

Objetos con arreglos y sobrecarga de operadores

Revisemos el siguiente ejemplo en el que se incluye una clase para representar conjuntos de números enteros. Recuerda que un conjunto es una colección de elementos en la que no hay elementos repetidos.

```
#include <iostream.h>

class Conjunto
{
    friend Conjunto operator +(Conjunto, Conjunto); //union de conjuntos
public:
    int Pertenece(int valor);
    void PideDatos();
    void Muestra();
private:
    int arr[20];
    int cant;
};

// Pide al usuario los elementos del conjunto, si se da un valor duplicado
// manda un mensaje de error
void Conjunto::PideDatos()
{
    int n, valor;

    cout<<"Cuantos valores hay en el conjunto ";
    cin>>n;
    cant = 0;
    while (cant < n)
    {
        cout<<"Teclea un elemento ";
        cin>>valor;
        if (Pertenece(valor) == 1)
            cout<<"Error, este elemento está duplicado"<<endl;
        else
        {
            arr[cant] = valor;
            cant++;
        }
    }
}

// Regresa 1 si el valor es elemento del conjunto y 0 si no lo es
int Conjunto::Pertenece(int valor)
{
    int encontro = 0;
    for (int i = 0; i < cant; i++)
        if (valor == arr[i])
```

```

        encontro = 1;
    return encontro;
}

// Muestra en la pantalla los elementos del conjunto
void Conjunto::Muestra()
{
    for (int i = 0; i < cant; i++)
        cout<<arr[i]<<endl;
}

// Funcion friend de la clase de los conjuntos (por eso puede usar los datos miembro
// de la clase de los conjuntos) que hace la union de dos conjuntos
Conjunto operator +(Conjunto A, Conjunto B)
{
    Conjunto Res;
    int i;

    // copio los valores del conjunto A al conjunto Res
    for (i=0; i<A.cant; i++)
        Res.arr[i] = A.arr[i];
    Res.cant = A.cant;

    // paso al conjunto Res los elementos del conjunto B que no estan en Res
    for (i=0; i<B.cant; i++)
        if (Res.Pertenece(B.arr[i]) == 0) // si el elemento no esta en el conjunto Res
            Res.arr[Res.cant++] = B.arr[i]; //nota que aquí se incrementa el contador cant de
Res
    return Res;
}

int main()
{
    Conjunto A, B, C;

    cout<<"Teclea los valores del conjunto A"<<endl;
    A.PideDatos();

    cout<<"Teclea los valores del conjunto B"<<endl;
    B.PideDatos();

    cout<<"La union de los conjuntos A y B es "<<endl;
    C = A + B;
    C.Muestra();

    return 0;
}

```

Ejercicio

Copia el ejemplo que se incluye en esta sección y pruébalo.

Agrega una función que realice la intersección de 2 conjuntos; sobrecarga el operador * para representar la intersección. Recuerda que la intersección de 2 conjuntos A y B es el conjunto

formado por todos los elementos que se encuentran en ambos conjuntos.

[ver solución](#)

Ligas sugeridas

<http://www.cplusplus.com/doc/tutorial/>

<http://www.cs.wustl.edu/~schmidt/C++/>

[*Regresar*](#)