

# Examen CPOO du 29 novembre 2021 9h15

- L'examen dure **1 heure**.
- Le barème est donné à titre **indicatif**.

## Avant de commencer

Cet examen consiste à **compléter un projet que nous vous fournissons**, puis à **rendre ce projet sur le LMS au format ZIP** au signal de votre surveillant (voir les instructions à la fin du sujet).



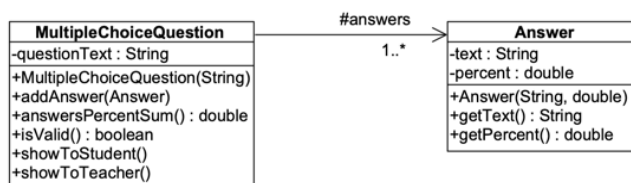
### Comment obtenir le projet à compléter ?

Ouvrez dans VS Code le projet que vous devrez compléter :

1. Téléchargez et décompressez l'archive [exam-2021-11-B.zip](#)
2. Identifiez le répertoire **exam-2021-11-B le plus interne** (celui qui contient directement **src/** et **lib/**).
3. Lancez VS Code et ouvrez ce répertoire (menu File > Open...).

## Exercice 1 : modéliser une question à choix multiple (8 points)

Cet exercice consiste à modéliser une question à choix multiple, modélisée par le texte de la question et une liste de réponses, dont chacune donne un pourcentage des points de la question. Les classes à écrire sont décrites dans le diagramme de classes suivant :



### La classe Answer (4 points)



1. Complétez et corrigez le classe **Answer** fournie.
2. Validez votre code à l'aide du test unitaire JUnit **AnswerTest**.



Point de vigilance : avez-vous pensé aux visibilités ?



## La classe `MultipleChoiceQuestion` (4 points)



La classe `MultipleChoiceQuestion` contient un texte (amorce de la question), et une liste de réponses possibles (objets de classe `Answer`), dont la somme des pourcentage de points doit atteindre 100%. Complétez maintenant cette classe :

- Ajoutez les attributs, le constructeur et trois premières méthodes :
  - `addAnswer(...)` ajoute une des réponses possibles
  - `answersPercentsSum()` retourne la somme des pourcentages des réponses possibles
  - `isValid()` retourne `true` si et seulement la somme des pourcentage est 100%
- Validez votre code à l'aide du test unitaire `JUnit MultipleChoiceQuestionTest`
- Ajoutez les méthodes restantes :
  - `showToStudent()` affiche dans la console le texte de la question puis tous les textes des réponses possibles ; chacune des réponses doit être précédée d'un numéro de réponse. Par exemple :
 

```
Combien de salles informatiques y a-t-il au CI ?
1. 0
2. 14
```
  - `showToTeacher()` affiche en plus le pourcentage de points devant chaque réponse. Voici un exemple de sortie (votre format d'affichage des pourcentages peut être différent) :
 

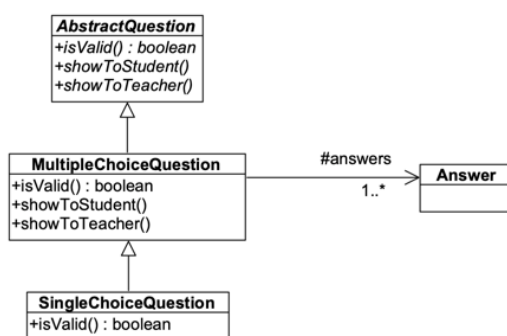
```
Combien de salles informatiques y a-t-il au CI ?
0.0 1. 0
100.0 2. 14
```
- Validez ces méthodes en dé-commentant le premier bloc de `QuestionsMain`, puis en l'exécutant et en vérifiant l'affichage dans la console.



Points de vigilance : avez-vous pensé à tous les attributs du diagramme ? Vos constructeurs sont-ils complets ? Avez-vous pensé aux visibilités ?

## Exercice 2 : sous-types de questions (8 points)

Cet exercice consiste à utiliser le sous-typage pour modéliser différents types de question, conformément au diagramme de classes suivant :



## La classe `SingleChoiceQuestion` (4 points)



- Écrivez maintenant la classe `SingleChoiceQuestion` conformément au diagramme UML. En particulier, `isValid()` retourne `true` si et seulement si il y a une seule réponse avec un pourcentage positif, et que ce pourcentage est de 100%.
- Validez votre classe en dé-commentant le deuxième bloc de `QuestionsMain`, puis en l'exécutant et en vérifiant l'affichage dans la console.



Points de vigilance : Vos constructeurs sont-ils complets ? Les relations UML sont-elles bien transcrites ? Avez-vous pensé aux visibilitées ?



#### La classe abstraite `AbstractQuestion` (4 points)



1. Écrivez maintenant la classe `AbstractQuestion` conformément au diagramme UML. Rappel : en UML l'italique signale les classes et méthodes abstraites.
2. Faites les modifications strictement nécessaires dans la ou les autres classes du diagramme pour réaliser cette interface.

### Exercice 3 : signalement d'anomalie par exception (4 points)



#### Pas de réponse avec pourcentage hors de [0, 100] ou sans texte



1. Quand le pourcentage de points d'une réponse n'est pas compris entre 0 et 100 ou quand son texte est `null` ou vide (`""`), on considère qu'il s'agit d'une anomalie. Complétez la classe correspondante pour empêcher la création d'objets à partir d'un paramètre de constructeur anormal. Chaque anomalie devra être signalée par une exception standard appropriée.
2. Validez votre code en commentant et dé-commentant la classe `ExceptionsMain` fournie.



Points de vigilance : avez-vous utilisé une classe standard appropriée pour signaler l'anomalie ? Avez-vous testé et vérifié que l'anomalie était signalée dans la console ? L'information fournie à l'utilisateur est-elle explicite ?

### Comment rendre votre travail ?



Voici les étapes à suivre :

1. Si nécessaire, **sauvegardez** tous vos fichiers VS Code : menu "File > Save All"
2. Faites une **nouvelle archive ZIP** du dossier `exam-2021-11-B` : cette archive contient les fichiers que vous avez modifié pendant l'examen.
3. **Déposez** cette archive dans le devoir *Dépôt de l'examen du 29 novembre 2021* :
  1. Cliquez sur "Ajouter un travail" tout en bas de la page.
  2. Ajoutez l'archive que vous venez de créer. **Ne confondez pas** avec l'archive d'origine.
  3. Cliquez sur "Enregistrer". Votre travail passe dans l'état "Remis pour évaluation", mais en cas d'erreur, vous pouvez encore le modifier en rendant une autre archive zip.