

Estructuras de Datos

Comparadores

Adriana Collaguazo Jaramillo, Mg.

Ejercicio

Implemente el método **findIntersection** de la clase LinkedList.

El método recibe una lista y retorna los elementos communes a la lista recibida y a la que invoca el método.

```
public List<E> findIntersection(List<E> otherList) {  
  
}
```

La Interface Comparator<E>

Provee orden para colecciones de objetos que no tiene un orden *natural*

`int` compare (T object1, T object2)

Retorna un entero:

Valor	Significado
Negativo	<code>object1 < object2</code>
Cero	<code>object1 = object2</code>
Positivo	<code>object1 > object2</code>

Esto permite implementar la búsqueda en nuestro TDA:

`LinkedList<E> findAll (Comparator<E> cmp, E object)`

<https://docs.oracle.com/javase%2F8%2Fdocs%2Fapi%2F%2F/java/util/Comparator.html>

Formas de Utilizar Comparadores

1. Creando una instancia de la clase Comparator

```
Comparator<Customer> cmpByCity = new Comparator<Customer>() {  
    @Override  
    public int compare(Customer o1, Customer o2) {  
        if (o1.getCity().equals(o2.getCity())) {  
            return 0;  
        } else {  
            return 1;  
        }  
    }  
};
```

```
// usamos una instancia que contiene el criterio o valor que queremos buscar  
Customer customerGye = new Customer();  
customerGye.setCity("Guayaquil");
```

```
LinkedList<Customer> results = allCustomers.findAll(cmpByCity, customerGye);
```

2. Usando una clase abstracta:

```
Customer customerGye = new Customer();  
customerGye.setCity("Guayaquil");
```

```
LinkedList<Customer> results = allCustomers.findAll(new Comparator<Customer>() {  
    @Override  
    public int compare(Customer o1, Customer o2) {  
        if (o1.getCity().equals(o2.getCity())) {  
            return 0;  
        } else {  
            return 1;  
        }  
    }  
}, customerGye);
```

Mediante una expresión lambda

```
Customer customerGye = new Customer();  
customerGye.setCity("Guayaquil");
```

```
LinkedList<Customer> results = allCustomers.findAll((Customer o1, Customer o2) -> {  
    if (o1.getCity().equals(o2.getCity())) {  
        return 0;  
    } else {  
        return 1;  
    }  
}, customerGye);
```