Université Paris Cité

# Compléments en Programmation Orientée Objet TP nº 3

Indication : avec le cours sous les yeux les trois premiers exercices doivent être fait rapidement.

# 1 Les bases du polymorphisme

## Exercice 1 : Définir le sous-typage entre types objet

On suppose déjà définies :

```
class A {}
class B {}
interface I {}
interface J {}
```

Voici une liste de déclarations :

```
class C extends I {}
interface K extends B {}
class C implements J {}
interface K implements B {}
class C extends A implements I {}
interface K extends I, J {}
class C extends A, B {}
class C implements I, J {}
```

Lesquelles sont correctes?

#### Exercice 2: Transtypage

```
System.out.println((byte) 257);
                                                                    System.out.println((char) 98);
                                                         14
  class A { }
                                                         15
                                                                    System.out.println((double) 98);
                                                                    System.out.println((char) 98.12);
                                                         16
  class B extends A { }
                                                         17
                                                                    System.out.println((long) 98.12);
                                                                    System.out.println((boolean) 98.);
                                                         18
6 class C extends A { }
                                                                    System.out.println((B) new A());
                                                         19
                                                         20
                                                                    System.out.println((C) new B());
8 public class Tests {
                                                         21
                                                                    System.out.println((A) new C());
      public static void main(String[] args) {
9
                                                         22
          System.out.println((int)true);
                                                         23 }
          System.out.println((int) 'a');
                                                         24
          System.out.println((byte) 'a');
```

Dans la méthode main() ci-dessus,

- 1. Quelles lignes provoquent une erreur de compilation?
- 2. Après avoir supprimé ces dernières, quelles lignes provoquent une exception à l'exécution?
- 3. Après les avoir enlevées, elles aussi, quels affichages provoquent les lignes restantes?

#### Exercice 3: Surcharge

```
1 class A {};
2 class B extends A {};
3 class C extends B {};
4 class D extends B {};
6 public class Dad {
      public static void f(A a, A aa) { System.out.println("Dad : A : A"); }
      public static void f(A a, B b) { System.out.println("Dad : A : B"); }
9 }
10 public class Son extends Dad {
      public static void f(A a, A aa) { System.out.println("Son : A : A"); }
11
      public static void f(C c, A a) { System.out.println("Son : C : A"); }
12
13
      public static void main(String[] args) {
14
          f(new B(), new A());
          f(new D(), new A());
16
17
           f(new B(), new D());
          f(new A(), new C());
18
19
      }
20 }
```

Dans la méthode main() ci-dessus,

- 1. Quels affichages provoquent les lignes 15 à 18?
- 2. Que se passe-t-il si on appelle f(new C()), new C())? f(new C(), new B())?
- 3. Dans la classe Son comment être sûr d'appeler les méthodes f de la classe Dad? Quels types de paramètres permettent d'appeler la fonction f avec signature (A,A)?

# 2 Les dessous du transtypage

### Exercice 4: Transtypages primitifs

Voici un programme (TranstypagesPrimitifs.java sur Moodle):

- 1. Compilez et exécutez ce programme (assurez-vous de comprendre la notation 9.2E11f).
- 2. Nous allons regarder superficiellement le code-octet produit : dans un terminal, allez dans le répertoire où se trouve TranstypagesPrimitifs.class et tapez la commande "javap -c -v TranstypagesPrimitifs". Le code-octet apparaît ainsi sous une forme désassemblée quasi lisible. Nous nous intéresserons en particulier au début de la partie Code :, qui correspond à la déclaration et l'initialisation de nos trois variables. On peut repérer l'appel à l'instruction suivante, println, par l'instruction getstatic dans le code-octet.
  - Il n'y a donc que 6 ou 7 lignes à regarder. Constatez que certaines variables sont initialisées par une séquence d'instructions comme : bipush 42; istore\_2, alors que d'autres ont la séquence ldc suivie de istore ou fstore (le i ou le f désigne clairement un type)
- 3. Nous allons nous intéresser à la façon dont sont fait les transtypages. Ajoutez une ligne avant l'instruction d'affichage : vint=vshort; et interpréter les opérations load, store qui apparaissent.

Avec les 3 variables présentes il y a théoriquement 6 transtypages, certains qu'il faut rendre explicites. Essayez les tous et complétez le tableau ci-dessous avec vos remarques.

Le tableau :	=	vint	vshort	vfloat
	vint	XXX		
	vshort		XXX	
	vfloat			XXX

Relevez, notamment, dans chaque cas:

- si ça compile directement, s'il ajouter un *cast* explicite, etc.;
- la nature des instructions ajoutées dans le code-octet (notez que les instructions de la forme f2i expriment un changement de type);
- l'affichage produit après conversion.

### Exercice 5: Transtypages d'objets (sur machine)

Même exercice que le précédent mais sur le programme suivant :

```
public class TranstypagesObjets {
   public static void main(String[] args) {
        Object vObject = new Object();
        Integer vInteger = 42;
        String vString = "coucou";
        System.out.println("vObject = " + vObject + ", vInteger = " + vInteger + ", vString = " + vString);
    }
}
```

Différence, vous ne verrez plus l'ajout de l'instruction **u2t** mais parfois celle de **checkcast**. Dans quels cas?

Dans certains cas vous aurez eu besoin, pour compiler, d'un *cast* explicite. Lesquels? Est-ce les-mêmes que dans la question précédente?

Dans certains cas, le programme quittera sur ClassCastException, lesquels?

#### Exercice 6: Transtypages mixtes (sur machine)

Faites le même travail sur le programme suivant. Remarquez les instructions, insérées par le compilateur, qui correspondent au boxing et à la vérification de types.