

TD-TP6 – R2.01

Semaine 11

Objectifs

Utiliser l'héritage et comprendre le chaînage des constructeurs.

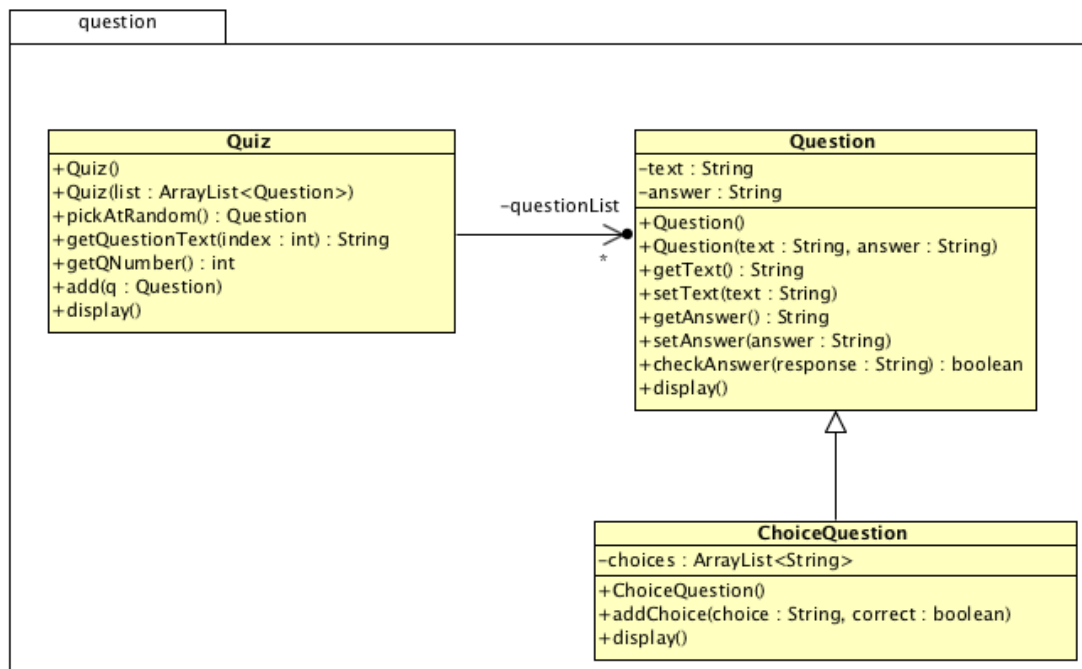
Observer le polymorphisme.

Utiliser la classe Scanner pour réaliser un dialogue.

1- L'application Quiz

L'application Quiz consiste à modéliser un questionnaire contenant des questions simples et des questions à choix multiples.

Une question est représentée par le texte de la question et le texte de la réponse.



Une question à choix multiple est une question mais avec un choix à faire parmi plusieurs réponses possibles (une seule réponse).

2- La classe Quiz

La classe **Quiz** permet de stocker une liste de questions. Pour cela elle a un attribut d'instance **questionList** qui sera implémenté par un **ArrayList**.

- Le constructeur sans paramètre crée l'attribut **questionList** en tant qu'**ArrayList**. Ce sera une liste vide au départ.
- Le deuxième constructeur prend en paramètre l'**ArrayList** de questions et l'utilise pour créer son attribut **questionList**.
- La méthode **add(Question q)** permet d'ajouter une question à la liste.
- La méthode **pickAtRandom()** retourne une question tirée au hasard dans la liste **questionList** (on peut faire une copie défensive).

- La méthode **getQuestionText(int index)** retourne le texte de la question dont d'indice est passé en paramètre.
- La méthode **getQNumber()** retourne le nombre de questions (taille de l'ArrayList).
- La méthode **display()** affiche toutes les questions du questionnaire.

3- La classe Question

La classe **Question** représente une question type avec un attribut « text » qui est le texte d'une question et un attribut « answer » qui est la réponse à la question.

- La méthode **boolean checkAnswer(String response)** vérifie que la réponse passée en paramètre correspond à l'attribut **answer** de la question et retourne en valeur true si c'est le cas et false sinon. Pour que la comparaison des chaînes de caractères ne tienne pas compte des majuscules ou des minuscules il faut utiliser la méthode **equalsIgnoreCase** de la classe **String**.
- La méthode **display()** affiche à l'écran le texte de la question.
- Le constructeur sans paramètre initialise le texte et la réponse avec une chaîne vide.
- Le deuxième constructeur prend en paramètre le texte de la question et la réponse pour initialiser les attributs d'instance.

Exemple de dialogue (avec un scanner)

```
$ java ScenarioQuestion
Who was the inventor of Java?
Your answer: James Gosling
true
$
```

4- La classe ChoiceQuestion

Il s'agit maintenant de spécialiser la classe **Question** pour avoir des questions à choix multiples en créant une sous-classe de la classe **Question**.

La classe **ChoiceQuestion** utilise un **ArrayList** de chaînes de caractères pour stocker les différents choix de réponses possibles à la question.

La réponse à une question à choix multiples sera le numéro de la réponse.

- Le constructeur **ChoiceQuestion()** crée l'ArrayList **choices** qui contiendra les choix de réponse.
- La méthode **addChoice(String choice, boolean correct)** permet d'ajouter une réponse possible à la liste **choices**. Le booléen **correct** permet de savoir si la réponse est la bonne. Ce booléen est à vrai si la réponse que l'on ajoute est la bonne ou false sinon. Si **correct** est vrai alors on appellera **setAnswer** pour stocker la bonne réponse qui sera le numéro de la réponse (indice+1 dans la liste car les choix sont numérotés à partir de 1).
- La méthode **display()** utilisera la méthode héritée de même nom pour afficher le texte de la question, puis elle affichera les réponses proposées.

Exemple de dialogue

```
$ java ScenarioQuestion
In which country was the inventor of Java born?
1: Australia
2: Canada
3: Denmark
4: United States
Your answer: 2
true
```

Etapes du travail

Remarque : les classes de l'application sont dans le package « question », les classes scenario sont à l'extérieur de ce package

1. Implémenter entièrement en Java la classe **Question** documentée au format Javadoc et en respectant la spécification donnée dans le diagramme de classes UML.
2. Ecrire une classe **ScenarioQuestion** (à l'extérieur du package question) qui fait quelques tests sur la classe **Question** et réalise un dialogue avec l'utilisateur pour lire au clavier la réponse à la question en utilisant la classe `java.util.Scanner` vu en cours.
3. Implémenter en Java la classe **ChoiceQuestion** documentée au format Javadoc et en respectant la spécification donnée dans le diagramme de classes UML.
4. Compléter la classe **ScenarioQuestion** pour créer vos questions à choix multiples et les intégrer au dialogue
5. Implémenter en Java la classe **Quiz** documentée au format Javadoc et en respectant la spécification donnée dans le diagramme de classes UML.
6. Implémenter une nouvelle classe **ScenarioQuiz** (à l'extérieur du package question) qui permet de définir un ensemble de questions et de créer le quiz.
7. Pour votre scenario, vous pouvez faire une boucle d'un certain nombre de fois qui à chaque pas de la boucle prendra une question au hasard (ou on) dans la liste de questions du quiz et vous demandera la réponse.

Pour aller plus loin :

- Vous pouvez sophistiquer la méthode d'analyse de la réponse donnée par l'utilisateur (explorer l'API de la classe String) : supprimer des espaces en trop, permettre une réponse plus courte, comme juste un nom sans le prénom pour un inventeur.
 - Dans votre classe Scenario vous pouvez compter les bonnes réponses et attribuer un score à l'utilisateur.
 - Vous pouvez ajouter d'autres sous-classes à la classe Question :
- par exemple une sous-classe de ChoiceQuestion : MultipleChoiceQuestion où plusieurs réponses sont possibles