Examen du cours "Programmation objet : concepts avancés"
Durée: 3 heures

Tout document autorisé.

Le soin apporté à la rédaction et à la présentation ainsi que la rigueur des réponses seront pris en compte dans la notation.

1 Scala : traits (60 minutes)

Question 1 *En moins de 5 lignes, expliquez à quels problèmes répondent les mécanismes de trait et de composition mixin de Scala ?* □

Question 2 *Voici un fragment de code écrit en* SCALA *:*

```
abstract class A {
  val default :Int
  def m (x :Int) :Int = default * x
}

trait B extends A {
  override def m (x :Int) :Int = super.m (x * 2)
}

trait C extends B {
  override def m (x :Int) :Int = super.m (x + 1)
}

trait D extends C {
  override def m (x :Int) :Int = super.m (x - 2)
}

trait E extends A {
  val default :Int = 1
}
```

Quel est le résultat de l'évaluation des expressions suivantes ? (Justifiez votre réponse.)

(i) `(new A { val default = 42 }).m (11)`

(ii) `(new A with D { val default = 42 }).m (2)`

(iii) `(new A with C with E).m (2)`

□

Question 3

```
trait Iterator[A] {
    def hasNext :Boolean
    def next :A
    def m[B >:A](that :Iterator[B]) :Iterator[B] = new Iterator[B] {
        def hasNext = Iterator.this.hasNext ∨ that.hasNext
        def next = if (Iterator.this.hasNext) Iterator.this.next else that.next
    }
}
```

(i) Quel nom "plus parlant" donneriez-vous à la méthode `m` *?*
(ii) Comment expliquez-vous l'annotation `[B >: A]` *?*
(iii) Quel avantage a l'utilisation d'un trait en SCALA *sur l'utilisation d'une interface en* JAVA *pour définir les itérateurs ?*
(iv) Pourquoi utilise-t-on des itérateurs ?

□

2 Java (30 minutes)

Voici un programme Java :

```
public class OneInteger {
  public int x;
}

public class TwoIntegers extends OneInteger {
  public int y;
}

public class OneIntegerOneFloat extends OneInteger {
  public float f;
}

public class Boom {
  public static void why_not (OneInteger[] a) {
      a[0] = new OneIntegerOneFloat ();
  }
  public static int why_not2 (TwoIntegers[] a) {
      return (a[0].y);
  }
  public static void main (String[] args) {
    TwoIntegers[] a = new TwoIntegers[1];
    why_not (a);
    System.out.println (why_not2 (a));
  }
}
```

Sans surprise, l'exécution de ce programme produit une erreur :

```
Exception in thread "main" java.lang.ArrayStoreException:
OneIntegerOneFloat
        at Boom.why_not(Boom.java:3)
        at Boom.main(Boom.java:10)
```

Cette erreur aurait pu être détectée par le compilateur. En effet, ce programme est accepté par tous les compilateurs de Java alors qu'il ne respecte pas les principes du sous-typage.

Question 4 *Expliquez en quoi les principes du sous-typage ne sont pas respectés dans ce programme Java.* □

Question 5 *Quel message d'erreur aurait dû produire le compilateur Java ?* □

Question 6 *D'après vous, quelle conséquence (désastreuse) cette erreur dans la conception de Java induit-elle sur les performances des programmes écrits dans ce langage ?* □

3 Modélisation objet (1h30)

On s'intéresse à la modélisation d'un compteur d'occurrences de mots dans un ensemble de documents. La description informelle du logiciel est formulée comme suit par un client :

> On souhaite un logiciel permettant à un ensemble d'utilisateurs de surveiller en temps réel le nombre d'occurrences d'un ensemble de mots qui apparaissent dans les données textuelles produites par un ensemble de sources de contenu.
>
> Les sources peuvent être des flux RSS, des documents PDF sur un système de fichier ou bien encore des documents hypertextes. Les utilisateurs peuvent rajouter ou supprimer de nouvelles sources à tout moment de l'exécution du logiciel.
>
> Les utilisateurs doivent pouvoir être prévenus par SMS, par email ou encore par message instantané. Ils doivent aussi formuler des critères d'alerte comme par exemple : « le nombre d'occurrences du mot "X" a augmenté de N% », « je n'ai pas reçu d'alerte dans les dernières N heures », ...
>
> Le logiciel doit être capable de monter en charge en se dupliquant sur de nouvelles machines signalées par un administrateur système.
>
> Un système de plugins doit permettre le rajout de nouveaux types de sources de contenu.

L'objectif de cet exercice est d'évaluer vos capacités d'analyse et de conception orientée objet. Dans la suite, sont attendues des réponses **argumentées**, **structurées** et **articulées logiquement**. Vous devez utiliser les procédés habituels de structuration d'un document : titre, sous-titre, énumération, figure, tableau, ... Une réponse non argumentée, même plausiblement juste, n'obtiendra aucun point.

L'utilisation de patrons de conception vus en cours, si elle est justifiée, est encouragée.

Question 7 *Donnez des diagrammes UML de cas d'utilisation résumant au mieux l'ensemble des besoins exprimés par le client. Est-ce que ces besoins exprimés sont tous raisonnables à votre avis ? Quels aspects du logiciel souhaiteriez-vous préciser pour en maximiser la faisabilité ? Justifiez vos réponses.* □

Question 8 *Énumérez et donnez une courte définition des concepts nécessaires à la modélisation de ce logiciel. On vise ici la complétude à un haut niveau d'abstraction, laissant de côté les cas particuliers et les détails. Vous ferez référence aux cas d'utilisation pour montrer que les objets qu'ils mettent en jeu sont des instances de ces concepts. Justifiez vos réponses.* □

Question 9 *Décrivez votre décomposition par objets de ce problème en donnant les relations entre les concepts énumérés dans la question précédente. Explicitez votre architecture ainsi que les dépendances entre les différents modules. Justifiez vos réponses.* □

Question 10 *Proposez un diagramme UML de classes servant de modèle logique à ce logiciel. En quoi votre modèle logique prend-il en compte les futures extensions potentielles du logiciel ? Justifiez vos réponses.* □