

DESCRIPCIÓ DEL DIAGRAMA DE CLASSES

Classe Usuari

Aquesta classe fa referència als usuaris que faran servir el programa. Aquests usuaris poden ser Admin o usuaris corrents. Els usuaris tenen un nom d'usuari, una contrasenya, una mà prioritaria (Esquerra o Dreta) i un identificador per saber si es Admin o no. Els primers 2 atributs son Strings, mentres que el 3r atribut es un Enum i l'últim és un booleà.

Les operacions són principalment *setters* i *getters*, amb una associació amb la classe Teclat. Destaquen les operacions:

- `isTeclat(id: Integer) :: boolean`
 - Retorna true si el teclat amb ID id existeix dins de l'associació.
- `addTeclat(id: Integer, teclat : Teclat)`
 - Afegeix un teclat a l'associació amb id.
- `eliminarTeclats(id : Integer)`
 - Elimina el teclat a l'associació amb id.
- `getTeclats(): HashMap<Integer, Teclat>`
 - Retorna el HashMap amb els teclats de l'usuari, junt amb el id corresponent

Classe Teclat

Aquesta classe representa un teclat, amb una string com a nom, el seu *layout* (expressat com una matriu de chars), el algorisme amb el que s'ha creat el teclat (instancia de la classe Algorisme) i la data de creació d'aquesta classe (expressat amb la classe Date). Cal destacar que el teclat no guarda el usuari, ja que el propi usuari, guarda els teclats en el seu Hashmap.

Les operacions de aquesta classe són principalment *setters* i *getters*. Algunes de les operacions destacables són:

- `addFrequencies(Frequencies frequencies) ::`
 - Associa la instancia de freqüència que es passa com a paràmetre a la instancia de teclat
- `getInfo() :: Vector<String>`
 - Retorna un vector amb la informació del teclat (com a diferents strings). En ordre del vector apareixen:
 - El nom del teclat
 - El nom del algoritme amb el que s'ha creat
 - La data de creació del teclat
 - El nom del alfabet de del input
 - I per últim cada fila de la matriu del layout

Classe Text (+ subclasses)

La classe text representa tots els textos que el sistema permet guardar. Com cada tipus de text té les seves pròpies funcionalitats, s'han classificat en dues subclasses: `TextPúblic` i `TextPredefinit`. Els textos públics són aquells que els usuaris poden penjar al sistema per a que altres usuaris els agafin com a input per als seus teclats. Per altre banda, els textos predefinits, son aquells que existeixen el el sistema de forma "predefinida". Només els usuaris determinats com a administradors poden gestionar aquestes instancies. Aquests textos ens permeten donar als usuaris que no tenen cap fitxer específic alguns exemples generals que poden arribar a usar com a input dels seus teclats.

Les funcionalitats principals s'implementen en les subclasses, ja que en la classe Text només trobem *setters* i *getters*.

En la classe TextPublic podem destacar la funcionalitat:

- `getUsuariUsername(): String`
 - Aquesta funció retorna el username que ha penjat el text públic

Classe Freqüència

La classe Freqüències és vital per al sistema. En ella es guarden i es processen les freqüències dels textos o els inputs que pot realitzar l'usuari per crear el seu teclat. Utilitza un TreeMap per guardar la llista de freqüències, es guarda el Alfabet al que pertany i també té un paràmetre on guarda el nombre de paraules que es troben (si es tracta d'un text d'on cal treure les freqüències). Aquesta classe és l'encarregada de processar els inputs per a la creació dels teclats, i per tant té diverses funcions per als diferents tipus de inputs que pot rebre. Algunes de les principals són:

- `genera(String s, Alfabet[] llistaAlfabets) ::`
 - és la funció encarregada de llegir un text (que rep en la variable string s). La funció té uns valors definits "prohibits", que la funció comprova en el text i si els detecta no els conta. La funció doncs, llegeix les paraules del text introduït i mentres detecta l'alfabet del propi text. Un cop acaba de processar tot el text, genera les freqüències de les paraules detectades
- `llegir(String t, Alfabet[] llistaAlfabets) ::`
 - és la funció encarregada de llegir l'input quan aquest són uns parells de paraules amb les seves freqüències. Aquesta funció processa el format amb el que s'ha introduït. També detecta l'alfabet del input, que es selecciona de l'array de Alfabets que té com a paràmetre

Classe Alfabet

Aquesta classe representa tots els alfabetes que es permeten en el sistema. L'usuari el tria de forma indirecte en el moment d'introduir el input per a la generació del teclat. Aquesta classe té com a atributs el nom de l'alfabet, i el alfabet en si (és a dir, els caràcters que conformen l'alfabet). La majoria de les funcionalitats són setters i getters, a excepció de la funcionalitat principal, que es la que s'utilitza per detectar el alfabet de la col·lecció d'alfabets. La funció és la següent:

- `determinaAlfabet(Character c, TreeMap<String, String> magatzemAlfabets) ::`
 - Aquesta funció és la responsable de anar revisant els caràcters que rep i els compara amb els dels alfabetes guardats en el sistema. Si detecta un dels alfabetes, l'assigna.

Classe Algorisme

La classe algorisme representa els possibles algorismes de creació de teclats que podria arribar a usar un Usuari. Aquesta classe és una classe abstracte, ja que no té cap instància i les seves funcions no tenen cap implementació. La implementació es realitza en les seves jerarquies (que són els dos algorismes implementats en el sistema). La única funció interessant mencionable és:

- `generarLayout(Frequencies frequencies, int columnes, int files) :: char[][]`
 - La implementació d'aquesta funció es troba en les seves subclasses, ja que cada una usa una estratègia de generació diferent

Classe BranchAndBoundEager

La subclasse de algorisme implementada en aquesta primera entrega. Es l'encarregada de generar el layout que se li proporciona al teclat un cop s'ha realitzat l'input complet. Els dos atributs principals i destacables són la matriu de distàncies i la de proximitats entre caràcters (aquestes proximitats es basen en les freqüències que es calculen en la classe freqüència). Aquestes matrius són matrius de integers, característiques del problema QAP que tractem de solucionar amb aquest algorisme amb l'estratègia Branch and Bound (per més informació sobre la resolució del algorisme, veure el document Estructures de dades i Algorismes). A part dels setters i getters, la funcionalitat que destaca és la següent:

- `generarLayout(Frequencies frequencies, int columnes, int files) :: char[][]`
 - Aquesta és la implementació de la funció mencionada en l'explicació de la classe Algorisme. És l'encarregada de generar el layout que li pertoca a partir de les freqüències mitjançant l'estratègia Branch and Bound (explicada en el document d'algorisme). La funció retorna directament la matriu de chars que pertoca al teclat amb totes les lletres col·locades segons la solució de l'algorisme

Classe SegonAlgorisme

Aquesta classe no esta implementada en la primera entrega, pero representa el segon algorisme de creació de teclats que podrà usar l'usuari en el futur. Cal destacar que la funcionalitat principal a implementar en les següents entregues serà la funció `generarLayout(Frequencies frequencies, int columnes, int files) :: char[][]`, que s'encarregarà de generar el layout seguint la estrategia que l'equip decideixi implementar.