

**Exercice 1.**

Q1) Quels résultats fournit le programme suivant ?

```
class A {
    public void affiche() {
        System.out.print("Je suis un A ");
    }
}

class B extends A {
}

class C extends A {
    public void affiche() {
        System.out.print("Je suis un C ");
    }
}

class D extends C {
    public void affiche() {
        System.out.print("Je suis un D ");
    }
}

class E extends B {
}

class F extends C {
}

class Poly {
    public static void main(String arg[]) {
        A a = new A(); a.affiche(); System.out.println();

        B b = new B(); b.affiche(); a = b; a.affiche(); System.out.println();

        C c = new C(); c.affiche(); a = c; a.affiche(); System.out.println();

        D d = new D(); d.affiche(); a = d; a.affiche(); c = d; c.affiche(); System.out.println();

        E e = new E(); e.affiche(); a = e; a.affiche(); b = e; b.affiche(); System.out.println();

        F f = new F(); f.affiche(); a = f; a.affiche(); c = f; c.affiche();
    }
}
```

Q2) Certaines possibilités d'affectation entre objets des types classes A, B, C,D, E et F ne figurent pas dans le programme ci-dessus. Pourquoi ?

**Exercice 2**

1) Réalisez la classe voiture avec un changement de vitesse, en y installant, tout d'abord, deux méthodes ne retournant rien, mais permettant, l'une d'incrémenter la vitesse, et l'autre de décrémenter la vitesse. Surchargez ensuite la méthode d'incrémentation de vitesse, en lui passant, en argument, le nombre de vitesses à incrémenter.

2) Réalisez le constructeur de la classe voiture, initialisant la vitesse à 0. Surchargez ce constructeur si l'on connaît la vitesse initiale.

## Exercice 3

Quels résultats fournit le programme suivant ?

```
class A {
    public void f(double x) {
        System.out.print("A.f(double=" + x + " ");
    }
}

class B extends A {
}

class C extends A {
    public void f(long q) {
        System.out.print("C.f(long=" + q + " ");
    }
}

class D extends C {
    public void f(int n) {
        System.out.print("D.f(int=" + n + " ");
    }
}

class F extends C {
    public void f(float x) {
        System.out.print("F.f(float=" + x + " ");
    }

    public void f(int n) {
        System.out.print("F.f(int=" + n + " ");
    }
}

class PolySur {
    public static void main(String arg[]) {
        byte bb = 1; short p = 2; int n = 3; long q = 4;
        float x = 5.f;          double y = 6.;

        System.out.println("*** A ** ");
        A a = new A(); a.f(bb); a.f(x);    System.out.println();

        System.out.println("*** B ** ");
        B b = new B(); b.f(bb); b.f(x); System.out.println();
        a = b; a.f(bb); a.f(x); System.out.println();

        System.out.println("*** C ** ");
        C c = new C(); c.f(bb); c.f(q); c.f(x); System.out.println();
        a = c; a.f(bb); a.f(q); a.f(x); System.out.println();

        System.out.println("*** D ** ");
        D d = new D(); d.f(bb); c.f(q); c.f(y); System.out.println();
        a = c; a.f(bb); a.f(q); a.f(y); System.out.println();

        System.out.println("*** F ** ");
        F f = new F(); f.f(bb); f.f(n); f.f(x); f.f(y); System.out.println();
        a = f; a.f(bb); a.f(n); a.f(x); a.f(y); System.out.println();
        c = f; c.f(bb); c.f(n); c.f(x); c.f(y);
    }
}
```

## Exercice 4

Indiquez si le code suivant est correct ou non. S'il ne l'est pas proposez une correction.

A.java

```
public class A {  
    public int i;  
    private int j;  
    void afficher(){  
        System.out.println(i+ j); }  
    public int getI() { return i;}  
    public int getJ() { return j;}  
    public void setI(int i) {this.i=i;}  
    public void setJ(int j) {this.j=j;}  
}
```

B.java

```
public class B extends A{  
    public int k;  
    void afficher() {  
        System.out.println(i+ j+k);}  
}
```

Test.java

```
public class Test {  
    public void main(String[] args) {  
        A a = new A();  
        a.i = 1;  
        a.j = 2;  
        a.afficher();  
        a = new B();  
        a.i = 3;  
        a.j = 4;  
        a.k = 5;  
        a.afficher();  
    }  
}
```

## Exercice 5

```
public class Parent {  
    int x;  
    Parent(int k) {x=k;}  
    int ajoute(int a) { return x+a; }
```

```

    public void moi() { System.out.println(" x = "+ x); }
}
public class Enfant1 extends Parent {
    int y;
    Enfant1 (int k, int l) { super(k); y=l; }
    int ajoute(int a) { return x+2*a;}
}
public class Enfant2 extends Enfant1 {
    int z ;
    Enfant2 (int k, int l, int m) { super(k, l); z= m; }
    int ajoute(int a) { return x+3*a;}
    public void moi() {
        super.moi();
        System.out.println(" z = "+ z);
    }
}
public class Essai{
    public static void main (String args[]) {
        int a =2;
        Parent p = new Parent(3);
        p.moi();
        System.out.println(" ajoute("+ a +" ) = "+ p.ajoute(a) );
        Enfant1 e1 = new Enfant1(3, 4);
        e1.moi();
        System.out.println(" ajoute("+ a +" ) = "+ e1.ajoute(a) );
        e1 = new Enfant2(3, 4, 5);
        e1.moi();
        System.out.println(" ajoute("+ a +" ) = "+ e1.ajoute(a) );
    }
}

```

1. Quels sont les attributs dont disposent les classes Enfant1 et Enfant2 ?
2. Ecrivez le résultat de l'exécution de la classe Essai.

## Exercice 6

Soit la déclaration de la méthode suivante :

```
public void test(int a) {}
```

Quelles sont les surcharges admises entre ces différentes possibilités ?

```
public void test() {}
```

```
public void test(double a) {}
```

```
public void test(int a, int b) {}
```

```
public int test(int a) {}
```

## Exercice 7

Les fichiers Java suivants ne compileront pas. Expliquez pourquoi.

### A.java

```
public class A {
    void A(int i) {
        System.out.println("Hello");
    }
    public static void main(String[] args) {
        A unA = new A(5);
    }
}
```

### Code 1 : Fichier Java : PrincipalTest.java

```
class Test {
    int a;
    Test (int b) {
        a = b;
    }
}

public class PrincipalTest {
    public static void main(String[] args) {
        Test unTest = new Test();
    }
}
```

### Code Java : fichier PrincipalTest.java

```
class Test {
    int a;
    int c;

    Test (int b) {
        a = b;
    }
    static int donneC() {
        return c;
    }
}

public class PrincipalTest {
    public static void main(String[] args) {
        Test unTest = new Test(5);
    }
}
```