

Nom et pr nom :

Groupe : AB

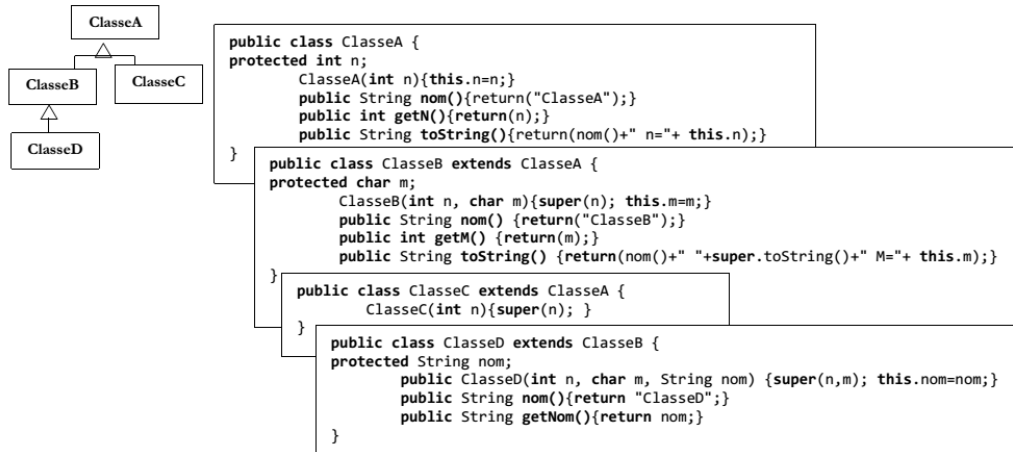
1. (10 points) Soient les deux classes suivantes. Qu'affiche le code suivant :

```
public class Classe1 {
    protected int j=3;
    public Classe1(){j++;
    System.out.println("const Mere "+j);}
    public String toString(){ return ("Mere, j= "+j);}
    public void test() {j++;
    System.out.println("Classe1.test : " + this);}

    public static void main(String[] args) {
        Classe1 m = new Classe2(1) ;
        System.out.println(m);
        m.test() ;
        Classe2 c= new Classe2(2);
        c.test();
        c.test1() ;
    }
}
class Classe2 extends Classe1{
    public int n ;
    public Classe2(int n) {this.n = n ;
    System.out.println("const fille "+ j+" "+n);}
    public String toString() { return super.toString()+ " n= "+n ; }
    public void test1() {
        this.n++;
        System.out.println("SC.test1 : " + this) ; }
    public void test() {
        super.test();
        System.out.println( "SC.test : " + n + " " + j);
    }
}
```

Solution: const Mere 4 (1pt)
const fille 4 1 (1pt)
Mere, j= 4 n= 1 (1pt)
Classe1.test : Mere, j= 5 n= 1 (1pt)
SC.test : 1 5 (1pt)
const Mere 4 (1pt)
const fille 4 2 (1pt)
Classe1.test : Mere, j= 5 n= 2 (1pt)
SC.test : 2 5 (1pt)
SC.test1 : Mere, j= 5 n= 3 (1pt)

2. (5 points) soit la hi rarchie de classes suivante :



1) Soit la déclaration suivante :

```
ClasseA a=new ClasseD(4, 'a', "essai");
```

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) <code>System.out.println(a);</code>	Ok (0,5pt)	Ok: ClasseD ClasseD n=4 M=a (1pt)
2) <code>System.out.println(a.getN());</code>	ok(0,5pt)	Ok: 4 (1pt)
3) <code>ClasseC c= a;</code>	Not ok(0,5pt)	
4) <code>ClasseC c=(ClasseC)a;</code>	ok(0,5pt)	Not ok: ClasseD cannot be cast to ClasseC (1pt)

3. (5 points) Soit la classe Personnel suivante :

```

public class Personnel {
    private String nom; /** son nom (une chaine de caractÃres) **/
    private String prenom; /**son prenom (une chaine de caractÃres) **/
    private int anneeNais; /**son annee de naissance (entier) **/
    ...}
  
```

Ãcrire la redÃfinition de la mÃthode equals qui permet de vÃrifier si deux personnels sont les mÃmes ou pas.

Solution:

```

public boolean equals(Object o) { //1pt
    if (o==this)return true; //1pt
    if (!(o instanceof Personnel)) return false; //1pt
    Personnel r= (Personnel)o; //0,5pt
    return(nom.equals(r.getNom())&& //1.5pt
           (prenom.equals(r.prenom))&&
           anneeNais==r.anneeNais);
}
  
```