

Nom et prénom :

Groupe : **CD**

1. (6 points) Soient les deux classes Mere et Fille suivantes. Qu'affiche le code suivant :

```
public class Mere {
    protected String m;
    public Mere(String m) {
        this.m=m;
        System.out.println("appel_const_mere");}
    public String toString() {return "Mere!";}
    public String toString(int pos) {return m.substring(pos);}
    public void methode() {System.out.println("methode_mere_"+ this+ "_" + m);}
}

public class Fille extends Mere{
    public int n;
    protected static int i;
    public Fille(String m, int n) {
        super(m);
        this.n=n;
        System.out.println("appel_const_fille"); }
    public String toString() {return "Fille!" + "_" + i;}
    public void methode() {
        System.out.println("methode_fille_" + this+ "_" + super.toString()+"_" + m
            +"_" + n); }
    public void test() { System.out.println("Fille.test");}
    public static int f(Fille u, int p) {
        u.n= 5+p;
        p--;
        i=u.n;
        return p;
    }

    public static void main(String args[]) {
        Fille fi = new Fille("java", 17) ;
        Fille.i = 6;
        fi.n = 4 + Fille.f(fi, Fille.i);
        System.out.println(fi) ;
        fi.methode();
        Mere e= new Mere("hello") ;
        System.out.println(e) ;
        e.methode() ;
        Fille f2 = new Fille("essai", 1) ;
        f2.test();
        e=f2;
        e.methode();
        System.out.println(e.toString(2));
    }
}
```

Solution:

appel const mere(0.5pt)
appel const fille (0.5pt)
Fille! i= 11 (0.5pt)
methode fille Fille! i= 11 Mere! java 9 (0.5pt)
appel const mere (0.5pt)
Mere! (0.5pt)
methode mere Mere! hello (0.5pt)
appel const mere (0.5pt)

appel const fille (0.5pt)
 Fille.test (0.5pt)
 methode fille Fille! i= 11 Mere! essai 1 (0.5pt)
 sai (0.5pt)

2. (7 points) Indiquez si les instructions suivantes sont correctes ou pas (ce qui se passe à la compilation et à l'exécution) et, pour les instructions qui vous semblent correctes, indiquez ce qui serait affiché, si il y a affichage.

	Compilation	Exécution
1) Mere h = new Fille("mere",5);	(ok)upcast	(ok) 1pt
Fille var=(Fille)h;	(ok) downcast 1pt	appel const Mere 0.5pt appel const fille 0.5pt
2) Mere m=new Mere("lundi"); Object obj=m; Fille ff= (Fille) obj;	(ok) (ok) upcast (ok) downcast 1pt	Erreur ClassCastException 1pt : l'objet obj ne pointe pas Fille ou fille de la classe Fille
3) Mere b = new Fille("mere",8); b.test();	ne compile pas car test n'est pas membre de Mere 2pt	

3. (7 points) Soient la classe A et la classe B suivantes :

```
public class A {
    protected int [] tab;
    public A(int[] tab) {
        this.tab=tab;} }
class B extends A {
    protected String val;
    public B(int t1, int t2, int t3) {
        //a finir...}
}
```

1. Finir d'écrire le constructeur de la classe B qui prend 3 entiers et permet :
- d'initialiser son tableau d'entier tab avec ces 3 entiers et
 - d'initialiser le membre val à une chaîne vide ;

Solution:

```
public B(int t1, int t2, int t3) {
    super(new int[] {t1,t2,t3}); //1pt
    val=""; //1pt
}
```

2. redéfinir la méthode equals au niveau de la classe A qui permet de vérifier si deux objets de classe A sont les mêmes ou pas. En effet, pour considérer deux objets de classe A égaux si ils ont deux tableaux avec la même taille et avec exactement les mêmes valeurs à la même position.
 exemple : si obj1 possède un tableau constitué des valeurs 1, 2, 3 et obj2 possède un tableau constitué des valeurs 2, 1, 3 votre méthode equals devra retourner false. En revanche : si obj1 possède un tableau constitué des valeurs 1, 2, 3 et obj2 possède un tableau constitué des valeurs 1, 2, 3 votre méthode equals devra retourner true.

Solution:

```
@Override
    public boolean equals(Object o) { //0.5pt
        if (o==this) return true; //0.5pt
        if (!(o instanceof A)) return false; //1pt
        A r= (A)o; //1pt downcast pour pouvoir acceder aux attributs d'un objet de la
            classe A
        if(r.tab.length!=tab.length) return false; //0.5pt
        else{
            for(int i=0; i<tab.length; i++) //0.5pt
            {if(r.tab[i]!=tab[i]) return false; //0.5pt
            }
            return true; //0.5pt
        }
    }
```