Estructura de Dades
Pràctica 1(2Conv): DLL Ordenada
curs 2022-23

Estudiant: Matías Larrosa
Professor/a: Marc Ruiz
Data de lliurament: 21/04/20210

Decisions de disseny

Com que l'únic que canviava d'aquesta classe respecte a la DLL eren els mètodes inserir el que he fet ha sigut crear una nova classe que estén de la principal (DLL), que es guarda un nou booleà menorMajor i que sobreescriu les funcions inserir(T) i inserir(int, T). Aquest booleà, guarda l'ordre de la llista ja sigui creixent (true) o decreixent (false).

```
public class DLLOrdenada<T extends Comparable<T>> extends DLL<T> {

    private boolean menorMajor;                //false: <, true: >

    public DLLOrdenada(boolean menorMajor){
        super();
        this.menorMajor = menorMajor;
    }

    @Override
    public void inserir(T data){...}

    @Override
    public void inserir(int posicio, T data) throws operacioImpossible {
        throw new operacioImpossible(posicio);
    }
}
```

Inserir(int, T) llença directament una excepció ja que no hem puc assegurar que aquella posició triada manté l'ordre de la llista, per tant, l'única manera de afegir elements en aquesta llista és mitjançant el mètode inserir.

Inserir:

El mètode inserir(T) recorre la llista fins a trobar un punt d'inflexió (final o data ja no segueix l'ordre), després de trobar aquest punt d'inflexió es decidirà si cal afegir l'element a la esquerra o a la dreta d'aquest punt d'inflexió. Fent els canvis de referencies necessaris.

Per al joc de proves he fet servir un altre JUnit test dins de DLLTest amb el nom ordenat().

```
boolean dreta;
if(menorMajor){
    while(poi.getFwd() != null && data.compareTo(poi.getData()) >= 0){
        poi = poi.getFwd();
    }
    dreta = data.compareTo(poi.getData()) >= 0;
}else{
    while(poi.getFwd() != null && data.compareTo(poi.getData()) <= 0){
        poi = poi.getFwd();
    }
    dreta = data.compareTo(poi.getData()) <= 0;
}
if(dreta){
    DLL<T> nnode = new DLL<>(data, poi);
    nnode.setFwd(poi.getFwd());
    poi.setFwd(nnode);
    if(nnode.getFwd() != null)
        nnode.getFwd().setBkw(nnode);
}else{
    if(poi.getBkw() == null){
        DLL<T> nnode = new DLL<>(this.data, bkw: this);
        if(getFwd() != null)
            getFwd().setBkw(nnode);
        nnode.setFwd(getFwd());
        this.data = data;
        setFwd(nnode);
    }else{
        DLL<T> nnode = new DLL<>(data, poi.getBkw());
        nnode.setFwd(poi);
        poi.getBkw().setFwd(nnode);
        poi.setBkw(nnode);
    }
}
```

Codi

```
package fase1.EstructuraDades;

import fase1.Excepcions.operacioImpossible;

public class DLLOrdenada<T extends Comparable<T>> extends DLL<T> {

    private boolean menorMajor;                    //false: >, true: <

    public DLLOrdenada(boolean menorMajor){
        super();
```

```
        this.menorMajor = menorMajor;
    }

    @Override
    public void inserir(T data){
        if(data != null){
            if (this.data != null) {
                DLL<T> poi = this;

                boolean dreta;
                if(menorMajor){
                    while(poi.getFwd() != null &&
data.compareTo(poi.getData()) >= 0){
                        poi = poi.getFwd();
                    }
                    dreta = data.compareTo(poi.getData()) >= 0;
                }else{
                    while(poi.getFwd() != null &&
data.compareTo(poi.getData()) <= 0){
                        poi = poi.getFwd();
                    }
                    dreta = data.compareTo(poi.getData()) <= 0;
                }
                if(dreta){
                    DLL<T> nnode = new DLL<>(data, poi);
                    nnode.setFwd(poi.getFwd());
                    poi.setFwd(nnode);
                    if(nnode.getFwd() != null)
                        nnode.getFwd().setBkw(nnode);
                }else{
                    if(poi.getBkw() == null){
                        DLL<T> nnode = new DLL<>(this.data, this);
                        if(getFwd() != null)
                            getFwd().setBkw(nnode);
                        nnode.setFwd(getFwd());
                        this.data = data;
                        setFwd(nnode);
                    }else{
                        DLL<T> nnode = new DLL<>(data, poi.getBkw());
                        nnode.setFwd(poi);
                        poi.getBkw().setFwd(nnode);
                        poi.setBkw(nnode);
                    }
                }

            } else
                this.data = data;

        }
    }

    @Override
    public void inserir(int posicio, T data) throws operacioImpossible
{
        throw new operacioImpossible(posicio);
    }
}
```