

## Exercice 1

```
int i,j;
for ( i = 0;; i++) {
    for ( j = 0;; j++)
        if (j++ < 100) //lorsque j devient 1, on sort de la boucle
            break;
    System.out.printf("%d %d ", i, j);
    if (i == 3) //lorsque i devient 3, on sort de la boucle
        break;
}
```

Le code précédent est équivalent à :

```
int i,j = 1;
for ( i = 0; i <= 3; i++) {
    System.out.printf("%d %d ", i, j);
}
```

Donc la sortie est :

0 1 1 1 2 1 3 1

## Exercice 2

```
class A {
    public int x;

    public A() {
        f();
    }

    public void f() {
        x += 8;
    }
}

class B extends A {

    public int x;

    public void f() {
        x += 5;
    }
}

public class Test {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
    }
}
```

```

        System.out.println(a.x);
        //Affiche 8 : x est initialisé à 0 puis incrémenté par 8 (appel de f())

        System.out.println(b.x);
        //Affiche 5 : x est initialisé à 0 puis incrémenté par 5 (appel de f())

        a = new B();
        System.out.println(a.x);
        //Affiche 0 : x est initialisé à 0 (n'est pas modifié par la méthode f() de B)

        System.out.println(b.x);
        //Affiche 5 (la valeur de x n'est pas modifiée)
        a.f();
        b.f();
        System.out.println(a.x);
        //Affiche 0 : (x n'est pas modifiée par la méthode f() de B)

        System.out.println(b.x);
        //Affiche 10 : x est incrémentée par 5 (appel de f())
    }
}

```

### Exercice 3

```

import java.util.Scanner;
public class GestionGarage {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        Service[] tabServices = new Service[3];
        tabServices[0] = new Service("Lavage", 60, 30);
        tabServices[1] = new Service("Vidange", 15, 20);
        tabServices[2] = new Service("Batterie", 10, 10);

        System.out.println("Liste des services");
        for (int i = 0; i < 3; i++)
            System.out.println(tabServices[i]);
        System.out.println("Veuillez saisir votre choix :");

        String choix = clavier.nextLine();
        System.out.println(choix);
        for (Service service : tabServices)
            if (service.getNom().equals(choix))
                System.out.println(service);
    }
}
class Service {
    private String nom;
    private int prix, temps;
    public Service(String nom, int temps, int prix) {

```

```

        this.nom = nom;
        this.prix = prix;
        this.temps = temps;
    }
    public String toString() {
        return nom + " —> temps : " + temps + "min, prix : " + prix + "Dh";
    }
    public String getNom() {
        return nom;
    }
    public int getPrix() {
        return prix;
    }
    public int getTemps() {
        return temps;
    }
}

```

## Exercice 4

```

public class Personne {
    private String nomConnexion, motPasse;

    public Personne(String nomConnexion, String motPasse) {
        this.nomConnexion = nomConnexion;
        this.motPasse = motPasse;
    }

    public String getNomConnexion() {
        return nomConnexion;
    }

    public String getMotPasse() {
        return motPasse;
    }

    public void setNomConnexion(String nomConnexion) {
        this.nomConnexion = nomConnexion;
    }

    public void setMotPasse(String motPasse) {
        this.motPasse = motPasse;
    }
}

```

```

public abstract class MaterielInfo {
    private String type;
    static int numeroInventaire=100;
}

```

```

    public MaterielInfo(String type) {
        this.type = type;
        numeroInventaire++;
    }

    public String getType() {
        return type;
    }

    public abstract void configuration();
}

```

```

public class Reseau extends MaterielInfo {
    private String nom, motPasse;
    private int nbreInterfaces;

    public Reseau(String type, String nom, String motPasse,
        int nbreInterfaces) {
        super(type);
        this.nom = nom;
        this.motPasse = motPasse;
        this.nbreInterfaces = nbreInterfaces;
    }

    public void configuration() {
        System.out.println("Reseau");
    }
}

```

```

public class PC extends MaterielInfo {
    Personne administrateur;
    public PC(String type, Personne administrateur) {
        super(type);
        this.administrateur = administrateur;
    }

    public void configuration() {
        System.out.println("PC");
    }
}

```

```

import java.util.InputMismatchException;
import java.util.Scanner;

public class GestionMaterielInfo {
    public static void main(String[] args) {
        // pour la saisie a partir du clavier
        Scanner clavier = new Scanner(System.in);
        int n;
    }
}

```

```

MaterielInfo[] equipements = new MaterielInfo[4];
equipements[0] = new Reseau("Reseau","routeur","0000",2);
equipements[1] = new Reseau("Reseau","switch","0000",24);
equipements[2] = new PC("PC", new Personne("root","fsoAdmin"));
equipements[3] = new PC("PC", new Personne("root","fsoAdmin"));

for (MaterielInfo equipement : equipements)
    System.out.println(equipement.getType());

System.out.println("Saisir un entier compris entre 0 et 3");
try {
    n = clavier.nextInt();
    equipements[n].configuration();
} catch (InputMismatchException exp) {
    System.out.println("Vous n'avez pas saisi un entier");
} catch (ArrayIndexOutOfBoundsException exp) {
    System.out.println("Vous avez depasse le tableau");
}
}
}

```