



/TheVampi



LAB 03

Pasar LAB 01 en Java usando P00

Estructura de
Datos

Martinez Rivera Luis Fernando
Ruiz Olvera Alfonso Emilio





Descripción del problema

Instrucciones:

1) Resuelve en Java la representación de las siguientes operaciones de conjuntos utilizando el paradigma orientado a objetos utilizando TDA

Determinar:

$A \cup B$

$B \cap C$

Entrada sean los conjuntos

$U = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$

$A = \{2,4,6,8,10,12,14\}$

$B = \{1,2,3,5,8,13\}$

$C = \{1,2,4,6,7,10,11,13\}$

Proyecto disponible para descargar en:

<https://github.com/TheVampi/DataStructures-TecNM/tree/main/Java/lab02ConjuntosPOO>

Código fuente

```
package lab02conjuntospoo;

import javax.swing.JOptionPane;

public class Lab02ConjuntosPOO {

    public static void main(String[] args) {
        String U = "1,2,3,4,5,6,7,8,9,10,11,12,13,14", A = "2,4,6,8,10,12,14", B = "1,2,3,5,8,13", C = "1,2,4,6,7,10,11,13";
        conjunto Obj1 = new conjunto();
        String Resultados = "AUB\n" + "[" + Obj1.union(A, B) + "]\n" + "BnC\n" + "[" +
        Obj1.interseccion(B, C) + "]\n" + "C\n" + "[" + Obj1.complementoResta(U, C) + "]\n" + "BUC\n" +
        "[" + Obj1.union(B, C) + "]\n" + "A∩C\n" + "[" + Obj1.interseccion(A, C) + "]\n" + "A'\n" + "[" +
        Obj1.complementoResta(U, A) + "]\n" + "B-A\n" + "[" + Obj1.complementoResta(B, A) + "]\n" + "C-
        B'\n" + "[" + Obj1.complementoResta(C, Obj1.complementoResta(U, B)) + "]\n";
        JOptionPane.showMessageDialog(null, "Dado los siguientes conjuntos resuelve las siguientes
        operaciones.\n" + "U= [" + U + "]\n" + "A= [" + A + "]\n" + "B= [" + B + "]\n" + "C= [" + C + "]\n" +
        Resultados, "02 Laboratorio", 1);
    }
}
```





```
}  
}  
  
class conjunto {  
  
    String[] conjunto_1, conjunto_2, conjunto_aux;  
    String conjunto_Resultante = "";  
  
    String union(String A, String B) {  
        String aux2 = "";  
        conjunto_Resultante = "";  
        conjunto_Resultante = complementoResta(A, B);  
        conjunto_Resultante += "," + B;  
        conjunto_aux = conjunto_Resultante.split("\\\\,");  
        conjunto_Resultante = "";  
  
        for (int i = 1; i < conjunto_aux.length; i++) {  
            for (int k = 1; k <= (conjunto_aux.length - 1); k++) {  
                if (Integer.parseInt(conjunto_aux[k - 1]) > Integer.parseInt(conjunto_aux[k])) {  
                    aux2 = conjunto_aux[k - 1];  
                    conjunto_aux[k - 1] = conjunto_aux[k];  
                    conjunto_aux[k] = aux2;  
                }  
            }  
        }  
        for (int i = 0; i < conjunto_aux.length; i++) {  
            conjunto_Resultante += conjunto_aux[i] + ",";  
        }  
        conjunto_Resultante = conjunto_Resultante.substring(0, conjunto_Resultante.length() - 1);  
        return conjunto_Resultante;  
    }  
  
    String interseccion(String A, String B) {  
        conjunto_Resultante = "";  
        conjunto_1 = A.split("\\\\,");  
        conjunto_2 = B.split("\\\\,");  
  
        for (int i = 0; i < conjunto_2.length; i++) {  
            for (int k = 0; k < conjunto_1.length; k++) {  
                if (conjunto_1[k].equals(conjunto_2[i])) {  
                    conjunto_Resultante += conjunto_2[i] + ",";  
                }  
            }  
        }  
        conjunto_Resultante = conjunto_Resultante.substring(0, conjunto_Resultante.length() - 1);  
        return conjunto_Resultante;  
    }  
}
```



```
String complementoResta(String U, String B) {  
    int aux = 0;  
    conjunto_Resultante = "";  
    conjunto_1 = U.split("\\,");  
    conjunto_2 = B.split("\\,");  
  
    for (int i = 0; i < conjunto_1.length; i++) {  
        for (int k = 0; k < conjunto_2.length; k++) {  
            if (conjunto_1[i].equals(conjunto_2[k])) {  
                aux = 1;  
                k = conjunto_2.length;  
            }  
        }  
        if (aux == 0) {  
            conjunto_Resultante += conjunto_1[i] + ",";  
        }  
        aux = 0;  
    }  
    conjunto_Resultante = conjunto_Resultante.substring(0, conjunto_Resultante.length() - 1);  
    return conjunto_Resultante;  
}
```

Capturas de ejecución

