

Exo 1 (5pt) : Quel est l'utilité des classes abstraites ?

Exo 2 (5pt) : Indiquer ce qui est affiché lors de l'exécution du programme suivant : 5pt

<pre>public class Base { public int a ; protected int b ; private int c; public Base(){ a+=1; System.out.println("C"+a); } public void dolt(){ System.out.println("G"+a); } }</pre>	<pre>public class Derived extends Base{ public int d=1; private int e; public Derived(){ super(); a+=d; System.out.println("A"+d); } public void dolt(){ d+=1; System.out.println("T"+d); super.dolt(); } public void dolt(int i){ System.out.println("S"+d); d+=i; dolt(); } }</pre>
<pre>public class Tester{ public int a=12; public static void main(String[] args){ Base x = new Base(); Derived y = new Derived(); x.dolt();y.dolt(); y.dolt(3); x=y; x.dolt();y.dolt(); y=(Derived)x; x.dolt();y.dolt();y.dolt(3); } }</pre>	

Exo 3 (10pt) : Donnez le code Java de la classe TV avec :

Attributs :

— on :boolean — chaine : int (>0 et <9999) — volume: int (>=0 et <=10)

Méthodes :

- void allumer() — void eteindre()
- void setChaine(int ch) // change de chaine,
- void chaineUp() //aller à la chaine suivante, — void chaineDown() // aller à la chaine précédente
- void volumeUp() // augmente le volume +1 — void volumeDown() //diminue le volume -1
- void reCall () // affiche la liste des 4 dernières chaines regardées

Remarque : les différentes opérations ne doivent opérer que si la TV est allumée.

Exo 1 ;

Utilité des classes abstraites

- En pratique, les classes abstraites permettent de définir des fonctionnalités (des comportements) que les sous-classes devront impérativement implémenter même si la classe abstraite n'est pas en mesure de fournir elle-même une implémentation pour ces méthodes.
- Les utilisateurs des sous-classes d'une classe abstraite sont donc assurés de trouver toutes les méthodes définies dans la classe abstraite dans chacune des sous-classes concrètes.
- Les classes abstraites constituent donc une sorte de contrat (spécification contraignante) qui garantit que certaines méthodes seront disponibles dans les sous-classes et qui oblige les programmeurs à les implémenter dans toutes les sous-classes concrètes.

Exo 2 ;

C1 ,C1 ,A1 ,G1 ,T2 ,G2 ,S2 ,T6 ,G2 ,T7 ,G2 ,T8 ,G2 ,T9 ,G2 ,T10,G2 ,S10,T14,G2 ,

Ex0 3

```
public class Tv{
    private boolean on ;
    private int chaine ;
    private int volume ;
    private [] int Tabrecall = new int[4];

    public Tv(){
        on = false ;
    }

    Public Tv(boolean on ,int chaine ,int volume){
        this.on = on ;
        this.chaine=chaine;
        this.volume=volume;
    }

    public void allumer(){
        on=true ;
        Tabrecall = new int[4];
    }
    public void eteindre() {
        on=false;
    }
    public void setChaine(int ch) {
        chaine = ch ;
        setReCall ();
    }
    public void chaineUp(){
        if ( !allumer)
            return ;
        chaine +=1 ;
        if(chaine >9999)
            chaine=1 ;
        setReCall ();
    }
}

    public void chaineDown(){
        if ( !allumer)
            return ;
        chaine -=1 ;
        if(chaine <0)
            chaine=9999;
        setReCall ();
    }
    public void volumeUp(){
        if ( !allumer)
            return ;
        volume+=1 ;
        if(volume>10)
            volume=10 ;
    }
    public void volumeDown(){
        if ( !allumer)
            return ;
        volume-=1 ;
        if(volume<0)
            volume=0 ;
    }
}
    public void reCall (){
        for(int i=0;i<4;i++)
            System.out.println(Tabrecall[i];
    }
}
void setReCall (){
    for(int i=0;i<3;i++)
    {
        Tabrecall[i+1]=Tabrecall[i];
    }
    Tabrecall[0]=chaine;
}
}
```