Université Denis Diderot Paris 7
UFR d'informatique
L3 Licence d'informatique

Examen de P00 (deuxième session)

(juin 2009)

durée deux heures ; les documents ne sont pas autorisés

2 pages

Le barème est donné à titre indicatif. (total sur 23)

1. *(3 points)* On considère le programme suivant :

```
class F {
   void f(){System.out.print("F.f() ");this.g();}
   void g(){System.out.println("F.g() ");}
}
class Fbis extends F{
   void f(){System.out.print("Fbis.f() ");this.g();}
   void g(){System.out.println("Fbis.g() ");super.f();}
}
```

   Le code suivant : `F f=new Fbis() ; ((F)f).f() ;`
   n'est pas correct, que provoque-t-il et pourquoi.

2. On considère les classes :

```
class A{public int i; public A(){i=0;}; public String toString(){return "i="+i;}}
class B extends A {public int i; public B(){i=10;}}
```

   (a) *(1 point)*

      i. Que produira le code suivant (erreur à la compilation, à l'exécution, et s'il n'y a pas d'erreur quelle en est la sortie) On justifiera la réponse :

```
A[] ta = new B[10];
for (int i = 0; i < ta.length; i++) {
    ta[i] = new A();
    ta[i].i = i;
}
System.out.println(ta[1]);
```

      ii. *(1 point)* Même question pour :

```
A[] tab = new B[10];
for (int i = 0; i < tab.length; i++) {
    tab[i] = new B();
    tab[i].i = -i;
}
System.out.println(tab[1]);
```

      iii. *(1 point)* Même question pour :

```
B[] tb=new B[10];
for (int i = 0; i < tb.length; i++) {
    tb[i]=new B();
    tb[i].i = -i;
}
A[] tac=tb;
System.out.println(tac[1]);
```

iv. *(3 point)* Redéfinir la méthode `toString` pour `B` de façon à ce que cette méthode retourne une chaîne de caractères donnant les valeurs de *toutes données membres d'un objet de B. Que se passe-t-il pour les trois questions précédentes ?*

v. (3 points) *On rappelle qu'en Java si `Y` est une classe dérivée de `X` alors `Y[]` est aussi dérivée de `X[]`. Ainsi si `tx` est déclarée comme un tableau de `X`, `tx` peut référencer un tableau de `Y`. Quel problème pose alors le code :*

```
Y[] y=new Y[1];
X[] x=y;
x[0]=new X();
System.out.println(y[1]);
```

S'il y a une erreur, celle-ci peut-elle être détectée à la compilation ?

Si au lieu de considérer des tableaux on considère des `List<X>` et des `List<Y>` a-t-on le même comportement (présence ou absence d'erreur) ?

(b) Considérons une suite u_n ($n \in N$) de réels ; on suppose que la valeur du terme u_n est donnée par la valeur retournée par $u(n)$ où u est donné par une méthode `double u(int n)`.

On veut dans la suite faire certaines manipulations sur les suites comme, par exemple, tester la croissance, la somme, la valeur moyenne, ... entre les termes i et k. Pour cela on définira une classe (abstraite) `Suite` dont une des méthodes sera la méthode `double u(int n)` qui sera abstraite. Pour utiliser les méthodes définies dans `Suite` il suffit alors de créer une classe dérivée de `Suite` dans laquelle u est redéfinie pour le u voulu.

i. *(3 points)* Ecrire une classe abstraite `Suite` contenant une méthode abstraite nommée `u` 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)`, une méthode `somme(int i, int j)` qui calculent respectivement la moyenne, le min, le max et la somme des valeurs de de u entre i et j.

ii. *(2 points)* En créant des classes dérivées de `Suite`, utiliser la méthode `somme` pour calculer la somme des m premiers termes pour les suites suivantes : $u_n = n$, $u_n = 2 * u_{n-1}$ ($u_0 = 1$) et $u_{n+2} = u_n + u_{n+1}$ ($u_0 = 0$ et $u_1 = 1$).
Est-il possible pour cela de ne pas créer explicitement une nouvelle classe ? Si oui comment ?

iii. *(2 points)* Définir une classe `Tab` (contenant un tableau de `double`) extension de `Suite` qui calcule la somme des éléments contenus dans le tableau en utilisant la méthode `somme` définie dans `Suite`. On ajoutera aussi des constructeurs à cette classe pour créer et initialiser un tableau `t` de taille n.

iv. *(2 points)* En considérant les suites de la forme $u_{n+2} = f(u_n, u_{n+1})$ (avec $u_0 = x_0$ et $u_1 = x_1$), créer une classe `SuiteTab` extension de `Suite` qui utilise un objet de la classe `Tab` pour calculer u_n : pour calculer u_n on remplira le tableau des données de `Tab` avec les valeurs successives de $u_0, \cdots, u_{n-1}$. Cette classe contiendra comme méthode abstraite une méthode `f`.

v. *(2points)* Utiliser `SuiteTab` pour calculer la somme des k premiers termes de la suite $u_{n+2} = u_n + u_{n+1}$ (avec $u_0 = 0$ et $u_1 = 1$).
Donner une solution ne créant pas explicitement de nouvelles classes.