Année 2018-2019

Feuille 3 - Héritage

1 Héritage

Considérez le programme suivant :

```
1  class A{
2          private int x;
3          public A(){ this(0); }
4          public A(int x){ this.x=x; }
5          public int getX() { return x; }
6          public void inc() { x++; }
7          public String toString() { return "A:"+x; }
8          public void f() { System.out.println("A:f:"+this); }
9
10         public static void main(String [] args){
11                 A aa = new A(1);
12                 A ab = new B();
13                 A ac = new C();
14                 B bb = new B(2,3);
15                 C cc = new C();
16                 aa.f();
17                 ac.f();
18                 bb.f();
19                 cc.f();
20                 aa.inc(); ac.inc(); bb.inc(); cc.inc();
21                 System.out.println(aa);
22                 System.out.println(ac);
23                 System.out.println(bb);
24                 System.out.println(cc);
25         }
26 }
27
28 class B extends A{
29         int y;
30         public B(int x, int y) { super(x); this.y=y; }
31         private void f() { System.out.println("B:f:"+this); }
32         public void g() { this.x++; }
33         public String toString() { return "B:"+getX()+",␣"+y; }
34 }
35
36 class C extends B{
```

```
37          public C() { super(0,0); }
38          public int x=5;
39          public void f() { System.out.println("C:f:"+this); }
40          public void g() { this.x++; }
41          public String toString() { return "C:"+getX()+" , "+y+" , "+x; }
42  }
```

1. Trouver et corriger toutes les erreurs de compilation.
2. Une fois corrigé, qu'affiche ce programme ?

2 Surcharge

Considérez le programme suivant :

```
1   class A{
2           int x=5;
3           public String toString() { return "A:"+x; }
4           public void f(B a) { System.out.println("A:f:"+a); }
5           public static void g(A a) { System.out.println("A:g:"+a); }
6   }
7
8   class B extends A{
9           public void f(A a) { System.out.println("B:f:"+a); }
10          public static void g(A a) { System.out.println("B:g:"+a); }
11          public void f(B b) { System.out.println("B:fB:"+b); }
12          public static void g(B b) { System.out.println("B:gB:"+b); }
13          public String toString() { return "B:"+x++; }
14          public static void main(String [] args){
15                  A aa = new A();
16                  A ab = new B();
17                  B bb = new B();
18                  g(aa);
19                  g(ab);
20                  A.g(bb);
21                  aa.f(ab);
22                  ab.f(bb);
23                  ab.f(ab);
24                  bb.f(aa);
25                  bb.f(bb);
26          }
27  }
```

1. Trouver et corriger toutes les erreurs de compilation.
2. Une fois corrigé, qu'affiche ce programme ?

3 La classe myNumber

Pour cet exercice on va écrire une classe abstraite `myNumber`. Le but de cette classe est de nous donner une manière unifiée de traiter des objets des classes qui représentent des nombres, comme la classe `Complex` et la classe `Rational`, qu'on a déjà vues. Donner la signature de la class `myNumber` avec

1. Deux méthodes `abstract` : la méthode `square`, qui retourn le carré de l'objet, et la méthode `add`, qui prend un autre objet de la class `myNumber` et retourn la somme avec l'objet actuel.
2. Une méthode static `squareAll` qui, étant donné un tableau de `myNumber`, calcule le carré de chaque élément et affiche le resultat (avec `System.out.println`).
3. Une méthode static `addAll` qui, étant donné un tableua de `myNumber`, calcule la somme de tous les éléments et le retourne.

Pour tester la classe, il faut programmer des sous-classes. Donner une implémentation basique de deux classes, `Complex` et `Rational` qui étendent `myNumber`.

— Donner une redéfinition des méthodes abstraites de `myNumber`.

— Donner une redéfinition de `toString`

Enfin, écrire un programme simple pour tester ces classes. Votre programme peut construire un tableau de `myNumber` et calculer le carré de chaque élément, puis la somme.

1. Peut-on mettre à la fois des objets `Rational` et `Complex` dans le tableau ?
2. Qu'est-ce qu'il va se passer si on tente de calculer la somme ?