

Modalités

- ✓ Durée : 2h.
- ✓ Vous pouvez utiliser une feuille de notes manuscrites — écrites à la main non photocopées — recto-verso de taille quelconque.
- ✓ Rendez l'annexe jointe si vous l'utilisez
- ✓ Dans ce devoir,
 - tous les accessoires nécessaires sont supposés déjà définis,
 - la classe `Liste.java` est directement utilisable : elle est supposée placée au bon endroit,
 - les listes d'éléments peuvent être représentées au choix par des `ArrayList` ou par des tableaux (Utilisez le même système de représentation pour tout l'examen).

Modalités

- ✓ Documents autorisés : une feuille manuscrite
- ✓ Durée : 2h

Questionnaires

Il s'agit d'évaluer des étudiants en utilisant des questionnaires informatisés.

1. Première étape

Une question du questionnaire comporte un énoncé ; par exemple :

```
"L'instruction (3) permet-elle de faire une/des instantiation/s ?" 1
```

Elle a aussi une valeur de vérité booléenne (`true` ou `false` suivant que l'énoncé est juste ou faux). Ici, cette valeur est `true`.

Elle est associée aussi à une réponse booléenne, donnée indirectement par l'étudiant (il entre oui pour une réponse qu'il considère comme juste et non dans le cas contraire, ce qui donne une réponse `true` ou `false`).

Elle est enfin dotée d'un score maximal (ici par exemple 0.5) qui est le nombre de points remportés si la réponse de l'étudiant correspond à la valeur de la question (la réponse attendue). Ici, si la réponse de l'utilisateur est oui, il obtient les 0.5 points de la question. Sinon, son score reste à 0.

1. Complétez le premier diagramme de l'annexe avec les déclarations des attributs (variables d'instances) nécessaires pour représenter une instance de la classe `Question`.
2. Écrivez la méthode `setReponse()` de la classe `Question` qui affiche l'énoncé de la question en demandant à l'utilisateur de rentrer oui ou non. Par exemple, pour la question vue plus haut :

```
L'instruction (3) permet-elle de faire une/des instantiation/s ? {oui,non} :
```

1 Voir plus loin (Questionnaire) pour comprendre instruction (3).

La méthode demande ensuite la réponse tant qu'elle ne vaut pas oui ou non et stocke le résultat booléen correspondant.

3. Écrivez la méthode `isReponseCorrecte()` qui restitue `true` si et seulement si la réponse de l'utilisateur est correcte, c'est-à-dire si elle correspond bien à la valeur de la question (la bonne réponse).
4. Écrivez la méthode `getScore()` qui restitue le score maximal de la question si la réponse de l'utilisateur est correcte et 0 sinon.

Un questionnaire comporte une liste de questions. Il a un énoncé global et une note maximale² atteinte quand toutes les réponses sont correctes.

L'énoncé global d'un questionnaire peut par exemple être la chaîne suivante :

```
"(1) Disque d1 = new Disque(new Point(-1,2),1);\n"+  
"(2) Point p1 = new Point(1,1);\n"+  
"(3) Segment s1 = new Segment(d1.getCentre(),p1);\n"+  
"(4) s1.getExt1().setX(0);\n"+  
"(5) d1.setCentre(new Point(p1));\n"+  
"(6) p1.setY(0);\n"+  
"(7) d1 = null;\n"+  
"(8) p1 = null;\n"+  
"\n"+  
"Indiquez si les énoncés suivants sont corrects pour les instructions précédentes." + "\n"
```

5. Complétez le premier diagramme de l'annexe avec les déclarations des attributs et les relations nécessaires pour représenter une instance de la classe `Questionnaire` et ses relations avec les instances de `Question`.
6. Écrivez la méthode `setReponses()` de `Questionnaire` qui affiche l'énoncé du questionnaire puis demande à l'étudiant les réponses à chaque question.
7. Écrivez la méthode `getNote()` de la même classe, qui calcule la note globale obtenue. Cette note est la somme des scores effectués sur les différentes questions divisée par la somme des scores maximaux possibles, et le tout multiplié par la note maximale du questionnaire.

2. Deuxième étape

On souhaite distinguer les réponses — propres aux étudiants — des questions — qui sont les mêmes pour tous les étudiants.

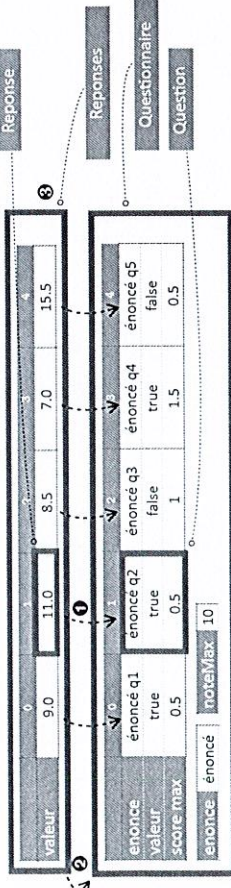
Dans ces conditions, une question ne comprend plus qu'un énoncé, une valeur booléenne et un score maximal (pas de réponse), un questionnaire contient toujours une liste de questions.

On ajoute alors une classe `Reponse` dont chaque instance correspond à une question (référence une question **Q**) et qui correspond à la réponse d'un étudiant donné. Sa valeur est la traduction booléenne de la réponse (oui ou non) de l'étudiant à la question (Elle correspond à réponse dans la première étape).

1. Complétez le deuxième diagramme de l'annexe en tenant compte des nouvelles spécifications pour les classes `Question` et `Reponse`.
2. Pour que les fonctionnalités de la première étape soient respectées, quelles méthodes doit contenir la classe `Reponse` ? Écrivez-les.

2 On peut vouloir une note sur 10 ou sur 20 par exemple.

Annexe



Pour respecter les fonctionnalités de la première étape, il faut ajouter une classe Reponses dont une instance (⊖) représente la collection des réponses données par un étudiant à un questionnaire. La classe Reponses référence ce questionnaire (⊕).

3. Complétez le deuxième diagramme de l'annexe pour ce qui concerne les spécifications des classes Questionnaire et Reponses et les relations entre instances.
4. Écrivez la méthode setReponses() de la classe Reponses. Cette méthode crée et référence la collection de réponses d'un étudiant au questionnaire. Chaque réponse de la collection doit correspondre à une question de même rang dans l'instance de questionnaire correspondante.

3. Troisième étape

En plus des questions à réponse booléennes (classe Binaire dans le diagramme), on souhaite désormais que le questionnaire puisse contenir des questions à réponse libre (classe TexteLibre), où l'étudiant répond par un texte à la question. La note associée à la réponse doit alors être entrée par un évaluateur (Elle ne peut pas être calculée automatiquement).

1. Traduisez ces nouvelles spécifications en spécifiant dans le troisième diagramme de l'annexe les déclarations d'attributs et les relations nécessaires (Ce diagramme ne comprend plus les questionnaires ni les collections de réponses).
2. Indiquez sans écrire de code comment doivent être réparties et définies les méthodes nécessaires pour...
 - ✓ permettre à l'étudiant d'entrer sa réponse.
 - ✓ restituer le score associé à une réponse.

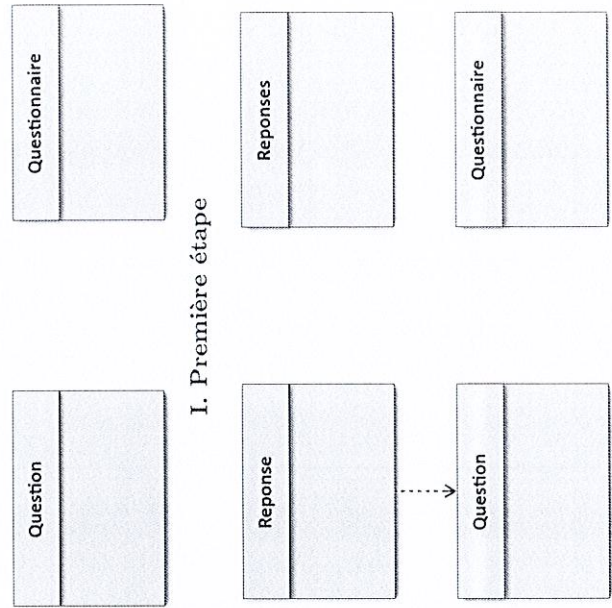
4. Bonus (3 points)

On souhaite ajouter aux questions des QCM (Questions à Choix Multiples) et des QCU (Questions à Choix Unique). Dans les deux cas, la question est constituée d'une liste de questions booléennes. Dans un QCM, potentiellement, plusieurs questions ont la valeur true (plusieurs énoncés sont justes) et l'étudiant n'a le score maximal que s'il ne sélectionne que ces questions³. Pour les QCU, une seule réponse est juste et l'étudiant ne peut sélectionner qu'une question. La réponse à la question n'est valable (score max) que si c'est la question contenant l'énoncé juste qui est sélectionnée.

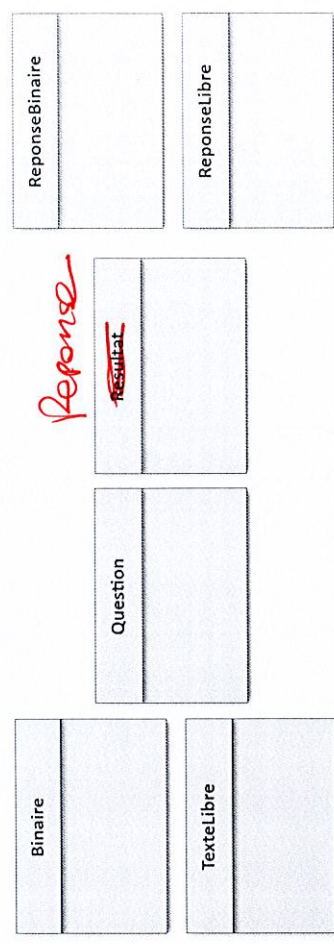
Sans programmer, dites comment représenter cette situation ?

³ Il existe plusieurs moyen d'évaluer une réponse à une question de ce type. Nous considérerons ici qu'il y a un code <eval QCM> qui permet de faire cela.

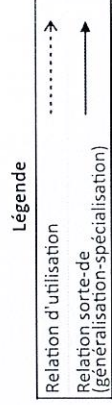
I. Première étape



II. Deuxième étape



III. Troisième étape

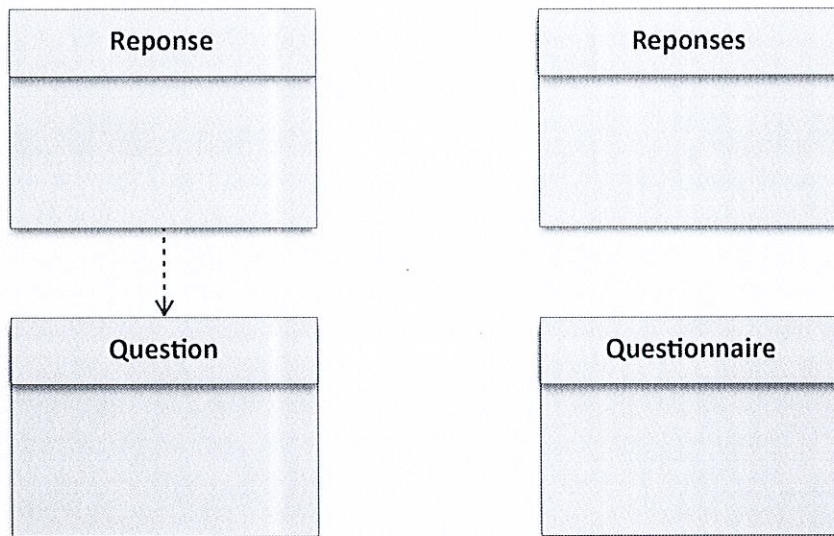


N° d'anonymat :

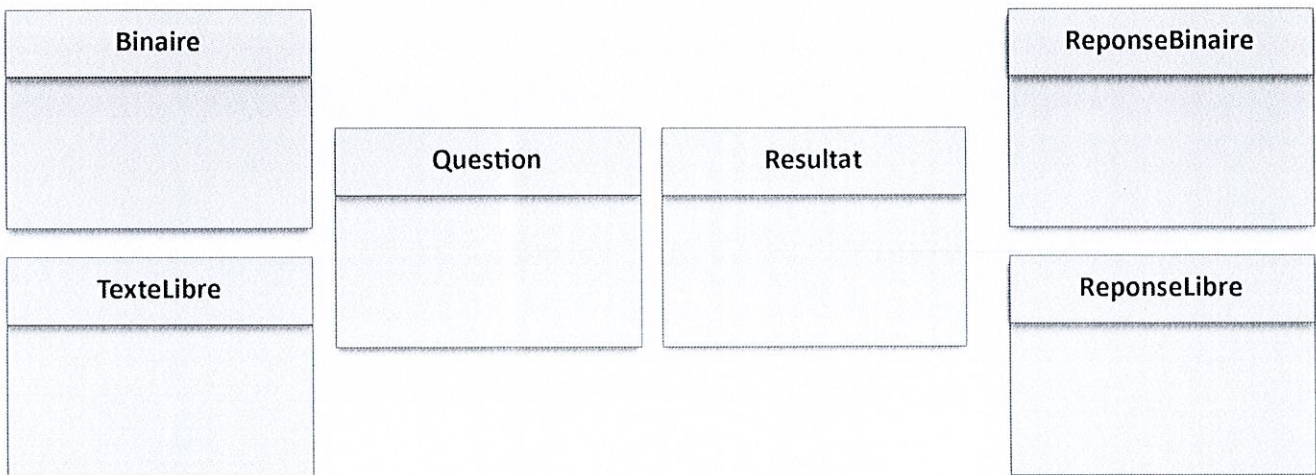
Annexe



I. Première étape

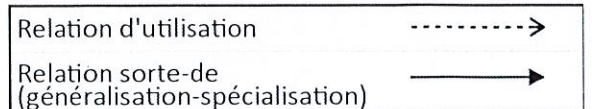


II. Deuxième étape



III. Troisième étape

Légende



N° d'anonymat :