

Exercice 01 (9 pts)

```
package p1;
public class Calculateur {
    public void afficher(){
        System.out.println ("Je suis un Calculateur");}
}
```

```
package p1;
public class Multiplicateur extends Calculateur{
    private int value ;
    public Multiplicateur(int value) {
        this.value= value;}
    public void produit(int a, int b){ a=a*b;}
    public static void carre(){value= value*value;}
    public void afficher(){
        System.out.println ("Je suis un Multiplicateur") ;}
    public void marche(){
        System.out.print ("ça marche! ");super.afficher();}
}
```

```
package p2;
import p1.* ;

public class Test {
    public static void main(String args[]){
        //Création d'un objet c de la classe Calculateur
        Calculateur c=new Calculateur() ;
        //Création d'un objet m de la classe Multiplicateur
        Multiplicateur m= new Multiplicateur(0) ;

    }
}
```

Dans le tableau ci-dessous :

- Si le code est correct, répondez par « oui », et donner le résultat affiché par l'exécution du code.
- Si le code est incorrect, répondez par « non », et donner le code corrigé, sinon, expliquer l'erreur.

Code	Correct ?	Affichage/Correction/Explication
Calculateur.afficher() ;	non	c.afficher() ;

<code>m.afficher();</code>	oui	Je suis un Multiplicateur
<code>m.camarche() ;</code>	oui	ça marche! Je suis un Calculateur
<code>int x=5; int y=10; m.produit(x,y); System.out.println(x);</code>	oui	5
<code>c=m; c.afficher();</code>	oui	Je suis un Multiplicateur
<code>c=m; c.camarche() ;</code>	non	<code>c=m; ((Multiplicateur)c).camarche() ;</code>
<code>m=c ;</code>	non	Il n'est pas possible d'utiliser une référence de la classe fille « Multiplicateur » pour un objet de la classe mère « Calculateur ».

Exercice 01 (11 pts)

<code>//1</code>
<code>public abstract class Employe {</code>
<code> static int nbr=0;</code>
<code> protected int numero;</code>
<code> protected String nom;</code>
<code> protected String prenom;</code>
<code>//2</code>
<code> public Employe (String nom, String prenom){</code>
<code> numero=nbr;</code>
<code> nbr++;</code>
<code> this.nom=nom;</code>
<code> this.prenom=prenom;</code>
<code> }</code>
<code>//3.a</code>
<code> public abstract double salaireBrut();</code>
<code>//3.b</code>
<code> public double retenues() {</code>
<code> if (salaireBrut ()<20000)</code>
<code> return salaireBrut ()*0.05;</code>
<code> else</code>
<code> return salaireBrut ()*0.1;</code>
<code> }</code>
<code>//3.c</code>
<code> public double salaireNet() {return salaireBrut ()-retenues ();}</code>
<code>//3.d</code>
<code> public String toString () {return "Numero: "+numero+", Nom: "+nom+", Prénom: "+prenom+"Salaire net: "+net();}</code>

Responsable de la matière : Dr. Sadek Benhammada

```
}
```

```
//4
public class Permanent extends Employe{
    static final double valeurPI=50;
    private int indice;
//5
    public Permanent(String nom, String prenom, int indice) {
        super(nom, prenom);
        this.indice=indice;
    }
//6
    public String getIndice() {return nom;}
    public void setIndice(int indice) {this.indice=indice;}
//7
    public double salaireBrut () {return indice*valeurPI;}
//8
    public String toString () {return super.toString()+" , Indice:"+this.indice;}
}
```

```
//9
public class Test {
    public static void main(String args[])
    {
        Permanent p=new Permanent("Ahmed", "Ali", 900);
        p.setIndice(1000);
        System.out.print(p.toString());
    }
}
```