

TD02: Héritage

Cette séance a pour but:

- De comprendre ce qui se passe lors de l'héritage
- de comprendre l'ordre d'accès des constructeurs

Exercices

1. Héritage et appel des constructeurs

On considère les classes A, B et Test suivantes:

```
1 public class A {
2     public A() {
3         System.out.println("> Constructeur de A()");
4         afficher();
5     }
6
7     public void afficher() {
8         System.out.println("> Affichage de A");
9     }
10 }
```

```
1 public class B extends A {
2     private int v;
3
4     public B() {
5         System.out.println("> Constructeur de B()");
6         v = 5;
7         afficher();
8     }
9
10    public void afficher() {
11        System.out.println("> v = " + v);
12    }
13 }
```

```
1 public class Test {
2     public static void main(String[] args) {
3         System.out.println("A objetAA = new A();");
4         A objetAA = new A();
5         System.out.println("A objetAB = new B()");
6         A objetAB = new B();
7         System.out.println("B objetBA = new A()");
8         B objetBA = new A();
9     }
10 }
```

L'objectif est de deviner, par l'analyse le résultat du code de la méthode main.

Question 1.1

Que produit la ligne 4 de la classe Test (`A objetAA = new A();`) et pourquoi ?

- ☐ une erreur à la compilation
- ☐ une erreur à l'exécution
- ☒ L'affichage > Constructeur de A() > Affichage de A
- ☐ L'affichage > Constructeur de A() > Affichage de A > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de A() > Affichage de A > v = 0 > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de A() > v = 0 > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de B() > v = 5

Résultat: Votre réponse est juste.

Lors de la ligne 4, on ne fait appel qu'à la classe A, que ce soit pour le constructeur ou la méthode afficher. La classe B pourrait ne pas exister, cela ne changerait rien à ce résultat.

Question 1.2

Que produit la ligne 6 de la classe Test (`A objetAB = new B();`) et pourquoi ?

- ☐ une erreur à la compilation
- ☐ une erreur à l'exécution
- ☐ L'affichage > Constructeur de A() > Affichage de A
- ☐ L'affichage > Constructeur de A() > Affichage de A > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de A() > Affichage de A > v = 0 > Constructeur de B() > v = 5
- ☒ L'affichage > Constructeur de A() > v = 0 > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de B() > v = 5

Résultat: Votre réponse est juste.

Ici, le résultat est un peu plus inattendu. La référence `objetA` est une référence vers A, mais l'instance créée est de type B.

Le constructeur de B appelle silencieusement le constructeur par défaut de A.

Dans le constructeur de A, il y a un appel à la méthode `affiche`. Or, comme l'objet qui est en train d'être construit est de type B, c'est l'implémentation la plus spécifique qui est appelée (point clé no 4 de l'héritage).

A ce moment là de l'appel de la méthode `affiche`, l'opérateur `new` a déjà été appelé, et les attributs ont été initialisés à leurs valeurs par défaut. C'est pourquoi on a l'affichage `v = 0`.

Ensuite, on revient au constructeur de B et l'on affiche les 2 lignes suivantes.

Question 1.3

Que produit la ligne 8 de la classe `Test` (`B objetBA = new A();`) et pourquoi ?

- ☒ une erreur à la compilation
- ☐ une erreur à l'exécution
- ☐ L'affichage > Constructeur de A() > Affichage de A
- ☐ L'affichage > Constructeur de A() > Affichage de A > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de A() > Affichage de A > v = 0 > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de A() > v = 0 > Constructeur de B() > v = 5
- ☐ L'affichage > Constructeur de B() > v = 5

Résultat: Votre réponse est juste.

Le polymorphisme en Java implique qu'un objet de type B est également de type A. Par contre, un objet de type A n'est **pas** de type B.

Question 1.4

Testez ce code sur machine (en commentant la dernière ligne de la méthode `main`).

Question 1.5

Dessinez le diagramme APO de ce qui se passe dans la méthode `main`.

Résultat: Votre réponse est juste.

ous pouvez vérifier votre réponse grâce à [ce document](#).

[Fermer]