 Generalitat de Catalunya Departament d'Ensenyament Institut Caparrella	Avaluació: M3-UF5 ( 2ona ordinària)	
	Curs: DAW1	Nota:
	Data: 05/ 06 / 2018	
	Matèria/Crèdit: M3 – Programació M3- UF5	
Cognoms i Nom:	Signatura	

## EXAMEN DAW M3 UF5: POO. Llibreries de classes fonamentals

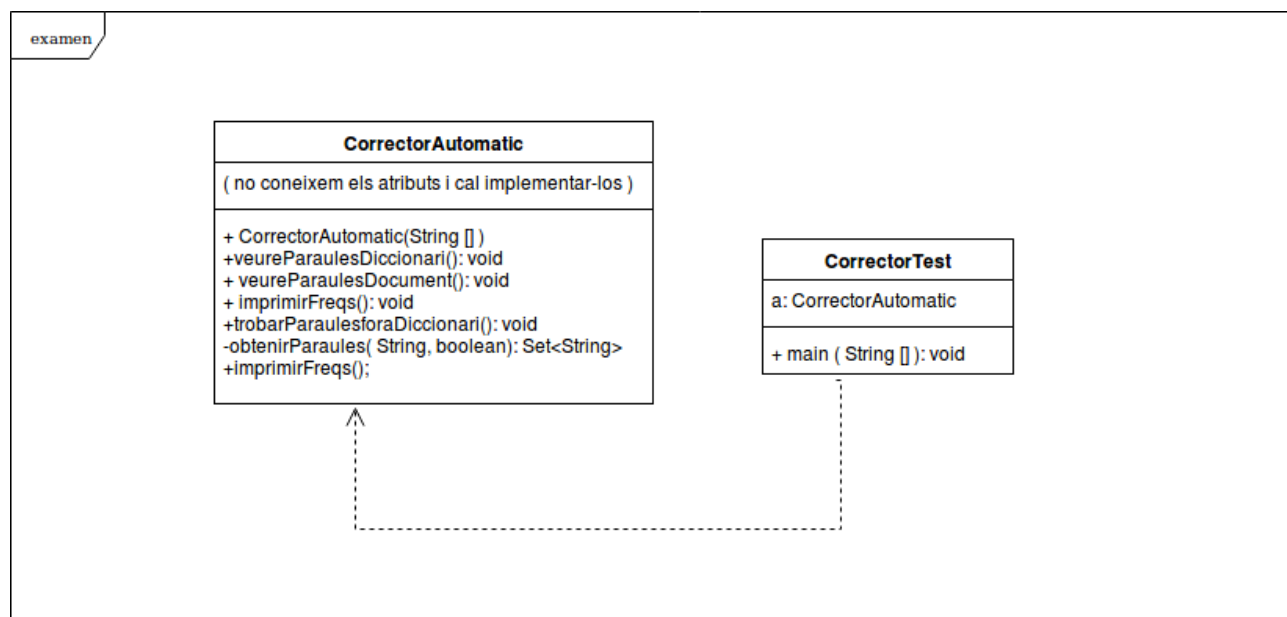
### Exercici 1 ( 10 punts )

Esteu realitzant una fase inicial d'un projecte dedicat a la creació d'un nou processador de textos. En el projecte s'estan implementant diferents funcions, concretament la que ens ocupa i volem implementar tracta sobre la típica eina de correcció de paraules.

Per fer-ho s'ha decidit implementar aquesta funcionalitat en diferents fases, i com us trobeu en la fase inicial d'implementació cal implementar un sistema que tingui en compte la utilització de dos fitxers externs escrits en text pla.

El requisit imposat pel vostre cap de programació és no utilitzar les col·leccions ArrayList ni tampoc utilitzar *arrays* convencionals o vectors tradicionals per emmagatzemar les variables principals. La resta de col·leccions de Java si us les permet utilitzar.

En tot cas us ha facilitat un esbós UML de les funcions que es pretenen implementar.





També se us facilita la part corresponent a la classe amb el programa principal que testearà la classe que si implementareu vosaltres anomenada **CorrectorAutomatic**

Fitxer: **CorrectorTest.java**

```
import java.io.FileNotFoundException;

public class CorrectorTest {

    public static void main(String argv[]) throws FileNotFoundException {

        try {
            CorrectorAutomatic a=new CorrectorAutomatic(argv);

            System.out.println("Diccionari carregat amb les següents paraules:");
            a.veureParaulesDiccionari();

            System.out.println("\n\nDocument carregat amb les següents paraules:");
            a.veureParaulesDocument();

            System.out.println("\n\nParaules detectades no trobades al diccionari:");
            a.trobarParaulesforaDiccionari();

            System.out.println("\n\nParaules ordenades del document i freqüència d'aparició:");
            a.imprimirFreqs();
        }

        catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```

En una conversa posterior amb el vostre cap us explica que la classe que ell us ha lliurat anomenada **CorrectorTest** utilitza 2 paràmetres externs:

- *Primer paràmetre* indica el nom del fitxer del diccionari de paraules.
- *Segon paràmetre* indica el nom d'un document de text.

Per comprovar el vostre programa disposeu de 2 fitxers que entrareu com a paràmetres:

Fitxer 1: **diccio.txt** ( correspon al 1er paràmetre passat a CorrectorTest )

La plaza tiene una torre,  
la dama una blanca flor.  
ha pasado un caballero

Fitxer 2: **poema.txt** ( correspon al 2on paràmetre passat a CorrectorTest ).

La plaza tiene una torre,  
la torre tiene un balcón,  
el balcón tiene una dama,  
la dama una blanca flor.



ha pasado un caballero  
-¡ quién sabe por qué pasó !-,  
y se ha llevado la plaza,  
con su torre y su balcón,  
con su balcón y su dama  
su dama y su blanca flor.

La sortida esperada de l'execució de:

```
java examen_UF5.CorrectorTest ../diccio.txt ../poema.txt
```

Diccionari carregat amb les següents paraules:

plaza,pasado,caballero,tiene,la,una,flor,ha,un,torre,dama,blanca,

Document carregat amb les següents paraules:

pasado,su,con,sabe,qué,quién,pasó,el,una,llevado,blanca,plaza,por,se,caballero,  
tiene,la,balcón,un,flor,ha,y,torre,dama,

Paraules detectades no trobades al diccionari:

su,con,sabe,qué,quién,pasó,el,llevado,por,se,balcón,y,

Paraules ordenades del document i freqüència d'aparició:

balcón	-> Repeticions: 4
blanca	-> Repeticions: 2
caballero	-> Repeticions: 1
con	-> Repeticions: 2
dama	-> Repeticions: 4
el	-> Repeticions: 1
flor	-> Repeticions: 2
ha	-> Repeticions: 2
la	-> Repeticions: 4
llevado	-> Repeticions: 1
pasado	-> Repeticions: 1
pasó	-> Repeticions: 1
plaza	-> Repeticions: 2
por	-> Repeticions: 1
quién	-> Repeticions: 1
qué	-> Repeticions: 1
sabe	-> Repeticions: 1
se	-> Repeticions: 1
su	-> Repeticions: 6
tiene	-> Repeticions: 3
torre	-> Repeticions: 3
un	-> Repeticions: 2
una	-> Repeticions: 3
y	-> Repeticions: 4



Consideracions a tenir en compte:

**1ª Consideració:** la funció **obtenirParaules**:

Aquesta funció rep 2 paràmetres:

- Primer paràmetre: Indicarà el nom del fitxer d'on es carregaran les paraules.
- Segon paràmetre: si és **true** indica que es carrega el diccionari ( del fitxer *diccio.txt* ), i a **false** indica que es carreguen les paraules del document ( del fitxer *poema.txt* ).

Aquesta funció separarà les paraules de la següent forma: Tot caràcter que sigui diferent a una lletra alfabètica minúscula, o bé una lletra alfabètica majúscula, o vocal accentuada esdivindrà un delimitador entre paraules, i per tant, no formarà part d'elles.

Per fer això possible, podeu utilitzar el mètode anomenat `useDelimiter` de la classe `Scanner` que permet indicar com a paràmetre un patró o expressió regular amb els caràcters que utilitzareu com a delimitadors.

Finalment tingueu en compte que `obtenirParaules` retornarà un conjunt de Strings segons el diagrama UML.

**2ª Consideració:** Els fitxers del diccionari (*diccio.txt*) i del document (*poema.txt*) s'han de llegir un sol cop i la funció responsable de fer-ho serà **obtenirParaules**.

**3ª Consideració:** Les excepcions `FileNotFoundException` seràn delegades des de les funcions de la classe `CorrectorAutomatic` fins ser tractades per la classe `CorrectorTest`.

**Es demana:**

Escolliu les col·leccions que siguin més adequades per als atributs membres de la classe `CorrectorAutomatic`. Tingueu en compte que per representar tant el diccionari com les paraules del document dins el `CorrectorAutomàtic` no ens importa l'ordre d'inserció i tampoc volem repetits.



Implementeu totes les funcions membres tenint en compte el diagrama UML proporcionat.

<b>Objectius a assolir:</b>	<b>Qualificació:</b>
Elecció atributs per representar el diccionari i les paraulesDocument.	1 punts
Constructora <i>CorrectorAutomatic</i>	1,5 punts
Funció <i>obtenirParaules</i> (part corresponent a la carrega de les paraules del diccionari) separat pels delimitadors indicats. Cal haver indicat correctament l'expressió regular utilitzada dins useDelimiter	1,5 punt
Funció <i>obtenirParaules</i> (part corresponent a la carrega de les paraules del document) separat pels delimitadors indicats i processament del recompte de paraules segons la freqüència en que apareixen en el document.	2 punts
mètode <i>trobarParaulesforaDiccionari</i>	2 punts
Mètode <i>veureParaulesDiccionari</i>	0,5 punts
Mètode <i>veureParaulesDocument</i>	0,5 punts
Mètode <i>imprimirFreqs</i>	1 punt

**NOTA:** No es valoraran els apartats si hi ha errors de compilació.

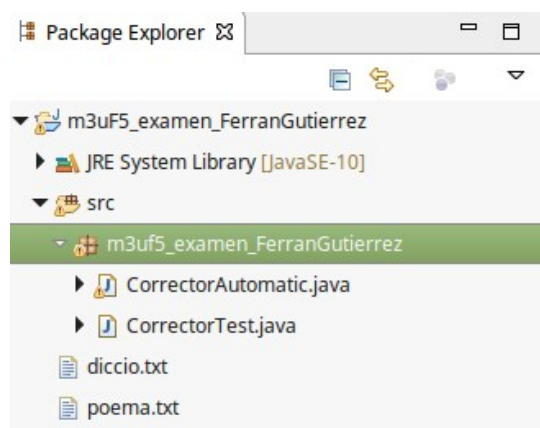
Tampoc es valoraran els apartats correctament si la classe *CorrectorTest* no proporciona els resultats esperats.



## **Format de Lliurament:**

Creeu un únic projecte anomenat **m3uf5Examen\_NomCognom** amb un únic package que s'anomenarà **examen\_NomCognom** i creeu dins un package anomenat m3uf5.

Us haurà de quedar tal i com apareix a la imatge d'exemple però ha de figurar el vostre nom tant al projecte com al package.



**Els programes realitzats han de poder compilar i executar-se.** Si els programes donen error de compilació i no es poden executar la seva puntuació serà 0 punts.

Per lliurar els exercicis seguiu aquests passos:

**1er pas)** Exporteu els exercicis fent clic amb **botó dret** a sobre del projecte **m3uf5Examen\_Nom-Cognom** i seleccionant la opció **Export...** i escolliu la opció **General – Archive File** . Us obrirà un quadre de diàleg i si premeu a sobre de Finish us generarà un fitxer .zip. Controleu en quina carpeta s'ha creat el vostre fitxer .zip

**2on pas)** Verifiqueu que el fitxer .zip l'exercici amb tots els fitxers .java com els fitxers .class. Si no és així torneu al 1er pas.

**3er pas)** Pugeu el vostre fitxer .zip que heu generat al eduportal i dins l'aula virtual a l'apartat **Lliurament examen M3UF5**