Master bio-info : Java

Année 2011-2012

TD5

Exercice 1 (Classification)

On prend les classes suivantes : `Etudiant`, `Personne`, `EtudiantTravailleur`, `Enseignant`, `EtudiantSportif` et `Travailleur`.

1. dessinez une arborescence cohérente pour ces classes en la justifiant,
2. où se situeront les champs suivants : `salaire`, `emploiDuTemps`, `anneeEtude`, `nom`, `age` et `sportPratique`.
3. Écrivez une méthode qui prend un tableau de personnes en argument et retourne la somme des salaires de ces personnes.

Exercice 2 On considère les en-tête de méthodes suivantes :

1. `int methode(int a, char b)`
2. `void methode()`
3. `int methode(int a)`
4. `int methode(char b, int a)`
5. `int methode(int b, char a)`
6. `void methode(int a, char b)`
7. `int methode(int a, char b, float c)`

Quelles combinaisons de méthodes peuvent être définies simultanément dans une même classe ?

- a Toutes,
- b 1, 2, 3, 4
- c 2, 3, 5, 6, 7
- d toutes sauf 5,
- e toutes sauf 5 et 6
- f rien de tout ça.

Exercice 3 (Liaison dynamique - 2)

Qu'affiche le programme suivant ?

```
class A {
    public String f(B obj) { return ("A et B");}
    public String f(A obj) { return ("A et A");}
}
class B extends A {
    public String f(B obj) { return ("B et B");}
    public String f(A obj) { return ("B et A");}
}
class test {
    public static void main (String [] args){
        A a1 = new A();
```

```
A a2 = new B();  
B b = new B();  
System.out.println(a1.f(a1));  
System.out.println(a1.f(a2));  
System.out.println(a2.f(a1));  
System.out.println(a2.f(a2));  
System.out.println(a2.f(b));  
System.out.println(b.f(a2));  
}  
}
```

correction

```
class A {
    public String f(B obj) { return ("A et B");}
    public String f(A obj) { return ("A et A");}
}
class B extends A {
    public String f(B obj) { return ("B et B");}
    public String f(A obj) { return ("B et A");}
}
class test {
    public static void main (String [] args){
        A a1 = new A();
        A a2 = new B();
        B b = new B();
        System.out.println(a1.f(a1));
        System.out.println(a1.f(a2));
        System.out.println(a2.f(a1));
        System.out.println(a2.f(a2));
        System.out.println(a2.f(b));
        System.out.println(b.f(a2));
    }
}
```

Correction :

```
A et A
A et A
B et A
B et A
B et B
B et A
```

Le choix de la méthode est fait dynamiquement pendant l'exécution, tandis que le type de l'argument est déterminé à la compilation.