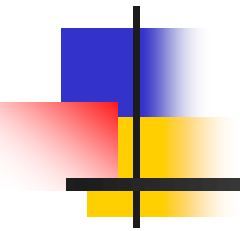


Programació Avançada

Aclariments Pràctica 3

Pràctica 3

Objectius:

Objectiu 1: Estructures de dades no lineals: Arbres binaris.

Objectiu 2: Fitxers en Java.

Lliurament:

Cal complir normativa, portada, índex, ...

→ 1.- Llistat imprès dels fonts (**Inici pràctica 4**)

→ 2.- Penjar el projecte al Moodle (**16/Novembre 8h**)

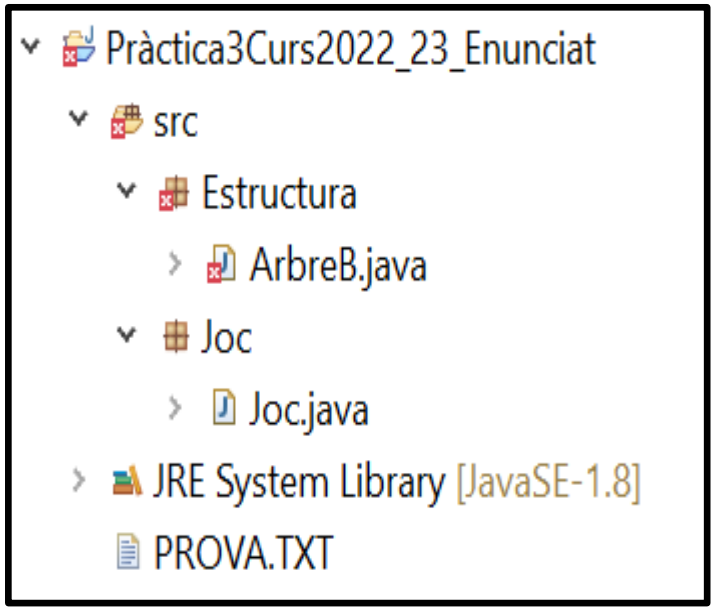
Consideracions:

Se **us dona un projecte JAVA** amb l'estructura que heu de seguir, **importeu-lo** al vostre espai de treball i canvieu-li el nom. Cal que l'identifiqueu amb: **Pràctica3NomCognom1&NomCognom2**

Pràctica 3

Consideracions:

Se **us dona un projecte JAVA** amb l'estructura que heu de seguir, **importeu-lo** al vostre espai de treball i canvieu-li el nom. Cal que l'identifiqueu amb: **Pràctica3NomCognom1&NomCognom2**



Heu de mantenir l'estructura del projecte. Programeu els mètodes seguint les pautes indicades

Pràctica 3

Enunciat: Joc Endevinador d'Animals

Execució

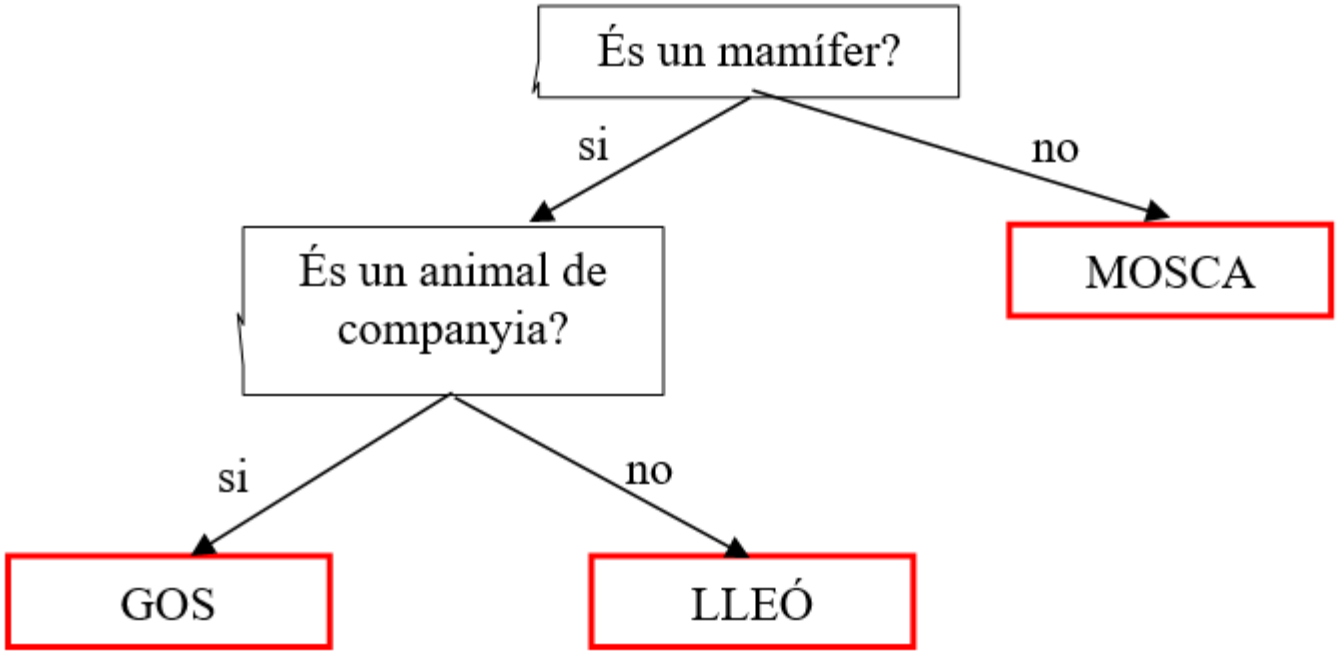
1.- El jugador pensa un animal (DOFÍ)

2.- S'inicia el joc.

Programa: És un mamífer?
Jugador: SI
Programa: És un animal de companyia?
Jugador: NO
Programa (ja no té més preguntes): És un LLEÓ?
Jugador: NO
Programa: Quin animal has pensat?
Jugador: DOFÍ
Programa: Diguem una pregunta que ajudaria a encertar l'animal que has pensat
Jugador: És un animal que viu al mar?

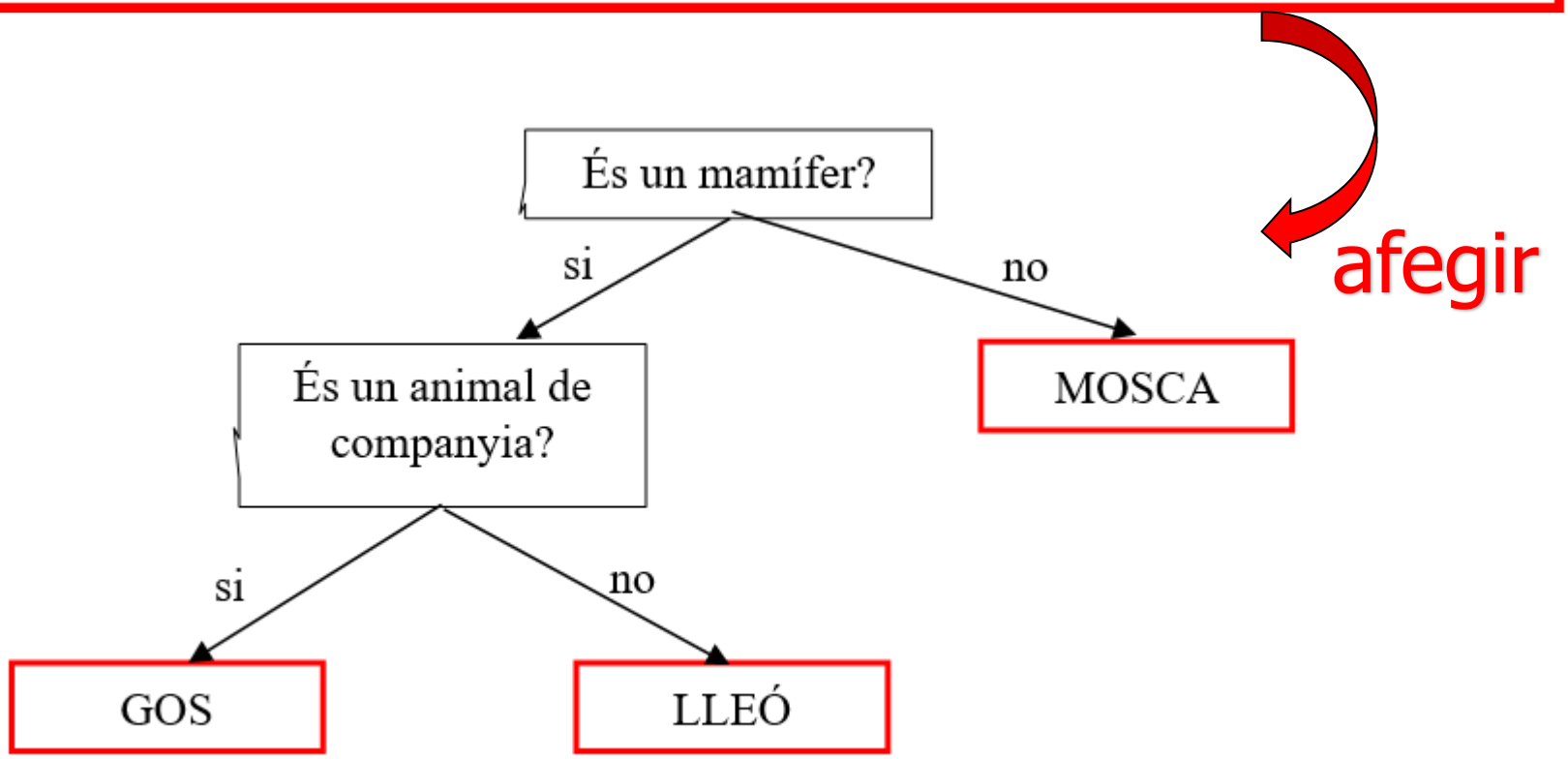
Pràctica 3

Implementació amb un Arbre Binari

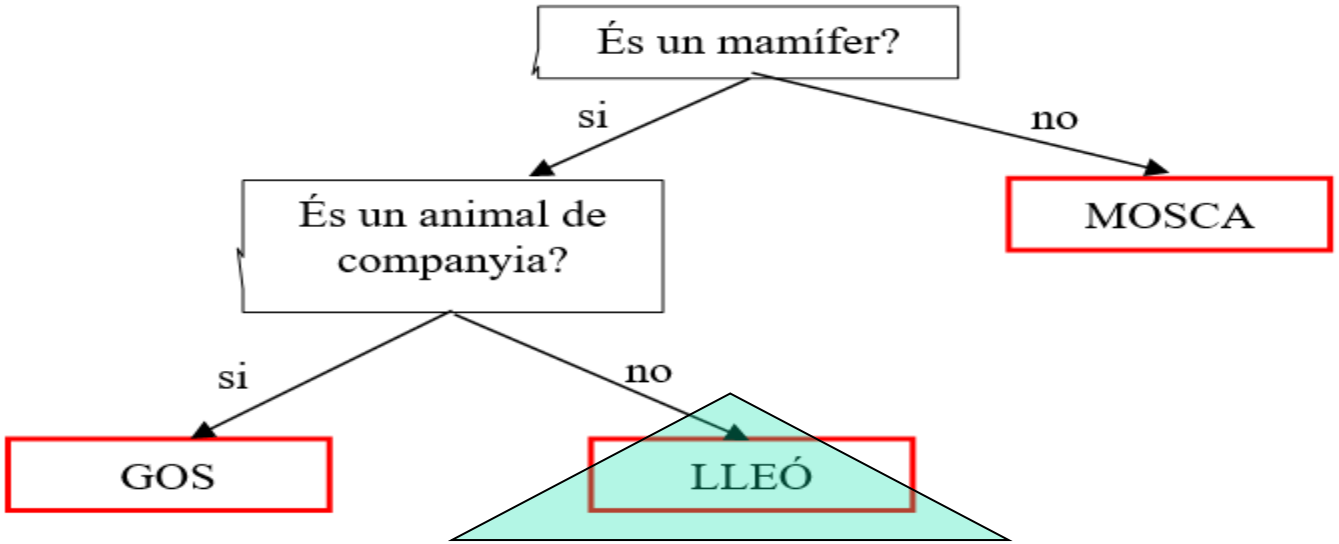


Pràctica 3

Programa: Quin animal has pensat?
Jugador: DOFÍ
Programa: Diguem una pregunta que ajudaria a encertar l'animal que has pensat
Jugador: És un animal que viu al mar?



Pràctica 3



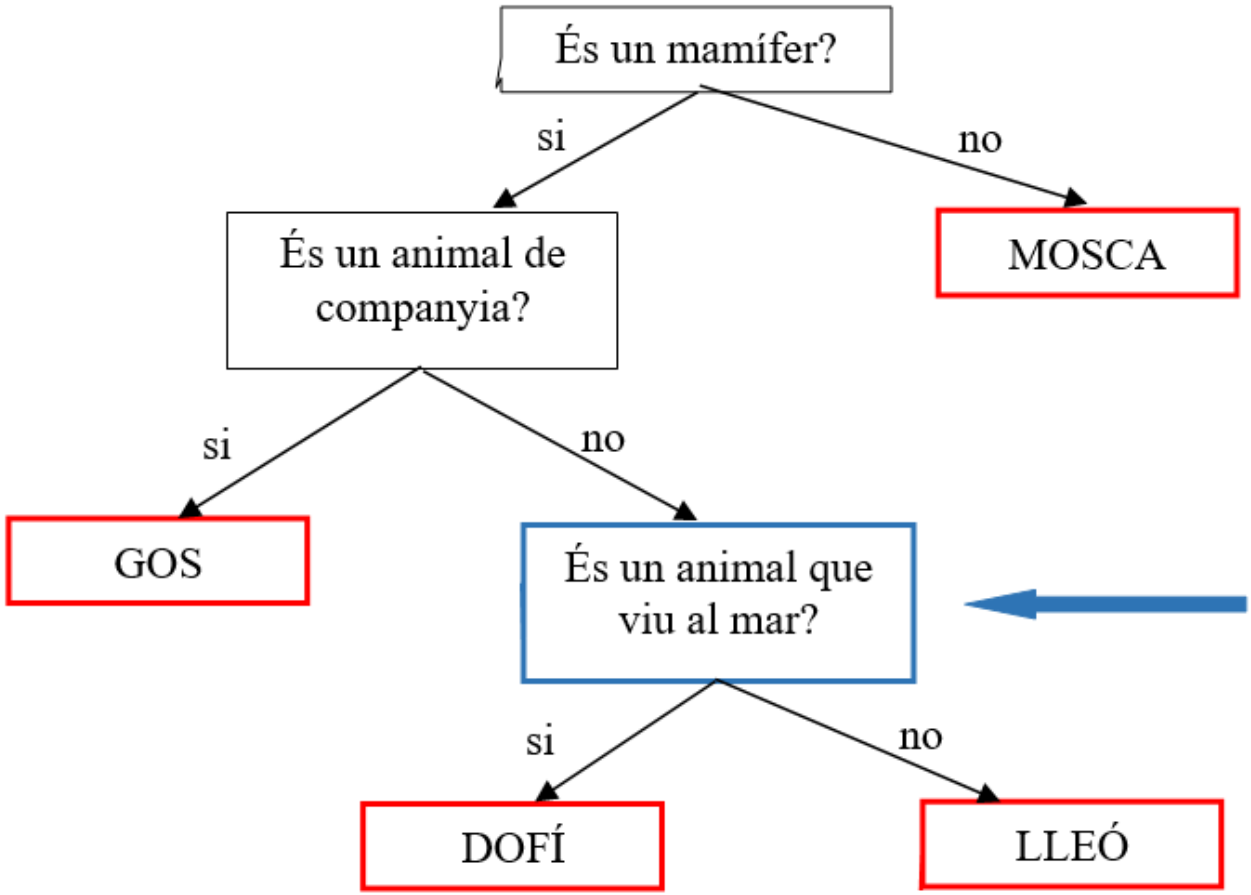
Execució

1.- El jugador pensa un animal (DOFÍ)

2.- S'inicia el joc.

Programa: És un mamífer?
Jugador: SI
Programa: És un animal de companyia?
Jugador: NO
Programa (ja no té més preguntes): És un LLEÓ?
Jugador: NO
Programa: Quin animal has pensat?
Jugador: DOFÍ
Programa: Diguem una pregunta que ajudaria a encertar l'animal que has pensat
Jugador: És un animal que viu al mar?

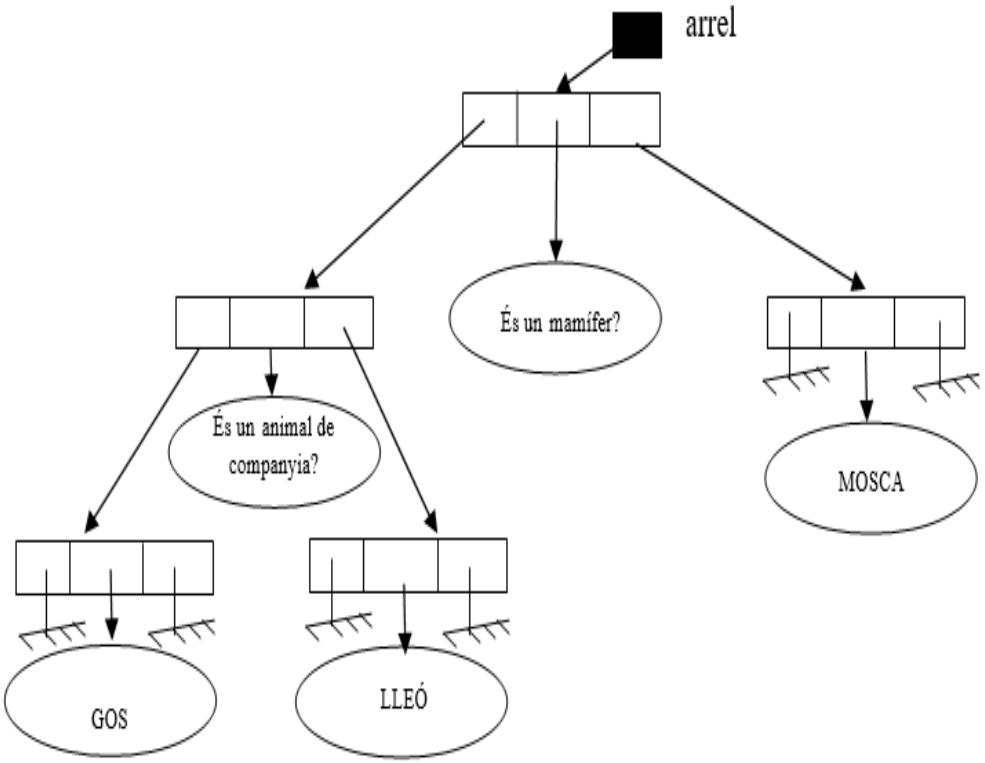
Pràctica 3



El node que tenia la resposta LLEO té la nova pregunta (**canviar la informació del node**). El fill esquerra d'aquest el nou animal i el fill dret la resposta incorrecte donada pel programa LLEÓ. **S'han creat dos nous nodes que són fulles**.

Pràctica 3. Representació

Opció 1: Treballada a classe de teoria



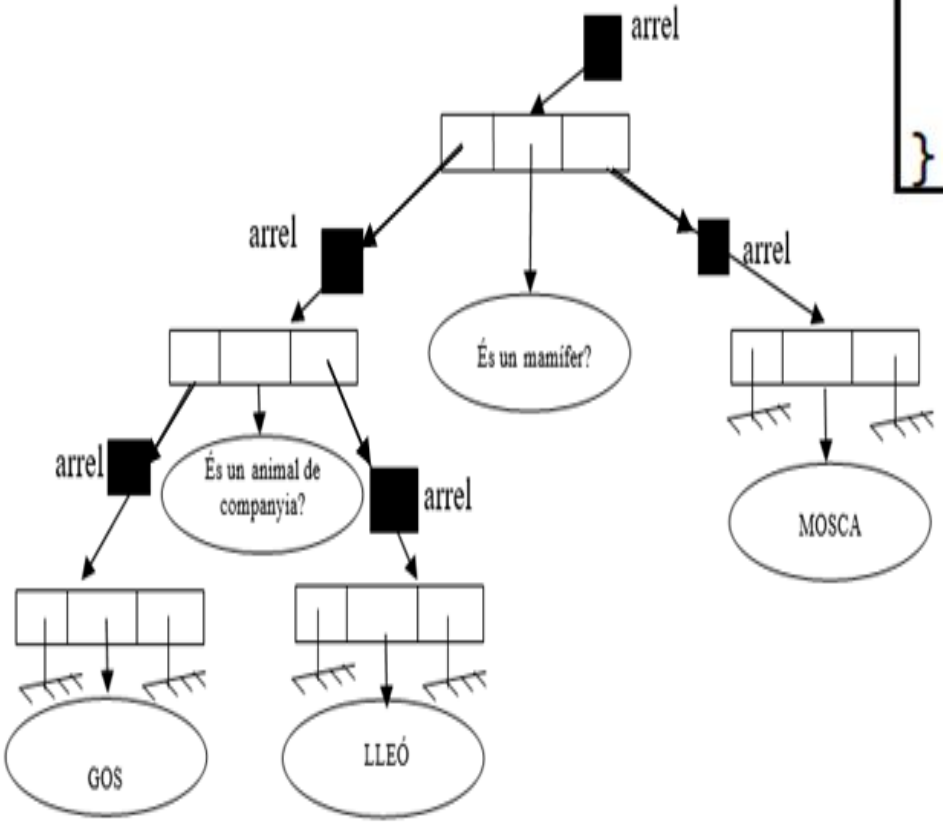

```
class NodeA {
    Object inf;
    NodeA esq, drt;

    NodeA() {
        this(null);
    }
    NodeA(Object o) {
        this(o, null, null);
    }
    NodeA(Object o, NodeA e, NodeA d) {
        inf = o;
        esq = e;
        drt = d;
    }
}
```

Pràctica 3. Representació

Opció 2: Treballar a pràctiques

```
private class ArbreB {
    private class NodeA {
        Object inf;
        ArbreB esq,drt;
    }
    //mètodes
}
```



Adoneu-vos de que els nodes fulla, les seves referències als fills són referències nul·les i no pas arbres binaris buits

Pràctica 3. Implementació

La Classe NodeA (dins la classe ArbreB)

```
private class NodeA {
    String contents;
    ArbreB yes, no;
    \\ Mètodes
}
```

```
NodeA(String contents) {
    //Constructor 1. Amb referencies nul.les als atributs yes i no
}
NodeA(String pregunta, ArbreB a1, ArbreB a2) {
    //Constructotr 2
}
```

Més mètodes. A decidir per l'estudiant el prototipus

Pràctica 3. Implementació

La Classe ArbreB

```
public class ArbreB {
    private class NodeA {
        String contents;
        ArbreB yes, no;
        NodeA(String contents) {
            //Constructor 1. Inicialitza als atributys yes i no a null
        }
        NodeA(String pregunta, ArbreB a1, ArbreB a2) {
            //Constructor 2. Crea el node i l'inicialitza amb els paràmetres
        }
    }
    // Atributs: Taula de 2 posicions
    private NodeA[] root;

    /* CONSTRUCTORS */
    public ArbreB(ArbreB a1, ArbreB a2, String pregunta) {
        //Constructor 1. Crea un arbre amb una pregunta i dos respostes
    }
    public ArbreB() {
        //Constructor 2. Crea un arbre buit
    }
    public ArbreB(String filename) throws Exception{
        //Constructor 3. Crea l'arbre amb el contingut donat en un fitxer
        //El paràmetre indica el nom del fitxer
    }
```

constructors

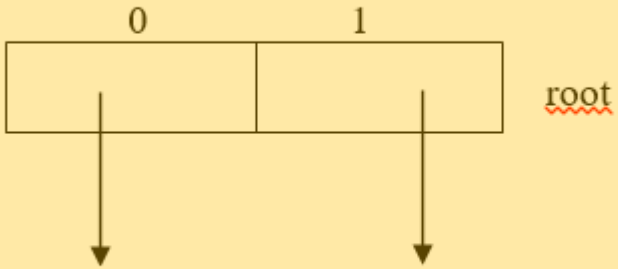
Pràctica 3. Implementació

La Classe ArbreB

```
public class ArbreB {
    private class NodeA {
        String contents;
        ArbreB yes, no;
        NodeA(String contents) {
            //Constructor 1. Inicialitza als atributys yes i no a null
        }
        NodeA(String pregunta, ArbreB a1, ArbreB a2) {
            //Constructor 2. Crea el node i l'inicialitza amb els paràmetres
        }
    }
    // Atributs: Taula de 2 posicions
    private NodeA[] root;

    /* CONSTRUCTORS */
    public ArbreB(ArbreB a1,
        //Constructor 1. Cre
    }
    public ArbreB() {
        //Constructor 2. Cre
    }
    public ArbreB(String fil
        //Constructor 3. Cre
        //El paràmetre indica el nom del fitxer
    }
```

private NodeA[] root;



Pràctica 3. Implementació

La Classe ArbreB

```
public class ArbreB {
    private class NodeA {
        String contents;
        ArbreB yes, no;
        NodeA(String contents) {
            //Constructor 1. Inicialitza als atributys yes i no a null
        }
        NodeA(String pregunta, ArbreB a1, ArbreB a2) {
            //Constructor 2. Crea el node i l'inicialitza amb els paràmetres
        }
    }
    // Atributs: Taula de 2 posicions
    private NodeA[] root;
```

Constructors sobrecarregats

```
    /* CONSTRUCTORS */
    public ArbreB(ArbreB a1, ArbreB a2, String pregunta) {
        //Constructor 1. Crea un arbre amb una pregunta i dos respostes
    }
    public ArbreB() {
        //Constructor 2. Crea un arbre buit
    }
    public ArbreB(String filename) throws Exception{
        //Constructor 3. Crea l'arbre amb el contingut donat en un fitxer
        //El paràmetre indica el nom del fitxer
    }
```

Pràctica 3. Implementació

La Classe ArbreB

Més mètodes (necessaris pel joc)

```
/* PUBLIC METHODS */
public boolean isEmpty () {
      //COMPLETE
}
/*sets the tree in its initial state: with the current node being the root*/
public void rewind () {
     //COMPLETE
}
/* True if the current node is an answer (a leaf) */
public boolean atAnswer () {
    /* COMPLETE */
}
/* move current to yes-descendant of itself */
public void moveToYes() {
    /* COMPLETE */
}

/* move current to no-descendant of itself */
public void moveToNo() {
    /* COMPLETE */
}
/* get the contents of the current node */
public String getContents () {
      //COMPLETE
}
/* Improves the tree at current with a new question and its yes-answer */
public void improve (String question, String answer) {
    /* COMPLETE */
}
```

Pràctica 3. Implementació

La Classe ArbreB

Més mètodes

- Visualitzar a pantalla el nom dels animals que conté l'arbre:

```
public void visualitzarAnimals ()
/*La implementació s'ha de fer, obligatòriament, invocant a un
mètode de la classe NodeA. És irrellevant l'ordre de
visualització*/
```

- Visualitzar a pantalla les preguntes que conté l'arbre:

```
public void visualitzarPreguntes ()
/*La implementació s'ha de fer, obligatòriament, invocant a un
mètode de la classe NodeA. És irrellevant l'ordre de
visualització*/
```

- Comptabilitza el nombre d'animals que conté l'arbre:

```
public int quantsAnimals ()
/*La implementació s'ha de fer, obligatòriament, invocant a un
mètode de la classe NodeA*/
```

- Calcula i retorna l'alçada de l'arbre. Recordeu que aquesta ve donada per la longitud del camí que va des de l'arrel de l'arbre a la fulla més llunyana:

```
public int alsada()
/*La implementació s'ha de fer, obligatòriament, invocant a un
mètode de la classe NodeA */
```

Pràctica 3. Implementació

Fitxers:

→ l'arbre de coneixement s'ha de poder emmagatzemar en un fitxer.

→ s'ha de poder carregar en un arbre a partir del fitxer.

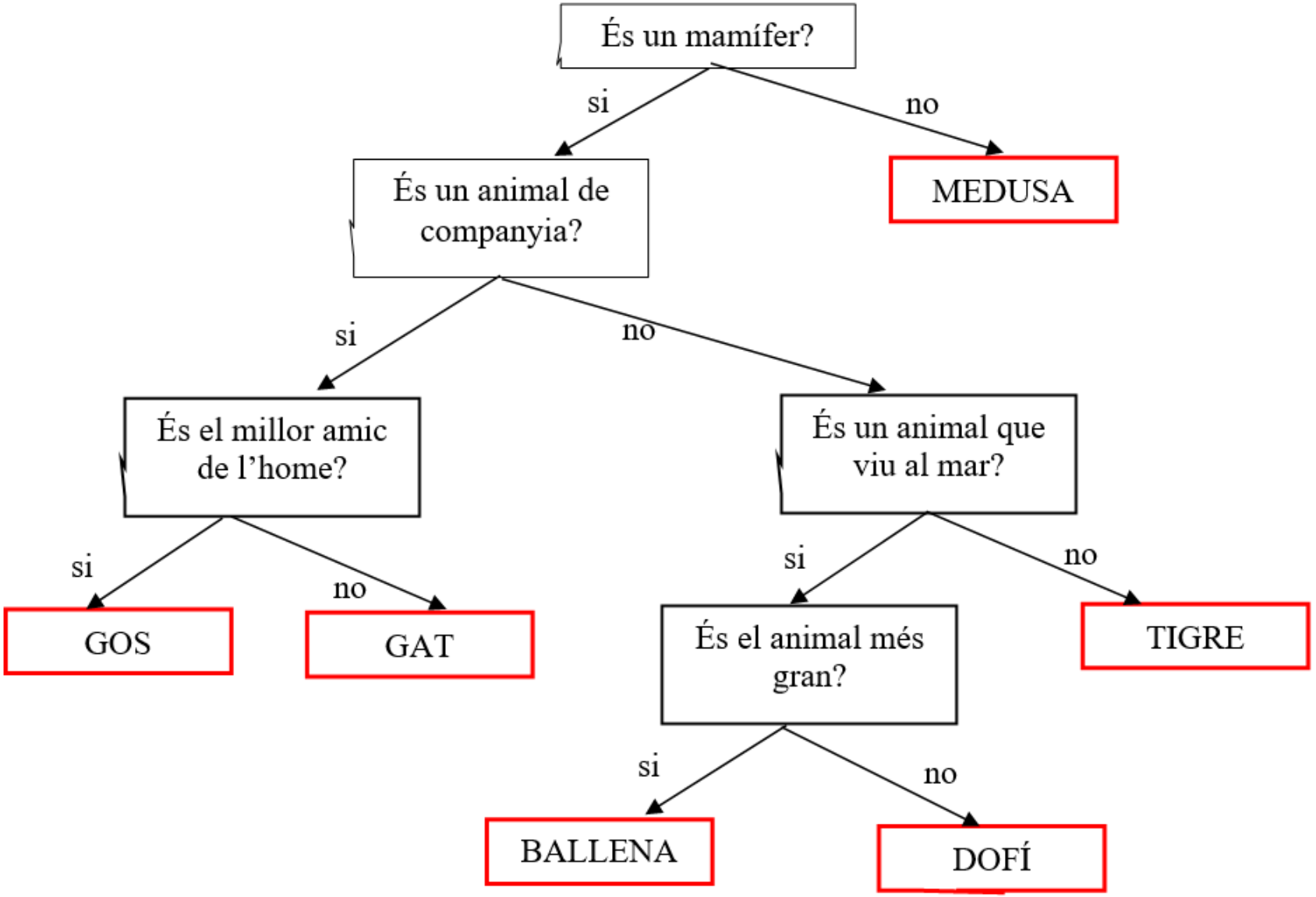
```
/* Writes, in preorder, the contents of the tree the root of which
 * is the implicit parameter */
public void preorderWrite(BufferedWriter buw) throws IOException{

    /* COMPLETE */
}
```

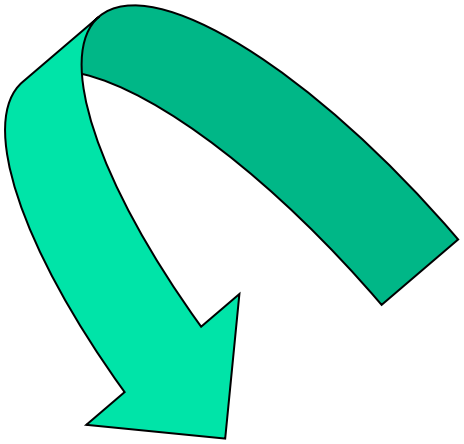
```
/*
 * Retrieves a tree from text file filename and returns its root. The file
 * has been saved using preorderWrite
 */
private NodeA loadFromFile(String filename){
    //Mètode recursiu
}
```

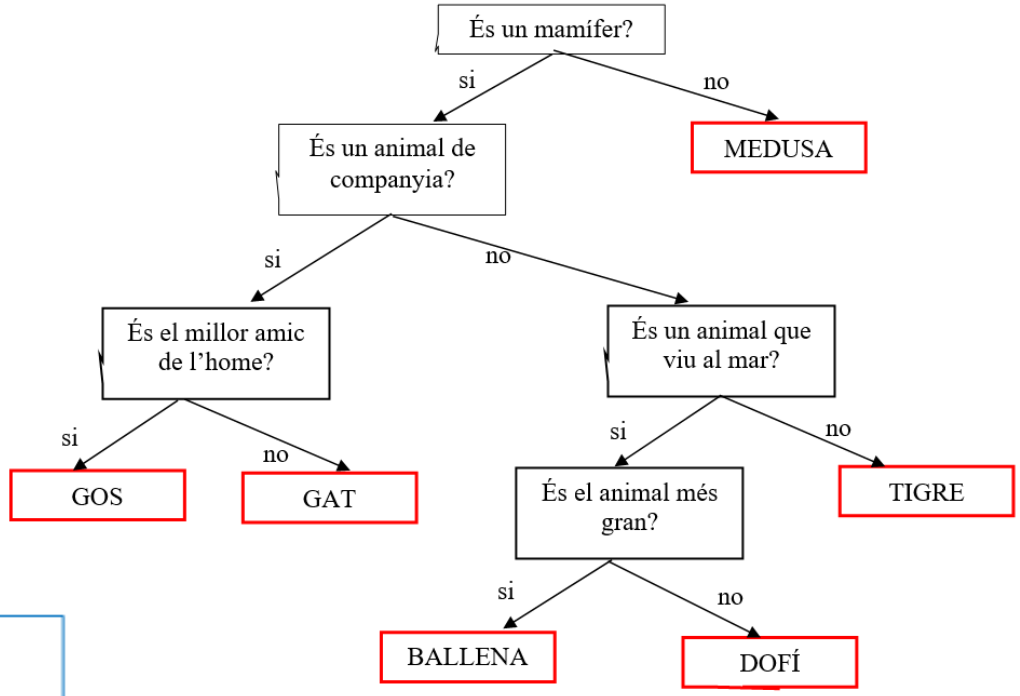
Invocat des del constructor

Emmagatzemar un ArbreB en un fitxer



Emmagatzemar ArbreB en un fitxer

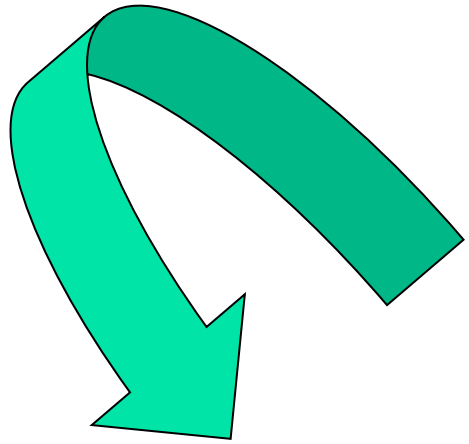




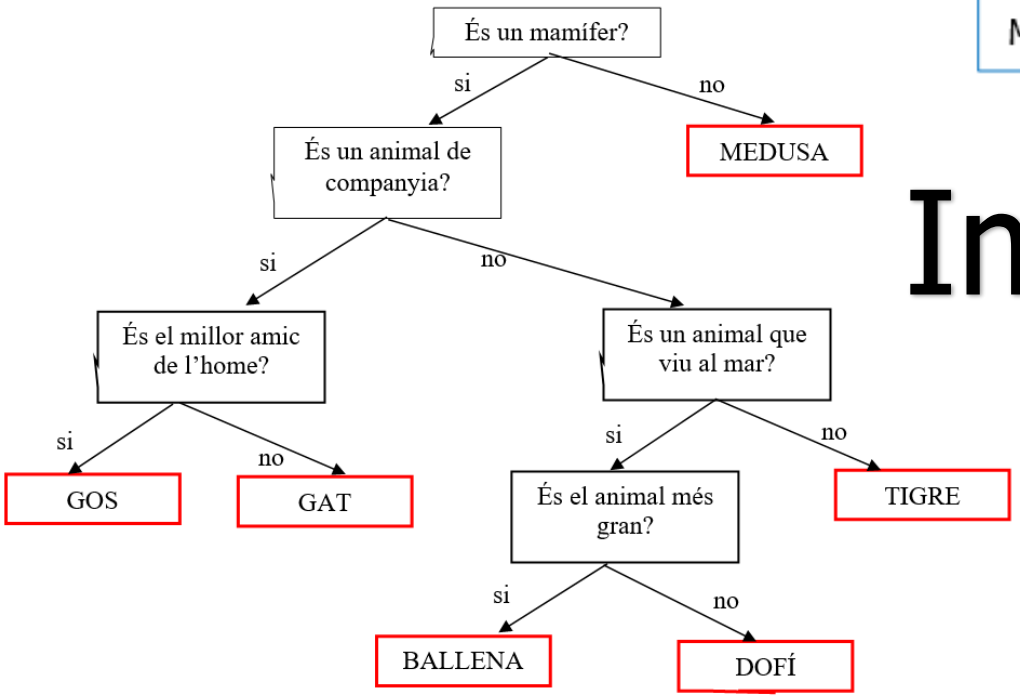

```
És un mamífer?
És un animal de companyia?
És el millor amic de l'home?
GOS
GAT
És un animal que viu al mar?
És el animal més gran?
BALLENA
DOFÍ
TIGRE
MEDUSA
```

Recorregut Preordre

Carregar un ArbreB a partir d'un fitxer



```
És un mamífer?
És un animal de companyia?
És el millor amic de l'home?
GOS
GAT
És un animal que viu al mar?
És el animal més gran?
BALLENA
DOFÍ
TIGRE
MEDUSA
```



Implementació recursiva