Contrôle de programmation Java n°2

durée 45 minutes – justifiez vos réponses

1. (4 points) On définit

```
1 class UnePaire {
2 private int a ;
3 private int b;
4 public UnePaire(int x, int y) { this.a = x ; this.b= y; }
5 public UnePaire(){}
6 public boolean equals(UnePaire p){ return ((this.a==p.a)&&(this.b==p.b));}
7 public void setA(int a){this.a=a;}
8 public void setB(int b){this.b=b;}
9 }
```

puis

```
1 class EssaiUnePaire {
2 public static void main(String args[]) {
3 unePaire p1; unePaire p2;
4 p1 = new UnePaire();
5 p2 = new UnePaire(3,4);
6 p1.setA(3);
7 p1.setB(4);
8 System.out.println (p1==p2);
9 System.out.println (p2.equals(p1)); }}
```

Quel affichage obtient-on ? Justifiez.

2. (8 points) On définit

```
1 class Truc {
2 public static int entier ;
3 public Truc(int a) { entier = a ; }
4 public Truc(){}
5 public String toString(){return("l'entier vaut "+ entier);}
6 }
```

```
1 class Machin extends Truc {
2 public boolean verite ;
3 public Machin(int a){
4         if (a==this.entier) verite=true; else verite=false;}
5 public String toString(){
6         if (verite) return (super.toString()+": c'est la vérité");
7         else return (super.toString()+": ce n'est pas la vérité");
8 }
9 }
```

```
1  class Chose extends Truc {
2  public boolean boo ;
3  private static int sauv = entier ;
4  public Chose(int a){
5  super(a);
6  if (a== sauv) boo=true;
7  else boo = false; }
8  public String toString(){
9  if (boo) return(super.toString());
10 else return("l'entier valait" + sauv + super.toString());
11 }
12 }
```

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

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

3. (8 points)

3.1 Ecrire une classe Pet dont les objets sont caractérisés par un attribut privé *nom* de type String et un attribut *genre* de type **boolean**. Ecrire un constructeur à 2 paramètres.

Réécrire la méthode toString() pour que s'affiche soit "il s'appelle " suivi du *nom* de l'objet si *genre* vaut **true**, soit "elle s'appelle " suivi du *nom* de l'objet si *genre* vaut **false**.

3.2 Ecrire une classe Chien, sous-classe de Pet, dont les objets sont caractérisés par un attribut *race* de type String. Ecrire un constructeur à 3 paramètres.

Réécrire la méthode toString() pour que, si *genre* vaut true, s'affiche "il s'appelle " suivi du *nom* de l'objet suivi de " et il aboie", sinon, s'affiche "elle s'appelle " suivi du *nom* de l'objet suivi de " et elle aboie".

Réécrire la méthode equals(Chien c) qui retourne **true** si l'objet a le même *nom* et est de la même *race* que le paramètre c.

3.3 Ecrire une classe Chat, sous-classe de Pet, dont les objets sont caractérisés par un attribut *poids* de type entier. Ecrire un constructeur à 3 paramètres.

Réécrire la méthode toString() pour que s'affiche le *nom* de l'objet Chat suivi de " ronronne fort" si *poids* est supérieur à 5, ou suivi de "ronronne" sinon.