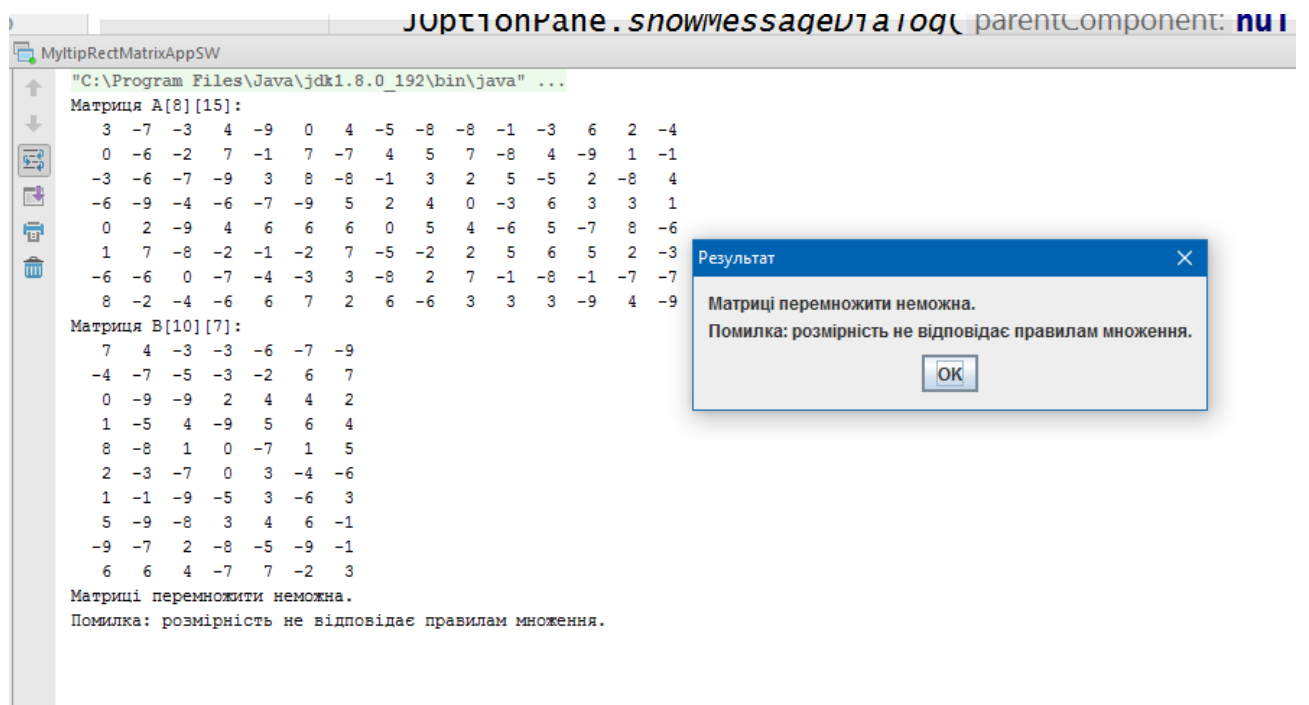
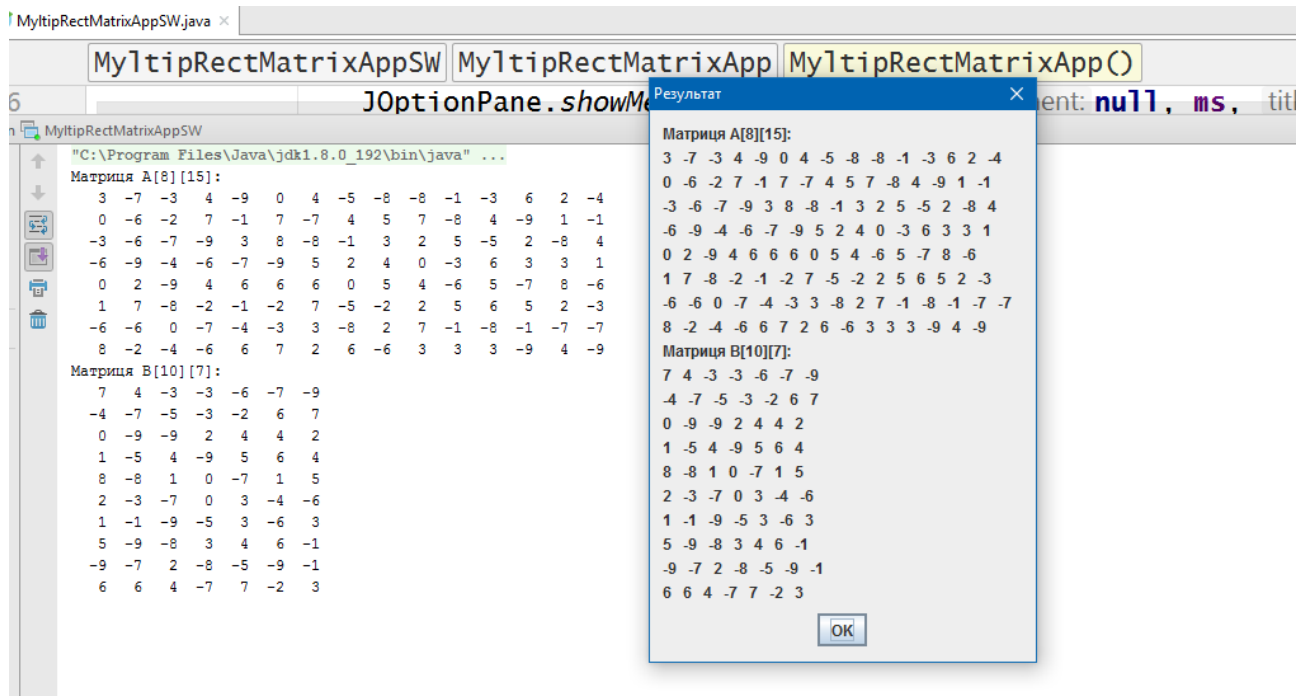
Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Введіть кількість стовпчиків матриці A[n][m] m=", a text input field containing the value "15", and "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Введіть кількість рядків матриці B[p][t] p=", a text input field containing the value "10", and "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Введіть кількість стовпчиків матриці B[p][t] t=", a text input field containing the value "7", and "OK" and "Cancel" buttons.



Тест 2. Вірне виконання

Input

Введіть кількість рядків матриці $A[n][m]$ $n=$

8

OK Cancel

Input

Введіть кількість стовпчиків матриці $A[n][m]$ $m=$

11

OK Cancel

Input

Введіть кількість рядків матриці $B[p][t]$ $p=$

11

OK Cancel

Input

Введіть кількість стовпчиків матриці $B[p][t]$ $t=$

5

OK Cancel

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 A[" + n + "][" + m + "]:");
            ms = "Матриця A[" + 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

```

MyltipRectMatrixAppSW - [C:\TEMP\MyltipRectMatrixAppSW] - [MyltipRectMatrixAppSW] - ...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyltipRectMatrixAppSW src MyltipRectMatrixAppSW
MyltipRectMatrixAppSW.java

MyltipRectMatrixAppSW MyltipRectM

MyltipRectMatrixAppSW
"C:\Program Files\Java\jdk1.8.0_192\bin\java" ...
Матриця A[6][9]:
-1 1 -8 3 -1 3 0 -6 -8
4 7 8 0 2 -2 4 8 7
7 -6 8 2 1 -1 0 -6 -8
-7 -8 1 -2 2 2 6 -5 6
-5 -6 8 2 -7 2 2 -6 2
-6 2 -5 -9 0 -2 5 7 -3
Матриця B[9][10]:
1 1 -3 1 0 -1 3 -3 6 -3
8 -5 6 -8 4 -7 3 4 8 -1
-1 -8 -4 -8 6 1 3 -8 1 -6
1 6 5 -7 -1 4 4 4 4 -6
-8 2 8 -3 -5 4 8 7 1 4
1 -6 6 4 7 -8 -9 -4 4 -6
-4 -1 -3 -2 8 0 0 -9 -2 6
-4 -3 0 8 0 0 3 3 -5 0
6 2 0 -9 4 -5 6 0 -9 0
Матриця C[6][10]:
5 58 66 73 -53 10 -113 46 119 10
28 -93 -10 -137 112 -56 157 -38 -29 -23
-80 -5 -77 -6 -22 103 -14 -108 109 -65
-56 26 -31 -41 52 18 -31 -90 -144 67
27 -33 -93 -76 95 9 -81 -158 -53 -67
-62 -50 -22 146 1 -18 -42 14 -87 142
Process finished with exit code 0

```

```

MyltipRectMatrixAppSW src MyltipRectMatrixAppSW
MyltipRectMatrixAppSW.java

MyltipRectMatrixAppSW MyltipRectM

MyltipRectMatrixAppSW
"C:\Program Files\Java\jdk1.8.0_192\bin\java" ...
Матриця A[6][7]:
-2 6 2 -1 6 4 -9
1 -2 -5 0 -2 -3 2
-4 1 8 2 -7 2 -5
-1 3 -1 4 8 5 6
3 -9 -5 1 -9 -1 -9
6 0 7 5 -6 -4 8
Матриця B[9][4]:
1 -9 -4 8
-5 8 4 -1
4 -3 -2 0
-9 5 -7 0
3 7 0 7
-2 -1 3 -6
-1 -5 -5 2
4 -7 6 5
5 -7 -6 -7
Матриці перемножити неможна.
Помилка: розмірність не відповідає правилам множення.
Process finished with exit code 0

```

```

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");

MyItpRectMatrixApp() {

    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 A[" + n + "][" + m + "]:");
    ms = "Матриця A[" + 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();
    }

}

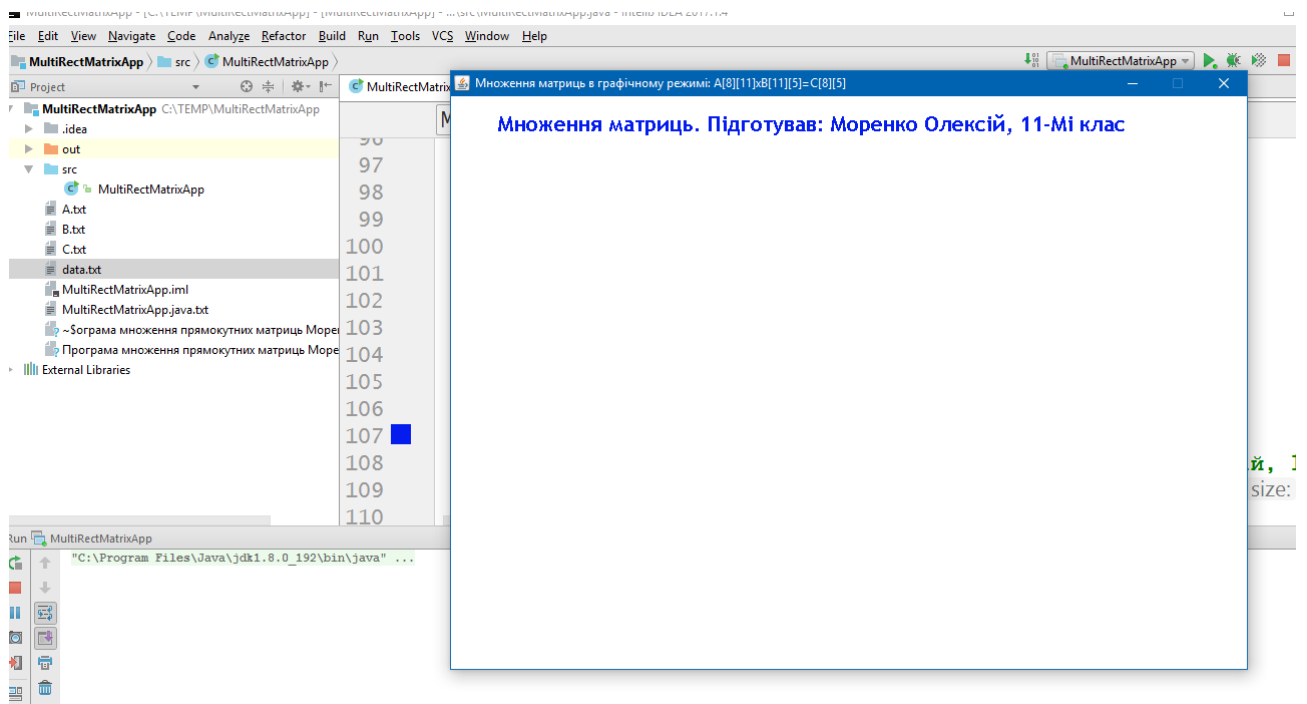
private static int rnd(int min, int max) {
    return (new Random()).ints(min, max).iterator().nextInt();
}
}
}

```

Графічна бібліотека java.awt.*

Введення/Виведення інформації на екран

Перший етап. Створення матриць $A[8][11]$ і $B[11][5]$. Виконання їх множення.
Результат - матриця $C[8][5]$. Дані записані у файли: A.txt, B.txt, C.txt, data.txt



A.txt

MultiRectMatrixApp > A.txt >	
Project	MultiRectMatrixApp.java x A.txt x B.txt x C.txt x data.txt x
MultiRectMatrixApp C:\TEMP\MultiRectMatrixApp	1 8
.idea	2 11
out	
src	3 8 2 1 8 2 8 7 8 4 1 7
MultiRectMatrixApp	4 1 4 6 6 2 5 7 3 6 6 1
A.txt	5 7 5 1 5 3 1 7 8 1 5 7
B.txt	6 5 2 5 8 8 6 6 5 7 6 2
C.txt	7 7 5 6 2 5 1 8 1 5 3 2
data.txt	8 2 2 6 6 6 2 8 7 2 2 8
MultiRectMatrixApp.iml	9 1 1 4 7 7 4 1 2 6 1 2
MultiRectMatrixApp.java.txt	10 1 5 6 1 4 7 7 2 8 2 2
~\$орамa множення прямокутних матриць More	11
Програма множення прямокутних матриць More	
External Libraries	

B.txt

MultiRectMatrixApp > B.txt >

Project

MultiRectMatrixApp C:\TEMP\MultiRectMatrixApp

.idea

out

src

MultiRectMatrixApp

A.txt

B.txt

C.txt

data.txt

MultiRectMatrixApp.iml

MultiRectMatrixApp.java.txt

~\$орамa множення прямокутних матриць More

Програма множення прямокутних матриць More

External Libraries

MultiRectMatrixApp.java x

A.txt x

B.txt x

C.txt x

data.txt x

1

11

2

5

3

5 7 4 5 3

4

1 3 7 4 6

5

8 3 4 7 4

6

5 3 6 7 8

7

1 7 2 2 8

8

5 8 3 8 2

9

3 6 3 5 1

10

8 1 8 3 7

11

1 6 6 6 4

12

1 4 6 2 5

13

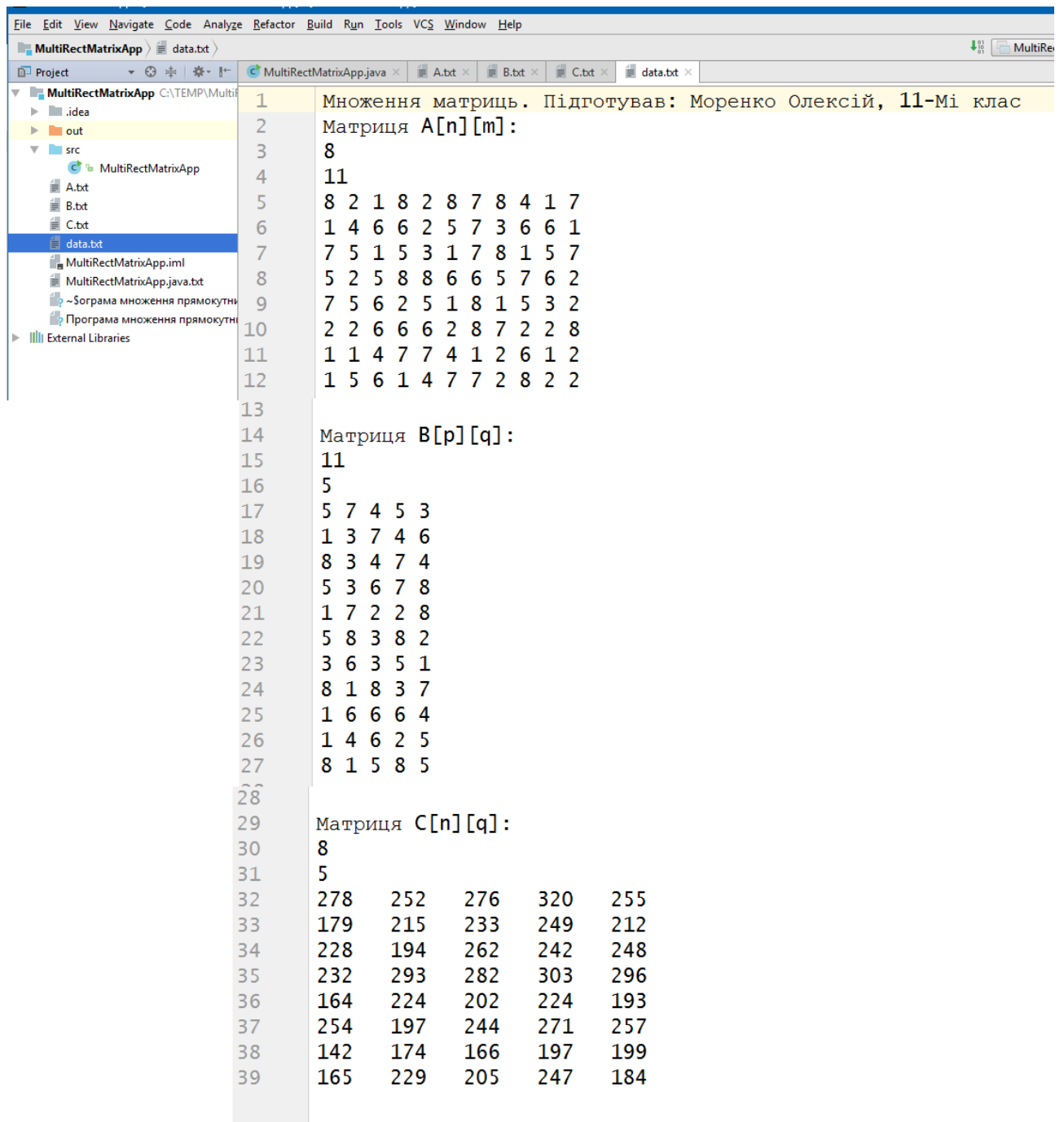
8 1 5 8 5

14

C.txt

MultiRectMatrixApp C.txt						
Project MultiRectMatrixApp.java A.txt B.txt C.txt data.txt						
MultiRectMatrixApp C:\TEMP\MultiRectMatrixApp	1	8				
.idea	2	5				
out	3	278	252	276	320	255
src	4	179	215	233	249	212
MultiRectMatrixApp	5	228	194	262	242	248
A.txt	6	232	293	282	303	296
B.txt	7	164	224	202	224	193
C.txt	8	254	197	244	271	257
data.txt	9	142	174	166	197	199
MultiRectMatrixApp.iml	10	165	229	205	247	184
MultiRectMatrixApp.java.txt						
~Сорема множення прямокутних матриць Море						
Програма множення прямокутних матриць Море						
External Libraries						

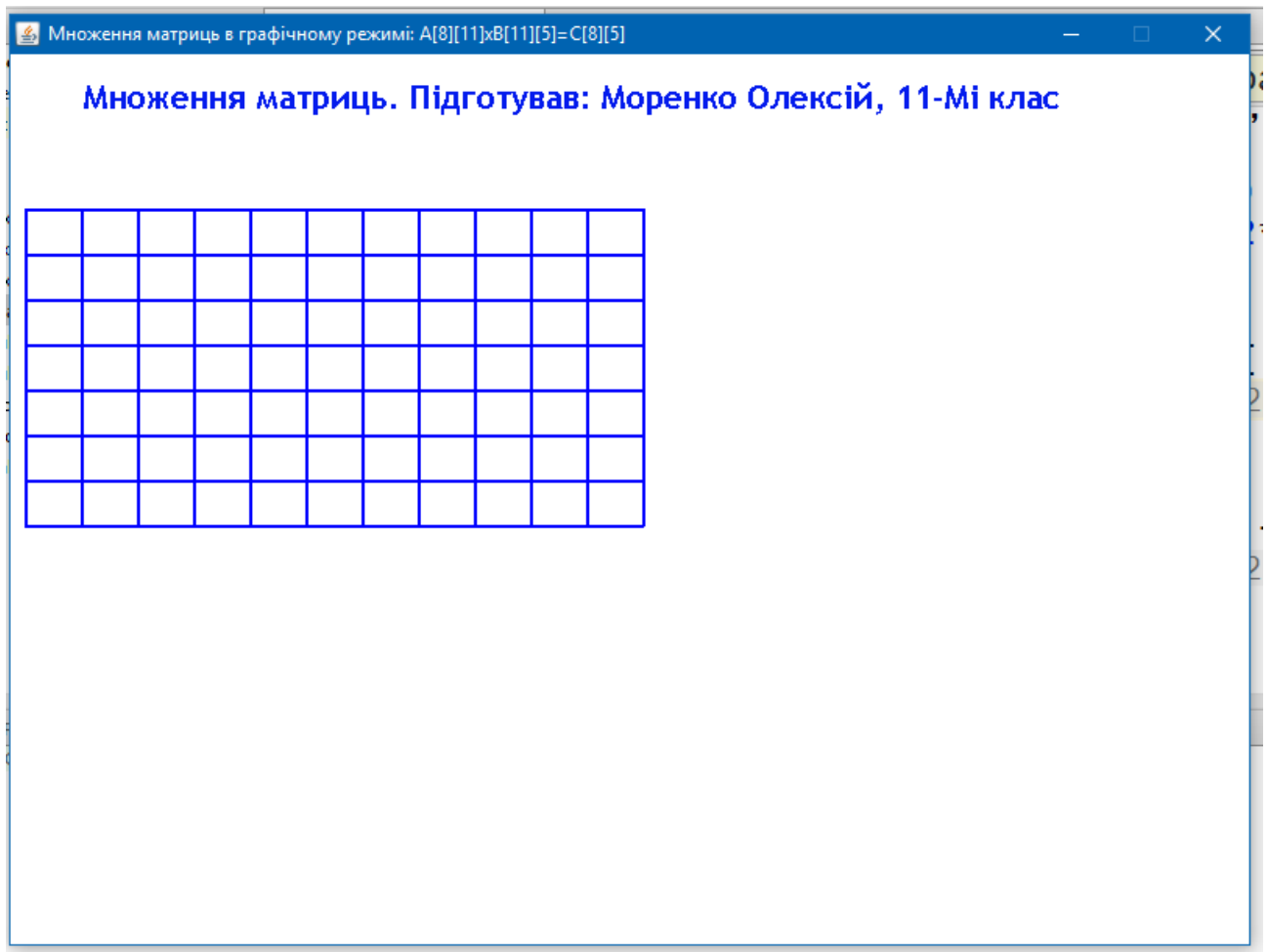
data.txt



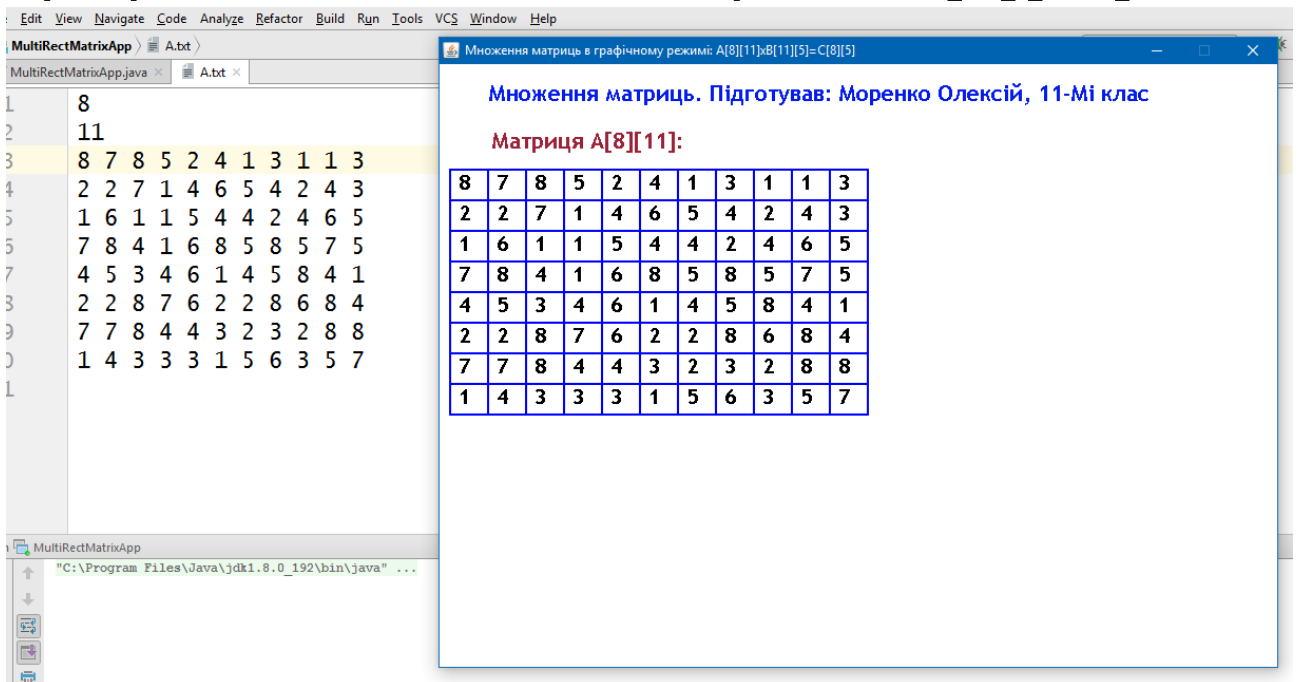
```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MultiRectMatrixApp > data.txt
Project MultiRectMatrixApp C:\TEMP\MultiRectMatrixApp
  .idea
  out
  src
    MultiRectMatrixApp
      A.txt
      B.txt
      C.txt
      data.txt
      MultiRectMatrixApp.iml
      MultiRectMatrixApp.java.txt
      ~$орграма множення прямокутних матриць
      Програма множення прямокутних матриць
  External Libraries

1  Множення матриць. Підготував: Моренко Олексій, 11-Мі клас
2  Матриця A[n][m]:
3  8
4  11
5  8 2 1 8 2 8 7 8 4 1 7
6  1 4 6 6 2 5 7 3 6 6 1
7  7 5 1 5 3 1 7 8 1 5 7
8  5 2 5 8 8 6 6 5 7 6 2
9  7 5 6 2 5 1 8 1 5 3 2
10 2 2 6 6 6 2 8 7 2 2 8
11 1 1 4 7 7 4 1 2 6 1 2
12 1 5 6 1 4 7 7 2 8 2 2
13
14 Матриця B[p][q]:
15 11
16 5
17 5 7 4 5 3
18 1 3 7 4 6
19 8 3 4 7 4
20 5 3 6 7 8
21 1 7 2 2 8
22 5 8 3 8 2
23 3 6 3 5 1
24 8 1 8 3 7
25 1 6 6 6 4
26 1 4 6 2 5
27 8 1 5 8 5
28
29 Матриця C[n][q]:
30 8
31 5
32 278 252 276 320 255
33 179 215 233 249 212
34 228 194 262 242 248
35 232 293 282 303 296
36 164 224 202 224 193
37 254 197 244 271 257
38 142 174 166 197 199
39 165 229 205 247 184
```

Малювання ліній



Графічно заповнено матрицю $A[8][11]$:



Графічно заповнені матриці A[8][11] і B[11][5]:

Множення матриць в графічному режимі: A[8][11]x8[11][5]=C[8][5]

Множення матриць. Підготував: Моренко Олексій, 11-Мі клас

Матриця A[8][11]:

1	1	6	2	5	1	1	5	7	7	6
7	6	1	7	1	3	7	5	2	8	1
8	6	3	4	6	1	4	2	3	7	6
1	7	3	3	6	6	7	6	4	2	5
7	4	4	7	1	3	8	6	3	5	1
3	4	8	3	7	3	3	4	3	8	2
8	1	3	8	2	5	6	5	2	3	6
6	3	5	4	5	3	5	4	3	7	4

Матриця B[11][5]:

4	8	1	5	7
2	7	5	1	6
5	5	4	5	2
7	6	5	6	6
6	4	3	4	5
2	5	2	3	8
2	2	6	8	6
1	8	2	6	2
4	1	4	3	3
4	6	1	1	6
6	1	6	5	6

Результат множення матриць виведено на графічний екран: Матриця C[8][5]

Множення матриць в графічному режимі: A[8][11]x8[11][5]=C[8][5]

Множення матриць. Підготував: Моренко Олексій, 11-Мі клас

Матриця A[8][11]:

1	5	3	8	2	4	7	1	2	1	8
1	4	1	6	2	1	2	4	1	7	5
1	6	1	1	2	8	8	4	4	2	2
3	4	3	3	1	5	8	8	5	5	8
2	5	8	1	7	2	8	5	5	7	4
8	8	5	8	1	2	7	2	8	8	5
4	1	6	6	3	8	3	5	4	8	7
5	3	2	8	8	4	1	4	3	2	5

Матриця B[11][5]:

7	3	4	2	7
1	3	2	1	7
8	8	8	2	1
4	2	8	5	1
7	2	5	4	3
1	4	2	1	5
8	1	6	5	3
3	2	6	1	2
2	2	1	4	4
3	5	6	4	7
6	3	5	2	5

Результат множення матриць - Матриця C[8][5]:

200	120	216	129	157
139	105	184	103	145
149	107	156	103	169
234	156	254	144	218
272	184	274	161	212
277	196	296	191	276
243	195	280	153	222
208	128	224	132	172

Лістинг проєкту

```
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;
```

```
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=8;
    public int m=11;
    public int p=11;
    public int q=5;
    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 of=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 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="Множення матриць. Підготував: Моренко Олексій, 11-Мі клас";
    g.setFont(new Font("Trebuchet Ms",Font.BOLD,21));
    g.drawString(text,50,miny+35);

```

////////////////////////////////////

```

ms=text;
ms+="\nМатриця A[n][m]:\n";
try {
    af.createNewFile();
    FileWriter writer=new FileWriter(af);
    text=n+"\n";
    text+=m+"\n";
    for (int i = 0; i < n ; i++) {
        for (int j = 0; j < m ; j++) {A[i][j]=rnd(1,9);
            text+=A[i][j]+" ";
        }
        text+="\n";
    }
    writer.write(text);
    writer.flush();
    writer.close();
    ms+=text;
} catch (IOException e) {
    e.printStackTrace();
}

```

////////////////////////////////////

```

ms+="\nМатриця B[p][q]:\n";
try {
    bf.createNewFile();
    FileWriter writer=new FileWriter(bf);
    text=p+"\n";
    text+=q+"\n";
    for (int i = 0; i < p ; i++) {
        for (int j = 0; j < q ; j++) {
            B[i][j]=rnd(1,9);

```

```

        text+=B[i][j]+" ";
    }
    text+="\n";
}
writer.write(text);
writer.flush();
writer.close();
ms+=text;
} catch (IOException e) {
    e.printStackTrace();
}

```

//

```

ms+="\nМатриця C[n][q]:\n";
try {
    cf.createNewFile();
    FileWriter writer=new FileWriter(cf);
    text=n+"\n";
    text+=q;
    for (int i = 0; i < n; i++) {
        text += "\n";
        for (int j = 0; j < q; j++) {
            C[i][j] = 0;
            for (int k = 0; k < m; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
            text += C[i][j] + " ";
        } ;
    }
    writer.write(text);
    writer.flush();
    writer.close();
    ms+=text;
} catch (IOException e) {
    e.printStackTrace();
}

```

//

```

try {
    of.createNewFile();
    FileWriter writer=new FileWriter(of);
    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(157, 35, 57));
text="Матриця A[8][11]:";
g.setFont(new Font("Trebuchet Ms",Font.BOLD,21));
g.drawString(text,minx+50,miny+60);
g.setColor(new Color(0,0,255));
dx= (int) ((maxx-minx)/(2*m));
dy= (int) ((maxy-miny-100)/(2*n));
x=minx+10; y=miny+70;
for (int i = 0; i <=n ; i++) {
    g.drawLine(x,y+i*dy,x+maxx-402,y+i*dy);
}

```

```

for (int i = 0; i <=m ; i++) {
    g.drawLine(x+i*dx,y,x+i*dx,y+maxy-365);
}
g.setColor(new Color(0,0, 0));
g.setFont(new Font("Trebuchet Ms",Font.BOLD,20));
x=x+dx/2-10; y=y+dy/2+5;
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(157, 35, 57));
text="Матриця B[11][5]:";
g.setFont(new Font("Trebuchet Ms",Font.BOLD,21));
g.drawString(text,minx+500,miny+60);
g.setColor(new Color(0,0,255));
dx= (int) ((maxx-minx)/(3*q));
dy= (int) ((maxy-miny-100)/(2*p));
x=minx+480; y=miny+70;
for (int i = 0; i <=p ; i++) {
    g.drawLine(x,y+i*dy,x+maxx-537,y+i*dy);
}
for (int i = 0; i <=q ; i++) {
    g.drawLine(x+i*dx,y,x+i*dx,y+maxy-365);
}
g.setColor(new Color(0,0, 0));
g.setFont(new Font("Trebuchet Ms",Font.BOLD,20));
x=x+dx/2-10; 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(157, 35, 57));
text="Результат множення матриць - Матриця C[8][5]:";
g.setFont(new Font("Trebuchet Ms",Font.BOLD,21));
g.drawString(text,minx+50,miny+325);
g.setColor(new Color(0,0,255));
dx= (int) ((maxx-minx)/(2*q));
dy= (int) ((maxy-miny-100)/(2*n));
x=minx+10; y=miny+335;
for (int i = 0; i <=n ; i++) {
    g.drawLine(x,y+i*dy,x+maxx-402,y+i*dy);
}
for (int i = 0; i <=q ; i++) {
    g.drawLine(x+i*dx,y,x+i*dx,y+maxy-365);
}
g.setColor(new Color(0,0, 0));
g.setFont(new Font("Trebuchet Ms",Font.BOLD,20));
x=x+dx/2-10; y=y+dy/2+7;
for (int i = 0; i <n ; i++) {
    for (int j = 0; j <q ; j++) {
        text= String.valueOf(C[i][j]);
        g.drawString(text,x+j*dx,y+i*dy);
    }
}

```



```
}
```

```
////////////////////////////////////
```

```
    public GrMultiRectMatrixApp() throws FileNotFoundException {  
        super("Множення матриць в графічному режимі:  
A[8][11]xB[11][5]=C[8][5]");  
        addWindowListener(this);  
        setLayout(null);  
        setLocation(80,80);  
        setSize(800,600);  
        this.setResizable(false);  
        show();  
    }  
}
```