

La clase conjuntoPlus.

Aplicando herencia crea una clase **conjuntoPlus** que hereda de la clase **conjunto** y que permita prohibir un intervalo de fracciones distintas para cada objeto que se instancie de dicha clase, así como llevar la cuenta del número de objetos **conjuntoPlus** creados.

Proporcione un **constructor** que cree un **conjuntoPlus** vacío con un intervalo de fracciones prohibidas determinado (indicado por parámetro) y un **constructor** sin parámetros que cree un **conjuntoPlus** vacío con el intervalo de fracciones prohibidas por defecto de $[0/1, 1/1]$.

La clase debe proporcionar un método **intervalo_por_defecto(f1, f2)** que permita cambiar el intervalo de fracciones prohibidas por defecto a la indicada por parámetro ($[f1, f2]$). El cambio solo podrá realizarse si en el momento de invocar dicho método no existe ningún objeto instanciado, de manera que si existe alguno el cambio no debe ser posible. Proporcione un método **cambiar_intervalo(f1, f2)** que permita modificar el intervalo de fracciones prohibidas para el objeto **conjuntoPlus** que lo invoca (al ejecutarse dicho método se deberán eliminar, si existieran, todas las fracciones de **conjuntoPlus** que estén dentro del nuevo intervalo prohibido). Sobrecargue los métodos **agregar()**, **ver()** y **operator <<** heredados de **conjunto** para que permitan, respectivamente, agregar sólo aquellas fracciones no prohibidas y visualizar el contenido de **conjuntoPlus** según el siguiente formato:

{ elementos del conjunto } – [intervalo fracciones prohibidas] (intervalo por defecto)

Ej: { $2/1$ $3/2$ $4/2$ } – $[0/1$ $1/1]$ ($0/1$ $1/1$) { $1/3$ } – $[1/5$ $1/4]$ ($0/1$ $1/1$)
 { } – $[1/1$ $5/1]$ ($0/1$ $1/1$) { $1/1$ } – $[2/1$ $2/1]$ ($7/2$ $9/2$)

Nota:

- No está permitido crear ningún método adicional distinto a los aquí pedidos, ni añadir ningún método adicional a las clases **conjunto** y **fracc** de las que hereda.
- La clase **conjunto** de la que hereda debe ser conjunto2 (versión dinámica).
- Los nombres de la clase y métodos deben ser los indicados en negrita.
- Todos los atributos deben ser **private** tanto en esta clase como de las que hereda.

Las clases utilizadas deben ser guardadas en una carpeta llamada conjuntoPlus1.

Implemente la clase **conjuntoPlus** para que el programa siguiente produzca exactamente la salida indicada:

PruebaPlus1.cpp

```

#include <cstdlib>
#include <iostream>
#include "fracc.hpp" //definicion de la clase fracc
#include "conjunto.hpp" //definicion de la clase conjunto
#include "ConjuntoPlus.hpp" //definicion de la clase ConjuntoPlus

using namespace std;

int main(int argc, char *argv[]) {
    fracc f1(4,2) f2(3);
    conjuntoPlus *p = new conjuntoPlus, *q;
    cout << "p: "; cout << *p << endl; //p: { } - [0/1 1/1] (0/1 1/1)
    conjuntoPlus::intervalo_por_defecto(fracc(5,5), fracc(6,5)); //cambio no permitido
    delete p;
    conjuntoPlus::intervalo_por_defecto(1, 2); //intervalo por defecto: [1/1, 2/1]
    p = new conjuntoPlus(3, fracc(8,3)); //p: { } - [3/3 4/3] (1/1 2/1)
    p->agregar(1); p->agregar(fracc(8,6));
    p->agregar(f1); //p: { 4/2 } - [3/3 4/3] (1/1 2/1)
    q = new conjuntoPlus(*p);
    cout << "p: "; cout << *p << endl; //p: { 2/1 } - [1/1 4/3] (1/1 2/1)
    delete p;
    conjuntoPlus::intervalo_por_defecto(fracc(0,1), 2); //cambio no permitido
    q->agregar(f2); q->agregar(fracc(6,2)); q->borrar(2);
    cout << "q: "; cout << *q << endl; //q: { 3/1 } - [1/1 4/3] (1/1 2/1)
    delete q;
    conjuntoPlus::intervalo_por_defecto(fracc(2,2), 2); //intervalo por defecto: [2/2, 2/1]

    conjuntoPlus a(fracc(2,4), f1), b;
    fracc x(1), y(7,6);
    cout << "a: "; a.ver(); cout << endl; //a: { } - [2/4 4/2] (2/2 2/1)
    cout << "b: "; b.ver(); cout << endl; //b: { } - [2/2 2/1] (2/2 2/1)
    cout << "a: "; a.conjunto::ver(); cout << endl; //a: { }
    cout << "a: " << a << endl; //a: { } - [1/2 2/1] (1/1 2/1)
    if (a.vacio()) cout << "a esta vacio\n"; //a esta vacio
    else cout << "a no esta vacio\n";
    cout << "Carga de a:\n";
    a.agregar(fracc(1,4)); a.agregar(fracc(2,6));
    a.agregar(); //introduce las fracciones 2/2, 7/1 y 2/4
    cout << "a: " << a << endl; //a: {1/4 1/3 7/1} - [1/2 2/1] (1/1 2/1)
    cout << "a: "; a.ver(); cout << endl; //a: {1/4 2/6 7/1} - [2/4 4/2] (2/2 2/1)
    cout << "a tiene " << a.getn() << " fracc\n"; //a tiene 3 fracc
    conjuntoPlus c(a); //c: {1/4 2/6 7/1} - [2/4 4/2] (2/2 2/1)
    a.borrar(7); a.borrar(2); //a: {1/4 2/6} - [2/4 4/2] (2/2 2/1)
    b=b=a;
    a.cambiar_intervalo(0, 1)
    cout << "a: " << a << endl; //a: { } - [0/1 1/1] (1/1 2/1)
    cout << "b: " << b << endl; //b: {1/4 1/3} - [1/2 2/1] (1/1 2/1)
    cout << "c: " << c << endl; //c: {1/4 1/3 7/1} - [1/2 2/1] (1/1 2/1)

    if (c.pertenece(x*2+fracc(10,2)))
        cout << "x*2+fracc(10,2) = " << x*2+fracc(10,2) << " esta en c\n"; //esta en c
    return EXIT_SUCCESS;
}

```

Crea una nueva carpeta **conjuntoPlus2** y copia en ella el contenido de **conjuntoPlus1**. Modifica en la carpeta **conjuntoPlus2** la clase **conjuntoPlus** de forma que admita un parámetro adicional nombre. Al **constructor** de **conjuntoPlus** habrá que pasarle obligatoriamente el nombre del **conjuntoPlus** creado.

```

class conjuntoPlus: public conjunto {
    char *nombre; //nombre del conjuntoPlus creado
    ...; //atributos de la version conjuntoPlus1
public:
    ...
};

```

Realiza los cambios necesarios en la clase **conjuntoPlus** para que el fichero **PruebaPlus1.cpp** (al que se añade en cada objeto **conjuntoPlus** creado su nombre) siga dando los mismos resultados de antes.