

Transtypage et héritage

Règles de transtypage ou de Casting (UpCasting + DownCasting)

- Dans une instruction d'affectation ($A = B;$), le compilateur vérifie que : le type de la variable A est le même que celui de B ou un type parent du type de la variable B.

"UpCasting" implicite/explicite : permet de convertir le type d'une référence vers un type parent.

UpCasting Exemple :

```
Object obj1 = new String(); // UpCasting implicite
```

```
Object obj2 = (Object) new String(); // UpCasting explicite
```

UpCasting

La conversion (implicite ou explicite) vers un type parent est toujours acceptée par le compilateur et elle ne posera aucun problème à l'exécution du programme.

Une référence d'un type parent peut, sans risque, lire/modifier les attributs ou invoquer les méthodes dont elle a accès, indépendamment si l'instance référencée est du même type ou un sous-type du type de la référence.

DownCasting

Permet de convertir le type d'une référence vers un sous type.

DownCasting implicite Exemple

```
String str1 = new Object(); // DownCasting implicite  
(erreur de compilation : Type mismatch: cannot convert from Object to String)
```

La conversion implicite vers un sous type est toujours refusée par le compilateur.

DownCasting explicite Exemple

```
String str2 = (String) new Object(); //DownCasting explicite
```

(erreur à l'exécution : java.lang.ClassCastException: java.lang.Object cannot be cast to java.lang.String)

La conversion explicite vers un sous type est toujours acceptée par le compilateur. Cependant à l'exécution du programme, la JVM va vérifier si le type de l'instance (Object) est le même ou un sous-type du type qu'on a spécifiée pour la conversion (String) : si ce n'est pas le cas, la JVM déclenchera une exception.

DownCasting explicite

Exemple

String str2 = (String) "chaîne"; //DownCasting explicite qui fonctionne

En voilà un exemple :

Considérons les classes suivantes :	Mettez une croix dans la case correspondant à la bonne réponse, si l'instruction proposée est insérée à l'endroit proposé		
		Erreur de compilation	Pas d'erreur
<pre>class A { void fairececi() { }; } class B extends A { void fairececi() { }} class C { void fairececi() { }} class Principale { public static void main(String[] args) { A a = new A(); B b = new B(); /*insérer ici*/ } }</pre>	A x = a;		X
	A x = b;		X
	B x = a;	X	
	B x = b;		X
	A x = (A) b;		X