PROJET JAVA

L'objectif de ce projet est d'implémenter un jeu de Questions / Réponses entre au moins 4 joueurs au début du jeu. Les joueurs sont éliminés à travers les différentes phases du jeu jusqu'à ce qu'il ne reste plus qu'un seul gagnant.

Le jeu se base sur une liste de questions auxquelles les candidats doivent répondre à tour de rôle. En fonction des réponses, un score est cumulé pour chaque joueur et les candidats ayant les meilleurs scores sont sélectionnés pour les phases suivantes.

Les questions traitent de H (H >=10) thèmes et sont de 3 types différents :
- QCM : Le joueur peut choisir une réponse entre plusieurs
- VF : Le joueur peut répondre uniquement par vrai / faux
- RC : Le joueur peut saisir une réponse courte.

De plus, les questions ont 3 niveaux de difficulté :
- 1 : Niveau facile
- 2 : Niveau moyen
- 3 : Niveau difficile

La réalisation de cette application s'appuie sur les spécifications fonctionnelles détaillées dans la section qui suit.

- ## Description fonctionnelle

Pour réaliser le jeu de Question / Réponse, il convient d'implémenter les classes suivantes :

Les classes des types des questions

Une question de type QCM est décrite par un texte, 3 variables de réponses proposées et une variable bonne réponse.

Une question de type VF est décrite par son texte et sa bonne réponse de type booléen.

Une question de type RC est décrite par son texte et une variable bonne réponse de type chaîne de caractères.

Les questions sont numérotées de manière séquentielle indépendamment de leurs thèmes.

Donner l'implémentation **des questions de telle sorte que pour chaque question** on ait :
- Un constructeur
- Une méthode **qui retourne une représentation textuelle** d'une question décrite par un numéro, un thème, un niveau de difficulté, un énoncé et ses champs de réponse en fonction de son type
- Une méthode **de saisie** qui permet la saisie d'une question d'un type donné.

Thèmes

Un thème est désigné par une chaîne de caractères (Sciences, Sport, Histoire, ..). On dispose d'au moins 10 thèmes différents. Lorsqu'un thème est sélectionné, un indicateur est positionné sur ce thème pour ne pas le sélectionner lors de la prochaine sélection.

Donner l'implémentation de la classe **Thèmes** comportant:
- Un constructeur
- Une méthode de **sélection d'un thème** qui demande à l'utilisateur un thème et renvoie son indice
- Une méthode de **sélection de cinq thèmes**
- Une méthode **qui retourne une représentation textuelle** de tous les thèmes et la valeur de leur indicateur.

Questions

Pour chaque thème, on dispose d'un ensemble de questions dont le nombre et les types sont variables. L'ensemble des questions relatives à un thème est stocké dans une liste chaînée de questions. A chaque fois, un indicateur de la question sélectionnée est mis à jour.

Le nombre de questions de chaque thème est donné aléatoirement entre 5 et 10.

Donner l'implémentation de la classe **Questions** contenant :
- Un constructeur
- Une méthode **d'ajout d'une question**
- Une méthode de **suppression d'une question par son rang dans la liste**

- Une (ou plusieurs) méthode(s) **de sélection d'une question** qui sélectionne une question pour un joueur selon une politique qui sera définie dans les différentes phases du jeu
- Une méthode qui **retourne une représentation textuelle** de tous les thèmes et de leurs questions.

Phase

Une partie de jeu se déroule en plusieurs phases. A chaque phase, le joueur ayant le plus faible score est éliminé et les autres sont sélectionnés pour la phase qui suit.

Définir une interface **Phase** comportant les méthodes suivantes :
- de sélectionner des joueurs
- de dérouler une phase de jeu

NB : Il est possible d'ajouter toute autre classe jugée nécessaire pour implémenter une partie de jeu à partir de l'interface Phase.

Joueur

Un joueur est décrit par un numéro, un nom, un score et un état (sélectionné, gagnant, super gagnant, éliminé ou en attente). Pour chaque réponse correcte, le score est incrémenté d'une valeur dépendant de la phase du jeu. Le numéro du joueur est un numéro qui commence à 100 et est incrémenté de 10 à chaque fois (100, 110, 120, 130, …).

Donner l'implémentation de la classe **Joueur** comportant :
- Un constructeur
- Une méthode **de saisie**
- Une méthode **qui retourne une représentation textuelle**
- Une méthode **de mise à jour du score**
- Une méthode **de changement de l'état**

Joueurs

L'ensemble des joueurs est stocké dans un tableau 20 joueurs candidats dont au moins 4 pourront participer au jeu.

Donner l'implémentation de la classe **Joueurs** comportant :

- Un constructeur
- Une méthode de sélection aléatoire d'un joueur du tableau
- Une méthode générant aléatoirement l'ensemble des joueurs participant au jeu
- Une méthode **qui retourne une représentation textuelle** de l'ensemble des joueurs participant au jeu

Remarques :
1. Pour des besoins de tests, il faut prévoir à l'intérieur des classes des constructeurs supplémentaires permettant de remplir par programme les structures nécessaires au déroulement du test afin d'éviter la saisie de données longue et fastidieuse.
2. Il faudra remplir avec des données correctes et sensées.
3. Pour les noms des joueurs, utiliser les lettres de l'alphabet A, B, C, …, Z.
4. S'il y a lieu surcharger et/ou redéfinir les méthodes et/ou créer des sous-classes.

- ## Application à réaliser

Phases du jeu

Le jeu de Questions / Réponses à réaliser comprendra trois phases où 4 joueurs s'affrontent autour de questions sur 10 thèmes:

Phase I :
Dans cette phase, le jeu se déroule entre 4 joueurs choisis aléatoirement. Un thème parmi 10 est sélectionné automatiquement du tableau dans un ordre séquentiel (un indicateur du dernier thème sélectionné est mis à jour, après le choix du 10ème thème, on revient au premier).

Une question de niveau facile est sélectionnée pour chacun des joueurs selon une politique Round-Robin (i.e de manière circulaire). Les 4 joueurs répondent à leurs questions séparément, le score est incrémenté de 2 si la réponse donnée est correcte.

Phase II :
Le jeu se déroule entre les trois joueurs gagnants de la phase I. Cette phase propose deux questions de niveau moyen pour chaque joueur (une question par thème choisi).

A ce niveau, les questions porteront sur uniquement 6 thèmes choisis aléatoirement.

A tour de rôle et de manière alternée, chaque joueur choisit un thème (ensuite un deuxième) qui est supprimé des choix aussitôt sélectionnés.

Une question de niveau moyen (dans le thème choisi) est sélectionnée selon la politique Round-Robin et présentée au joueur.

Le score du joueur qui donne la bonne réponse est incrémenté de 3.

Phase III :

Dans cette phase, le jeu se déroule entre les deux joueurs gagnants de la phase II. Les deux questions de niveau difficile porteront sur trois thèmes choisis par le concepteur du jeu.

Une question de niveau difficile (dans le thème choisi) est sélectionnée selon la politique Round-Robin et présentée au joueur.

Le score du joueur donnant la bonne réponse est incrémenté de 5.

Description de l'application

L'application doit, de manière générale, permettre de réaliser les actions suivantes :

1. Afficher les 10 thèmes choisis
2. Créer une liste de questions pour chaque thème
3. Afficher toutes les questions d'un niveau n donné par thème
4. Ajouter une question à la liste pour un thème donné
5. Supprimer une question de numéro n de la liste pour un thème donné
6. Créer le tableau de joueurs et afficher leurs états
7. Lancer une partie du jeu avec 4 joueurs choisis en affichant toutes les étapes du déroulement du jeu
8. Quitter le jeu
9. D'autres actions peuvent être ajoutées si nécessaire.

- **Fonctionnalités optionnelles BONUS**

Gestion de conflits

Pour chaque joueur un timer régis par un Thread est associé. Il démarre lorsque son joueur obtient la main pour répondre à la question, et s'arrête dès que la réponse est fournie.

En cas d'égalité de scores entre les joueurs à une phase donnée, ils seront départagés grâce aux valeurs des timers. Ainsi, les joueurs ayant été les plus rapides seront ceux qui se qualifient à la phase d'après.

Dans le cas d'égalité, à la fois, des scores et timers, proposer jusqu'à trois questions supplémentaires pour les départager. Après cela, faire une sélection aléatoire pour passer à l'étape suivante.

Le grand jeu

Faire en sorte que l'application permette de considérer 3 groupes de 4 joueurs

- Implémenter une partie de jeu pour chaque groupe de 4 joueurs
- Récupérer les numéros des trois joueurs gagnants
- Implémenter le grand jeu entre les trois joueurs gagnants, la même politique est reprise commençant à la phase II qui comportera trois thèmes au choix et une question par joueur.
- Affichant toutes les étapes du déroulement du jeu.