

---

# PROGRAMMATION ORIENTÉE OBJET

## TD1 : Classes et Objets

2022-2023

---

### Exercice 1

Que fournit le programme suivant ?

```
class A
{ public A (int coeff)
{ nbre *= coeff ;
nbre += decal ;
}

public void affiche ()
{ System.out.println ("nbre==>" + nbre + "<==decal==>" + decal) ;
}
private int nbre = 20 ;
private int decal ;
}

public class InitChmp
{ public static void main (String args [])
{ A a = new A (5) ; a.affiche() ;
}
}
```

### Exercice 2

Pour la classe D définie comme suit :

```
class D {
public int x ;
public D() {x=3 ; } ;
public D( int a){this() ; x=x+a ;} ;
public D( int a, int b){this(b) ; x= x-a ;}}
```

Qu'affichera le code suivant ? Et Pourquoi ?

```
D a=new D(5,6) ;
System.out.println(a.x) ;
```

### Exercice 3

Pour la classe Interro définie comme suit :

```
class Interro {
public static int s;
public int j;
public Interro() {s++; j=s; }
}
```

qu'affichera le code suivant ? Expliquez les étapes que vous avez suivi.

```
Interro test=new Interro(); Interro test1=new Interro(); Interro test2= test;
System.out.println(test2.s + ? et ? + test2.j);
```

## Exercice 4

Que fournit le programme suivant ?

```
class Entier
{ public Entier (int nn)      { n = nn ; }
  public void incr (int dn) { n += dn ; }
  public void imprime ()     { System.out.println (n) ; }
  private int n ;
}
public class TstEnt
{ public static void main (String args[])
{ Entier n1 = new Entier (2) ; System.out.print ("n1==") ; n1.imprime() ;
Entier n2 = new Entier (5) ; System.out.print ("n1==") ; n2.imprime() ;
n1.incr(3) ;                      System.out.print ("n1==") ; n1.imprime() ;
System.out.println ("n1==n2_est" + (n1 == n2)) ;
n1 = n2 ; n2.incr(12) ;           System.out.print ("n2==") ; n2.imprime() ;
System.out.print ("n1==") ; n1.imprime() ;
System.out.println ("n1==n2_est" + (n1 == n2)) ;
}
}
```

## Exercice 5

Que fournit le programme suivant ?

```
class Util
{ public static void Échange (int a, int b) // ne pas oublier stat
{ System.out.println ("début_Échange:" + a + " " + b) ;
int c ;
c = a ; a = b ; b = c ;
System.out.println ("fin_Échange:" + a + " " + b) ;
}
}
public class Échange
{ public static void main (String args[])
{ int n = 10, p = 20 ;
System.out.println ("avant_appel:" + n + " " + p) ;
Util.Échange (n, p) ;
System.out.println ("apres_appel:" + n + " " + p) ; } }
```

## Exercice 6

Voici un programme complet utilisant la méthode permute :

```
class Point
{
  public Point(int abs, int ord)
  { x = abs ; y = ord ;
  }
  public void permute (Point a) // methode d'Échange les coordonnees
  // du point courant avec celles de a
  { Point c = new Point(0,0) ;
  c.x = a.x ; c.y = a.y ; // copie de a dans c
  a.x = x ; a.y = y ; // copie du point courant dans a
  x = c.x ; y = c.y ; // copie de c dans le point courant
  }
  public void affiche ()
  { System.out.println ("Coordonnees:" + x + " " + y) ;
  }
  private int x, y ;
}
public class Permute
{ public static void main (String args[])
{ Point a = new Point (1, 2) ;
Point b = new Point (5, 6) ;
a.affiche() ; b.affiche() ;
a.permute (b) ;
a.affiche() ; b.affiche() ; } }
```

Que va afficher ce programme ?

## Exercice 7

Soit la définition de classe suivante :

```
class A
{ public void f (int n, float x)      { ..... }
  public void f (float x1, float x2)  { ..... }
  public void f (float x, int n)      { ..... }
}
```

Avec ces déclarations :

```
A a ; short p ; int n1, n2 ; float x ;
```

Quelles sont les instructions correctes et, dans ce cas, quelles sont les méthodes appelées et les éventuelles conversions mises en jeu ?

```
a.f(n1, x) ;      a.f(x, n1) ;      a.f(p, x) ;      a.f(n1, n2) ;
```

## Exercice 8

Quels résultats fournit ce programme ?

```
class A
{ public void f(int n, float x)
{ System.out.println ("f(int_n, _float_x)_n=" + n + "x=" + x) ;
}
private void f(long q, double y)
{ System.out.println ("f(long_q, _double_y)_q=" + q + " _y=" + y) ;
}
public void f(double y1, double y2)
{ System.out.println ("f(double_y1, _double_y2)_y1=" + y1 + " _y2=" + y2) ;
}
public void g()
{ int n=1 ; long q=12 ; float x=1.5f ; double y = 2.5 ;
System.out.println ("——_dans_g_") ;
f(n, q) ;
f(q, n) ;
f(n, x) ;
f(n, y) ;
}
}
public class SurdfAcc
{ public static void main (String args[])
{ A a = new A() ;
a.g() ;
System.out.println ("——_dans_main") ;
int n=1 ; long q=12 ; float x=1.5f ; double y = 2.5 ;
a.f(n, q) ;
a.f(q, n) ;
a.f(n, x) ;
a.f(n, y) ;
}
}
```