Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №4  
з дисципліни «Математичні методи в психології»

«Перевірка статистичних гіпотез (непараметричні критерії)»

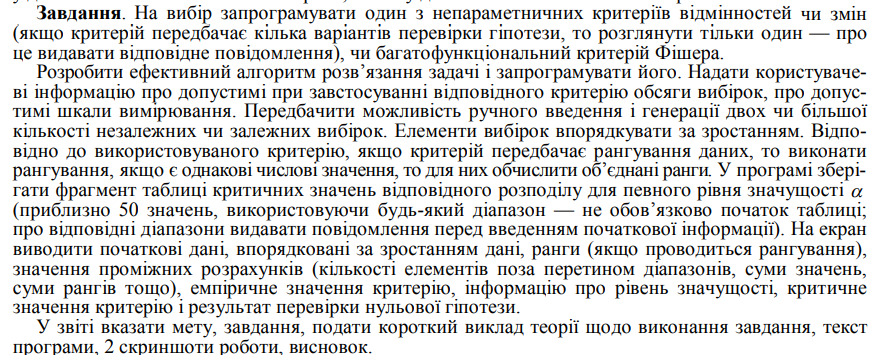
Варіант 8

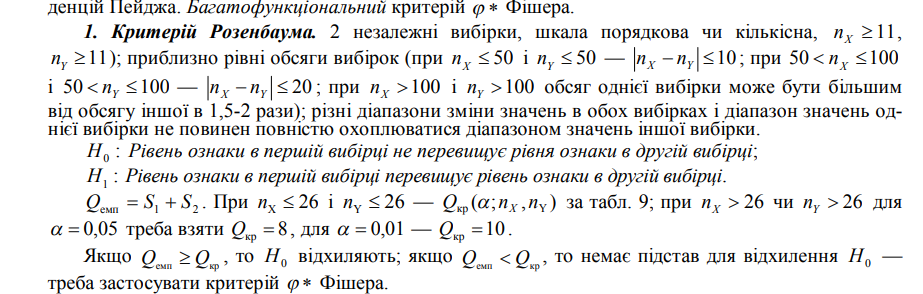
Виконав: студент групи ТР-12 Руденко В.І

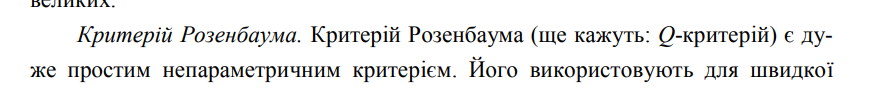
**Мета:**

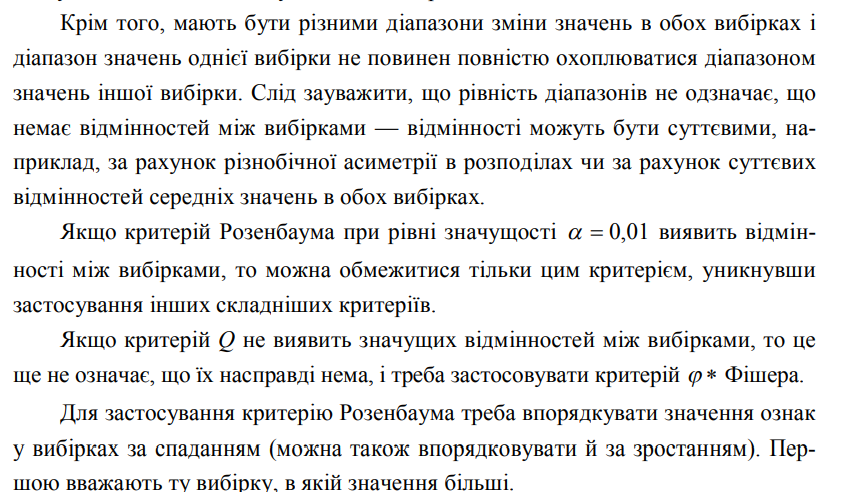
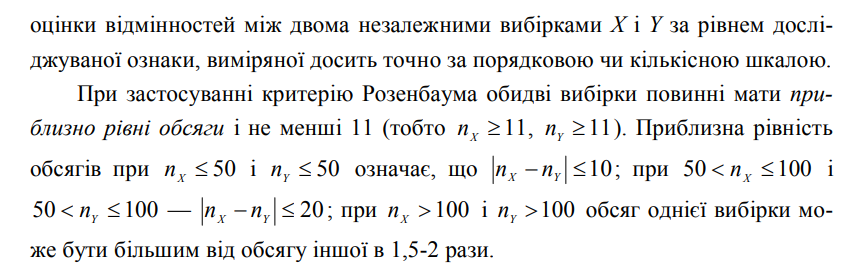
Розробити програмні засоби для перевірки статистичних гіпотез про: відмінності у двох чи більше незалежних вибірках; зміни у двох чи більше залежних вибірках

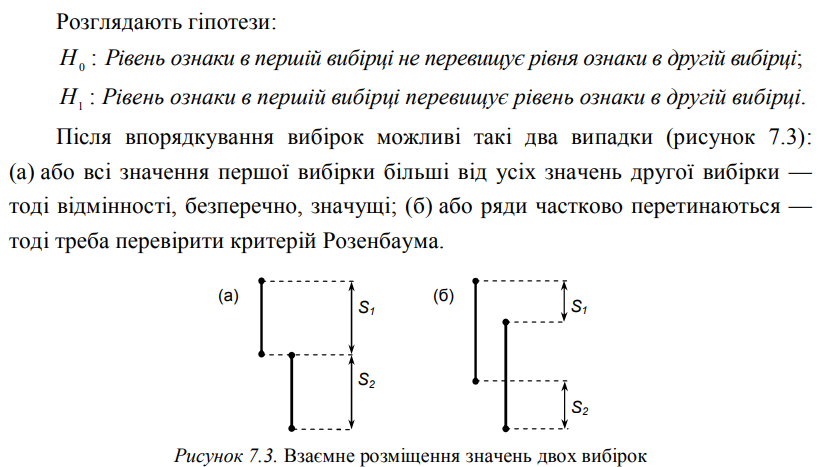
**Завдання:**

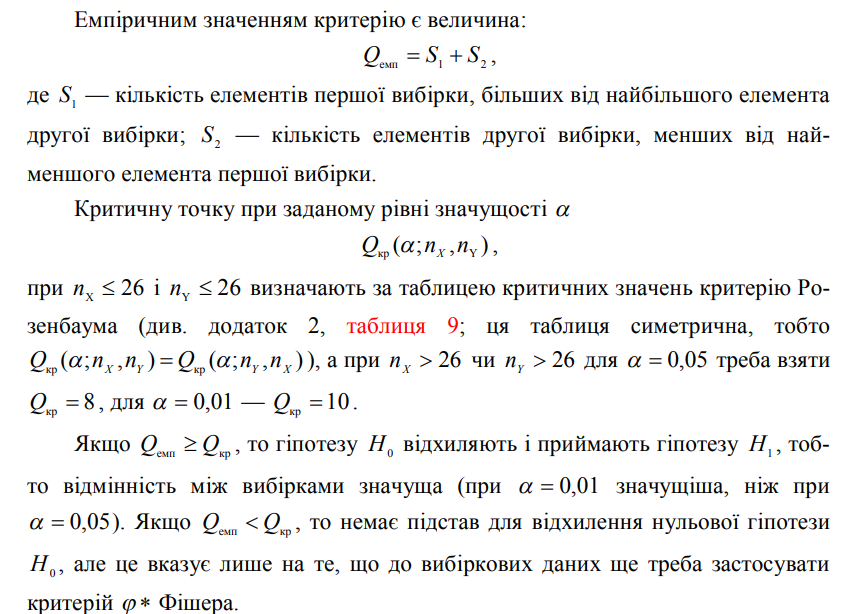


**Теорія:**









**Хід Роботи:**

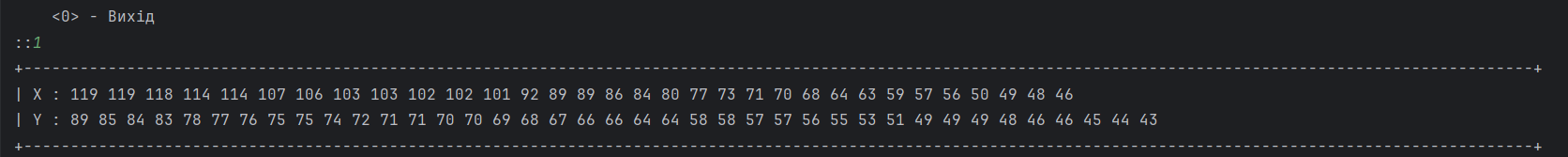


Table 1 Генерація Значень

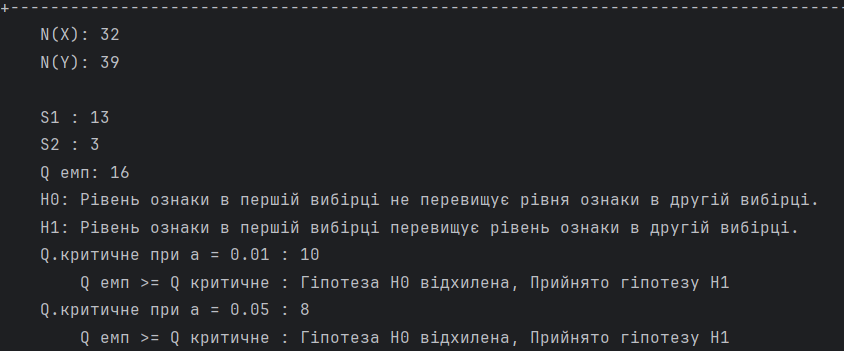


Table 3 Результати опрацювання

**Висновок:**

В рамках лабораторної роботи було розроблено програмні засоби для застосування критерію Розенбаума для перевірки статистичних гіпотез про відмінності між незалежними вибірками та зміни у залежних вибірках

**Код Програми:**  
*package LR4;  
  
import java.util.\*;  
  
class Data  
{  
 public final String hypothesisH0 = "H0: Рівень ознаки в першій вибірці не перевищує рівня ознаки в другій вибірці.";  
 public final String hypothesisH1 = "H1: Рівень ознаки в першій вибірці перевищує рівень ознаки в другій вибірці.";  
 public final int[][] QCriticalFor001 = new int[][]  
 {  
 {9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 12, 12, 12},  
 {9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12},  
 {9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 11, 11, 11},  
 {9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 11},  
 {9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10},  
 {9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10},  
 {10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10},  
 {10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10},  
 {10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10},  
 {10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9},  
 {11, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9},  
 {11, 11, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9},  
 {11, 11, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9},  
 {12, 11, 11, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9},  
 {12, 11, 11, 10, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9},  
 {12, 12, 11, 11, 10, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9}  
 };  
 public final int[][] QCriticalFor005 = new int[][]  
 {  
 {6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8},  
 {6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8},  
 {6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 7, 7, 6, 6, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8},  
 {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8},  
 {8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7},  
 {8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7},  
 {8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7},  
 {8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7},  
 {8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7},  
 {8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7}  
 };  
}  
  
enum UIState {  
 Exit,  
 RandomGeneration,  
 ManualInput,  
 InputWait;  
  
 public static UIState getByValue(int value) {  
 switch (value) {  
 case 0:  
 return Exit;  
 case 1:  
 return RandomGeneration;  
 case 2:  
 return ManualInput;  
 default:  
 return InputWait;  
 }  
 }  
}  
  
public class Main  
{  
 public static void main(String[] args)  
 {  
 System.out.println("<><><><>");  
 ConsoleUI();  
 }  
  
 private static void ConsoleUI()  
 {  
 Scanner scan = new Scanner(System.in);  
 UIState CurrentProgramPart = UIState.InputWait;  
 while(true)  
 {  
 System.out.print("Лабораторна робота 4??\n\t<1> - Згенерувати вибірки\n\t<2> - Ручний ввід\n\t<0> - Вихід\n::");  
 CurrentProgramPart = UIState.getByValue(scan.nextInt()) ;  
 switch (CurrentProgramPart)  
 {  
 case UIState.Exit:  
 return;  
 case UIState.InputWait:  
 System.out.println("Невіриний ввід....");  
 break;  
 default:  
 MainRun(CurrentProgramPart,scan);  
 break;  
 }  
 }  
 }  
  
 private static void MainRun(UIState currentProgramPart, Scanner scan)  
 {  
 scan.nextLine();  
 Integer[] ArrayX = new Integer[11];  
 Integer[] ArrayY = new Integer[11];  
 Data data = new Data();  
 switch (currentProgramPart)  
 {  
 case UIState.RandomGeneration:  
 Random rand = new Random();  
 rand.setSeed(System.currentTimeMillis());  
 int MinValue = 11,  
 MaxValue = 40,  
 Nx=0,  
 Ny=0;  
 do {  
 Nx = rand.nextInt(MaxValue) + MinValue;  
 Ny = rand.nextInt(MaxValue) + MinValue;  
 }while(!CheckNSize(Nx,Ny));  
 int YMin = 0;  
 int YMax = 0;  
 int XMin = 0;  
 int XMax = 0;  
 while(true)  
 {  
 YMin = rand.nextInt(MaxValue) + MinValue;  
 XMin = rand.nextInt(MaxValue) + MinValue;  
 YMax = rand.nextInt(MaxValue) + MaxValue;  
 XMax = rand.nextInt(MaxValue) + MaxValue;  
 if(YMin < XMin && YMax < XMax)  
 {  
 break;  
 }  
 }  
 ArrayX = ArrayGen(new Integer[Nx], rand,XMin,XMax);  
 ArrayY = ArrayGen(new Integer[Ny], rand, YMin,YMax);  
 break;  
 case UIState.ManualInput:  
  
 while(true)  
 {  
 System.out.print("Введіть розмірність вибірки X\n::");  
 Nx = scan.nextInt();  
 System.out.print("Введіть розмірність вибірки Y\n::");  
 Ny = scan.nextInt();  
 if(!CheckNSize(Nx, Ny))  
 {  
 System.out.println("НЄ...Не задовільняє умові розмірів....");  
 continue;  
 }  
 break;  
 }  
 ArrayX = new Integer[Nx];  
 ArrayY = new Integer[Ny];  
  
 do  
 {  
 System.out.println("Заповнення вибірок...X");  
 for(int i = 0; i < Nx; i++)  
 {  
 System.out.print("X[" + (i+1) + "] \n::");  
 ArrayX[i] = scan.nextInt();  
 }  
 System.out.println("Заповнення вибірок...Y");  
 for(int i = 0; i < Ny; i++)  
 {  
 System.out.print("Y[" + (i+1) + "] \n::");  
 ArrayY[i] = scan.nextInt();  
 }  
 int NXMin = Arrays.stream(ArrayX).min(Comparator.naturalOrder()).orElse(0);  
 int NYMin = Arrays.stream(ArrayY).min(Comparator.naturalOrder()).orElse(0);  
 int NXMax = Arrays.stream(ArrayX).max(Comparator.naturalOrder()).orElse(0);  
 int NYMax = Arrays.stream(ArrayY).max(Comparator.naturalOrder()).orElse(0);  
 if(NYMin > NXMin && NYMax > NXMax)  
 {  
 break;  
 }  
 if(NXMin > NYMin && NXMax > NYMax)  
 {  
 break;  
 }  
 System.out.println("Не задовільняє умову (Одна вибріка повністю покриває іншу)...");  
 }while(true);  
 break;  
 }  
 Arrays.sort(ArrayX, Collections.reverseOrder());  
 Arrays.sort(ArrayY, Collections.reverseOrder());  
 Arraysprint(ArrayX,ArrayY);  
 int S1Max = Arrays.stream(ArrayY).max(Comparator.naturalOrder()).orElse(0);  
 int S1= 0;  
 for (int num : ArrayX)  
 {  
 if (num > S1Max)  
 {  
 S1++;  
 }  
 }  
 int S2Min = Arrays.stream(ArrayX).min(Comparator.naturalOrder()).orElse(0);  
 int S2= 0;  
 for (int num : ArrayY)  
 {  
 if (num < S2Min)  
 {  
 S2++;  
 }  
 }  
 System.out.println("\tS1 : "+S1+"\n\tS2 : "+S2);  
 int Q = S1+S2;  
 System.out.println("\tQ емп: "+Q);  
 System.out.println("\t"+data.hypothesisH0);  
 System.out.println("\t"+data.hypothesisH1);  
  
 int Qcriticalfor001;  
 if(ArrayX.length <= 26 && ArrayY.length <= 26)  
 {  
 Qcriticalfor001 = data.QCriticalFor001[ArrayX.length-11][ArrayY.length-11];  
 }  
 else  
 {  
 Qcriticalfor001 = 10;  
 }  
 System.out.println("\tQ.критичне при a = 0.01 : "+Qcriticalfor001);  
 System.out.printf("\t\t%s\n",String.format((Qcriticalfor001<=Q?"Q емп >= Q критичне : Гіпотеза H0 відхилена, Прийнято гіпотезу H1":"Q емп < Q критичне : Немає підстав для відхилення гіпотези H0, треба застосувати критерій Фішера")));  
 int Qcriticalfor005;  
 if(ArrayX.length <= 26 && ArrayY.length <= 26)  
 {  
 Qcriticalfor005 = data.QCriticalFor005[ArrayX.length-11][ArrayY.length-11];  
 }  
 else  
 {  
 Qcriticalfor005 = 8;  
 }  
 System.out.println("\tQ.критичне при a = 0.05 : "+Qcriticalfor005);  
 System.out.printf("\t\t%s\n",String.format((Qcriticalfor005<=Q?"Q емп >= Q критичне : Гіпотеза H0 відхилена, Прийнято гіпотезу H1":"Q емп < Q критичне : Немає підстав для відхилення гіпотези H0, треба застосувати критерій Фішера")));  
 }  
  
 private static void Arraysprint(Integer[] arrayX, Integer[] arrayY)  
 {  
 String border = "+-----";  
 int charsize = Math.max(arrayX.length, arrayY.length);  
 for(int i = 0; i < charsize; i++)  
 {  
 border += "----";  
 }  
 border += "+";  
 System.out.println(border);  
 System.out.print("| X : ");  
 for(int i = 0; i < arrayX.length; i++)  
 {  
 System.out.print(arrayX[i] + " ");  
 }  
 System.out.println();  
 System.out.print("| Y : ");  
 for(int i = 0; i < arrayY.length; i++)  
 {  
 System.out.print(arrayY[i] + " ");  
 }  
 System.out.println();  
 System.out.println(border);  
 System.out.println("\tN(X): "+arrayX.length);  
 System.out.println("\tN(Y): "+arrayY.length);  
 System.out.println();  
 }  
  
 private static Integer[] ArrayGen(Integer[] array, Random rand, int min, int max)  
 {  
 Integer[] result = new Integer[array.length];  
 for (int i = 0; i < array.length; i++)  
 {  
 result[i] = rand.nextInt(max)+min;  
 }  
 return result;  
 }  
  
 private static boolean CheckNSize (int Nx, int Ny)  
 {  
 if(Nx < 11 || Ny < 11)  
 {  
 return false;  
 }  
 if(Nx <= 50 && Ny <= 50)  
 {  
 if(Math.abs(Nx-Ny) <= 10)  
 {  
 return true;  
 }  
 else return false;  
 }  
 if(Nx <= 100 && Ny <= 100 )  
 {  
 if(Math.abs(Nx-Ny) <= 20)  
 {  
 return true;  
 }  
 else return false;  
 }  
 if(Nx > 100 && Ny > 100)  
 {  
 if((Nx/Ny <= 2|| Ny/Nx <= 2))  
 {  
 return true;  
 }  
 else return false;  
 }  
 return false;  
 }  
}*