Université Abderrahmane Mira de Bejaia
Faculté des Sciences Exactes
Département d'informatique
Année universitaire : 2022/2023

**2ème Année Licence Informatique RN (S4)** Module : **Programmation Orientée Objet**

# TD2 : Le principe d'héritage

Veuillez consulter les documents sur ma class-room et suivre ma chaine YouTube :

https://www.youtube.com/watch?v=jW8M3nFb-es&list=PLQ4KaU9Gz0_G3jJTsq6ddAU0Bp8-gvSzA

**Les questions/ exercices avec * sont à faire individuellement**

## *Exercice 1

Soit le programme suivant, une classe Voiture dérivant d'une autre classe Véhicule (les notations ne correspondent pas à celles d'un véritable langage de programmation).

```
class Vehicule {
rouler() {
}
seGarer() {
}
}
class Voiture extends Vehicule {
seGarer() {
}
klaxonner() {
}
rouler () {
}
}
```

Soit une106, une instance de Voiture et monVehicule, une instance de Vehicule.

- Peux-t-on faire une106.rouler() ? Expliquez ?
- Si l'on fait une106.seGarer(), est-ce la méthode de la classe Vehicule ou de la classe Voiture qui est appelée ?
- Peux-t-on faire monVehicule.klaxonner() ? Pourquoi ?

## Exercice 2

Soient des personnes caractérisées par leurs nom, prénom, numéro de téléphone et le nombre d'enfants ; des étudiants qui sont des personnes qui suivent un cursus ; des salariés qui sont des personnes qui reçoivent un salaire.

- Donner la hiérarchie des classes (sous forme de schéma).
- Quels sont les membres (variables et méthodes) hérités ? (Utiliser un tableau pour expliquer).
- On vous donne ci-dessous les classes Personne et Etudiant. Ecrire la classe Salarie qui hérite de Personne et qui possède un constructeur ayant comme paramètre le nom et le salaire.

```
1 public class Personne {
2 protected String nom;
3 protected String prenom;
4 protected String nuTel;
5 protected int nbEnfants;
6 public Personne(String n,String p,String t) {
7 nom=n ; prenom=p ; nuTel=t ; nbEnfants=0;
```

```
8 }
9 public Personne (String n) {nom=n;}

10 public String getNom( ) { return nom; }
11 public void ajouterEnfant( ) { nbEnfants++; }
12 }
13 public class Etudiant extends Personne {
15 private String cursus;
16 public Etudiant(String n,String p,String t,String c) {
17 super (n,p,t) ; cursus=c ;
18 }
19 public boolean estEnL2( ) { return cursus.equals("L2") ; }
20 }
```

- Ecrire une méthode prime() qui retourne le montant de la prime accordée pour les enfants, à savoir 5% du salaire par enfant. Dans quelle classe mettre cette méthode ?
- Trouver les erreurs dans la méthode main ci-dessous :

```
1 public class TestPersonne {
2 public static void main (String [ ] args ) {
3 Personne p = new Personne ("Albert" ) ;
4 p.ajouterEnfant( ) ; p.ajouterEnfant( ) ;
5 double primP = p.prime( ) ;
6 System.out.println(p.getNom( ) + p.getSalaire( ) ) ;
7 Etudiant e = new Etudiant("Mahmoud" , "L2" ) ;
8 e.ajouterEnfant( ) ;
9 double primE = e.prime( ) ;
10 System.out.println(e.getNom( )) ;
11 boolean enL2 = e.estEnL2( ) ;
12 Salarie s = new Salarie("Lilia") ;
13 System.out.println(s.getNom( )) ;
14 }
15 }
```

***Exercice 3**

Soit une hiérarchie de classes du domaine de la biodiversité

- Dessiner l'arbre d'héritage et dire ce qu'affiche le programme suivant :

```
1 public class Plante {
2 public String toString( ) { return "Je suis une Plante" ; }
3 }
4 public class Arbre extends Plante { }
5 public class Fleur extends Plante {
6 public String toString( ) { return "Je suis une Fleur" ; }
7 }
8 public class Marguerite extends Fleur {
9 public String toString( ) { return "Je suis une Marguerite" ; }
10 }
public class Chene extends Arbre { }
12 public class Rose extends Fleur { }
13 public class MainPlante {
14 public static void main (String [ ] args ) {
15 Plante p = new Plante( ) ; System.out.println(p) ;
16 Arbre a = new Arbre( ) ; System.out.println(a ) ;
17 Fleur f = new Fleur( ) ; System.out.println(f) ;
18 Marguerite m = new Marguerite( ); System.out.println(m) ;
```

```
19 Chene c = new Chene( ) ; System.out.println(c) ;
20 Rose r = new Rose( ) ; System.out.println(r) ;

21 }
22 }
```

- En tirer des conclusions sur l'héritage et la redéfinition de méthode.
- Qu'affiche le code suivant :

```
1 Plante p2 = new Arbre( ) ;
2 System.out.println(p2) ;
3 Plante p3 = new Fleur( ) ;
4 System.out.println(p3) ;
5 Plante p4 = new Marguerite( ) ;
6 System.out.println(p4) ;
7 Plante p5 = new Rose( ) ;
8 System.out.println(p5) ;
9 Plante p6 = new Chene( ) ;
10 System.out.println(p6) ;
```

**Exercice 4**

Les exemples suivants sont-ils corrects ? Justifier votre réponse

**Exemple1:**

```
class A {
public void f () {
System.out.println("Bonjour.") ;
}
class B extends A {
private void f() {
System.out.println("Bonjour les amis. ") ;
}
}
```

**Exemple2:**

```
class A {
public int f(int a) {
return a++;
}
class B extends A {
public boolean f(int a) {
return (a==0);
}
}
```

**Exemple3:**

```
class A {
public int f( int a) {
return a++;
}
class B extends A {
public int f( int a, int b) {
return a+b;
}
class test {
B obj = new B();
int x = obj.f(3) ;
```

```
int y = obj.f(3,3) ;
}
}
```

**Exemple4:**

```
class A {
public int f( int a) {
return a++;
}
}
class B extends A {
public int f( int a, int b) {
return a+b;
}
class test {
A obj = new B();
int x = obj.f(3) ;
int y = obj.f(3,3) ;
}
}
```

**Exercice 5**

Donnez le résultat d'exécution du programme suivant, détectez l'erreur sinon et expliquez ?

```
class Shape {
private String color;
public Shape(String color) {
System.out.print("Shape");
this.color = color;
}
}
class Rectangle extends Shape {
public Rectangle() {
System.out.print("Rectangle");
}
}
public class TestConstructor {
public static void main(String[] args) {
new Rectangle();
}
}
```

**Exercice 6**

Soit le programme suivant, Indiquer la/les réponses correcte(s) si l'instruction proposée est insérée en ligne 14. Si l'instruction est correcte, cliquer sur ok et afficher le résultat

```
1 class Geek {
2 public void say() {
3 System.out.println("hello!");
4 }
5 }
6 class Sheldon extends Geek {
7 public void say() {
8 System.out.println("bazinga!");
9 }
10 public static void main(String[] args) {
11 Geek g = new Sheldon();
12 Sheldon s = new Sheldon();
```

```
13 Geek g2 = new Geek();
14 /* à remplacer */
15 }
16 }
```

| Instruction | Erreur de compilation | Ok affichage |
|---|---|---|
| g.say(); | ☐ | ☐ |
| g2.say(); | ☐ | ☐ |
| s.say(); | ☐ | ☐ |
| ((Sheldon) g).say(); | ☐ | ☐ |
| ((Sheldon) g2).say(); | ☐ | ☐ |
| ((Geek) s).say(); | ☐ | ☐ |

***Exercice 7**

Indiquer l'affichage du programme suivant :

```
class A {
public void affiche() {
System.out.println ("Je suis un A") ;
} }
class B extends A { }
class C extends A
{ public void affiche()
{ System.out.println ("Je suis un C") ;
} }
class D extends C
{ public void affiche()
{ System.out.println ("Je suis un D") ;}}
class E extends B {}
class F extends C {}
public class Test
{ public static void main (String arg[])
{ A a = new A() ; a.affiche() ;
B b = new B() ; b.affiche() ;
C c = new C() ; c.affiche() ;
D d = new D() ; d.affiche() ;
E e = new E() ; e.affiche() ;
F f = new F() ; f.affiche() ;
}
}
```

**Exercice 8**

Quels résultats fournit ce programme ?

```
class A {
public int a = 5;
public A(int a) {
this.a = a;
}
public void afficherVariables() {
System.out.println("a = " + a);
}
}
class B extends A {
```

```
public int a = 6;
public B(int b) {
super(2 * b);
a = b;
}
public void afficherVariables() {
System.out.println("this.a= " + a+ "
super.a= "+super.a );
}
}
class Alphabet {
public static void main (String[] args)
A[] as = new A[2];
as[0] = new A(1);
as[1] = new B(2);
for (int i = 0; i < as.length; i++) {
as[i].afficherVariables();
System.out.println("-----");
}
}
}
```

***Exercice 9**

À quel affichage conduit l'exécution du programme suivant ?

```
class A{
A a;
A(){ this.a = this; }
A(A a){ this.a = a; }
void m(){
if(this == this.a) System.out.println("Ahah!");
else System.out.println("Héhé!");
}
}
class B extends A{
B o;
B(){ super(); this.a = this; this.o = (B) this; }
void m(){ System.out.println("Ohoh!"); }
public static void main(String[] toto){
A u = new A();
A i = new A(u);
A b = new B();
u.m();
i.m();
b.m();
((B) b).o.m();
}
}
```

**Exercice 10**

Le code suivant compile-t-il? Si non, indiquez les erreurs affichées par le compilateur et proposez des corrections. A quel affichage conduit l'exécution du programme (éventuellement corrigé) ?

```
class A{
int i;
A(){
```

```
this.i = 0;
}
void m(){System.out.println("A");}
}
class B extends A{
int i = 2;
void m(){System.out.println("B");}
public static void main(String[] toto){
A a = new A();
B b = new B();
System.out.println(b.i);
System.out.println(a.i);
System.out.println(((A) b).i);
b.m();
a.m();
}
}
```

***Exercice 11**

Indiquez quelles lignes de la méthode main sont incorrectes.

```
package toto;
public class C {
public C tutu;
private int i = 0;
int j = 1;
static protected int k = 2;
}
package toto.titi;
public class Test extends toto.C {
public static void main(String[] t) {
toto.C m = new toto.C();
m.tutu = m;
m.tutu.i = 4;
m.j = 5;
m.tutu.k = 6;
}
}
```