

Le « gros » exo : le questionnaire (v3)

Vous avez à votre disposition un fichier Questionnaire1.txt

Il est structuré comme une série de deux lignes :

- La première contient une question
- La seconde contient la réponse à la question.

L'objectif est d'utiliser ces informations pour proposer à l'utilisateur un questionnaire et de vérifier la (ou les) réponse (s) de l'utilisateur.

Les étapes suivantes sont proposées :

1. Lecture et affichage du fichier (interface texte)

- a. Lire le fichier texte (avec Scanner) et l'afficher (avec `System.out.println()`)
- b. (optionnel) Permettre à l'utilisateur de choisir un fichier à l'aide de `JFileChooser`

2. Représentation des données (question-réponse et liste de questions-réponses)

- a. Définition d'une représentation adaptée (lien entre question et réponse, numérotation)
Pour cela définir une classe `Item` comportant au moins deux attributs de type `String` représentant la question et sa réponse. On peut y adjoindre un id numérique. Si ces attributs sont `private`, il faudra prévoir des accesseurs.
Il faut créer un constructeur pour cette classe, un `toString()` et une méthode booléenne `testerReponse()` qui prendra en paramètre une chaîne représentant la réponse d'un utilisateur et qui indiquera si cette réponse est bonne (égale à la réponse associée à la question) ou pas. On peut aussi introduire la méthode `obtenirBonneReponse()`.
- b. Définition d'une représentation adaptée d'une liste d'instances d'`Item`.
Pour cela, définir une classe qui encapsulera cette liste (sous la forme d'un `ArrayList` par exemple).
Par suite, il faudra que cette classe contienne la méthode de lecture du fichier texte et utilise le contenu du fichier texte pour initialiser la liste.
Adapter en conséquence la ou les méthode(s) d'affichage de la liste d'`Item`.

3. Lecture et affichage du fichier (interface graphique)

- a. Permettre à l'utilisateur de choisir un fichier à l'aide de `JFileChooser` (cf. 1.b)
- b. Lire le fichier texte (avec Scanner) (cf. 1.a) dans une liste d'`Item`.
- c. Afficher les questions et leurs réponses dans un `JTextArea`.
- d. Construire une interface graphique de navigation (Choisir un fichier, Afficher, Quitter)
 - Par exemple avec des boutons

4. Stockage des données

- a. Définir une classe permettant de créer une base de données comportant une table qui contiendra les données d'une liste d'instances d'`Item`. (A choix `SQLite` ou `MySQL` ou `Oracle`).
- b. Recopier les données d'un fichier texte dans la table de votre base de données.
- c. Adapter en conséquence les classes existantes pour qu'elles obtiennent leurs données à partir de la base de données.

5. Interactions avec l'utilisateur

- a. Choisir une question, la poser à utilisateur et lui donner le résultat de la vérification**
Cela peut être toujours la même question (la première) ou une question tirée au hasard.
- b. Présenter successivement toutes les questions à l'utilisateur**
Il sera utile de prévoir un bouton pour abandonner le questionnaire.
- c. Changer l'ordre des questions aléatoirement.**
- d. Tenir à jour un décompte de erreurs et bonnes réponses de l'utilisateur**
Au minimum à afficher en fin de questionnaire et en cas d'abandon.
- e. Introduire une possibilité pour l'utilisateur de ne pas répondre à certaines questions**
Prévoir de modifier le décompte et de donner la possibilité à l'utilisateur de revenir sur ces questions.
- f. Stocker les réponses à un questionnaire dans la base de données et cela pour différents utilisateurs.**

6. (optionnel) Évolutions envisageables

- a. Créer un questionnaire à partir d'une sélection parmi les questions-réponses présentes dans la base de données.**