

Implemente el TAD Conjunto en Java como colección de elementos no ordenados sin repetición con los constructores ConjuntoVacio e Inicializa con un valor. Defina las operaciones esVacio, unión, intersección, pertenece, diferencia, diferenciaSimetrica y elimina.

Representación Informal:

Dado que puede añadirse varias copias de un mismo elemento al conjunto y el conjunto resultante es el mismo, hay que definir ecuaciones entre los constructores que expresen esta propiedad. Además, puesto que hay que comparar elementos del conjunto es necesario tener un método **iguales** entre los elementos del conjunto que cumpla las propiedades:

Reflexiva: iguales(e,e) es True

Simétrica: Si iguales(e,e1) es True entonces también lo es iguales(e1,e)

Transitiva: Si iguales(e,e1) es True e iguales (e1,e2) es True entonces iguales (e,e3) también es True.

Las operaciones son:

- conjuntoVacio Crea el conjunto vacío.
- esVacio Decide si un conjunto es vacío.
- pertenece Decide si un elemento está en un conjunto.
- añadir Añade un elemento a un conjunto.
- unión Da la unión de dos conjuntos.
- intersección Da la intersección de dos conjuntos.
- diferencia Da la diferencia de dos conjuntos.
- diferenciaSimetrica Da la diferencia simétrica de dos conjuntos: elementos de ambos conjuntos excepto los elementos comunes.
- borrar Elimina un elemento del conjunto.
- igual Decide si dos conjuntos son iguales.
- Incluido Decide si un conjunto está incluido en otro.
- Cardinal Indica el número de elementos que tiene un conjunto.

Representación Formal:

Los constructores son ConjuntoVacio e Inicializar(e)

- *ConjuntoVacio -> Conjunto
- *Inicializar(Elemento) -> Conjunto
- Añadir(Conjunto,Elemento) -> Conjunto
- Borrar(Conjunto,Elemento) -> Conjunto
- EsVacio(Conjunto) -> Boolean
- Pertenece(Conjunto, Elemento) -> Boolean
- Unión(Conjunto,Conjunto) -> Conjunto
- Interseccion(Conjunto,Conjunto) -> Conjunto
- Diferencia(Conjunto,Conjunto) -> Conjunto
- DiferenciaSimetrica(Conjunto,Conjunto) -> Conjunto
- Igual (Conjunto, Conjunto) -> Boolean
- Incluido(Conjunto Conjunto) -> Boolean
- Cardinal(conjunto) -> Natural

Semántica: para todo e, e1 de tipo elemento y todo C de tipo conjunto.

- Añadir(Añadir(C,e),e) $\Rightarrow$ Añadir(C,e)
- Añadir(Añadir(C,e),e1) $\Rightarrow$ Añadir(Añadir(C,e1),e)
- Borrar(ConjuntoVacio,e) $\Rightarrow$ ConjuntoVacio
- Borrar(Añadir(C,e1),e) $\Rightarrow$ si igual(e,e1) entonces
Borrar(C,e)
Sino
Añadir(Borrar(C,e),e1)
- pertenece(ConjuntoVacio,e) $\Rightarrow$ False
- pertenece(Añadir(C,e1),e) $\Rightarrow$ si igual(e,e1) entonces True
else pertenece(C,e)
- esVacio(ConjuntoVacio) $\Rightarrow$ True
- esVacio(Añadir(C,e)) $\Rightarrow$ False
- unión(ConjuntoVacio,C) $\Rightarrow$ C
- unión(Añadir(C,e),D) $\Rightarrow$ Añadir(unión(C,D),e)
- Interseccion(ConjuntoVacio,C) $\Rightarrow$ ConjuntoVacio
- Interseccion(Añadir(C,e),D) $\Rightarrow$ si pertenece(D,e) entonces
Añadir(intersección(C,D),e)
Sino intersección(C,D)
- Diferencia(C,ConjuntoVacio) $\Rightarrow$ C
- Diferencia(Añadir(C,e),D) $\Rightarrow$ si pertenece(D,e) entonces diferencia(C,D)
Sino añadir(diferencia(C,D),e)
- DiferenciaSimetrica(C,D) $\Rightarrow$ diferencia(unión(C,D),intersección(C,D))
- Igual(C,D) $\Rightarrow$ si unión(diferencia(C,D),diferencia(D,C)) es
ConjuntoVacio entonces True sino False
- Incluido(C,D) $\Rightarrow$ si diferencia(C,D) es ConjuntoVacio entonces true
sino false
- Cardinal(ConjuntoVacio) $\Rightarrow$ Cero
- Cardinal(Añadir (C,e)) $\Rightarrow$ si pertenece(C,e) entonces Cardinal(C)
Sino Sucesor(Cardinal(c))

Elabore un proyecto en java que incluya la implementación del TADConjunto y una clase principal que haga uso de las operaciones de TADConjunto o compruebe los axiomas.

Redefina el método toString para que muestre los elementos actuales del conjunto.

Es libre de cambiar la especificación si considera necesario. Solo documente el hecho.