

Examen (1h30, sans documents)

- En cas de doute sur le sujet, notez vos choix sur la copie. Aucune réponse ne sera donnée par les surveillants.
- Il est conseillé de lire complètement le sujet avant de commencer à y répondre !
- Ne pas mettre les commentaires de documentation (sauf si nécessaires à la compréhension).
- Barème indicatif (sur 20 points) :

Exercice	1	2	3
Points	8	6	6

Exercice 1 Répondre de manière concise et précise aux questions suivantes.

1.1 Pour écrire une exception `NouvelleException`, on hésite entre hériter de `RuntimeException`, de `Exception` ou de `Object`. Indiquer les conséquences à hériter de l'une ou l'autre de ces classes.

1.2 Comparer *interface* et *classe abstraite* : points communs et différences.

1.3 Définir et comparer *surcharge* et *redéfinition*.

1.4 On considère la signature suivante d'une méthode `m` générique :

```
<T> void m(Set<? super T> l1, Set<T> l2, Set<? extends T> l3);
```

Expliquer ce que signifient les mots-clés **extends** et **super** et donner un exemple d'appel de cette méthode qui illustre l'intérêt de les avoir utilisés ici.

1.5 Expliquer ce qu'est un gestionnaire de placement dans l'API Swing.

1.6 Dessiner un diagramme de classe UML faisant apparaître les éléments suivants de l'API Swing : `ActionListener`, `ActionEvent`, `Event`, `addActionListener`, `JButton` et `actionPerformed`.

1 Questions à réponses courtes

Exercice 2 : Questionnaire et questions à réponses courtes

On souhaite réaliser un questionnaire composé de plusieurs questions. On ne considèrera que des questions à réponse courte (QRC) mais on pourrait envisager d'autres types de questions comme par exemple des questions à choix multiple, des questions graduées, etc. Chaque question à réponse courte est caractérisée par le texte de la question (texte) et la réponse attendue (réponse).

On veut pouvoir itérer sur toutes les questions d'un questionnaire avec le `foreach` de Java comme dans l'exemple du listing 1.

2.1 Donner le diagramme de classe d'analyse qui fait apparaître `Question`, `QRC` et `Questionnaire` en utilisant les rubriques requêtes, commandes et constructeurs.

2.2 Écrire `Question` (en Java).

2.3 Écrire `Questionnaire` (en Java), en se limitant à ce qui est nécessaire à l'exécution du listing 1.

2.4 Écrire `QRC` (en Java).

Listing 1 – La classe ExempleQuestionnaire

```
1 public class ExempleQuestionnaire {
2     public static void main(String[] args) {
3         // Création d'un questionnaire
4         Questionnaire qTOB = new Questionnaire();
5         qTOB.ajouter(new QRC("Accessible_à_tous_", "public"));
6         qTOB.ajouter(new QRC("Non_modifiable_", "final"));
7         qTOB.ajouter(new QRC("2_*_3_+_4_", "10"));
8
9         // Afficher le texte des questions
10        for (Question question : qTOB) {
11            System.out.println(question.getText());
12        } } }
```

Exercice 3 : Amélioration des QRC

Nous souhaitons apporter deux améliorations à nos QRC.

1. Une QRC peut avoir plusieurs réponses justes. La réponse attendue n'est donc pas unique.
2. Pour une QRC, il peut y avoir des erreurs fréquentes. Dans ce cas, on souhaite pouvoir expliquer à l'utilisateur l'erreur commise. Par exemple, pour la 3ème question, on peut lui dire « attention, la multiplication est prioritaire sur l'addition » s'il répond « 14 ». Bien sûr, il y a autant de messages explicatifs que de mauvaises réponses classiques.

3.1 Pour la deuxième question du questionnaire du listing 1, les étudiants répondent souvent **static** au lieu de **final**. Quelle explication fournir, i.e. que signifie **static** en Java ?

3.2 Donner les attributs de QRC à ajouter ou modifier (en précisant leur type) pour prendre en compte ces évolutions.

3.3 On considère le questionnaire du listing 1 qui contient 3 questions. Le scénario suivant décrit une utilisation possible de ce questionnaire, par exemple au travers de Moodle :

1. L'étudiant lance le questionnaire.
2. Le questionnaire affiche le texte de la première question.
3. L'étudiant propose "public".
4. Le questionnaire informe l'étudiant que la réponse est correcte
5. Le questionnaire affiche le texte de la deuxième question.
6. L'étudiant propose "static".
7. Le questionnaire informe l'étudiant que la réponse est incorrecte et lui affiche l'indication associée à "static" .
8. L'étudiant abandonne et retourne réviser son cours.

Dessiner un diagramme de séquence compatible avec ce scénario.

3.4 D'après ce scénario, quelles méthodes faudrait-il ajouter sur Questionnaire et Question ?