

Durée : 2h

CONTRÔLE

Exercice 1 (4 pts)

1. Combien d'instances de la classe A va créer le code suivant ?

```
A a1, a2, a3, a4, a5;  
a1 = new A();  
a2 = a1;  
a3 = new A();
```

- A) 5 B) 2 C) 3 D) aucune

2. Soient les classes A et B définies comme suit :

```
1 class A {  
2     public int x;  
3     public A() { x=5;}  
4 }  
5 class B extends A {  
6     public B() { x++;}  
7     public B(int i){this(); x+=i;}  
8 }
```

Qu'affichera le code suivant ?

```
B b1 = new B();  
B b2 = new B(20);  
System.out.println(b1.x + " et " + b2.x);
```

- A) 6 et 25 B) 5 et 25 C) 6 et 26 D) 1 et 21

3. Soient les classes A et B définies comme suit :

```
1 class A {  
2     public int x;  
3     public A() { x=5;}  
4 }  
5 class B extends A {  
6     public B() { x++;}  
7     public B(int i){super(); x--;}  
8 }
```

Qu'affichera le code suivant ?

```
B b1 = new B();  
B b2 = new B(20);  
System.out.println(b1.x + " et " + b2.x);
```

- A) 1 et 4 B) 1 et 5 C) 6 et 4 D) 6 et 5

4. Soient les classes A et B définies comme suit :

```
1 class A {  
2     protected static int x=0;  
3     public int f(){ x+=2; return 5*x;}  
4 }  
5 class B extends A {  
6     public int f(){ x+=3; return 5*x;}  
7 }
```

Qu'affichera le code suivant ?

```
B b1 = new B();  
B b2 = new B();  
System.out.println(b1.f() + " et " + b2.f());
```

- A) 10 et 10 B) 15 et 15 C) 15 et 30 D) 10 et 20

Exercice 2 (13 pts)

On veut développer une application pour gérer les tickets de caisse concernant des achats des clients d'un supermarché. Le supermarché propose un certain nombre d'articles aux clients. Un article est décrit par une référence (de type chaîne de caractères), une désignation et un prix unitaire.

1. Définir une classe **Article** comportant (4 pts) :
 - a. un constructeur d'initialisation ;
 - b. les accesseurs (getteurs et setters) des attributs **désignation** et **prix** ;
 - c. une méthode **equals(Article)** qui vérifie si deux articles sont identiques (ils ont les mêmes référence et désignation).

Un client peut acheter un ou plusieurs articles avec des quantités différentes. Un achat est décrit par un numéro, l'article et la quantité achetés.

2. Définir une classe **Achat** comportant (4 pts) :
 - a. un constructeur d'initialisation ;
 - b. une méthode **montantAchat()** qui renvoie le prix total d'un achat ;
 - c. une méthode **equals(Achat)** qui vérifie si deux achats ont les mêmes numéro et produit ;

Un ticket de caisse est délivré à un client après le paiement de ses achats. Un ticket de caisse comporte un numéro (généré automatiquement), la date à laquelle le ticket est établie (date du jour) et la liste des achats réalisés par le client (utiliser **ArrayList**).

3. Définir une classe **TicketCaisse** comportant (5 pts) :
 - a. un constructeur par défaut ;
 - b. une méthode **ajout(Achat)** qui permet d'ajouter un achat à la liste d'achats.
 - c. une méthode **supp(Achat)** qui permet de supprimer un achat de la liste d'achats.
 - d. une méthode **montantTotal()** qui renvoie le montant total des achats réalisés ;

Exercice 3 (3 pts)

Écrire un programme Java permettant déterminer l'élément le plus fréquent d'un tableau d'entiers. Le programme doit aussi fournir le nombre d'occurrences de l'élément dans le tableau. Par exemple, pour le tableau suivant {2, 8, 6, 8, 1, 6, 7, 2, 1, 8}, l'élément le plus fréquent est 8 et sa fréquence d'apparition est 3.