МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ



**Звіт**

До лабораторної роботи №1

З дисципліни:

«Теорія прийняття рішень»

На тему:

**«Дослідження методів представлення бінарних**

**Відношень та операцій над ними»**

Виконав:

Студент групи КН-38

Макара С.В.

Прийняла:

Заяць М.М.

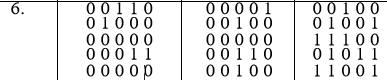
**Львів-2017**

**Мета:** Вивчення та порівняння різних способів представлення бінарних відношень. Засвоєння практичних навиків маніпулювання бінарними відношеннями в різних представленнях.

**Завдання:**

Реалізувати функції-члени для виконання наступних дій над відношеннями: перетину, об’єднання, різниці, симетричної різниці, доповнення, знаходження оберненого відношення, композиції, звуження.

Реалізувати функцію-член перевірки умови включення одного відношення в інше.

 За допомогою отриманої програми здійснити обчислення виразу: **K** = (**P**°**Q**) \ **R***d*. Значення відношень **P**,**Q**,**R** згідно варіанту.

Варіант №6:

**Текст програми:**

import java.util.Arrays;

import java.util.Scanner;

import java.util.StringTokenizer;

public class firsl {

static int[][] inputt(int nnum)

{

int [][] M = new int[nnum][nnum];

Scanner put = new Scanner(System.in);

for(int i = 0;i<nnum;i++)

for(int j = 0;j<nnum;j++)

{

M[i][j] = put.nextInt();

}

return M;

}

static void outputt(int[][] M) {

for (int i = 0; i < M.length; i++){

for (int j = 0; j < M[i].length; j++) {

System.out.print(M[i][j] + " ");

}

System.out.println();

}

}

static void choice(int [][] A, int [][] B, int [][] C)

{

int [][] R = null;

int ch, ch2 = 0;

do{

System.out.println("Яку дію над відношеннями виконати?:\n1 - Перетин\n"

+"2 - Об'єднання\n3 - Різниця\n4 - Симетрична різниця\n5 - Доповнення\n"

+"6 - Обернене відношення\n7 - Композиція\n8 - Двоїста матриця\n9 - Звуження\n10 - Включення\n"

+ "11 - Значення виразу");

Scanner put = new Scanner(System.in);

ch = put.nextInt()-1;

if(ch<1||ch>9) System.out.println("Повторіть вибір!");

} while(ch<1||ch>9);

switch (ch) {

case 0:

R = petetun(A, B); break;

case 1:

R = plus(A, B); break;

case 2:

R = minus(A, B); break;

case 3:

R = sminus(A, B); break;

case 4:

do{

System.out.println("1 - А\n2 - B");

Scanner putt = new Scanner(System.in);

ch2 = putt.nextInt();

if(ch2 == 1)

R = dopovv(A);

else if(ch2 == 2)

R = dopovv(B);

}while(ch2<1||ch2>2);

break;

case 5:

do{

System.out.println("1 - А\n2 - B");

Scanner putt = new Scanner(System.in);

ch2 = putt.nextInt();

if(ch2 == 1)

R = revers(A);

else if(ch2 == 2)

R = revers(B);

}while(ch2<1||ch2>2); break;

case 6:

R = compose(A, B); break;

case 7:

do{

System.out.println("1 - А\n2 - B");

Scanner putt = new Scanner(System.in);

ch2 = putt.nextInt();

if(ch2 == 1)

R = doubleOne(A);

else if(ch2 == 2)

R = doubleOne(B);

}while(ch2<1||ch2>2); break;

case 8:

do{

System.out.println("1 - А\n2 - B");

Scanner putt = new Scanner(System.in);

ch2 = putt.nextInt();

if(ch2 == 1)

R = littleone(A);

else if(ch2 == 2)

R = littleone(B);

}while(ch2<1||ch2>2); break;

case 10:

R = solution(A, B, C); break;

case 9:

if(minip(A, B)==1)

System.out.println("В включене в А");

else

System.out.println("В не включене в А");

break;

default: break;

}

if(ch!=9) outputt(R);

}

static int [][] petetun(int [][] A, int [][] B)

{

int [][] R = new int[A.length][A.length];

for(int i = 0;i<A.length;i++)

for(int j = 0;j<A.length;j++)

R[i][j] = A[i][j]&B[i][j];

return R;

}

static int[][] plus(int[][] A, int[][] B)

{

int [][] R = new int[A.length][A.length];

for(int i = 0;i<A.length;i++)

for(int j = 0;j<A.length;j++)

R[i][j] = A[i][j]|B[i][j];

return R;

}

static int[][] minus(int[][] A, int[][] B) {

int [][] R = new int[A.length][A.length];

for(int i = 0;i<A.length;i++)

for(int j = 0;j<A.length;j++)

if(A[i][j]==1&&B[i][j]==0)

R[i][j]=1;

else R[i][j]=0;

return R;

}

static int[][] sminus(int[][] A, int[][] B)

{

return plus(minus(A, B), minus(B, A));//(A\B)U(B\A)

}

static int[][] dopovv(int[][] A)

{

int [][] R = new int[A.length][A.length];

for (int i = 0; i < A.length; i++)

for (int j = 0; j < A.length; j++)

{

if (A[i][j] == 1)

R[i][j] = 0;

else R[i][j] = 1;

}

return R;

}

static int[][] revers(int[][] A)

{

int [][] R = new int[A.length][A.length];

for (int i = 0; i < A.length; i++)

for (int j = 0; j < A.length; j++)

R[i][j] = A[j][i];

return R;

}

static int[][] compose(int[][] A, int[][] B)

{

int [][] R = new int[A.length][A.length];

for (int i = 0; i < A.length; i++)

for (int j = 0; j < A[i].length; j++)

{

R[i][j] = 0;

for (int k = 0; k < A.length; k++)

{

R[i][j] += A[i][k] \* B[k][j];

}

}

for (int i = 0; i < A.length; i++)

for (int j = 0; j < B.length; j++)

if(R[i][j] > 0)

R[i][j] = 1;

return R;

}

static int [][] littleone(int [][] A)

{

Scanner read = new Scanner(System.in);

System.out.println("Кількість елементів звуженої матриці:");

int con = read.nextInt(), count\_i = -1, count\_j;

int [] temp = new int[con];

int [][] R = new int[con][con];

System.out.println("Порядкові номери множин:");

for(int i = 0;i<con;i++)

{

temp[i] = read.nextInt()-1;

}

for(int i = 0;i<con;i++)

{

++count\_i;

if(temp[count\_i]==i)

{

count\_j = 0;

for(int j = 0;j<con;j++)

if(temp[count\_j] == i)

{

R[count\_i][count\_j] = A[i][j];

count\_j+=1;

}

}

}

return R;

}

public static int minip(int [][] A, int [][] B)

{

for(int i = 0;i<A.length;i++)

for(int j = 0;j<A.length;j++)

if(A[i][j] == 0 && B[i][j]==1)

return 0;

return 1;

}

static int [][] doubleOne(int [][] A)

{

return revers(dopovv(A));

}

private static int[][] solution(int [][] P, int [][] Q, int [][] R)

{

return minus(compose(P, Q), dopovv(R));

}

public static void main(String[] args) {

int numm = 5;

//int[][] A = inputt(numm);

//int[][] B = inputt(numm);

int[][] A = { { 0, 0, 1, 1, 0 }, { 0, 1, 0, 0, 0 }, { 0, 0, 0, 0, 0 }, { 0, 0, 0, 1, 1 }, { 0, 0, 0, 0, 0 } };

int[][] B = { { 0, 0, 0, 0, 1 }, { 0, 0, 1, 0, 0 }, { 0, 0, 0, 0, 0 }, { 0, 0, 1, 1, 0 }, { 0, 1, 0, 0, 0 } };

int[][] C = { { 0, 0, 1, 0, 0 }, { 0, 1, 0, 0, 1 }, { 1, 1, 1, 0, 0 }, { 0, 1, 0, 1, 1 }, { 1, 1, 0, 0, 1 } };

int chh = 1;

do{

System.out.println("Матриця А:"); outputt(A);

System.out.println("Матриця B:"); outputt(B);

System.out.println("Матриця C:"); outputt(C);

choice(A, B, C);

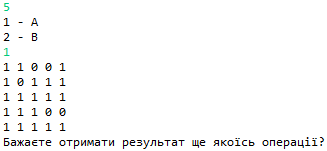
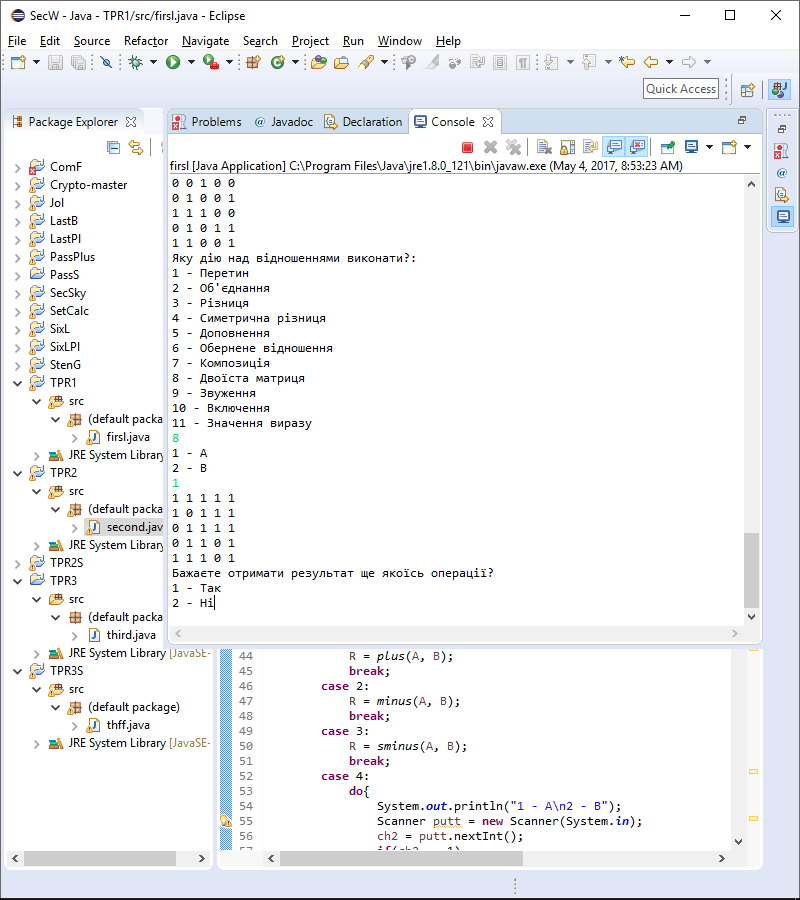
System.out.println("Бажаєте отримати результат ще якоїсь операції?"); System.out.println("1 - Так"); System.out.println("2 - Ні");

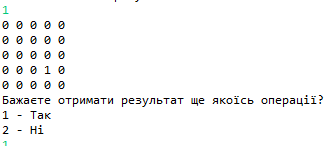
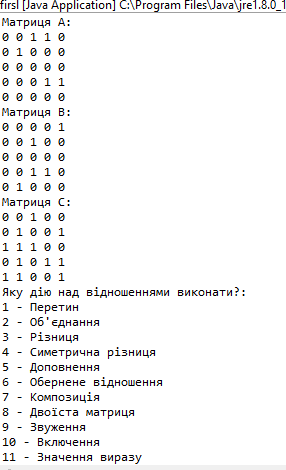
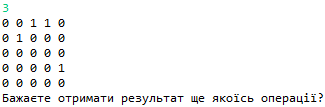
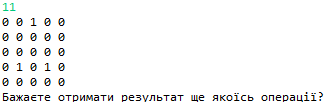
Scanner scn = new Scanner(System.in); chh = scn.nextInt();

}while(chh!=2);

}

}

******Приклади роботи програми:**

****

*Рис.1. Результати тестових прикладів*

**Висновки:** На даній лабораторній роботі мною було вивчено способи представлення бінарних відношень. Набуто навиків маніпулювання бінарними відношеннями в різних представленнях.