

PROGRAMACIÓ AVANÇADA

TRIMESTRE 1. CURS 2022/23

PRÀCTICA 3

.....

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

Objectiu 2: Fitxers en Java.

Durada: Una sessió

Lliurament: Llistat imprès dels fonts i penjar el projecte al Moodle.

Data Lliurament: Abans de la sessió de la Pràctica 4.

Aquesta pràctica s'implementarà en Java usant l'entorn de programació Eclipse.

Enunciat. Endevinador senzill



Un vell joc...

Fa uns quants anys es va fer popular la versió informàtica d'un joc anomenat *animal-vegetable-mineral*. La dinàmica d'aquest joc era molt simple: el jugador -una persona- pensava el nom d'un animal, d'un vegetal o d'un mineral. El programa feia preguntes que només es podien respondre dient si o no fins que creia saber què era el que el jugador havia pensat, llavors responia amb la seva solució i preguntava si havia encertat o no. En el cas que el programa hagués encertat, el joc tornava a començar amb el jugador pensant un altre nom. Si no era així -el programa no havia encertat- llavors demanava al jugador quina era la resposta correcta i una pregunta que, resposta afirmativament, li hagués permès endevinar el que el jugador pensava. El programa incorporava la pregunta i la resposta -allò que el jugador havia pensat- al seu *coneixement* i el joc tornava a començar. Evidentment el joc admet múltiples variants, per exemple restringir el joc a només noms d'animals. **Endevinador d'animals.** Adoneu-vos que a mesura que es va jugant el coneixement va augmentant. En cada errada -no encert- el programa aprèn de les seves errades.

... implementable amb l'ajut d'un arbre binari

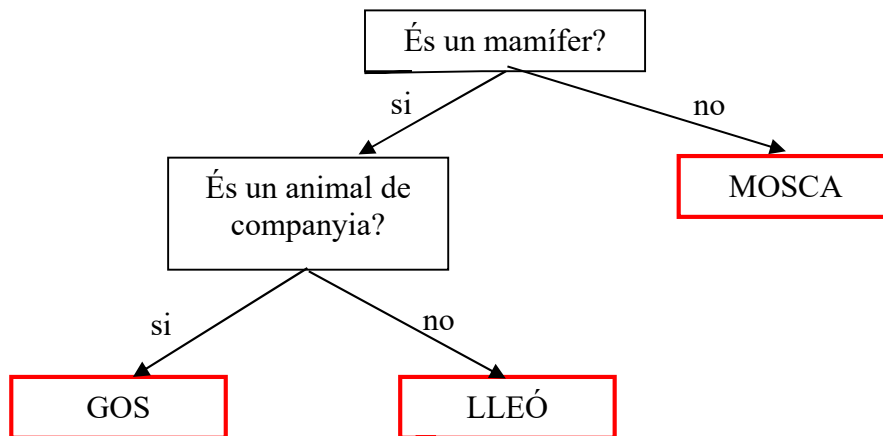
Una possible implementació d'aquest joc està basada en un arbre binari que conté, en cada node intern (no-fula), una pregunta, de tal manera que si aquesta es respon

afirmativament es visita el **fill esquerre** i si es respon **negativament** es visita el **fill dret**.

Cada node fulla conté una resposta i quan s'arriba a un d'aquests es dona la resposta que conté perquè ja no hi ha més preguntes per fer. Adoneu-vos que tots els nodes de l'arbre, o tenen dos fills (tenen una pregunta) -no hi han nodes amb només un fill- o són fulles (tenen una resposta).

Quan **no** es produeix **encert**, l'arbre s'amplia amb dos nodes, un que conté la **pregunta suggerida per l'usuari** i altre amb la **resposta** a donar en el cas que la pregunta es respongui afirmativament -allò que l'usuari estava pensant-. Com a resposta a donar en el cas que la pregunta es respongui negativament es col·loca la resposta que el programa havia donat però que no era un encert.

Exemple. Arbre de coneixement en un determinat moment



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?

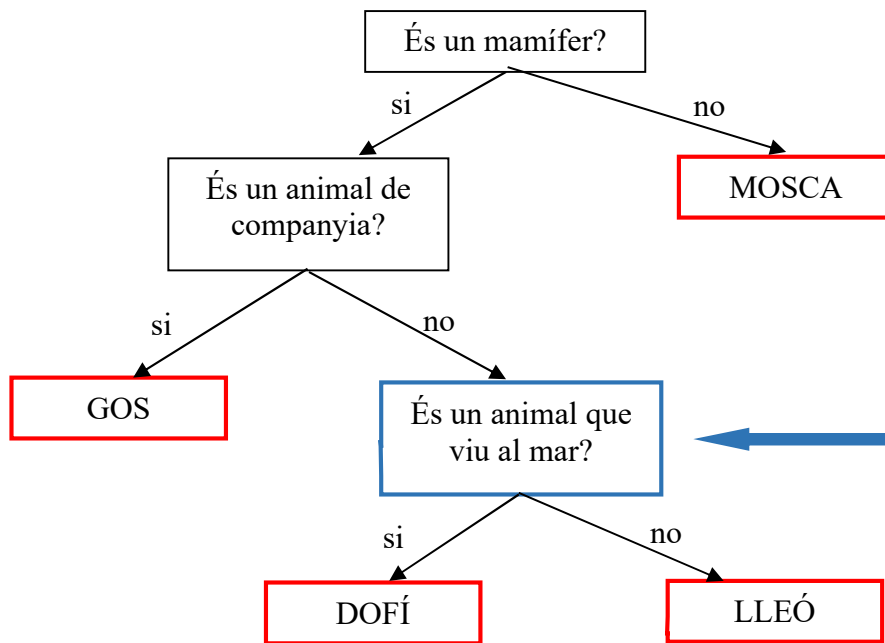
Jugador: DOFÍ

Programa: Diguem una pregunta que ajudaria a encertar l'animal que has pensat

Jugador: És un animal que viu al mar?



Cal ampliar l'arbre amb un nou node intern. Aquest té la pregunta indicada per l'usuari i com a resposta afirmativa (fill esquerre) un node fulla amb l'animal pensat



El node que tenia la resposta LLEÓ 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.

Representació de l'Arbre Binari mitjançant nodes

Opció 1: Treballada a classe de teoria

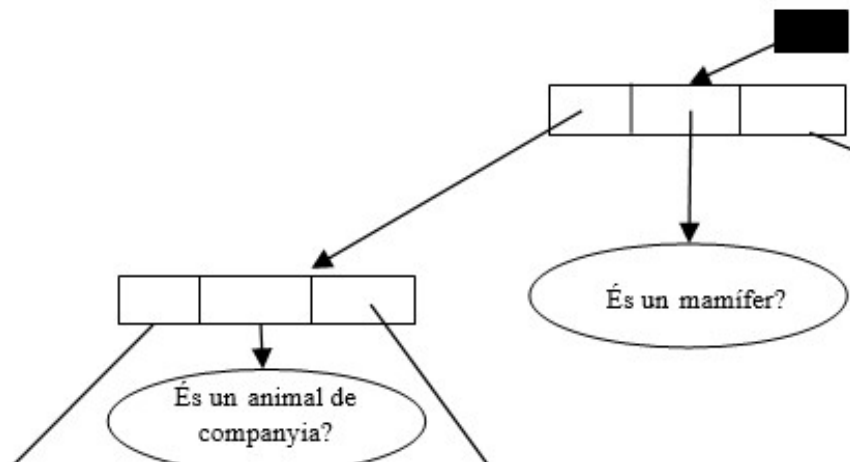
Cada node té tres atributs, un per emmagatzemar la informació i altres dos per referenciar a altre objecte node.

```

class NodeA {
    Object inf;
    NodeA esq, dret;

    NodeA() {
        this(null);
    }
    NodeA(Object o) {
        this(o, null, null)
    }
    NodeA(Object o, NodeA e
  
```

L'arbre binari té únicament un atribut, de tipus NodeA per referenciar a l'arrel de l'arbre. Si l'arbre és buit la referencia és nul·la.

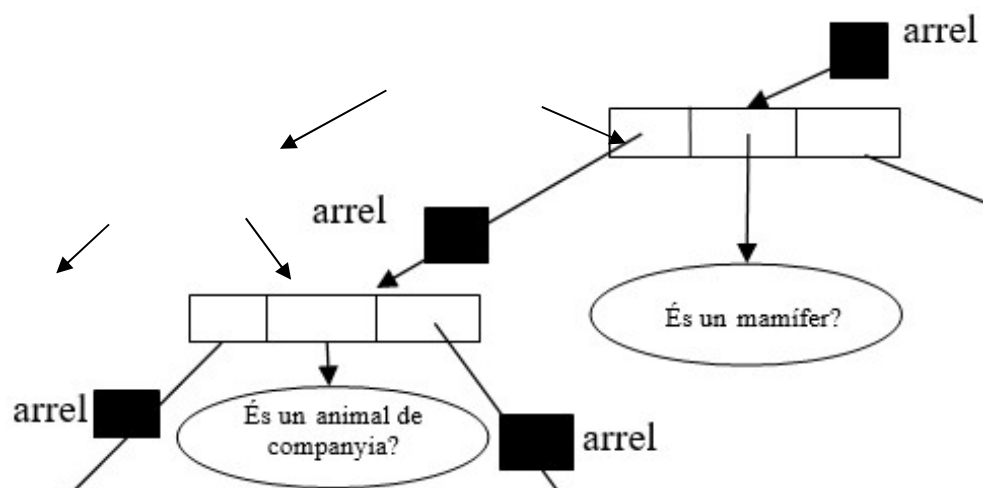


Opció 2: Treballar en aquesta pràctica

Cada node té tres atributs, un per emmagatzemar la informació i altres dos per referenciar a un objecte **arbre binari**. La classe es declara dins de la pròpia classe arbre binari.

```
private
public class Arbre
private class
Object inf
ArbreB esq
}
```

L'arbre binari, en aquest cas també té únicament un atribut, de tipus NodeA per referenciar a l'arrel de l'arbre. Si l'arbre és buit la referència és nul·la.



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

Implementació del joc

La vostra tasca principal consisteix en **completar** les implementacions dels mètodes de les classes. **No podeu afegir cap atribut** i heu de seguir les pautes indicades.

La Classe NodeA

En aquest cas la informació que emmagatzema cada node de l'arbre és un objecte de la classe String, i les referències als fills en direm **yes** i **no** en lloc de esq i drt i són declarades **com a referències a arbres binaris**. La classe NodeA té una visibilitat privada dins de la classe ArbreB.

```
private class NodeA {  
    String contents;  
    ArbreB yes, no;  
    // mètodes  
}
```

Mètodes

La classe té mètodes constructors i d'altres **mètodes que haureu d'afegir** per implementar les funcionalitats demanades en la classe arbre binari.

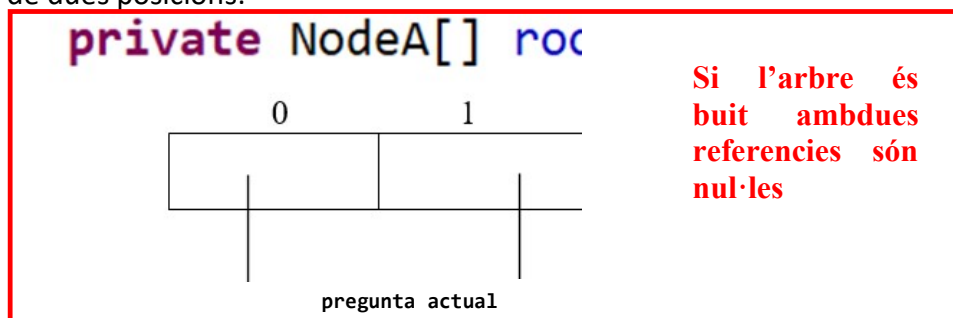
```
NodeA(String contents) {  
    //Constructor 1. Amb referències nul·les a  
}  
NodeA(String pregunta, ArbreB a1, ArbreB a2) {
```

La Classe ArbreB

La classe ArbreB a més de tenir l'atribut que referencia a l'arrel té un **segon atribut** per emmagatzemar la referència al node actual en l'avançament del joc. Inicialment quan s'inicia al joc té la referència a l'arrel i així com s'avança en el joc es va actualitzant.

Atributs

Dues referències a NodeA, **una que sempre s'ha de mantenir a l'arrel** de l'arbre i una **segona al node actual** en l'avançament de joc. Ambdues referències estan englobades en una taula de dues posicions:



Mètodes

- Mètodes constructors. Diferents maneres de crear un objecte arbre binari.
- Mètode afegir que permet ampliar l'arbre de coneixement afegint una nova pregunta amb la seva resposta afirmativa.
- Mètodes de consulta.
- Mètodes per avançar dins de l'arbre.
- Mètode per emmagatzemar l'arbre de coneixement en un fitxer.

Constructors

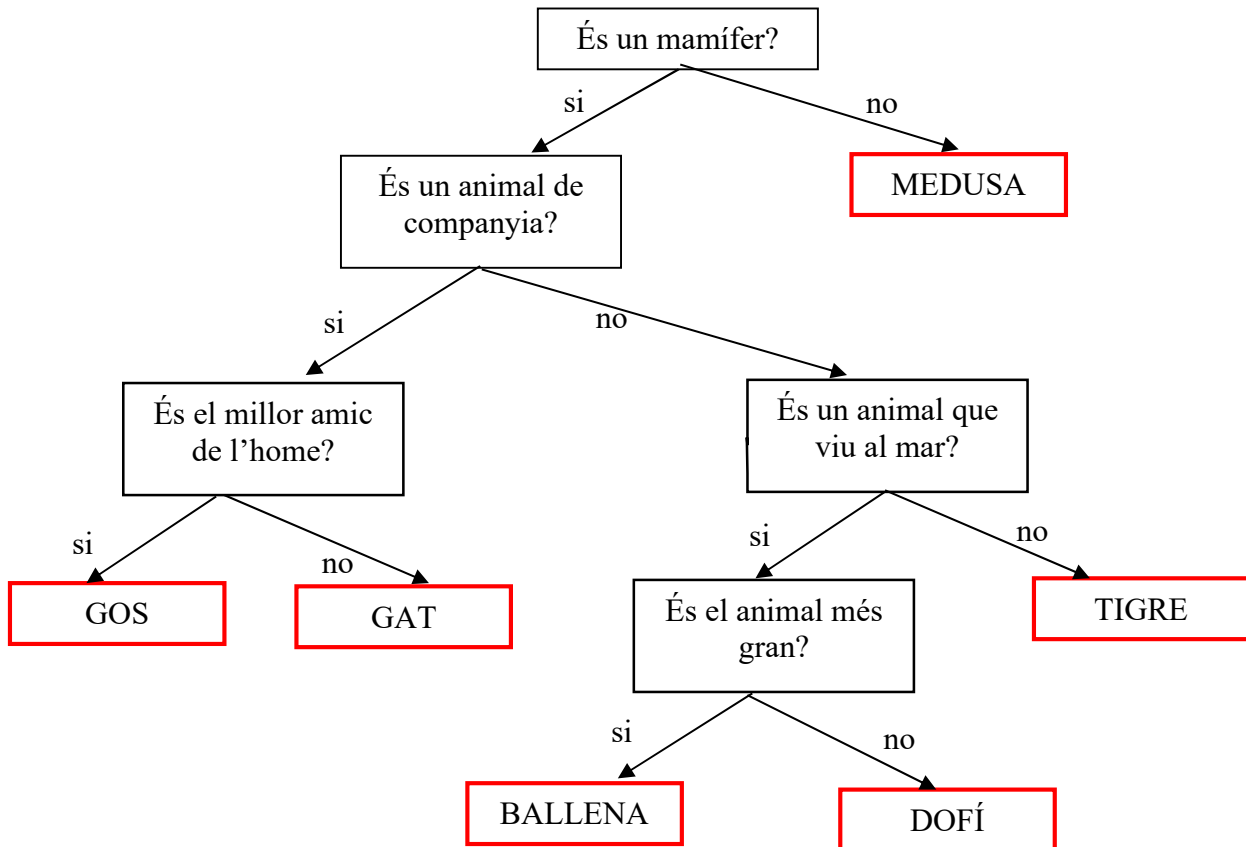
```
/* CONSTRUCTORS */

public ArbreB(ArbreB a1, ArbreB a2, String pregu
    //Constructor 1. Crea un arbre amb una pregu
}
public ArbreB() {
    //Constructor 2. Crea un arbre buit
}
```

Una part important de la pràctica és la part destinada a **desar** el contingut de l'arbre per a, posteriorment, poder-lo reconstruir en el **tercer constructor**. Per tant és necessari incloure a la classe un mètode, en direm **save**, que s'encarrega d'emmagatzemar l'arbre en un fitxer de text. Aquest mètode invoca a un mètode privat, anomenat **preorderWrite** que **imprescindiblement ha de fer una implementació recursiva**.

```
private void preorderWrite(BufferedWriter bu
    //Imprescindible que la implementació si
}
/* Saves contents of tree in a text file */
public void save(String filename) throws Exc
    // Invocar al mètode privat preorderWrit
    BufferedWriter buw = null;
    try {
        buw = new BufferedWriter(new FileWri
        this.preorderWrite(buw);
        buw.close();
```

El mètode **preorderWrite** ha de desar l'arbre en un **arxiu de text** on cada línia sigui el contingut (pregunta o resposta) d'un node. L'ordre és el determinat pel recorregut en **preordre** de l'arbre. Si una línia **acaba amb el caràcter ?** indica que aquell contingut correspon a un node intern (**pregunta**), si no hi acaba és una fulla (**resposta**). Per exemple, l'arbre següent:



Donaria lloc a un arxiu de text amb aquest contingut (nom fitxer PROVA.TXT):

```
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
```

Totes les preguntes han de finalitzar amb el caràcter interrogant de pregunta **?**, si l'usuari no li posa quan l'afegeix a l'arbre de coneixement, automàticament el programa li ha de posar. I referent al nom dels animals NO pot acabar amb el caràcter interrogant, si l'usuari li posa al fer la inserció, el programa automàticament li ha de treure. Aquest caràcter final és la diferenciació, quan s'emmagatzema en un fitxer l'arbre, entre node intern i fulla.

Tingueu present que, per la seva construcció, en aquesta mena d'arbres tots els nodes interns tenen exactament dos descendents (un node, o té dos fills o és una fulla).

El **constructor 3** ha de recuperar el coneixement del fitxer donat i construir l'arbre binari. Aquest constructor ha d'invocar a un mètode privat, en direm **loadFromFile** que en faci la càrrega fent una implementació recursiva. El mètode ha de tenir el següent prototipus:

```
/*  
 * Retrieves a tree from text file filename and re1  
 * has been saved using preorderWrite  
 */  
private NodeA loadFromFile(String filename){
```

Mètodes per gestionar el joc

La classe ofereix una col·lecció de mètodes, de visibilitat pública, **per gestionar pròpiament el joc. Imprescindible fer les implementacions tenint en compte les indicacions donades en forma de comentaris.**


```
/* PUBLIC METHODS */
public boolean isEmpty() {
    //COMPLETE
}
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 yes-descendant of itself */
public void moveToNo() {
    //COMPLETE
}
/* get the contents of the current node */
```

Més mètodes

La classe ofereix una col·lecció de mètodes, de visibilitat pública, per consultar informació vers l'arbre de coneixement:

➔ 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 */
```

El punt d'entrada en execució i les proves

Cal que implementeu el **main** per jugar al joc descrit.

Una típica sessió d'aquest joc podria anar de la següent manera:

Exemple 1: arbre de coneixement buit

Welcome to wonderful GUESS THE ANIMAL

Vols carregar un fitxer? no

Per començar cal introduir una primera pregunta amb dues res|

Indica la pregunta de l'arrel de l'arbre: és un mamífer

Indica el nom de l'animal de la resposta afirmativa: tigre

Indica el nom de l'animal de la resposta negativa: medusa

JUGUEM!!!!

ÉS UN MAMÍFER? si

ÉS UN ANIMAL DE COMPANYIA? si

Em sembla que ja ho se!!!!. Podria ser un/a GAT? no

Uuuups HE FALLAT!!!

plis, diguem el nom de l'animal que has pensat: gos

plis, diguem una pregunta que correspon a aquest animal:

Vols jugar altre cop? si

ÉS UN MAMÍFER? si

ÉS UN ANIMAL DE COMPANYIA? no

Em sembla que ja ho se!!!!. Podria ser un/a TIGRE? no

Uuuups HE FALLAT!!!

plis, diguem el nom de l'animal que has pensat: dofí

plis, diguem una pregunta que correspon a aquest animal:

Vols jugar altre cop? si

Vols jugar altre cop? **si**

ÉS UN MAMÍFER? **si**

ÉS UN ANIMAL DE COMPANYIA? **si**

ÉS EL MILLOR AMIC DE L'HOME? **no**

Em sembla que ja ho se!!!. Podria ser
Gooooood. ENCERTAT!!!

Vols jugar altre cop? **si**

ÉS UN MAMÍFER? **si**

ÉS UN ANIMAL DE COMPANYIA? **no**

ÉS UN ANIMAL QUE VIU AL MAR? **si**

ÉS L'ANIMAL MÉS GRAN DE TOTS? **no**

Em sembla que ja ho se!!!. Podria ser

Exemple 2. Carregant el fitxer PROVA.TXT. Es visualitzen dades vers l'arbre construït:

Welcome to wonderful GUESS THE A

Vols carregar un fitxer? **si**

Nom del fitxer: **PROVA.TXT**

HAS CARREGAT UN FITXER.

L'arbre de coneixement té els se

GOS

GAT

BALENA

Vols visualitzar les preguntes de l'arbre de

ÉS UN MAMÍFER?

ÉS UN ANIMAL DE COMPANYIA?

ÉS EL MILLOR AMIC DE L'HOME?

ÉS UN ANIMAL QUE VIU AL MAR?

ÉS L'ANIMAL MÉS GRAN DE TOTS?

ÉS UN MAMÍFER? NO

Em sembla que ja ho se!!!!. Podria ser un/a MEDUSA

Uuuups HE FALLAT!!!

plis, diguem el nom de l'animal que has pensa

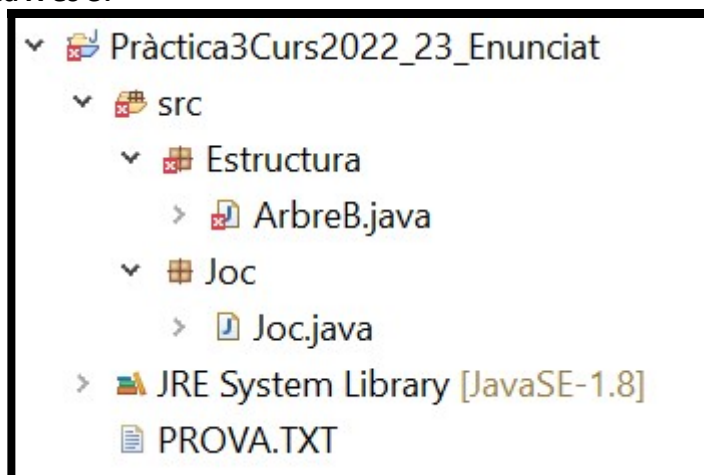
plis, diguem una pregunta que correspon a aqu

Quan l'usuari decideix carregar el coneixement d'un fitxer, el programa mostra a pantalla el contingut d'animals i alçada de l'arbre que s'ha construït. També si l'usuari vol, es poden llistar a pantalla les preguntes que emmagatzema l'arbre i a partir d'aquí iniciar el joc.

Organització

Cadascuna de les pràctiques que desenvolupareu en aquesta assignatura ha de ser un projecte diferent. Les pràctiques obligatòriament s'han de fer en grups de 2 estudiants.

Els noms dels vostres projectes, en les pràctiques d'aquesta assignatura han de seguir el següent patró: PràcticaXCognom1Nom1&&Cognom2Nom2. En aquesta pràctica X és 3.



Què se us subministra?

→ Fitxer amb l'enunciat de la pràctica. Document que esteu llegint.

→ Document amb aclariments. Explicació de classe.

→ Se us subministra un arxiu comprimit que conté una carpeta que al seu torn conté un projecte Eclipse. **Descomprimiu l'arxiu i importeu el projecte** al vostre workspace (**Import** → **General** → **Existing Projects into Workspace**). Implementeu el demanat respectant l'estructura del projecte (organització del codi en paquets). Canvieu el nom del projecte i **doneu-li un nou respectant l'organització indicada prèviament**.

→ Dins del projecte hi trobareu un arxiu de text **PROVA.TXT** amb el format explicat i que es pot carregar per jugar amb ell.

Què s'ha de lliurar i com?

S'ha de lliurar la carpeta que conté el **projecte** Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar el **llistat en paper** del codi desenvolupat. El format de lliurament d'aquest codi ha de seguir el patró indicat en la presentació de l'assignatura: amb portada, índex, número de pàgina, tabulació ... En **aquest llistat** cal que indiqueu quina ha estat **la distribució de la feina entre els dos estudiants. És a dir, el grau de participació de cada membre del grup en la realització d'aquesta activitat. Explica les principals dificultats en el seu desenvolupament, valoració del codi lliurat i funcionament de la pràctica.**

On s'ha de lliurar?

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

El lliurament en paper es farà directament a la professora a l'inici de la pràctica 4.

Quan s'ha de lliurar?

El lliurament a la plataforma es podrà fer fins el dia i hora indicats. Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.

Tots els grups → 16 de novembre a les 8h

Pautes de correcció

- Completar constructors de la classe NodeA amb els dos mètodes indicats: **0.5 punts**
- Completar la classe ArbreB amb els mètodes constructors i mètodes que gestionen el joc : **2.5 punts**
- Completar la classe ArbreB amb els mètodes que gestionen fitxers: **3 punts**
- Afegir mètodes a la classe ArbreB (visualitzarPreguntes, visualitzaAnimals, quantsAnimals i alsada): **2 punts**
- Joc: **main** i procediments: **2 punts**. Es valorarà la robustesa i la descomposició funcional aplicada (procediments ajudants).

Es valorarà l'eficiència i el seguiment de les pautes de programació indicades en el propi enunciat.

Si el joc no funciona la qualificació màxima serà de 4.