

Contrôle de programmation Java n°1

durée 1h30 – justifiez vos réponses

1. (3 points) On définit

```
1 class DeuxTableaux {
2 private int [] ta ;
3 private static int [] tb;
4 public DeuxTableaux(int nA, int nB)
5     { this.ta = new int [nA]; this.tb= new int [nB]; }
6 public DeuxTableaux(){ }
7 public String affichage (int i)
8     {return ("ta["+ i+"] _vaut_" + this.ta[i] + " , _tb["+i+"] _vaut_" +this.tb[i]);}
9 public void setTa(int x,int i){this.ta[i]=x;}
10 public void setTb(int x, int i){this.tb[i]=x;}
11 public int [] getTa(){return this.ta;}
12 public int [] getTb(){return this.tb;}
13 }
```

puis

```
1 class EssaiDeuxTableaux {
2 public static void main(String args []) {
3     DeuxTableaux p1; DeuxTableaux p2;
4     p1 = new DeuxTableaux(3,5);
5     p2 = new DeuxTableaux(3,4);
6     for (int k=0;k<p1.getTa().length;k++) p1.setTa(1,k);
7     for (int k=0;k<p1.getTb().length;k++) p1.setTa(1,k);
8     for (int k=0;k<p2.getTa().length;k++) p2.setTa(2,k);
9     for (int k=0;k<p2.getTb().length;k++) p2.setTa(2,k);
10    System.out.println (p1.getTb().length);
11    System.out.println (p1.affichage(0));
12    System.out.println (p2.affichage(0)); }
```

Quel affichage obtient-on ? JUSTIFIEZ.

2. (5 points) On définit

```
1 class Truc {  
2 public int entier ;  
3 public Truc(int a) { entier = a ; }  
4 public Truc(Truc t) { this.entier = t.entier ; }  
5 public boolean test( Truc t ) { return (t.entier==this.entier) ;}}
```

2.1 Comment appelle-t-on ce qui est défini en ligne 3 ?

2.2 L'instruction suivante est-elle valide (JUSTIFIEZ)?

```
Truc machin = new Truc();
```

2.3 Quel sera l'affichage provoqué par l'exécution de la méthode suivante (JUSTIFIEZ)?

```
1 public static void main(String args []) {  
2 Truc y = new Truc(1);  
3 Truc z = y;  
4 Truc w = new Truc(y);  
5 if (z==y) System.out.println ( "1" ) ;  
6 if (w==y) System.out.println ( "2" ) ;  
7 if (z.test(y)) System.out.println ( "3" ) ;  
8 if (w.test(y)) System.out.println ( "4" ) ;  
9 }
```

3. (5 points) Un intervalle d'entiers est défini par la donnée de ses bornes entières inférieure `bInf` et supérieure `bSup`. Par exemple, si `bInf=2` et `bSup=7` l'intervalle sera l'ensemble d'entiers $\{2,3,4,5,6,7\}$ et si si `bInf=5` et `bSup=3` l'intervalle sera vide.

Définir la classe `Intervalle` avec

- ses attributs privés `int bInf, bSup`
- ses accesseurs,
- un constructeur à deux paramètres entier,
- une **méthode d'instance** `estVide` qui renvoie `true` si un intervalle est vide et `false` sinon,
- une **méthode de classe** `longueur` qui renvoie le nombre d'entiers d'un intervalle,
- une **méthode d'instance** `union` à un paramètre de type `Intervalle` qui renvoie le plus petit intervalle contenant l'objet et le paramètre.

4. (7 points) Un cheval de course est déterminé par son nom, le tableau de ses résultats aux courses de l'année et la moyenne de ses gains.

Pour construire une classe représentant un cheval de course on va tout d'abord construire une classe pour maintenir à jour le tableau de ses résultats : on aura besoin d'un tableau et d'un entier représentant le nombre de courses déjà réalisées, sachant qu'il courra un maximum de 40 courses. Le calcul des gains suit la règle suivante : 10000€ pour la première place, 5000€ pour la deuxième place, 2000€ pour la troisième place et rien pour les autres.

4.1 Définir la classe `TableauDeCourses` caractérisée par un tableau de 40 bytes et un byte `nbreCourse` avec

- des attributs privés `tabResult` et `nbreCourse`,
- des accesseurs en lecture seulement,
- une méthode d'instance `ajouterValeur(byte n)` qui ajoute la valeur `n` à l'attribut `tabResult` de l'objet si la valeur de l'attribut `nbreCourse` le permet, ne fait rien sinon

4.2 Définir la classe `ChevalDeCourse` caractérisée par un nom, un tableau des résultats et la moyenne des gains des courses enregistrées dans le tableau avec

- ses attributs privés `String nom; TableauDeCourses tCourse ; float gainMoyen ;`
- un constructeur avec un seul paramètre de type `String`¹
- un accesseur en lecture pour l'attribut `gainMoyen`
- une méthode d'instance `ajouterGain (int s)` qui ajoute la somme `s` à l'objet et met l'objet à jour.

4.3 A-t-on besoin d'une méthode qui renvoie le gain moyen d'un objet `ChevalDeCourse` ?

¹Ne pas oublier les initialisations de tous les attributs