

ESTRUCTURAS DE DATOS

Curso 2025/26

PRÁCTICA 1

Conjuntos e iteradores

Instrucciones

- Se debe completar en una sesión.
- Práctica individual.
- Lee el enunciado completo antes de comenzar. Los comentarios incluidos en el código también proporcionan información útil y necesaria.
- Al finalizar la práctica debes entregar el código desarrollado.
- La práctica será APTA si se superan todos los test de validación proporcionados.

Ejercicio 1

Dado un iterador `it` a una colección de enteros, implementa un método que devuelva un conjunto con aquellos elementos que son múltiplos de cualquier otro elemento de la colección. Los ceros (0) no se considerarán. Ejemplos:

```
iterador a {2, 5, 3} → {}  
iterador a {4, 2, 6, 0} → {4, 6}  
iterador a {2, 5, 1, 3} → {2, 5, 3}
```

```
public static Set<Integer> multiplos (Iterator<Integer> it);
```

Ejercicio 2

Dados dos conjuntos de enteros, el método debe modificarlos de tal forma que al finalizar en el primer conjunto (cuadrados) deben quedar los elementos que son el cuadrado de otro número de cualquiera de los dos conjuntos de entrada, y en el segundo conjunto (noCuadrados) aquellos enteros que no son el cuadrado de ningún otro número. Ejemplos:

```
separate ({1,2,3},{4,5,6}) → {4},{5,6,1,2,3}  
separate ({1,2,3},{5,2}) → { }, {1,2,3,5}  
separate ({16, 4, 1, 25}, {2,1,4}) → {16, 4, 1}, {25, 2}
```

```
public static void separate (Set<Integer> cuadrados, Set<Integer>  
noCuadrados)
```

Ejercicio 3

Dado un iterador al inicio de una colección de elementos, implementar un método que divida dicha colección en el mínimo número de conjuntos. El método debe devolver los conjuntos obtenidos. Ejemplos:

```

iterador a (1, 2, 1, 2, 3) → {{1, 2, 3}, {1, 2}}
iterador a (2, 2, 2) → {{2}, {2}, {2}}
iterador a (1, 2, 3, 2, 3, 4, 5, 1, 8, 2, 6) → {{1, 2, 3, 4, 5, 8, 6},
{2, 3, 1}, {2}}

```

```

public static<T> Collection<Set<T>> divideInSets (Iterator<T> it);

```

Ejercicio 4

Dado un conjunto de elementos `u` y una lista de conjuntos `col`, implementa un método que identifique y devuelva aquellos dos conjuntos de la lista cuya unión forma el conjunto `u`. Si no hay 2 conjuntos de la lista que formen `u`, entonces devolverá una colección vacía. Si hubiera más de dos conjuntos cuya unión sea `u`, devolverá los primeros que encuentre.

Ejemplos:

```

u=["orange", "apple", "banana", "melon"],
col={{"orange", "banana"}, {"orange"}, {"melon", "apple"}, {"apple",
"banana"}}
→ {{"orange", "banana"}, {"melon", "apple"}}

```

```

u=["orange", "apple", "banana", "melon"],
col={{"orange", "banana"}, {"orange"}, {"melon"}, {"apple"}, {"apple",
"banana"}}
→ {}

```

```

public static<T> Collection<Set<T>> coverageSet2 (Set<T> u,
ArrayList<Set<T>> col)

```