

ANS Elbląg

**Instytut Informatyki Stosowanej im. Krzysztofa
Brzeskiego**

Programowanie obiektowe I – laboratorium

Studium Stacjonarne, sem. 3, 2022/2023

Sprawozdanie nr : 4,

nr grupy: 1,

dzień: wtorek,

godz. 12:00.

Data wykonania ćwiczenia: 08.11

Data oddania sprawozdania: 14.11

Nazwisko i imię: Kuczawski Kacper

Nr albumu: 20195

Nazwa pliku : lab4_kuczawski_kacper20195

1. a) Utworzyć klasę publiczną **Lab4Zad1**. W metodzie **main()** tej klasy zdefiniować tablicę tablicę **x[8]** liczb typu **double**. Dodać do tej klasy metody statyczne **readTab1D**, **printTab1D**. Metody te mają posiadać nagłówki o postaci

```
static void readTab1D( double x[]);
static printTab1D( double x[]);
```

W metodzie **main()** wczytać i wydrukować tablicę **x** (do wyznaczenia liczby elementów tablicy wewnętrz z każdej z metod zastosować pole **length**).

b) utworzyć kopię tablicy **x** jako **xKopia** przy zastosowaniu metody **copyOf** klasy **Arrays**
c) powiększyć tablicę dwukrotnie (nazwa **xDoubleSize**), wypełnić liczbami pseudolosowymi i wydrukować.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class Lab4Zad1{
    public static void main(String args[]){
        double x[]=new double[8];
        Lab4Zad1.readTab1D(x);
        Lab4Zad1.printTab1D(x);
        Random r = new Random();
        double[] xKopia=Arrays.copyOf(x,8);
        double[] xDoubleSize=new double[16];
        int l = xDoubleSize.length;
        for(int i=0;i<l;i++){
            xDoubleSize[i]=r.nextDouble(100);
        }
        Lab4Zad1.printTab1D(xDoubleSize);
        double[] z= Arrays.copyOfRange(x,4,8);
        Lab4Zad1.printTab1D(z);
    }
    static void readTab1D(double tab[]){
        int n;
        n=tab.length;
        Scanner sc=new Scanner(System.in);
        for(int i=0; i<n; i++){
            System.out.print("Podaj " + (i+1) + " wartoSC:");
            tab[i]=sc.nextDouble();
        }
    }
    static void printTab1D(double tab[]){
        for(double element:tab) System.out.println(element);
    }
}
```

2. Utworzyć klasę publiczną **Lab4Zad2**. Dodać do tej klasy metody statyczne **readTab2D**, **printTab2D**. Metody te mają posiadać nagłówki o postaci

```
static void readTab2D( double y[][]);
static printTab1D( double y[][]);
```

W metodzie **main()** zdefiniować tablicę **tabX[3][3]**, następnie wczytać tę tablicę i wydrukować. W poszczególnych metodach wyznaczyć liczbę wierszy i liczbę wierszy przy zastosowaniu pola **length**.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class Lab4Zad2{
    public static void main(String args[]){
        double tabX[][]=new double[3][3];
        Lab4Zad2.readTab2D(tabX);
        Lab4Zad2.printTab2D(tabX);
    }
    static void readTab2D(double tab[][]){
        int w, k;
        k=tab.Length;
        w=tab[0].Length;
        Scanner sc=new Scanner(System.in);
        for(int i=0; i<w; i++){
            for(int j=0; j<k; j++){
                System.out.print("Podaj tab[" + (i+1) + "][" + (j+1) + "]: ");
                tab[i][j]=sc.nextInt();
            }
        }
    }
    static void printTab2D(double tab[][]){
        for(double x[]:tab){
            for(double y: x){
                System.out.println(y+" ");
            }
            System.out.println();
        }
    }
}
```

3. a) Utworzyć klasę publiczną **Lab4Zad3**. W metodzie **main()** tej klasy zainicjować tablicę **tab[8]** typu **int**. Napisać metodę **sorttab** realizującą sortowanie tablicy niemalejąco i nierosnąco. Metoda otrzymuje tablicę i zwraca jej dwie posortowane wersje. W metodzie do sortowania niemalejąco wykorzystać metodę **sort** klasy **Arrays**. Aby uzyskać tablicę posortowaną nierosnąco, odwrócić kolejność elementów. Metoda **sorttab** posiada nagłówek o postaci

```
static int [] readTab1D( double x[], double xSortAsc[]);
```

Z postaci nagłówka wynika, że tablicę posortowaną niemalejąco należy zwrócić przez parametry, a posortowaną nierosnąco jako wartość zwracaną metody. Wydrukować posortowane tablice.

b) wyszukać zadany element w tablicy posortowanej niemalejąco przy użyciu metody **binarySearch** i wydrukować jego indeks,

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
public class Lab4Zad3{
    public static void main(String args[]){
        int[] tab = new int[8];
        int[] xSortAsc=new int[8];
        int element = 5;
        Lab4Zad3.readTab1D(tab);
        System.out.println("Początkowa tablica:");
        Lab4Zad3.printTab1D(tab);
        xSortAsc= Lab4Zad3.sorttab(tab,xSortAsc);
        System.out.println("Tablica po sortowaniu rosnąco:");
        Lab4Zad3.printTab1D(tab);
        System.out.println(element+" zostało znalezione na miejscu:
"+Arrays.binarySearch(tab,element));
        System.out.println("Tablica po sortowaniu malejąco:");
        Lab4Zad3.printTab1D(xSortAsc);
    }
    static int[] sorttab(int tab[], int xSortAsc[]){
        int i,l,element=5;
        l = tab.Length;
        Arrays.sort(tab);
        for(i=0;i<l;i++){
            xSortAsc[i]=tab[(l-1)-i];
        }
        return xSortAsc;
    }
    static void printTab1D(int tab[]){
        int dl=tab.Length;
        for(int i=0;i<dl;i++){
            System.out.println("x["++(i+1)+"]: "+tab[i]);
        }
    }
    static void readTab1D(int tab[]){
    }
```

```

Scanner sc=new Scanner(System.in);
int dl=tab.Length;
for(int i=0;i<dl;i++){
    System.out.printf("x["++(i+1)+"]: ");
    tab[i]=sc.nextInt();
}
}

```

Wyniki Zadania 1	Wyniki Zadania 2	Wyniki Zadania 3
Podaj 8 wartości: 3.0 5.0 9.0 7.0 1.0 2.0 3.0 8.0 55.09747760210667 47.79625644591691 2.789784337810808 45.23666194217327 23.360924060178633 13.817321866167742 55.783177431467614 52.00038907959139 89.84601350632568 82.77374809726615 25.385887397973896 89.12272138523076 53.38082767952507 78.62476157158929 84.24103097841477 34.654122880276525 1.0 2.0 3.0 8.0	Podaj tab[1][1]: 4 Podaj tab[1][2]: 8 Podaj tab[1][3]: 1 Podaj tab[2][1]: 2 Podaj tab[2][2]: 9 Podaj tab[2][3]: 4 Podaj tab[3][1]: 5 Podaj tab[3][2]: 2 Podaj tab[3][3]: 9 4.0 8.0 1.0 2.0 9.0 4.0 5.0 2.0 9.0	x[1]: 5 x[2]: 0 x[3]: 3 x[4]: 9 x[5]: 7 x[6]: 1 x[7]: 8 x[8]: 4 Tablica po sortowaniu rosnąco: x[1]: 0 x[2]: 1 x[3]: 3 x[4]: 4 x[5]: 5 x[6]: 7 x[7]: 8 x[8]: 9 5 zostało znalezione na miejscu: 4 Tablica po sortowaniu malejąco: x[1]: 9 x[2]: 8 x[3]: 7 x[4]: 5 x[5]: 4 x[6]: 3 x[7]: 1 x[8]: 0

4. a) Utworzyć klasę publiczną **Lab4Zad4**. W metodzie **main()** tej klasy stworzyć obiekt **a** typu **BigInteger** przy zastosowaniu konstruktora (liczba 23452347654222) oraz obiekt **b** stosując metodę **valueOf**.

- b) Obliczyć dla **a** i **b** sumę, różnicę, iloraz, iloczyn oraz **a modulo b**. Wydrukować wyniki.
- c) Utworzyć obiekt **c** typu **BigInteger** i obliczyć wartość wyrażenia **c*(a+b-100)** . Wydrukować wynik.

Treść:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
public class Lab4Zad4{
    public static void main(String args[]){
        BigInteger a= new BigInteger("23452347654222");
        BigInteger b=BigInteger.valueOf(100);
        System.out.println("a: "+a);
        System.out.println("b: "+b);
        System.out.println("a + b = "+a.add(b).toString());
        System.out.println("a - b = "+a.subtract(b).toString());
        System.out.println("a * b = "+a.multiply(b).toString());
        System.out.println("a / b = "+a.divide(b).toString());
        System.out.println("a mod b = "+a.mod(b).toString());
        BigInteger c = new BigInteger("15000");
        BigInteger pom = a.add(b);
        BigInteger dzialanie=c.multiply(pom.add(BigInteger.valueOf(100)));
        System.out.printf("\n"+c+"*("+a+"+"+b+"-100)="+dzialanie);
    }
}
```

Wyniki:

```
a: 23452347654222
b: 100
a + b = 23452347654322
a - b = 23452347654122
a * b = 2345234765422200
a / b = 234523476542
a mod b = 22

15000*(23452347654222+100-100)=351785214816330000
Process finished with exit code 0
```