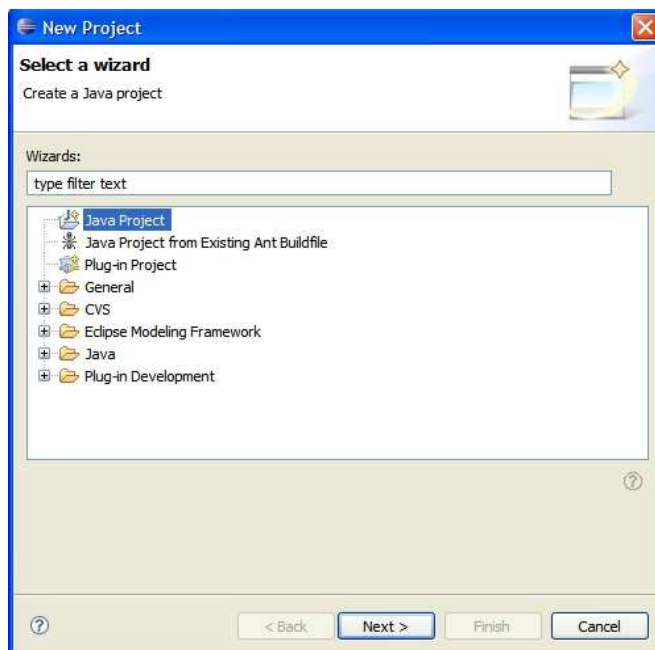


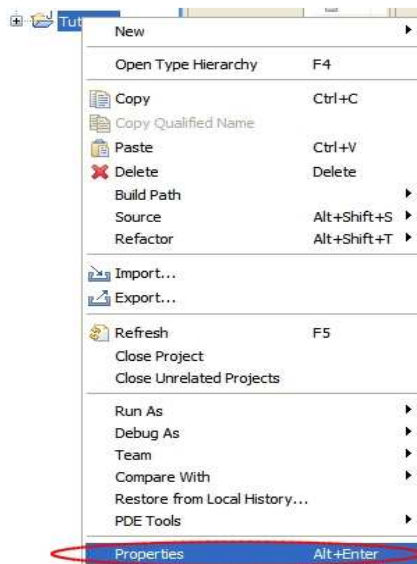
```
/*  
 * Tutorial de Connection à MYSQL en Java, via un driver JDBC  
 *  
 * Auteur : Kartono Loïc  
 *  
 */
```

Avant de se lancer comme des « bourrins » (lol) dans la programmation pure et dure, je vous propose dans un premier temps de commencer par télécharger le driver JDBC, à cette adresse : <http://codeur.invfoteam.com/download/mysqlJDBC.zip>

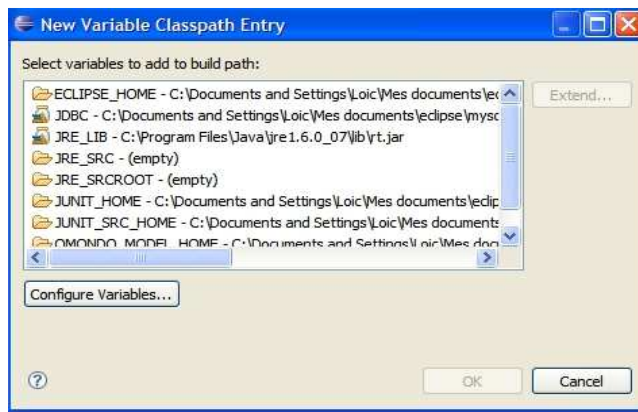
Dans Eclipse, nous allons créer un nouveau projet :



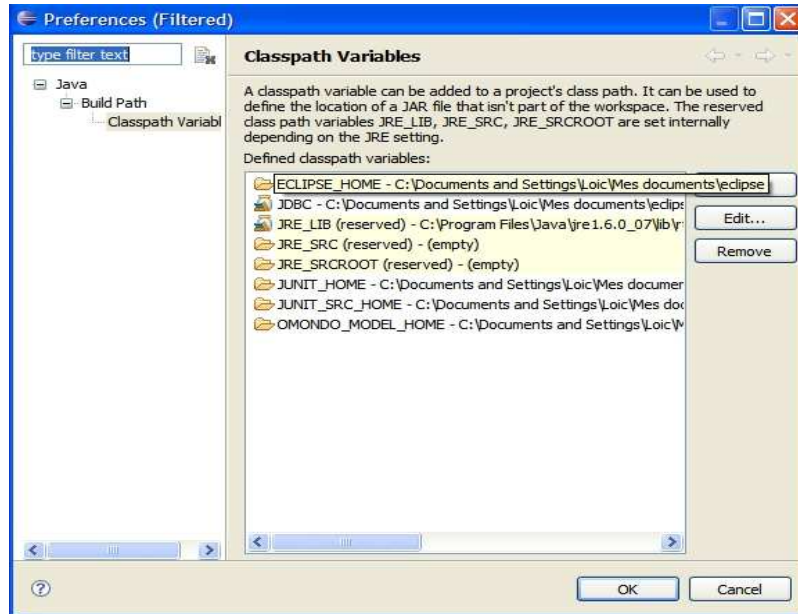
Il faut maintenant inclure notre driver JDBC (que vous avez normalement téléchargé précédemment), au CLASSPATH de notre nouveau projet. Pour voir la démarche à suivre, c'est par ici :



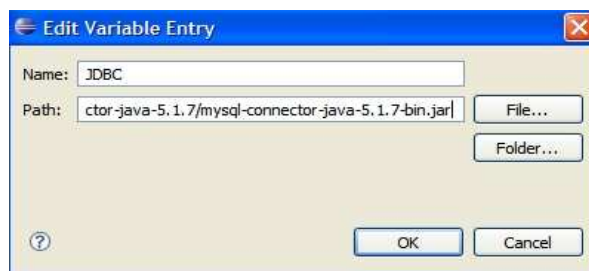
Après avoir cliqué sur **Properties**, une fenêtre apparaît. Dans la barre latérale gauche de cette fenêtre, cliquez sur **Java Build Path**. Cliquez ensuite sur le bouton **Add variables**. La fenêtre suivante devrait apparaître :



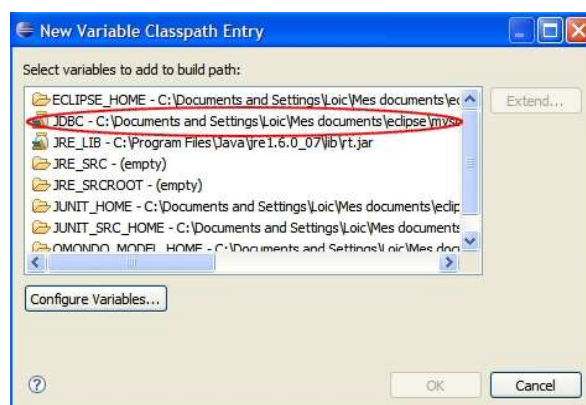
Pour ajouter une nouvelle entrée à la liste des variables intégré à Eclipse, il faut cliquer sur le bouton **Configure variables**. La fenêtre suivante apparaît :



Cliquez ensuite sur le bouton **new**. Dans le champ **Name**, entrez le nom que vous voulez pour désigner votre nouvelle variable (Je l'appellerai ici **JDBC**). Cliquez ensuite sur le bouton **File**, et allez sélectionner votre driver JDBC (\*.jar) :



Cette action nous a permis d'ajouter une variable externe à Eclipse. Maintenant, nous allons l'intégrer à notre Projet. Si les choses, on correctement été faites, vous devriez voir apparaître votre variable dans la fenêtre suivante :



Double cliquez sur cette dernière et valider. La variable est maintenant intégrée au projet.

Ha ! Voilà enfin la partie qui nous intéresse. Pour se connecter à MySQL en java c'est très simple :

```
/*
 * Toutes les classes de JDBC sont dans le package java.sql. Il faut donc l'importer dans
 * tous les programmes utilisant JDBC.
 */
import java.sql.*;

public class TutoSQL {

    /**
     * @param args
     */
    public static void main(String[] args){
        /* Il y a 4 classes importante à connaître : DriverManager, Connection, ResultSet
         * et Statement (et PreparedStatement)
         * Déclaration des variables utiles à la Connection
         */
        Connection laConnection;
        Statement transmission;
        ResultSet leResultat;

        /* Ici, obligation d'implémenter une structure de type try...catch pour nous
         * permettre de gérer les erreurs de Connection
         */
        try{
            // Pour se connecter à une base de donnée, il faut charger le driver
            // qui fait le lien entre le programme et la base de donnée
            Class.forName("com.mysql.jdbc.Driver");
            /* On charge et on configure le driver de la base de donnée à l'aide de la
             * classe DriverManager. On insert les informations de Connection ici.
             * sql = le nom de ma base
             * root = l'utilisateur par défaut
             * "" = le mot de passe (ici j'en ai pas mis)
             */
            laConnection =DriverManager.getConnection("jdbc:mysql://localhost/sql","root","");
            // transmission contiendra nos requêtes et les transmettra à la base de données
            transmission = laConnection.createStatement();
            /* Exécution d'une requête : le ResultSet nous permet de parcourir les
             * informations retournées par la base de données dans le cas d'une sélection
             * de données
             */
            leResultat = transmission.executeQuery("select * from maTable");
            /* Pour afficher le resultat de la requête renvoyé par MySQL, il suffit de
             * bouclé sur notre ResultSet.
             *
             * leResultat.getString("nom") → nom = champs de la table maTable
             */
            while(leResultat.next()){
                System.out.println("Nom : " + leResultat.getString("nom") + "Age : " +
                    leResultat.getInt("age"));
            }
        }catch(Exception e){
            System.out.print("Impossible de se connecter à la base de données");
        }
    }
}
```

Dans ce tutorial, vous avez appris à importer une variable au CLASSPATH d'un projet, vous connectez à une base de données MySQL et à récupérer le resultat d'une requête pour ensuite l'afficher.

Merci à toutes et à tous, et à un prochain tuto...

