Université Badji Mokhtar annaba
Faculté de technologie

Département d'informatique
2ème année LMD,S4

Année 2023/2024

Durée : 1h30

12/05/2024

Corrigé type Examen : programmation orientée objet

Documents non autorisés.

Exercice 1:(04 points) :

Pour chacune de ces affirmations, indiquez si elle est vraie ou fausse:

1. La surcharge en java :

a) Vrai 0,5 ptt

b) Faux 0,25 ptt

c) Vrai 0,25 ptt

2. Le mot-clé super :

a) Vrai 0,25 ptt

b) Vrai 0,25 ptt

c) Faux 0,5 ptt

3. Le mot-clé final peut être utilisé pour :

a) Vrai 0,5 ptt

b) Vrai 0,25 ptt

c) Vrai 0,25 ptt

4. Le constructeur en java :

a) Vrai 0,5 ptt

b) Vrai 0,25 ptt

c) Faux 0,25 ptt

Exercice 2: (04 points) :

- Corrigez les erreurs et donnez le résultat produit par ce programme :

Les erreurs sont mentionnées en rouge (2,25 ptt): la première solution ou la deuxième solution ou la troisième solution:

```
package p1;
0,25 public class Classe1 {
  private int n = 1;
0,75 protected int b = 6;
0,25 static private String s1;
public int getN() {
  return n; }
public void setN(int i) {
  n = i; }
public int methode1(Classe1 c)
{ c.setN(n + c.getN());
return n + c.getN(); }
public static String methode1(String
s2){
                return s1 + s2;
}
    }
```

```
package p2;
import p1.Classe1;// ou import p1.*;
0,5

public class Classe2 extends Classe1
{
     private int a;
     private String chaine;

   public int methode1(Classe1 c) {
          c.setN(b + c.getN());
          return b + c.getN();
     }
}
```

```
package p2;
 import p1.Classe1; //ou import p1.*;
0,5
public class Test {
public static void main (String []
args){
Classe1 c1 = new Classe1();
Classe1 c2 = new Classe1();
Classe2 c3 = new Classe2();
System.out.println(c2.getN());
System.out.println(c1.methode1(c2));
System.out.println(c2.getN());
System.out.println(c1==c3);
System.out.println(c3.methode1(c3));
          }
     }
```

```
package p1;
0,25 public class Classe1 {
  private int n = 1;
0,25 public int b = 6;
0,25 static    private String s1;
public int getN() {
  return n; }
public void setN(int i) {
  n = i; }
public int methode1(Classe1 c)
{ c.setN(n + c.getN());
return n + c.getN(); }
public static String methode1(String
s2){
                    return s1 + s2;
}
    }
```

```
package p2;
import p1.Classe1;// ou import p1.*;
0,5

public class Classe2 extends Classe1
{
      private int a;
      private String chaine;

   public int methode1(Classe1 c) {
            c.setN(b + c.getN());
            return b+ c.getN();
      }
}
```

```
package p2;
 import p1.Classe1; //ou import p1.*;
0,5
public class Test {
public static void main (String []
args){
Classe1 c1 = new Classe1();
Classe1 c2 = new Classe1();
Classe2 c3 = new Classe2();
System.out.println(c2.getN());
System.out.println(c1.methode1(c2));
System.out.println(c2.getN());
System.out.println(c1==c3);
System.out.println(c3.methode1(c3));
            }
      }
```

```
package p1;
0,25 public class Classe1 {
  private int n = 1;
 private int b = 6;
0,25 static    private String s1;
public int getN() {
  return n; }
public void setN(int i) {
  n = i; }
public int getB() {
      return b;
} 0,25
public int methode1(Classe1 c)
{ c.setN(n + c.getN());
return n + c.getN(); }
public static String methode1(String
s2){
                    return s1 + s2;
}
    }
```

```
package p2;
import p1.Classe1;// ou import p1.*;
0,5

public class Classe2 extends Classe1
{
      private int a;
      private String chaine;

   public int methode1(Classe1 c) {
            c.setN(getB() 0,25 +
c.getN());
            return getB() 0,25+
c.getN();
}
```

```
package p2;
 import p1.Classe1; //ou import p1.*;
0,5
public class Test {
public static void main (String []
args){
Classe1 c1 = new Classe1();
Classe1 c2 = new Classe1();
Classe2 c3 = new Classe2();
System.out.println(c2.getN());
System.out.println(c1.methode1(c2));
System.out.println(c2.getN());
System.out.println(c1==c3);
System.out.println(c3.methode1(c3));
            }
      }
```

Résultat de l'affichage (1,75 ptt):

System.out.println(c2.getN());	1 0,25
System.out.println(c1.methode1(c2));	3 0,5
System.out.println(c2.getN());	2 0,25
System.out.println(c1==c3);	false 0,5
System.out.println(c3.methode1(c3));	13 0,25

Exercice 3 : (12 points)

I. La classe joueur (3,5):

```
public class Joueur {
    private String nom; 0,25 ptt
    private int age;0,25 ptt
    private String position;0,25 ptt
    private String statut;0,25 ptt // ou private boolean statut;

    public Joueur(String nom, int age, String position, String statut //ou boolean statut )0,25 ptt {
        this.nom = nom;0,25 ptt
        this.age = age; 0,25 ptt
        this.position = position;0,25 ptt
        this.statut = statut;0,25 ptt
    }

    public double calculerPrimeJouesGagnes(int joues, int gagnes)0,25 ptt {
        double prime=0.0;
        if (statut=="principal")// ou if (statut)0,25 ptt
            prime = 100000 * (gagnes / joues); 0,25 ptt
            else prime = (100000 * (gagnes / joues)*0.5);// ou (100000 * (gagnes / joues))/2; 0,25 ptt

        return prime; 0,25 ptt
    }
}
```

II. La classe équipe (4,25) :

```
public class Equipe {
    private String nomEquipe; 0,25 ptt
    private String villeEquipe; 0,25 ptt
    private int matchsJoues;0,25 ptt
    private int matchsGagnes;0,25 ptt
    private int nombreJoueurs;0,25 ptt
    public Equipe(String nomEquipe, String villeEquipe, int matchsJoues, int matchsGagnes) 0,25 ptt {
        this.nomEquipe = nomEquipe; 0,25 ptt
        this.villeEquipe = villeEquipe; 0,25 ptt
        this.matchsJoues = matchsJoues; 0,25 ptt
        this.matchsGagnes = matchsGagnes; 0,25 ptt
        this.nombreJoueurs = 25; 0,25 ptt
    }

    public void comparerVictoires(Equipe autreEquipe)0,25 ptt {
       if (this.matchsGagnes > autreEquipe.matchsGagnes) 0,25 ptt{
            System.out.println("L'équipe " + this.nomEquipe + " a plus de victoires que l'équipe " +
autreEquipe.nomEquipe); 0,25 ptt
        } else if (this.matchsGagnes < autreEquipe.matchsGagnes)0,25 ptt {
            System.out.println("L'équipe " + autreEquipe.nomEquipe + " a plus de victoires que l'équipe
" + this.nomEquipe); 0,25 ptt
        } else {
            System.out.println("Les deux équipes ont le même nombre de victoires.");0,25 ptt
        }
    }
    }
```

Ou la version statique:

```
      public static void comparerVictoires(Equipe equipe1, Equipe equipe2) 0,25 ptt{
          if (equipe1.matchsGagnes > equipe2.matchsGagnes) 0,25 ptt{
              System.out.println("L'équipe " + equipe1.nomEquipe + " a plus de victoires que l'équipe
  " + equipe2.nomEquipe);0,25 ptt
          } else if (equipe1.matchsGagnes < equipe2.matchsGagnes) 0,25 ptt{
              System.out.println("L'équipe " + equipe2.nomEquipe + " a plus de victoires que l'équipe
  " + equipe1.nomEquipe);0,25 ptt
          } else {
```

```
            System.out.println("Les deux équipes ont le même nombre de victoires.") 0,25 ptt;
        }
    }
```

III. La classe match de football (2,75):

```
public class MatchDeFootball {
    private Equipe equipeA; 0,25 ptt
    private Equipe equipeB;0,25 ptt
    private int butsEquipeA; 0,25 ptt
    private int butsEquipeB; 0,25 ptt

    public MatchDeFootball(Equipe equipeA, Equipe equipeB, int butsEquipeA, int butsEquipeB) 0,25
ptt{
        this.equipeA = equipeA;0,25 ptt
        this.equipeB = equipeB; 0,25 ptt
        this.butsEquipeA = butsEquipeA; 0,25 ptt
        this.butsEquipeB = butsEquipeB; 0,25 ptt
    }

    public boolean estMatchNul() 0,25 ptt{
        return butsEquipeA == butsEquipeB;0,25 ptt// ou if (butsEquipeA == butsEquipeB) return true;
else return false; 0,25 ptt
    }
}
```

IV. Classe capitaine (1,5)

```
public class Capitaine extends Joueur 0,5 ptt{
    private int nombreMatchsCarriere; 0,25 ptt

    public Capitaine(String nom, int age, String position, String statut, int nombreMatchsCarriere)0,25
ptt {
        super(nom, age, position, statut);0,25 ptt
        if (nombreMatchsCarriere >= 100) {
            this.nombreMatchsCarriere = nombreMatchsCarriere;0,25 ptt
        }
    }

}
```