

# Masquage de membres de classes, surcharge/redéfinition de méthodes, appel de méthodes

## Exercice 1 - Masquage de champs

- 1 Le code suivant est-il valide? Si oui, quel est le type de la valeur affichée ?
- 2 A combien de `x` a t-on accès dans la classe `B` ? Comment modifier le code de la méthode `printX` pour qu'elle affiche l'autre valeur de `x` (donner si possible plusieurs manières) ?

```
class A {
    int x;
}

class B extends A {
    String x;

    public void printX() {
        System.out.println(x);
    }
}
```

On rajoute la classe `Test`.

- A combien de `x` a t-on accès dans la classe `Test` ? Qu'affiche la méthode `main` suivante ?

```
class Test {
    public static void main(String[] args) {
        System.out.println(new B().x);
    }
}
```

- 1 Quel est le type de la valeur affichée ?
- 2 Comment modifier `main()` pour qu'elle affiche l'autre valeur de `x` (donner si possible plusieurs manières) ?

## Exercice 2 - Masquage de champ statique

- 1 Quel est le type de la valeur affichée ?
- 2 A combien de `x` a t-on accès dans la classe `B` ? Comment modifier le code de la méthode `printX` pour qu'elle affiche l'autre valeur de `x` (donner si possible plusieurs manières) ?

```
class A {
    static int x;
}

class B extends A {
    static String x;

    public static void printX() {
        System.out.println(x);
    }
}
```

### Exercice 3 - Redéfinition

```
class A {
    protected int x() {
        return 3;
    }
}
class B extends A {
    public int x() {
        return 4;
    }
    public void printX() {
        System.out.println(x());
    }
}
class Test {
    public static void main(String[] args) {
        B b = new B();
        System.out.println(b.x());
    }
}
```

- 1 Quelle est la valeur affichée par `printX()` ?
- 2 Combien de méthodes `x()` sont accessibles dans `B` (i.e. à combien de méthodes `x` une méthode de `B` a-t-elle accès ?) ? S'il y en a plusieurs, donner pour chacune un moyen d'accès ?
- 3 Mêmes questions, mais en se plaçant dans `Test`.
- 4 Donnez le nom du mécanisme utilisé dans `B` à propos de la méthode `x()`.

### Exercice 4 - Surcharge, redéfinition, appel de méthode

Dans les exemples de classes suivants vous commencerez par dire où se trouvent les erreurs de compilation. Vous les expliquerez et retirerez les méthodes en provoquant. Vous indiquerez ensuite où sont les surcharges et les redéfinitions. Pour les appels de méthodes vous indiquerez également les méthodes appelées.

```
class A {
    public void a() { ... }
    void b() { ... }
    void c() { ... }
    void c(A a) { ... }
    void c(B b) { ... }
    static void d() { ... }
    void e() { ... }
    int f() { ... }
    void g() throws RuntimeException { ... }
    void h() throws IOException { ... }
}
class B extends A {
    public void a() { ... }
    protected void b() { ... }
    public void c(A a) { ... }
    static void d() { ... }
    static void d(A a) { ... }
    char f() { ... }
    void g() throws ArrayIndexOutOfBoundsException { ... }
    void h() { ... }
}
class C {
    public static void main(String[] args) {
```

```
A aa = new A();
A ab = new B();
B bb = new B();
aa.a(); ab.a(); bb.e(); aa.d(); bb.d();
aa.c(aa); aa.c(ab); aa.c(bb); ab.c(aa); ab.c(ab);
ab.c((B)ab);
        ab.c(bb); ab.c((A)bb);
((B)ab).c(aa);
bb.c(aa); bb.c(ab); bb.c(bb);
((A)bb).c(aa); bb.c(ab); bb.c(bb);
}
}
```