

Exercice P1 – Quiz

Le but de cet exercice est de créer un simple quiz. Une version exécutable du programme se trouve dans votre répertoire de classe. Le diagramme de classes se trouve en dernière page.

Partie Unimozzer

1. Créez le projet **ExerciceP1** dans Unimozzer.
2. Créez la classe `Question` avec les attributs suivants :
 - a. `question` contient le texte représentant la question elle-même.
 - b. `answer` contient la réponse correcte à la question.
 - c. `total` contient le nombre total de fois qu'un utilisateur a répondu à la question.
 - d. `correct` contient le nombre de réponses correctes données à la question.
3. Ajoutez des accesseurs et des manipulateurs pour tous les attributs.
4. Ajoutez un constructeur qui crée une nouvelle question et en initialise les attributs.
5. Ajoutez la méthode `getPercentage` qui retourne le pourcentage de réponses correctes données à la question.
6. Ajoutez la méthode `toString` qui retourne une chaîne de caractères de la forme suivante : "<question>, <answer>"
Les parties entre < > sont à remplacer par les valeurs réelles des attributs.
7. Ajoutez la classe `Quiz` qui gère une liste de questions `alQuestions`. Il y a un deuxième attribut `randomQuestion` qui contient la question à laquelle l'utilisateur est censé répondre actuellement.
8. Ajoutez la méthode `generateRandomQuestion` qui choisit aléatoirement une des questions et la stocke dans l'attribut `randomQuestion`.
9. Ajoutez la méthode `getRandomQuestion` qui retourne la question stockée dans `randomQuestion`.
10. Ajoutez la méthode `indexOfEasiestQuestion` qui retourne l'index de la question la plus facile, c.-à-d. l'index de la question avec le plus haut pourcentage de réponses correctes.
11. Ajoutez la méthode `indexOfAnswer` qui retourne l'index de la première question avec la réponse indiquée en paramètre.

Partie NetBeans

12. Ajoutez les méthodes `isEmpty`, `toArray`, `get`, `add` et `remove` à la classe `Quiz`.
13. Créez l'interface graphique de la page suivante. Pensez à renommer les variables des composants graphiques auxquels vous devez accéder à partir du code Java.
14. Ajoutez un attribut `quiz` à la classe `MainFrame` pour lui permettre de communiquer avec le modèle et initialisez-le avec un objet de la classe `Quiz`.
15. Quand le programme est lancé, seulement le panneau d'édition (le panneau inférieur) est visible.
16. Le bouton **Add** permet d'ajouter une nouvelle question au quiz.
17. Le bouton **Modify** permet de modifier la question sélectionnée dans la liste graphique.
18. Le bouton **Search Easiest Question** permet d'afficher les détails de la question avec le plus haut pourcentage de réponses correctes.

19. Le bouton **Show Question** permet de chercher et d'afficher la question correspondant à la réponse indiquée dans le champ de texte **Edit answer**. S'il n'y en a pas, le champ de texte **Edit question** est vidé. Ainsi il est possible de vérifier s'il y a déjà une question portant sur un certain sujet.
20. Le bouton **Delete Selected** supprime la question sélectionnée de la liste.
21. Quand la sélection de la liste graphique change, les valeurs des attributs de la question sélectionnée sont utilisées pour remplir les champs de texte du panneau d'édition.
22. Ajoutez une méthode `updatePlayPanel` à la classe `Quiz`. Elle choisit une question aléatoire et l'utilise pour actualiser toutes les étiquettes du panneau de jeu.
23. Le bouton **Play Quiz** cache le panneau d'édition et affiche le panneau de jeu, mais uniquement si le quiz contient au moins une question. Si c'est le cas toutes les étiquettes du panneau de jeu doivent être correctement initialisées (cf. exécutable).
24. Le bouton **Submit** permet à l'utilisateur de vérifier sa réponse. Dans tous les cas il faut indiquer si la réponse est correcte, actualiser les statistiques de la question, puis générer et afficher une nouvelle question aléatoire.
25. Le bouton **Edit Quiz** permet à tout moment de cacher le panneau de jeu et de retourner au panneau d'édition.

Interface graphique à réaliser

The interface is divided into two main sections by a horizontal separator.

Top Section (Edit/View Mode):

- Header: `<Question>`
- Field: Answer (text input)
- Button: Submit
- Header: `<Comment>`
- Field: Percentage of correct answers: `<percentage>`
- Button: Edit Quiz

Bottom Section (Edit/Play Mode):

- Header: Edit question
- Field: (text input)
- Header: Edit answer
- Field: (text input)
- Buttons: Add, Modify, Search Easiest Question, Show Question
- Header: Play Quiz (button)
- List (on the right):
 - Item 1
 - Item 2
 - Item 3
 - Item 4
 - Item 5
- Button: Delete Selected

