

Examen de Programmation

(deuxième session 2011)

Durée deux heures; tous les documents sont interdits

Le barème est donné à titre indicatif (total 33).

3 pages

1. Qu'entend-on par *erreur à la compilation*? Qu'entend-on par *erreur à l'exécution*? (1pt)
2. A quel moment une erreur à l'exécution est-elle détectée? A quel moment une erreur à la compilation est-elle détectée? En java, comment se manifeste cette détection (pour une erreur à la compilation, pour une erreur à l'exécution)? (1pt)
3. Donner un exemple (simple) de programme Java qui provoque une erreur à la compilation? Mêmes question pour une erreur à l'exécution. (1pt)
4. Le morceau de programme suivant provoque-t-il une erreur? Si oui est-ce une erreur à l'exécution ou à la compilation? (1pt)

```
int i; int j=1;
if (j>0) i=0; else i=1;
j= j/i;
```

Mêmes questions pour le programme : (1pt)

```
int i=0; int j=1;
j= j/i;
```

5. Toute *variable* d'un programme java a-t-elle un type unique? Si oui à quel moment (compilation ou exécution) et comment est-il déterminé? Mêmes questions pour une *expression*? (2 pts)
6. En java une variable déclarée comme étant d'un type A peut-elle référencer un objet d'un type autre que A? Si oui à quelle(s) condition(s)? (2pts)
7. Qu'est-ce que la *signature* d'une méthode? Qu'est-ce que la surcharge d'une méthode? (1pt)
8. On considère :

```
class A{
    int i=1;
    void f(){System.out.println("f de A:i="+i);}
}
class B extends A{
    int i=2;
    int j=3;
    void g(){System.out.println("g de B:j="+j);}
}
```

- (a) Soit le code suivant :

```
B b=new B(); A a=b;
System.out.println("a.i="+a.i);
System.out.println("b.i="+b.i);
```

Ce code provoque-t-il une erreur (si oui précisez s'il s'agit d'une erreur au moment de la compilation ou de l'exécution)? Si ce n'est pas le cas quel en est la sortie. On justifiera la réponse en précisant en particulier les types des variables et des objets référencés. (1pt)

(b) Mêmes questions pour le code suivant : (1pt)

```
A abis= new A();B bbis=abis;
System.out.println("abis.i="+abis.i);
System.out.println("bbis.i="+bbis.i);
```

(c) Mêmes questions pour le code suivant : (1pt)

```
A ater= new A();B bter=(B)ater;
System.out.println("ater.i="+ater.i);
System.out.println("bter.i="+bter.i);
```

(d) Mêmes questions pour le code suivant : (1pt)

```
B b=new B(); A a=b;
System.out.println("a.j="+a.j);
System.out.println("b.j="+b.j);
```

(e) Mêmes questions pour le code suivant : (1pt)

```
A a=new B();
a.g(); a.f();
```

(f) Mêmes questions pour le code suivant : (1pt)

```
B b=new B(); A a= b;
b.g();a.f();
```

(g) Après l'instruction `A a=new B();` est-il possible d'appeler la méthode `g` de la classe `B` à partir de `a`? Si oui comment et que s'affichera-t-il? (1pt)

(h) Après l'instruction `B b=new B();` est-il possible d'accéder à la variable `i` de la classe `A` à partir de `b`? Si oui comment? (1pt)

(i) On ajoute à la classe `B` la méthode :

```
void h(){System.out.println("h de B:i="+i);}
```

Le code suivant : `B b=new B(); b.h();` provoque-t-il une erreur à la compilation? à l'exécution? Si ce n'est pas le cas, qu'affichera-t-il? (1pt)

(j) Modifier la méthode `h` de la classe `B` de façon à ce que dans tous les cas elle affiche la valeur de la variable `i` de la classe `A`. (1pt)

(k) On ajoute à la classe `B` la méthode suivante :

```
void f(){g();}
```

Que provoque le code suivant `A a= new B(); a.f();` (erreur à la compilation, à l'exécution, affichage...)? (1pt)

Que provoque le code suivant `A a= new B(); ((A)a).f();` (erreur à la compilation, à l'exécution, affichage...)? (1pt)

(l) Est-il possible d'ajouter une méthode `f1` dans la classe `B` qui appelle `f` de `A`? Si oui comment? (1pt)

9. Dans cet exercice on s'intéresse à l'implémentation de *pires* (les piles sont définies par les opérations *empiler*, *dépiler* et *estVide*).

(a) Définir une interface `Pile` pour des piles de `Object` (1pt)

(b) Définir une classe `PileTab` qui implémente une `Pile` en utilisant un tableau de taille fixe pour stocker les données de la pile (1pt)

(c) On considère la classe :

```
static class Cellule{
    public Object d;
    public Cellule suivant;
    Cellule(Object o, Cellule s){d=o;suivant=s;}
    Cellule(Object o){this(o,null);}
    Cellule(){this(null,null);}
}
```

Définir une classe `PileCell` qui implémente une `Pile` en utilisant la classe `Cellule` pour stocker les données de la pile (2pts)

- (d) Quel(s) problème(s) concernant le comportement attendu des piles pose le code suivant : `Pile p= new PileTab(); Pile q = p;` Comment y remédier ? (2pts)
- (e) Au lieu de considérer des piles de `Object`, on veut réaliser des piles paramétrées par un type quelconque `T`. Décrire pour cela une implémentation utilisant un tableau de taille fixe et une implémentation utilisant la classe `Cellule`. Donner un exemple d'utilisation de ces implémentations pour le type `Integer` (2pts)
- (f) Si ces piles peuvent être utilisées dans des *threads* différentes, quel(s) problème(s) peuvent se poser ? Comment les résoudre ? (2pts)