

TP 2 : Classe et Objet

L'objectif de ce TP est de modéliser les QCM (questionnaires à choix multiples). Un QCM contient plusieurs questions. Une question est représentée par un texte qui pose la question et par une seule réponse possible. Ces réponses possibles peuvent être des bonnes réponses ou des mauvaises réponses.

Le programme principal affiche les questions en numérotant les réponses possibles. L'utilisateur indique quelle réponse il choisit. Lorsque toutes les questions ont été posées, le programme donne une note à l'utilisateur. Pour commencer, choisissez la notation suivante: chaque bonne réponse compte 1 point ; la note est la moyenne des réponses multipliée par 20, et arrondie à l'entier le plus proche.

Les classes que vous allez écrire devront pouvoir faire fonctionner le programme principal donné en Annexe.

Partie 1 : Classe Question

Écrire une classe Question représentant une des questions avec les réponses possibles. Vous conserverez l'indice de la bonne réponse dans l'état de la question.

Pour ranger les réponses à la question vous utiliserez la classe `ArrayList<String>` qui représente une liste de String, en quelque sorte un tableau de String, dont la dimension n'est pas fixe. Après sa création un `ArrayList` est vide ; ensuite vous pouvez ajouter des éléments dans la liste par la méthode `add`. Le premier élément ajouté aura l'indice 0, le 2ème l'indice 1, et ainsi de suite. Voici quelques exemples d'utilisation de cette classe :

```
// Création de la liste des réponses
private ArrayList<String> reponses = new ArrayList<String>();

...
// Ajouter une réponse dans la liste
this.reponses.add("Une île de l'océan Atlantique");
// Obtenir la i + 1 ème réponse (la 1ère réponse est this.reponses.get(0)) :
String r = this.reponses.get(i);
// Nombre de réponses dans la liste : this.reponses.size();
// Parcourir toutes les réponses par une boucle for-each :
for (String reponse : reponses) {
// Parcourir toutes les réponses par une boucle for "ordinaire" :
for (int i = 0; i < reponses.size(); i++) {
... reponses.get(i) ...
```

Définissez les méthodes suivantes :

- ✓ `public Question(String enonce)` //Un constructeur de la classe
- ✓ `public void ajouterReponse(String reponse, boolean bon)` // Ajoute une réponse possible à la question. Elle prend en paramètre `reponse`, `bon` true si c'est une bonne réponse.

- ✓ `public void afficher(String enteteQuestion) // Permet d'afficher la question sur l'écran. Elle prend en paramètre entete Question ce qu'il faut afficher juste avant d'afficher la question.`
- ✓ `public boolean lireReponseAuClavier() // Lit la réponse au clavier de l'utilisateur et retourne true si et seulement si la réponse est bonne.`
- ✓ `public String toString() // surcharge la méthode toString`

Partie 2 : Classe Questionnaire

Écrivez une classe Questionnaire. Un questionnaire est une liste de questions. Pour ranger les questions du questionnaire vous utiliserez aussi la classe ArrayList :

```
// Création de la liste des questions
private ArrayList<Question> questions = new ArrayList<Question>();
...
// Ajouter une question dans la liste
this.questions.add(question);
// Obtenir la i + 1 ème question :
Question q = this.questions.get(i);
// Nombre de questions dans la liste : this.questions.size();
```

Définissez les méthodes suivantes :

- ✓ `public Questionnaire(String titre) // Crée un questionnaire avec un titre. Elle prend en paramètre titre, titre du questionnaire.`
- ✓ `public void ajouterQuestion(Question question) //Ajoute une question au questionnaire. Elle prend en paramètre question`
- ✓ `public boolean resteDesQuestions() //Indique s'il reste des questions à poser. return true si et seulement s'il reste des questions à poser.`
- ✓ `public void reinitialiser() // Réinitialise le QCM. La prochaine question à poser sera la première question.`
- ✓ `public int resultat() // Retourne le résultat obtenu par l'utilisateur`
- ✓ `public void afficherQuestionSuivante() // Affiche la question suivante du questionnaire.`
- ✓ `public void lireReponseQuestion() // Lit la réponse au clavier la réponse de l'utilisateur à la question qui vient d'être affichée.`
- ✓ `public String titre() // Retourne le titre du QCM.`
- ✓ `public String toString()`

Partie 3 : Test final

Recopiez la Classe TestQcm donnée en annexe afin de testez les classes Question et Questionnaire.

Ajoutez d'autres questions à ce questionnaire ou créez votre propre questionnaire.

Annexe Classe Test QCM

```
public class TestQCM{
    public static void main(String[] args) {
        Questionnaire questionnaire = new Questionnaire("QCM de géographie");
        Question question = new Question ("Qu'est-ce que Java ?");
        question.ajouterReponse("Une île de l'océan Atlantique", false);
        question.ajouterReponse("Une île de l'océan Indien", true);
        questionnaire.ajouterQuestion(question);
        question = new Question("Combien d'habitants en France ?");
        question.ajouterReponse("Moins de 20 millions", false);
        question.ajouterReponse("Entre 20 et 50 millions", false);
        question.ajouterReponse("Plus de 50 millions", true);
        questionnaire.ajouterQuestion(question);
        /* Pour indiquer que la prochaine question à poser sera la 1ère question et initialiser le total
        des points obtenus par l'utilisateur*/
        questionnaire.reinitialiser();
        System.out.println(questionnaire.titre());
        System.out.println();
        while (questionnaire.resteDesQuestions()) {
            // Affiche une question et lit la réponse de l'utilisateur
            questionnaire.afficherQuestionSuivante();
            questionnaire.lireReponseQuestion();
        }
        System.out.println();
        System.out.println("Votre note : " + questionnaire.resultat());
        // Prépare le questionnaire pour la prochaine fois
        questionnaire.reinitialiser(); }}
}
```