→ 22.- Aquest any 2006 ha estat gloriós pel Barça, ha guanyat la lliga per segon any consecutiu i, la Champions, després de 14 anys !. El seu president, el Sr. Joan Laporta està eufòric, tant es així que ha decidit que dedicarà una partida pressupostària a la confecció d'un sistema d'informació per informatitzar tots els aspectes del club.
El Barça té molts socis, que durat l'any han d'anar pagant unes quotes que li donen dret de gaudir d'uns grans avantatges, l'import total de la quota de soci depèn del tipus de soci, és a dir, si té o no abonament a alguna de les seccions del club. El club té vàries seccions, la de futbol, la de bàsquet, la de handbol,... , estant cadascuna d'elles representada per un dirigent de secció a la Junta de Govern del Club. Els diferents equips del Barça ho són d'una secció, cada equip està format per un conjunt de jugadors dels que ens interessa emmagatzemar la seva nacionalitat, que pot ser més d'una, per qüestions reglamentàries.
Cal que estudieu en detall les classes i completeu els mètodes indicats:

import edNolineals.*; } **Sentències importades en tots els arxius**

Comparable.java

```
public interface Comparable{
        public boolean MenorQue(Comparable c);
        public boolean MajorQue(Comparable c);
 }
```

Soci.java

```
public class Soci  implements Comparable{
        private long numeroSoci; //identificador
        private String nom;
        private Cua* quotes; //conté objectes Rebut,
                             // que són els que té pendents de pagament el soci
        public Soci(long numeroSoci, String nom){
                this.numeroSoci=numeroSoci; this.nom=nom;quotes=new CuaEnll(); }
        public long getNumeroSoci(){return numeroSoci;}
        public String getNom(){ return nom;}
        public void addQuota(Rebut t){ /*sentències Exercici 1*/ }
        public void remQuota(Rebut t) throws Exception { /*sentències Exercici 1*/ }
        public int quantesQuotesPendents(){ /*sentències Exercici 1*/ }
        public boolean MenorQue(Comparable c){
                if (c instanceof Soci) return (numeroSoci < ((Soci) c).numeroSoci);
                else return false;  }
        public boolean MajorQue(Comparable c){
                if (c instanceof Soci) return (numeroSoci > ((Soci) c).numeroSoci);
                else return false;  }
} // fi classe
```

G *de la col.lecció Cua trobaràs a l'annex del final del dossier l'especificació de la interfície

Rebut.java

```
public class Rebut {
        private int identificador;
        private int quantia;
        private int any;
        public Rebut(int identificador, int quantia, int any){
                this.identificador=identificador; this.quantia=quantia;
                this.any=any;
        }
        public int getIdentificador(){return identificador;}
        public int getQuantia(){ return quantia;}
        public int getAny(){ return any;}
        public boolean equals(Object c){
                if (!(c instanceof Rebut)) return false;
                Rebut v=(Rebut)c;
                return identificador==v.identificador && any==v.any && quantia==v.quantia;
        }
} //fi classe
```

SociAmbAbonament.java

```
public class SociAmbAbonament extends Soci{
        private Seccio abonada;
        private char tipusAbonament  //A, B ó C;
        private String seient;
        public SociAmbAbonament(long numeroSoci, String nom, Seccio abonada, char tipus, String
        cadira){
                this(numeroSoci, nom); this.abonada=abonada; tipusAbonament=tipus;
                seient=cadira;
        }
        public Seccio getSeccio(){ return abonada;}
        public char getTipusAbonament(){ return tipusAbonament;}
        public String getSeient(){ return seient;}
} //fi classe
```

Jugador.java

```
public class Jugador implements Comparable{
        private String nom;
        private String nacionalitats[]; //màxim 3
        public Jugador(String nom){
                this.nom=nom;
                String []p={null,null,null}; nacionalitats=p;
        }
        public String getNom(){ return nom;}
        public void addNacionalitat(String a) throws Exception{ /*sentències 2*/ }
        public boolean MenorQue(Comparable c){
                if (c instanceof Jugador) return (nom.compareTo(((Jugador) c).nom)<0);
                else return false;  }
        public boolean MajorQue(Comparable c){
                if (c instanceof Jugador) return (nom.compareTo(((Jugador) c).nom)>0);
                else return false;  }
} //fi classe
```

Equip.java

```
public class Equip implements Comparable{
        private int codi;  //identificador
        private String nom;
        private class Node{
             Jugador jug;
             Node seg;
             public Node(Jugador jug, Node seguent){this.jug=jug;seg=seguent;} //fi constructor
        } // fi classe privada
        private Node jugadors; // seqüència enllaçada lineal dels jugadors fitxats per l'equip.
                               // NO TÉ NODE CAPÇALERA
        private int quants; //cardinalitat de la seqüència anterior
        public Equip(String nom, int codi){
                /*sentències Exercici 3*/
        }
        public Equip(String nom, Acb jugadors, int codi){
                /*sentències Exercici 3*/
        }
        public int getCodi(){ return codi;}
        public String getNom(){ return nom;}
        public int getQuantsJugadors(){ return quants;}
        public void addJugador(Jugador jug) throws Exception { /*sentències
         Exercici 3*/ }
        public void addJugador(String nom, String nacionalitat) throws Exception { /*sentències
        Exercici 3*/ }
        public void remJugador(Jugador jug) throws Exception{ /*sentències Exercici 3*/}
        public boolean MenorQue(Comparable c){
                if (c instanceof Equip) return (codi<((Equip) c).codi);else return false;  }
        public boolean MajorQue(Comparable c){
                if (c instanceof Equip) return (codi>((Equip) c).codi);else return false;}
} // fi classe
```

Acb.java

```
public interface Acb {
        void   Inserir (Comparable e) throws Exception;
        void   Esborrar (Comparable e) throws Exception;
        boolean   Membre (Comparable e);
        Comparable  Arrel () throws Exception;
        Acb FillEsq ()throws Exception;
        Acb  FillDret ()throws Exception;
        boolean ArbreBuit ();
        void  Buidar();
} //fi interfície
```

AcbEnll.java

```
public class AcbEnll implements Acb{
        private class Node{
                Equip inf;  Node esq,dret;
                public Node(Equip m, Node e, Node d){inf=m;esq=e;dret=d;}
        } //fi classe privada
        private Node arrel;
        public AcbEnll(){arrel=null;}
        /*implementació de totes les operacions de la interfície */
} // fi classe
```

Les operacions de la interfície les suposeu implementades **idènticament** a les treballades a classe.

Seccio.java

```
public class Seccio {
        private String nom;
        private  Acb equips; //magatzem dels equips d'aquesta secció
        public Seccio(String nom){equips=new AcbEnll(); this.nom=nom; }
        public void addEquip(Equip p) throws Exception{ equips.Inserir(p);}
        public void remEquip(Equip p) throws Exception{ equips.Esborrar(p);}
        public String equipMesJugadors(){ /*sentències Exercici 4*/ }
        public boolean equals(Object o){
        /* sentències Exercici 4. Redefinició del mètode equals*/
        }
        public String trobaNomEquipMajorCodi() throws Exception{ /*sentències Exercici
        4*/ }
}
```

Barça.java

```
public class Barça {
        private Acb[]socis; /*l'Acb de la primera posició emmagatzema els socis amb un
        número de soci acabat en 0, el de la posició 1 els acabats en 1, ..... Els
        Acbs emmagatzemen objectes Soci. */
        private Pila* seccions; /* Magatzem de les diferents seccions del club */
        public Barça(){ /*sentències Exercici 5*/ }
        public void addSoci(Soci p) throws Exception{ /*sentències Exercici 5*/ }
        public void remSoci(Soci q) throws Exception{ /*sentències Exercici 5*/}
}
```

*de la col.lecció Pila trobaràs a l'annex del final del dossier l'especificació de la interfície

Exercicis:

➔ 1.- Implementeu els mètodes pendents de la classe Soci, els tres són mètodes que gestionen els rebuts del soci, un afegeix un nou rebut (addQuota), l'altre l'elimina (remQuota) i el darrer mètode (QuantesQuotesPendents) determina quantes quotes li resten per pagar al soci. Recordeu que tal com s'especifica en el codi, el magatzem de quotes només emmagatzema les que estan pendents de pagament.

```
public void addQuota(Rebut t){ /*No cal controlar la repetició de rebut*/ }
public void remQuota(Rebut t) throws Exception{
    /*Llançament d'excepció si el rebut no està al magatzem*/ }
public int quantesQuotesPendents(){
    /*mètode consultor, el magatzem ha de quedar intacte */ }
```

➔ 2.- Implementa el mètode pendent de la classe Jugador, **public void** addNacionalitat(String a) **throws** Exception, que ha d'afegir una nova nacionalitat al jugador. Cal controlar la repetició de nacionalitat, en cas de repetició cal llançar una excepció. També cal controlar la disposició d'espai dins del magatzem.

➔ 3.- Implementa tots els mètodes pendents de la classe Equip.

- **Constructors:**

```
public Equip(String nom, int codi){
        /*de moment no té jugadors assignats*/
}
public Equip(String nom, Acb jugadors, int codi){
        /*els jugadors de l'equip ens arriben dins d'un magatzem Acb, el
        constructor s'ha d'encarregar de posar-los en el magatzem de la
        classe que té aquesta funcionalitat. Som usuaris de la interfície Acb
        */
}
```

- **Manipulació:**

```
public void addJugador(Jugador jug) throws Exception {
    //Afegeix un jugador a l'equip. Cal controlar la repetició
 }
public void addJugador(String nom,String nacionalitat) throws Exception {
        // Sobrecarrega del mètode anterior. Ha de tenir la mateixa
        // funcionalitat
}
public void remJugador(Jugador jug) throws Exception{
        // Ha d'esborrar del magatzem el jugador. Cal controlar la no
        // existència
}
```

➔ 4.- Implementeu els mètodes pendents de la classe Seccio:

```
public String equipMesJugadors(){
    /*Cal determina l'equip de la Secció que té més jugadors contractats,
    concretament ens interessa el nom d'aquest equip. Si en hi ha més d'un amb
    el mateix nombre és irrellevant el que es retorni */
}
```

```
public boolean equals(Object o){
        /* Dues seccions són la mateixa, són iguals si a més de tenir el mateix nom
        de Secció, l'arbre que conté els equips és idèntic, és a dir no basta que
        tinguin els mateixos elements, aquests dins de l'arbre han d'estar situats al
        mateix lloc. La implementació ha de ser recursiva*/
}
public String trobaNomEquipMajorCodi() throws Exception{
        /*ha de trobar el nom de l'equip de la secció amb número de codi
        d'equip més elevat. La implementació NO pot ser recursiva. Cal
        llançar una excepció si la secció no té cap equip. */
}
```

→ 5.- Implementeu els mètodes pendents de la classe Barça:

```
public Barça(){ /*construcció de magatzem buits*/ }
public void addSoci(Soci p) throws Exception{
        //Afegir un nou soci al magatzem. Si ja existeix es llança una excepció
}
public void remSoci(Soci q) throws Exception{
        //Donar de baixa un soci del magatzem. Llança una excepció si no hi és
}
```

→ 23.- L'any 1992, mentre a Barcelona es celebraven les olimpíades, a EEUU van llançar una forta campanya publicitària amb l'objectiu de popularitzar la música i balls Country. Amb aquesta promoció, el line dance (ball en línia) no només va arribar a Anglaterra, es va estendre ràpidament per tota Europa, i com no!, també va arribar al nostre país. Avui per avui aquesta tipologia de balls es seguida per moltes persones, tant es així que hi ha vàries associacions i locals on posar en pràctica aquests tipus de balls. L'Associació AVECA, amb l'objectiu de donar un bon servei als seus socis, ha decidit de informatitzar tota la informació que té en aquest àmbit de cara a facilitar-ne la consulta d'informació.

El disseny de classes realitzat per un soci que, professionalment es dedica al desenvolupament d'aplicacions informàtiques, és el que trobareu a continuació, de totes maneres per clarificar conceptes cal que es tinguin en compte els següents punts:

→ Tot ball country té:

- una coreografia (conjunt de passos que el descriuen),
- una música associada

→ Existeixen diferents tipus de balls, en funció de si es ballen en parella ó individualment, i de com es ballen, tots en una mateixa direcció ó bé els uns a davant dels l'altres. Tipus de balls:

1.- Line dance: balls en línia, tots en línia i en la mateixa direcció.

2.- Contradance: és un tipus de ball en line dance però en aquest cas les files estan encarades per creuar-se.

3.- Parella: tots en la mateixa direcció però amb parella (home-dona).

4.- Parella Big Circle: amb parelles (home-dona) però formant un cercle.

Com de balls country n'existeixen milers, es necessari trobar una bona manera d'emmagatzemar-los, per a què, garanteixi un bon temps de resposta en les operacions de consulta que puguin realitzar els socis de l'entitat.