

## TD - Séance n°6

### Contrôle Continu

Aucun documents autorisés.

**Exercice 1** Question de cours :

1. Expliquez en quelques phrases l'utilité du mot clé `protected`, rappelez le contexte de son utilisation.
2. Expliquez en français l'utilité et les règles d'utilisation du mot clé `super`
3. Donnez un exemple de quelques lignes qui utilise l'instruction :  
`for (int x:y) i+=(x>0)?1:-1;`  
donnez la valeur finale de `i` correspondant à votre exemple.
4. La méthode `toString()` par défaut est elle `final` ? est elle `statique` ?

**Exercice 2** On s'intéresse à la hiérarchie de classe où `B` hérite de `A`. Dans chacune des classe on redéfinit, ou pas, des méthodes de `A`. Le but de cet exercice est de prédire quelles sera le code qui sera exécuté pour les exemples mentionnés dans le `main`.

```
public class A{
    public String x="x de A";
    public void f(){System.out.println("f de A");}
    public void g(){System.out.println(this.x);}
    public void h(){System.out.println(this.x);}
    public void k(){System.out.print("via k :");this.f();}
    public void l(A x) {x.f();}
}
public class B extends A{
    public String x="x de B";
    public void f(){System.out.println("f de B");}
    public void h(){System.out.println(this.x);}
    public void l(B x) {x.k();}
}
public class Test{
    public static void main(String [] args){
        A a=new A();
        B b=new B();
        A ab=new B();
```

```

// série 1
System.out.println(a.x);
System.out.println(b.x);
System.out.println(((A)b).x);
System.out.println(ab.x);

// série 2
a.f();
b.f();
((A)b).f();
ab.f();

// série 3
a.g();
b.g();
((A)b).g();
ab.g();

// série 4
a.h();
b.h();
((A)b).h();
ab.h();

// série 5
a.k();
b.k();
((A)b).k();
ab.k();

// série 6
a.l(a);
a.l(b);
a.l(ab);
b.l(a);
b.l(b);
b.l(ab);
ab.l(a);
ab.l(b);
ab.l(ab);
}
}

```

**Exercice 3** On souhaite modéliser les différents moyens de paiements proposés par une banque à une personne à l'aide d'une hiérarchie de classe basée sur une classe mère MP. Vous vous contenterez ici de désigner la banque et la personne par leurs noms.

Les événements qui peuvent potentiellement interagir avec les moyens de paiements sont :

- `boolean retrait(..)`,
- `void dépôt(..)`,
- `boolean vérificationIdentité(..)`,
- `boolean bilanMensuel()`

La méthode `bilanMensuel` sera appelée à chaque fin de mois, il n'est donc pas nécessaire de se préoccuper de ce qui se passe en fin de mois dans les autres méthodes.

Comme nous n'avons pas encore vu en TD/TP les classes abstraites, on supposera que celles de ces méthodes qui n'ont pas de sens dans la classe de base y sont définies sur le modèle suivant :

```
System.out.println(
    "La méthode xxx n'a pas de sens sur la classe de base"
);
return yyy; // où yyy est une valeur sans importance
```

On considère dans notre modélisation :

- Les **comptes épargnes réglementés**. Ils sont utilisés pour déposer de l'argent. Ils proposent tous un taux de rémunération mensuel identiques (car réglementés). La somme disponible est uniquement celle qu'on y a déposée. On supposera, contrairement à la réalité, qu'elle n'est pas limitée.
- Les **cartes bleues**. Elles peuvent être vues comme une ligne de crédit dont le plafond est fixé à une certaine hauteur, négociée entre la banque et son client. Elles ne servent qu'à payer (avec la méthode `retrait`) et il n'y a pas de problèmes tant que le plafond n'est pas atteint.

Chaque carte est associée à un autre moyen de paiement, ainsi à la fin du mois la ligne de crédit sera restaurée si la somme due peut être prélevée du moyen de paiement associé.

Si on souhaite déposer de l'argent sur une carte bleue, ce n'est pas exactement ce qui se passe, on dépose en fait la somme sur le compte associé.

- Les **comptes courants**. Ce sont des instruments sur lesquels le client peut avoir un solde négatif du moment que le découvert reste inférieur à un seuil. Eventuellement, à la fin du mois il payera simplement des pénalités mensuelles proportionnelles à un taux (fixé à la tête du client)

1. Définissez les différentes classes, et écrivez uniquement leurs constructeurs.
2. Ecrivez les méthodes qui ont un sens dans la classe de base MP
3. Ecrivez la méthode `retrait` de la classe modélisant les comptes courants

4. Ecrivez la méthode **dépot** dans la classe modélisant les cartes bleues. Dans le cas où une carte bleue est associée à une autre carte bleue que se passe t'il ? Y a t'il un risque ? (si oui précisez lequel mais ne vous en occupez pas ici, si non précisez également)
5. (**facultative**) Pour cette dernière question on considère une liste quelconque de moyens de paiement, et on s'interroge sur la bonne façon d'effectuer le bilan mensuel global. En particulier on ne veut pas donner des intérêts (ou faire payer des pénalités) plusieurs fois pour le même mois. Esquissez une solution en l'expliquant d'abord en français, puis seulement si vous en avez le temps vous écrirez les grandes étapes du code.