

Questions :

1) Quel est le rôle de la liaison dynamique en Java ?

La liaison dynamique est la propriété de recherche permettant à la machine virtuelle de localiser la classe d'appartenance lors de l'appel d'une méthode d'instance.

2) Quel est la différence entre un Statement et un PreparedStatement ?

Si vous voulez exécuter un objet Statement plusieurs fois, le temps d'exécution sera réduit si vous utilisez plutôt un objet PreparedStatement.

L'objet PreparedStatement fournit une instruction SQL dès sa création contrairement à l'objet Statement. L'avantage est que l'instruction SQL sera directement envoyée au SGBD, où elle sera compilée.

3) Donnez trois raisons d'utiliser un tableau en JAVA ?

- Gestion optimisée de la mémoire
- Facilité la manipulation d'ensemble à partir de la référence du tableau.
- Effectuer des opérations d'ensemble sur les données.

Exercice 1 :

```
Public class SousChaine{
    Public boolean trouve (String chaine ,String souschaine){
        int cpt=1;
        Int c=chaine.indexOf(souschaine);
        While(c!=-1){
            c=chaine.indexOf(souschaine,c) ;
            cpt=cpt+1 ;
        }
        If(cpt>=20){
            return true ;
        }
        Else
            Return false;
    }
}
```

Exercice 2 :

```
package insertproduit;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class InsertProduit {

    public void insert (Compte [ ] comptes){

        try { Class.forName ("com.mysql.jdbc.Driver").newInstance() ;
            Connection con= DriverManager.getConnection ("jdbc:mysql
://127.0.0.1/Employee","root","") ;
```

```

        PreparedStatement p = con.prepareStatement
            ("insert into comptes values(?,?,?,?)") ;
        for(int i=0;i<comptes.length;i++){
            p.setString(1,comptes[i].getPrenom()) ;
            p.setString (2,comptes[i].getNom()) ;
            p.setInt (3,comptes[i].getAge()) ;
            p.setInt (4,comptes[i].getAge()) ;

            p.executeUpdate ( ) ;
        }

        p.close () ;
        con.close ();
    }

    catch (Exception er) { er.printStackTrace ( ) ; }

}

}

```