

## Examen de Programmation Orientée Objets (première session 2011-2012)

Durée deux heures; tous les documents sont interdits  
Le barème est donné à titre indicatif (total 23).

2 pages

1. Qu'entend-on par *erreur à la compilation*? Qu'entend-on par *erreur à l'exécution*? (0,5pt)
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)? (0,5pt)
3. 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*? (1 pt)
4. 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)? (1 pt)
5. Qu'est-ce que la *signature* d'une méthode? Qu'est-ce que la surcharge d'une méthode? (1 pt)
6. On considère :

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

- (a) Soit le code suivant :

```
X b=new Y(); X 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 quelle 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)

```
Y b=new Y(); X a=b;
System.out.println("a.j="+a.j);
System.out.println("b.j="+b.j);
```

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

```
X ater= new X();Y bter=(Y)ater;
System.out.println("ater.i="+ater.i);
System.out.println("bter.i="+bter.i);
```

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

- (e) Après l'instruction `Y b=new Y();` est-il possible d'accéder à la variable `i` de la classe `X` à partir de `b`? Si oui comment? (1pt)
- (f) On ajoute à la classe `Y` la méthode :
- ```
void h(){System.out.println("h de B:i="+i);}
```
- Le code suivant : `Y b=new Y(); b.h();` provoque-t-il une erreur à la compilation? à l'exécution? Modifier la méthode `h` de la classe `Y` de façon à ce que dans tous les cas elle affiche la valeur de la variable `i` de la classe `X`. (1pt)
- (g) Est-il possible d'ajouter une méthode `f1` dans la classe `Y` qui appelle `f` de `X`? Si oui comment? (1pt)
7. On considère une fonction `f` des entiers dans les réels (la méthode `f` est définie pour des paramètres de type `int` et retourne un `double`), on veut faire certaines manipulations et calculs sur une telle fonction. Pour cela on va créer une classe abstraite `Fonc` contenant la méthode abstraite `f` et diverses méthodes correspondant aux calculs que l'on veut réaliser. Pour utiliser cette classe pour une fonction particulière il suffira ensuite de créer une extension non abstraite pour laquelle la méthode `f` sera redéfinie par un code correspondant à cette fonction particulière. (De cette façon, la méthode abstraite `f` joue le rôle de "paramètre" de la classe abstraite).
- (a) (3 points) Ecrire une classe abstraite `Fonc` contenant une méthode abstraite nommée `f` ayant comme argument un entier et retournant un `double`, une méthode `moyenne(int i, int j)`, une méthode `min(int i, int j)`, une méthode `max(int i, int j)` et une méthode `somme(int i, int j)` qui calculent respectivement la moyenne, le min, le max et la somme des valeurs de `f` pour toutes les valeurs entières comprises entre `i` et `j`.
- (b) (3 points) En créant des classes dérivées de `Fonc`, utiliser la méthode `somme` pour calculer la somme des  $m$  premiers termes pour les suites suivantes :
- $u_n = n$
  - $u_n = 2 * u_{n-1}$  ( $u_0 = 1$ )
  - $u_{n+2} = u_n + u_{n+1}$  ( $u_0 = 0$  et  $u_1 = 1$ )
- Est-il possible pour cela de ne pas créer explicitement une nouvelle classe? Si oui comment?
- (c) (2 points) Définir une classe `Tab` (contenant un tableau `t` de `double`) extension de `Fonc` qui, à partir de ce tableau calcule la moyenne, le min, le max et la somme des éléments entre l'indice `i` et l'indice `j` contenus dans le tableau `t` en utilisant les méthodes `moyenne`, `min`, `max` et `somme` définies dans `Fonc` et en définissant de façon appropriée la méthode abstraite `f` de `Fonc`. On ajoutera aussi des constructeurs à cette classe pour créer et initialiser ce tableau `t` de taille `n`.
- (d) (2 points) Utiliser `Tab` pour calculer la somme des  $k$  premiers termes de la suite  $u_{n+2} = u_n + u_{n+1}$  (avec  $u_0 = 0$  et  $u_1 = 1$ ). Le calcul devra être linéaire (en fonction de  $n$ ). Donner une solution ne créant pas explicitement de nouvelles classes.
- (e) (2 points) Etant donné une fonction `f`, au lieu de calculer la somme de `i` à `j` directement à partir d'une extension de la classe `Fonc` comme dans la question (a), on veut maintenant utiliser la classe `Tab` : pour calculer  $\sum_{k=i}^{k=j} f(k)$  on remplit un tableau de taille `t` contenant les valeurs `f(i)`, `f(i+1)`, ..., `f(j)` et on calcule la somme telle que définie par `Fonc`. Définir une classe abstraite `TabFonc` ayant `f` comme méthode abstraite et une méthode `somme(int i, int j)` qui calcule la somme  $\sum_{k=i}^{k=j} f(k)$  en remplissant un tableau utilisé par la classe `Tab`.