Master bio-info : Java

Année 2013-2014 TP 3

Exercice 1 [Classe Personne]

1. Écrivez une classe `Personne` (constructeur et attributs) permettant de décrire complètement une personne, sachant que l'on souhaite avoir autant d'informations que dans la phrase suivante :

   "LeBlanc Pierre, sexe masculin, né(e) en 1965, célibataire" ou "LeNoir Bernadette, sexe féminin, né(e) en 1945, marié(e)"

2. On veut ajouter une méthode `age` qui calcule l'âge approximatif d'une personne en fonction de l'année courante. Pour cela,
   - Ajoutez un attribut `anneeCourante`, et ajouter une méthode `incrémenteAnneeCourante()`.
   - Écrivez la méthode `age()`.

3. Ajoutez à la classe `Personne` un attribut conjoint et examinez les conséquences que cela peut avoir sur l'ensemble du code.

   Attention : on ne peut-être marié qu'à une seule personne !

4. Ajoutez une méthode `boolean marier(Personne p)` qui permet de marier une personne à une autre. On vérifiera que les deux mariés ont plus de 18 ans, sinon le mariage n'aura pas lieu et on retournera `false`.

5. Écrivez une méthode `boolean marieAvec(Personne p)`, telle que `p1.marieAvec(p2)` retourne `true` si les personnes `p1` et `p2` sont mariés ensemble. Écrivez une autre méthode `boolean marieAvec(Personne p1, Personne p2)`. Que peut-on en dire ? Quelle méthode `marieAvec` vous paraît préférable ?

6. Écrivez un autre constructeur qui créera une personne et la mariera en même temps à la personne donnée en argument.

7. Ajoutez une méthode `boolean divorcer()` qui annule le précédent mariage s'il y avait lieu, retourne `false` si la personne n'était pas mariée, et `true` sinon.

8. Écrivez une méthode `Personne enfanter(String prenom, Char sexe)` qui créera une nouvelle personne de même nom que la mère, de sexe masculin ou féminin selon la lettre ('M' ou 'F') donnée en argument.

   Attention : A l'heure actuelle, seule une femme peut enfanter !

9. Soit la classe `Compte`, permettant de gérer un compte en banque :

```
class Compte {
    private int solde;

    public Compte() { solde = 0; }
    public int getSolde() { return solde;}
    public void credit(int montant) { solde += montant; }
    public void debit(int montant) { solde -= montant; }
}
```

On suppose maintenant que toute personne a un attribut `compte` de type `Compte`. Complétez le code suivant (dans `Personne`).

```
//Vole a p sa monnaie. Modifie la monnaie de p.
    public void vole(Personne p){ ... }

    //Donne sa monnaie a p. Modifie la monnaie de p.
    public void donne(Personne p){ ... }

    //Depose la monnaie sur le compte
    public void deposeMonnaie(){ ... }

    //Retire de l'argent du compte pour avoir de la monnaie
    public void retireMonnaie(int montant){ ... }
```

Exercice 2 [instances] La classe A est définie par

```
public class A{}
```

Combien d'instances de la classe A crée le code suivant ?

```
A x,u,v;
x=new A();
A y=x;
A z=new A();
```

(a) Aucune
(b) Cinq
(c) Trois
(d) Deux

Exercice 3 [this()] Pour la classe B définie comme suit :

```
class B {
public String s = "";
public B(){s = s + "Ciao";}
public B(int i) {this(); s += ("Bonjour "+i);}
}
```

qu'afficheront les instructions suivantes ?

```
B monB=new B(2003);
System.out.println(monB.s);
```

(a) erreur de compilation
(b) erreur d'exécution
(c) CiaoBonjour 2003
(d) Bonjour 2003

Exercice 4 [statique/dynamique] Qu'affichera le code suivant ?

```
class C {
public static int i;
public int j;
public C() {i++; j=i; }

public static void main(String[] args){
C x=new C(); C y=new C(); C z= x;
System.out.println(z.i + " et " + z.j);
}
}
```

(a) 2 et 2
(b) 1 et 1
(c) 2 et 1
(d) 1 et 3

Exercice 5 [] Qu'affichera le code suivant ?

```
public class D {
int a = 0 ;
public void f(int a) {
System.out.print(a + " ") ;
}
public void g(int a) {
System.out.println(this.a) ;
}
public static void main(String[] args) {
D p = new D();
p.f(12);
p.g(13);
}
}
```

(a) 12 13

(b) 12 0
(c) 0 13
(d) 0 0

Exercice 6 [égalité]
Qu'affichera le code suivant ?

```
public class E{
    public int x;
    public E(int x){
        this.x = x;
    }
    public static void main(String[] args){
        E a = new E(1);
        E b = new E(2);
        E c = new E(1);
        System.out.println(a.x + " "+ b.x + " " + c.x);
        System.out.println((b==c) + " " + (b.x == c.x));
        System.out.println((a==c) + " " + (a.x == c.x));
        b = a;
        System.out.println((a==b)+" " + (a.x == b.x));
    }
}
```

Exercice 7 [visibilité, statique/dynamique] Soit la classe F suivante

```
public class F{
    int a;
    private int b = 10;
    public static int c = 0;

    public F(int a){
        this.a = a;
        c = c + a;
    }

    private int getA(){
        return a;
    }
    public int getB(){
        return b;
    }
    public static int getC(){
        return c;
```

```
    }
    public void affiche (){
        System.out.println("a= "+ getA()+ ", b= "+  getB()+", c= "+  getC());
    }

}
```

Dans le code suivant, quelles lignes provoqueront des erreurs à la compilation ? Une fois celles-ci commentées, qu'affichera ce programme ?

```
public class FTest{
    public static void main(String[] args) {
        F p = new F(2);
        F q = new F(3);
        F r;
        p.affiche();
        q.affiche();
        r.affiche();

        System.out.println("p.getA() = " + p.getA());
        System.out.println("p.getB() = " + p.getB());
        System.out.println("p.getC() = " + p.getC());

        System.out.println(" F.getA() =" + F.getA());
        System.out.println(" F.getB() = " + F.getB());
        System.out.println("F.getC() = " + F.getC());

        System.out.println("p.a = " + p.a);
        System.out.println("p.b = " + p.b);
        System.out.println("p.c= " + p.c);

        System.out.println("F.a = " + F.a);
        System.out.println("F.b = " + F.b);
        System.out.println("F.c= " + F.c);
    }
}
```