Énoncés 5

Exercices sur les références et les objets imbriqués

1 Exercice : combien d'objets créés ?

Dans cet exercice, nous allons utiliser la classe `Compte` présentée dans le cours sur les classes. Cette classe comporte le nom du titulaire sous forme d'une chaîne de caractères.

Chaque question comporte une méthode main et le but de l'exercice est de dire à la lecture combien d'objets de type `Compte` sont créés au cours de l'exécution du programme et combien restent accessibles dans la méthode `main` à la fin de l'exécution.

Notez que l'on ne compte ici que les objets instance de `Compte`. Il y a d'autres objets impliqués dans les programmes : les chaînes de caractères qui sont des objets instances de `String`.

Question 1

```
public class Combien1{
    public static void main(String[] args){
        Compte c1, c2;
        c1 = new Compte("lulu",101);
        c2 = new Compte("lili",505);
        System.out.println(c1.getTitulaire());
        System.out.println(c2.getTitulaire());
    }
}
```

— Combien de comptes sont créés lors de l'exécution du programme ?
— Combien de comptes sont utilisables à la fin du programme ?
— Qu'affiche ce programme ?

Celui-là est trop facile, c'est juste un échauffement.

Question 2

```
public class Combien2{
    public static void main(String[] args){
        Compte c1, c2;
        c1 = new Compte("lulu",101);
```

```
        c2 = new Compte("lili",505);
        c1 = c2;
        System.out.println(c1.getTitulaire());
        System.out.println(c2.getTitulaire());
    }
}
```

— Combien de comptes sont créés lors de l'exécution du programme ?
— Combien de comptes sont utilisables à la fin du programme ?
— Qu'affiche ce programme ?

Question 3

```
public class Combien3{
    public static void main(String[] args){
        Compte c1, c2, c3;
        c1 = new Compte("lulu",101);
        c2 = new Compte("lili",505);
        c3 = c2;
        c1 = c3;
        c2 = c1;
        System.out.println(c1.getTitulaire());
        System.out.println(c2.getTitulaire());
        System.out.println(c3.getTitulaire());

    }
}
```

— Combien de comptes sont créés lors de l'exécution du programme ?
— Combien de comptes sont utilisables à la fin du programme ?
— Qu'affiche ce programme ?

Question 4

```
public class Combien4{
    public static void main(String[] args){
        Compte c1;
        Compte[] tab = new Compte[3];
        c1 = new Compte("lulu",101);
        for (int i=0; i<3; i++){
            tab[i] = c1;
        }
        System.out.println(c1.getTitulaire());
        for (int i=0; i<3; i++){
            System.out.println(tab[i].getTitulaire());
        }
    }
}
```

— Combien de comptes sont créés lors de l'exécution du programme ?
— Combien de comptes sont utilisables à la fin du programme ?

— Qu'affiche ce programme ?

Question 5

```
public class Combien5{
    public static void main(String[] args){
        Compte c1;
        Compte[] tab = new Compte[3];
        c1 = new Compte("lulu",101);
        for (int i=0; i<3; i++){
            tab[i] = new Compte("lulu",101);
        }
        System.out.println(c1.getTitulaire());
        for (int i=0; i<3; i++){
            System.out.println(tab[i].getTitulaire());
        }
    }
}
```

— Combien de comptes sont créés lors de l'exécution du programme ?
— Combien de comptes sont utilisables à la fin du programme ?
— Qu'affiche ce programme ?

2 Exercice : dessiner la mémoire

Pour cet exercice, vous allez dessiner des structures de données au moyen de petits schémas analogues à ceux du cours, comportant la pile et le tas. Les références pourront être représentées par des flèches ou des adresses.

Pour chacun des programmes suivants, la question est la même : dessinez l'état de la mémoire à la fin de l'exécution de la méthode main.

Dans la plupart des cas, les classes ont été simplifiés à l'extrême : il faudrait ajouter des méthodes pour avoir des classes réalistes.

Question 1 *tableaux*

```
class Dessin1{
    public static void main(String[] args){
        int[] t1, t2, t3, t4, t5;
        t2 = new int[4];
        t3 = new int[4];
        t4 = new int[4];
        for (int i=0; i<t3.length; i++){
            t3[i] = 10+i;
            t4[i] = 10+i;
        }
        t5 = t4;
    }
}
```

Question 2 *objets*

```
class Dessin2{
    public static void main(String[] args){
        C1 var1, var2, var3;
        C2 cou1, cou2, cou3, cou4, cou5, cou6;
        var1 = new C1(0);
        var2 = new C1(1);
        cou2 = new C2(null,null);
        cou3 = new C2(var1,var2);
        cou4 = new C2(var1,var1);
        cou6 = new C2(var1, new C1(2));
    }
}
class C1{
    private int x;
    C1(int i){
        x=i;
    }
}
class C2{
    private C1 premier, second;
    public C2(C1 p, C1 s){
        premier=p;
        second=s;
    }
}
```

Question 3 *tableaux d'objets*

Même question pour le programme suivant.

```
class Dessin3{
    public static void main(String[] args){
        C1[] t1, t2, t3;
        C1 var = new C1(0);
        t1=new C1[3];
        t2=new C1[3];
        t3=new C1[3];
        for (int i=0; i<t1.length; i++){
            t1[i] = var;
            t2[i] = new C1(0);
        }
        t1[1].setX(1);
        t2[1].setX(1);
        t3 = t2;
    }
}
```

Question 4 *objets contenant des tableaux*

Même question pour le programme suivant.

```
class Dessin4{
    public static void main(String[] args){
        C3 pl1, pl2, pl3, pl4;
        C1 cl1;
        C1[] tab;
        cl1 = new C1(0);
        pl1 = new C3(null);
        pl2 = new C3(new C1[3]);
        pl3 = new C3(new C1[3]);
        pl4 = new C3(pl3.getTab());
        tab = pl3.getTab();
        tab[1] = cl1;
        tab[2] = cl1;
    }
}
class C3{
    private C1[] tab;
    public C3(C1[] t){
        tab = t;
    }
    public C1[] getTab(){
        return tab;
    }
}
```

3 Exercice d'imbrication : jeu d'Othello

Le jeu d'Othello est une jeu de plateau. Il y a un plateau de 8x8 cases sur lesquelles les joueurs posent alternativement des jetons qui ont un côté blanc et l'autre côté noir. En cours de jeu, les jetons peuvent être retournés si bien que la couleur visible change.

Si vous ne connaissez pas ce jeu, consultez les pages suivantes :

— `https://fr.wikipedia.org/wiki/Othello_(jeu)` l'article wikipedia sur ce jeu.
— `https://www.youtube.com/watch?v=Z5EN-cbgo-4` les règles du jeu en vidéo.
— `https://www.jeux-gratuits.com/jeu-gratuit-reversi-othello.html` le jeu en ligne.

Nous n'allons pas faire le jeu complet mais esquisser deux classes, une pour les jetons, l'autre pour le plateau de jeu. Les opérations à implémenter dans cette première esquisse seront :

— créer un plateau de jeu (vide ou avec la configuration intiale à 4 jetons).
— créer un jeton.
— déposer un jeton sur le plateau avec une certaine couleur visible.
— retourner un jeton déjà posé sur le plateau.
— réaliser un `toString` pour chaque classe.

Un nouvel objet java instance de la classe des jetons sera créé lorsqu'un jeton est posé sur le plateau. En revanche, retourner un jeton ne doit pas créer de nouvel objet, juste modifier un objet existant.

Parmi les questions à traiter : comment référencer une case, comment représenter une case du plateau, comment représenter en Java une case vide.