



# ISI5

## Développement d'interfaces Homme-Machine

**Manuele Kirsch Pinheiro**

Maître de conférences en Informatique  
Centre de Recherche en Informatique  
Université Paris 1 – Panthéon Sorbonne

[Manuele.Kirsch-Pinheiro@univ-paris1.fr](mailto:Manuele.Kirsch-Pinheiro@univ-paris1.fr)

<http://mkirschp.free.fr>



## Présentation

- **Contenu prévisionnel**
  - Architecture en 3 couches (3-tier)
  - JDBC
  - Déploiement des applications

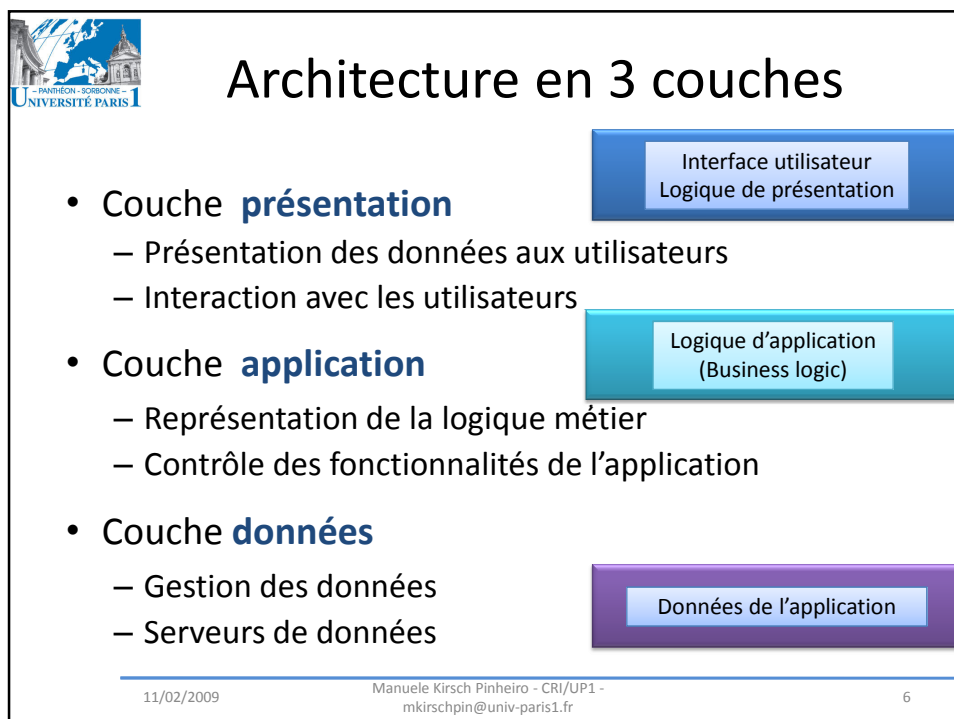
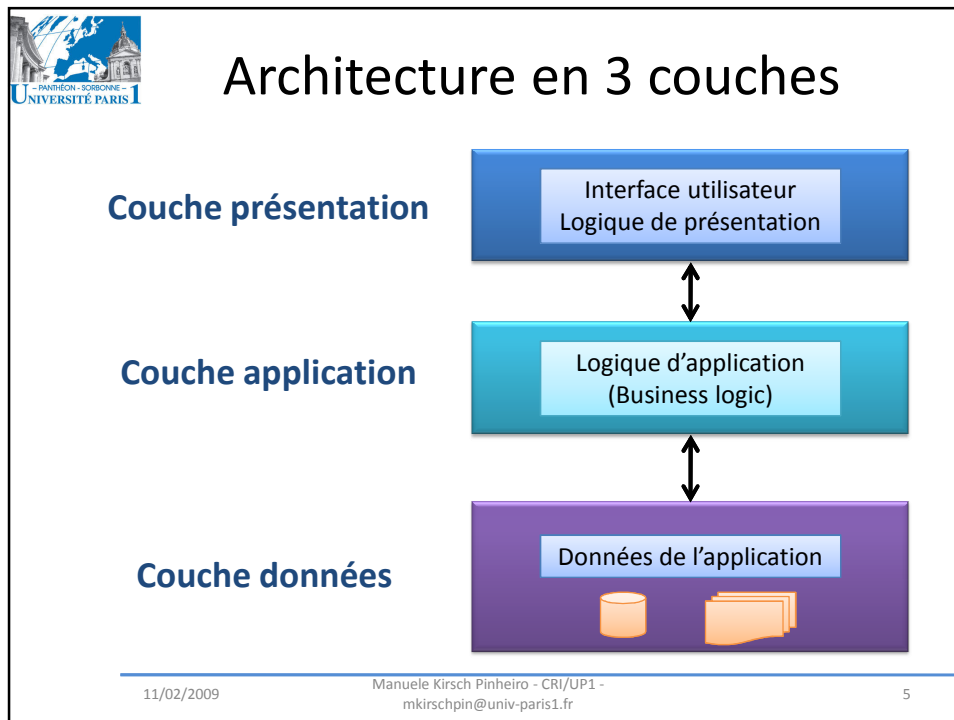



## Architecture en 3 couches



## Architecture en 3 couches

- Termes liés
  - 3-tier,  $n$ -tier, multi-tier architecture
- **Objectif**
  - Une réelle séparation entre l'interface utilisateur, la logique de l'application (métier) et les données
- **Structure**
  - Couche **présentation** : interface
  - Couche **application** : logique métier
  - Couche **données** : données de l'application






## Architecture en 3 couches

- **Avantages**
  - Modularité
  - Indépendance
    - Chaque couche peut évoluer et être modifiée sans que cette évolution affecte les autres couches
    - Changements technologiques sont possibles
- **Usage**
  - Développement d'applications réparties
  - Programmation Client/Server

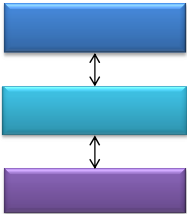
---

11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr 7

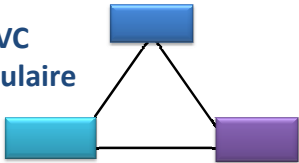


## Architecture en 3 couches

- **Comparaison avec le modèle MVC**
  - Extension naturelle, même principe : division des responsabilités
  - La couche de présentation ne communique jamais avec la couche des données




**3-tier  
Linéaire**



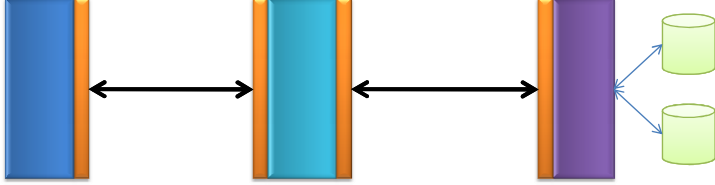
**MVC  
Triangulaire**

---


11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr 8

 **Architecture en 3 couches**


- Accès aux couches
  - Chaque couche doit être isolée des autres
  - Usage des design patterns
    - Facade



11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr 9

 **Connexion aux bases de données  
JDBC**


11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr 10



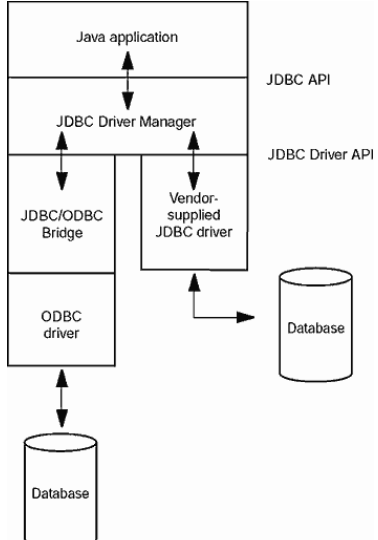
# JDBC

- API d'accès aux bases de données
  - Paquetage **java.sql.\***
  - Connexion à une base de données
  - Manipulation d'une base de données
    - *Query*
    - *Update*
- Historique
  - Première version en 1996 (J2SE 1.1)
  - JDBC 2 avec J2SE 1.2 en 1998
  - JDBC 4 avec J2SE 6

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
11

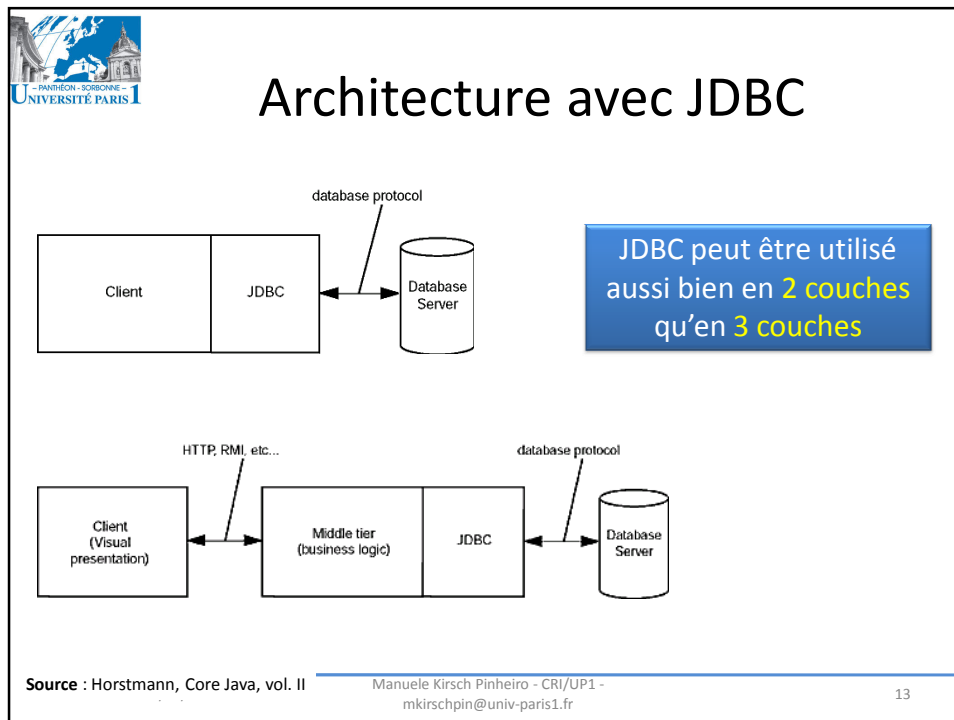


# JDBC



- Les applications accèdent à la BD par un **driver (JDBC Driver)**
  - Apache Derby : derbyclient.jar
  - PostgreSQL : postgresql-8.3-604.jdbc3.jar
- Les drivers sont proposés par les **fournisseurs** du SGBD
- Ces drivers ne sont pas forcément 100% pure Java

Source : Horstmann, Core Java, vol. II
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
12




## Connexion JDBC

- Avant tout, il faut avoir une base de données disponible... 😊
  - Exemple :
    - Base de données « Dictionnaire » sur Apache Derby
    - Sous NetBeans : onglet Services → JavaDB → Create...

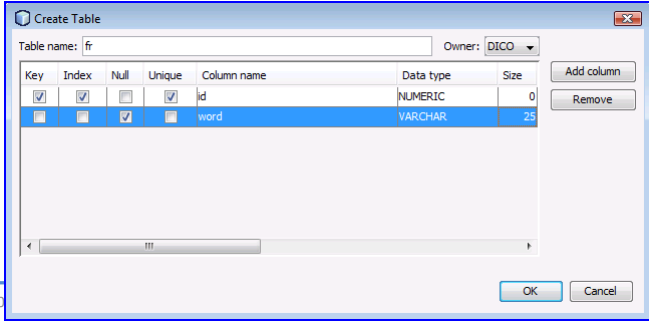
11/0

14




## Connexion JDBC

- Création d'une table sous NetBeans
  - Onglet Services
  - Database connection → Connect
  - Database connection → Table, puis Create Table

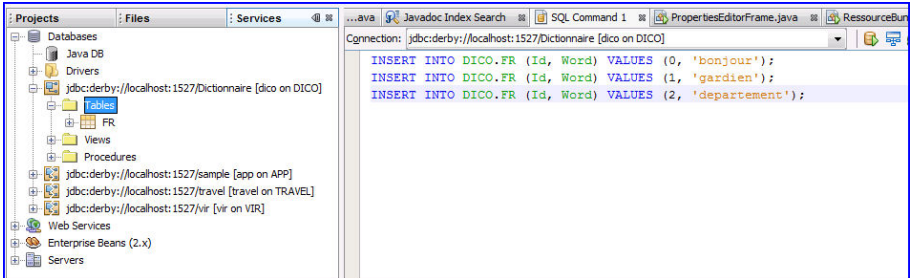


11/0 15




## Connexion JDBC

- Insertion de quelques données sous NetBeans
  - Onglet Services
  - Database connection → Connect
  - Database connection → Table, puis Execute Command



11/0 15





## Connexion JDBC

- Connexion à la BD passe par un URL JDBC
  - *jdbc:driver:other stuff*
- Paramètres de connexion propres au driver et à la base

jdbc:derby://localhost:1527/Dictionnaire

↓  
**Driver**


↓  
**DB Server**

↓  
**Port**

↓  
**DataBase**

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr
17



## Connexion JDBC


- Connexion à travers la classe **DriverManager**
  - Chargement dynamique du driver
  - Entrée des paramètres
  - Création de la connexion → classe **Connection**

```

Class.forName(driver);
Connection conn =
  DriverManager.getConnection(bdd, login, pass);
  
```

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr
18



## Connexion JDBC

- Connexion
  - Charge
  - Entrée
  - Création


```

//on peut lire ces informations sur des propriétés
String driver = "org.apache.derby.jdbc.ClientDriver";
String bdd = "jdbc:derby://localhost:1527/Dictionnaire";
String login = "dico";
String pass = "dico";
Connection conn = null;

try {
    //Charge le pilote
    Class.forName(driver);
    //Crée la connexion
    conn = DriverManager.getConnection(bdd, login, pass);
    ...
} catch (ClassNotFoundException cnfe) { ... }
} catch (SQLException sqle) { ... }

```

11/02/2009



## Connexion JDBC

- Toute connexion ouverte doit être fermée !
  - `conn.close ()`
  - Usage de la clause *finally* est suggérée


```

try {
    Class.forName(driver);
    conn = DriverManager.getConnection(bdd, login, pass);
    ...
} catch (ClassNotFoundException cnfe) { ... }
} catch (SQLException sqle) { ... }
} finally {
    if (conn != null)
        conn.close();
}

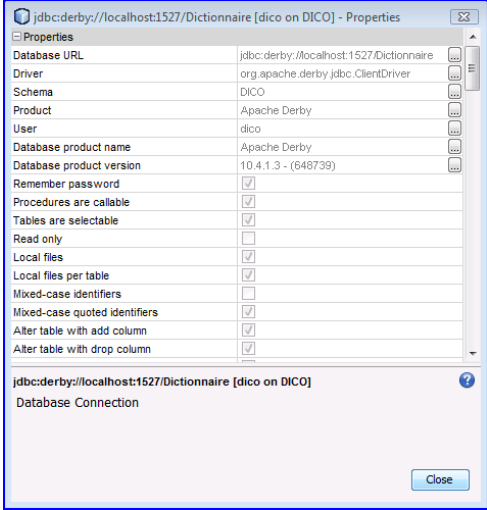
```

11/02/2009


mkirschpin@univ-paris1.fr

 **Connexion JDBC**

- Sous NetBeans, on peut obtenir les paramètres de connexion
  - Onglet Services, Database connection, puis Proprieties



11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr 21

 **Connexion JDBC**

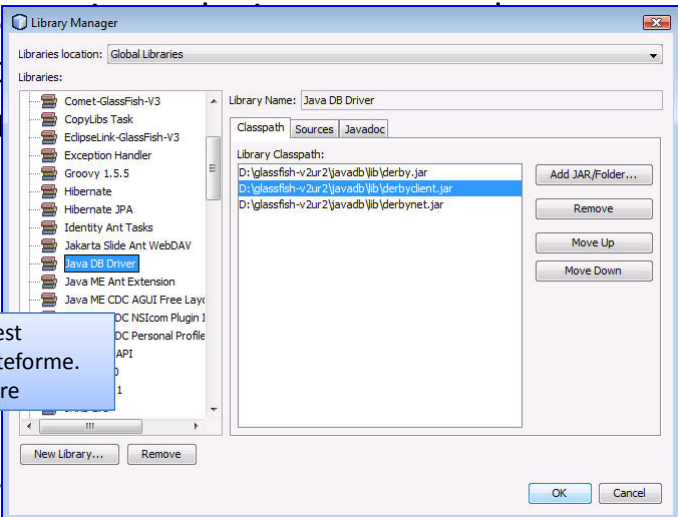
- On doit garantir que le .jar correspondant au driver est accessible
  - Classpath
  - Projet

11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr 22

## Connexion JDBC

- On doit garantir que le driver est disponible dans la plateforme
  - Classpath
  - Projet

On vérifie si le driver est disponible dans la plateforme.  
On l'ajoute si nécessaire



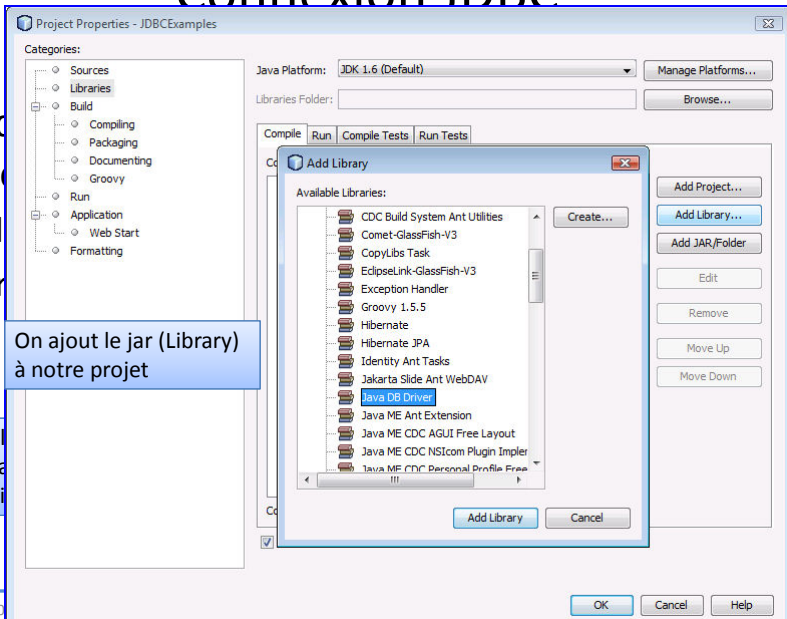
11/02/2009

## Connexion JDBC

- On doit garantir que le driver est disponible dans la plateforme
  - Classpath
  - Projet

On ajoute le jar (Library) à notre projet

On vérifie si le driver est disponible dans la plateforme.  
On l'ajoute si nécessaire



11/02/2009




## Manipulation BD

- La manipulation de la base de données se fait à travers la classe **Statement**
  - Statement statement = conn.**getStatement**()
  - Une query SELECT
    - ResultSet rs = statement.**executeQuery** (sql)
  - Un update (INSERT, UPDATE...)
    - int rs = statement.**executeUpdate** (sql)
  - Générique
    - boolean rs = statement.**execute** (sql)



## Manipulation BD

- Les résultats d'une requête SELECT sont accessibles par un **ResultSet**
  - resultSet.**next**() :
    - avance sur l'ensemble des n-uplets
  - **getString** (colName) / getString(n) :
    - récupère la valeur sur la colonne '*colName*' ou la *n*-ème colonne
  - Autres types de données
    - getInt, getFloat, getByte, getDate, getObject...




## Manipulation BD

- Les résultats d'une requête SELECT sont accessibles par un **ResultSet**
  - resultSet.**next()** :
    - avance sur l'ensemble des n-uplets
  - **getString** (colName) :
    - récupère la valeur de la colonne
  - Autres types de données :
    - **getInt, getFloat, getDouble, getBoolean, getDate, getTime**

```
String sql = "SELECT * FROM TABLE";
try {
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(sql);

    while (rs.next()) {
        String col = rs.getString("Colonne");
        ...
    }
} catch (SQLException ex) { ... }
```

11/02/2009



## Manipulation BD


- L'exécution d'une requête INSERT, UPDATE, DELETE... indique le nombre de lignes (n-uplets) affectées
  - int rs = statement.**executeUpdate** (sql)

```
String pattern = "INSERT INTO TABLE (Col1, Col2...) VALUES (val1, val2... )";
try {
    Statement stat = conn.createStatement();
    rows = stat.executeUpdate(sql);
    ...
} catch (SQLException ex) { ... }
```

11/02/2009

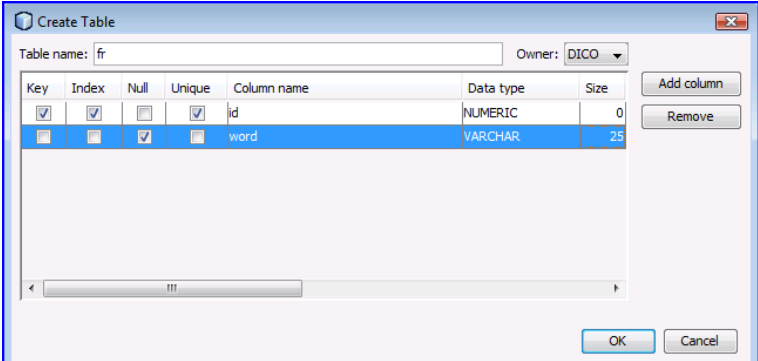
Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr

28




## Exemple

- Base de données **Dictionnaire** sous Apache Derby
- Table **FR (Id, Word)**
  - int id
  - String (VarChar) Word



29



## Exemple

- Classe **DicoBase** : connexion et accès à la BD


```

public boolean connectDB() {
    boolean ok = false;
    try {
        //Charge le pilote
        Class.forName(driver);
        //Crée la connexion
        conn = DriverManager.getConnection(bdd, login, pass);
        System.out.println("Connexion à la BDD");
        ok = true;
    } catch (ClassNotFoundException cnfe) {
        Logger.getLogger(DicoBase.class.getName()).
            log(Level.SEVERE, "Driver inconnu ", cnfe);
    } catch (SQLException sqle) {
        Logger.getLogger(DicoBase.class.getName()).
            log(Level.SEVERE, "Connexion au driver impossible", sqle);
    }
    return (ok);
}

```

Connexion

11/02/2009      Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr      30



## Exemple

- Classe **DicoBase** : connexion et accès à la BD

```

public boolean connectDB() {
    boolean ok = false;
    try {
        //Charge le pilote
        Class.forName(driver);
    }
}

public void closeDB() {
    if (conn != null) {
        try {
            conn.close();
            System.out.println("Cloture de la connexion à la BDD");
        } catch (SQLException ex) {
            Logger.getLogger(DicoBase.class.getName()).
                log(Level.SEVERE, "Erreur de cloture: ", ex);
        }
    }
}


```

Connexion

Déconnexion

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
31



## Exemple

- Classe **DicoBase** : connexion et accès à la BD

```

public ArrayList<String> getWords() {
    ArrayList<String> words = new ArrayList();
    String sql = "SELECT * FROM FR";

    try {
        Statement stat = conn.createStatement();
        ResultSet rs = stat.executeQuery(sql);

        while (rs.next()) {
            String aWord = rs.getString("Word");
            words.add(aWord);
        }
    } catch (SQLException ex) {
        Logger.getLogger(DicoBase.class.getName()).log(Level.SEVERE, null, ex);
    }
    return words;
}


```

Select

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
32





## Exemple

- Classe **DicoBase** : connexion et accès à la BD

```

public int getLastId () {
    int lastId = -1;
    String sql = "SELECT Id FROM FR";

    try {
        Statement stat = conn.createStatement();
        ResultSet rs = stat.executeQuery(sql);

        while (rs.next()) {
            int id = rs.getInt("Id");
            if (id>lastId)
                lastId = id;
        }

    } catch (SQLException ex) {
        Logger.getLogger(DicoBase.class.getName()).log(Level.SEVERE, null, ex);
    }
    return lastId;
}


```

Select

ct

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
33



## Exemple

- Classe **DicoBase** : connexion et accès à la BD

```

public int insertWord (int id, String word) {
    int rows = -1;
    String pattern = "INSERT INTO FR (Id, Word) VALUES ({0}, '{1}')";
    String sql = MessageFormat.format(pattern, id, word);
    System.out.println (sql);


    try {
        Statement stat = conn.createStatement();
        rows = stat.executeUpdate(sql);
    } catch (SQLException ex) {
        Logger.getLogger(DicoBase.class.getName()).log(Level.SEVERE, null, ex);
    }
    return rows;
}

```

Insert

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
34



## Exemple

- Classe Main

Connexion à la BDD  
 bonjour  
 gardien  
 departement  
 INSERT INTO FR (Id, Word) VALUES (3, 'salut')  
 bonjour  
 gardien  
 departement  
 salut  
 Cloture de la connexion à la BDD

```

public class Main {
    /**...*/
    public static void main(String[] args) {
        DicoBase dico = new DicoBase();
        dico.connectDB();

        ArrayList<String> words = dico.getWords();
        for (String aWord : words) {
            System.out.println (aWord);
        }


        int newone = dico.getLastId();
        dico.insertWord(newone+1, "salut");

        words = dico.getWords();
        for (String aWord : words) {
            System.out.println (aWord);
        }

        dico.closeDB();
    }
}

```

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr
35



## Déploiement des applications

- Une fois l'application terminée, elle doit être déployée chez le client
- Le déploiement doit être le plus facile et pratique possible
- Méthodes :
  - Usage des fichier Jar
  - Usage des ressources

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr
36



## Fichiers Jar

- Les fichiers .jar permettent le déploiement sur un fichier unique
- Fichier Jar
  - Compression Zip
  - Structure répertoire
    - Fichier class
    - Ressources : audio, images, textes
    - Librairies
  - Manifeste
    - Description du contenu du Jar

11/02/2009

Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr

37




## Commande jar

- Format général :
  - jar cvf JarFile.jar file1 file2 ...
  - jar cvf JarFile.jar dir
- Options :
  - m        manifeste
  - t        affiche la table de matières
  - u        met à jour le Jar
  - jar -cf JarFileName.jar -m manifest.mf ...

11/02/2009

Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr

38



## Exemple

- JarTest
- ihmexamples
- tempconverter
- converterawt
- converterswing

```

jar cvf ihmexamples.jar ihmexamples Resources*.properties


```

```

C:\Users\kirsch\Documents\NetBeansProjects\JarTest>jar cvf ihmexamples.jar ihmexamples Resources*.properties
manifest ajout
ajout : ihmexamples/ (entrée = 0) (sortie = 0) (0% stocké)
ajout : ihmexamples/tempconverter/ (entrée = 0) (sortie = 0) (0% stocké)
ajout : ihmexamples/tempconverter/CelciusConverter.class (entrée = 990) (sortie = 533) (46% compressés)
ajout : ihmexamples/tempconverter/converterawt/ (entrée = 0) (sortie = 0) (0% stocké)
ajout : ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$1.class (entrée = 968) (sortie = 439) (54% compressés)
ajout : ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2.class (entrée = 995) (sortie = 460) (53% compressés)
ajout : ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$3.class (entrée = 995) (sortie = 458) (53% compressés)
ajout : ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$4.class (entrée = 750) (sortie = 413) (44% compressés)
ajout : ihmexamples/tempconverter/converterawt/TempConverterAwTGUI.class (entrée = 4504) (sortie = 2045) (54% compressés)

```

11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - 39  
mkirschpin@univ-paris1.fr



## Exemple

- JarTest
- ihmexamples
- tempconverter
- converterawt
- converterswing

```


jar tvf ihmexamples.jar ihmexamples Resources*.properties

```

```

C:\Users\kirsch\Documents\NetBeansProjects\JarTest>jar -tvf ihmexamples.jar
Manifest-Version: 1.0
Created-By: 1.6.0_18-b02
Main-Class: ihmexamples/tempconverter/converterawt/TempConverterAwTGUI
Classes:
  ihmexamples/
  ihmexamples/tempconverter/
  ihmexamples/tempconverter/CelciusConverter.class
  ihmexamples/tempconverter/converterawt/
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$1.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$3.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$4.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUI.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUIv2$1.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUIv2$2.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUIv2$3.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUIv2$4.class
  ihmexamples/tempconverter/converterawt/TempConverterAwTGUIv2.class
  ihmexamples/tempconverter/converterswing/
  ihmexamples/tempconverter/converterswing/TempConverterSwingGUIv2$1.class
  ihmexamples/tempconverter/converterswing/TempConverterSwingGUIv2$2.class
  ihmexamples/tempconverter/converterswing/TempConverterSwingGUIv2$3.class
  ihmexamples/tempconverter/converterswing/TempConverterSwingGUIv2$4.class
  ihmexamples/tempconverter/converterswing/TempConverterSwingGUIv2.class
  ihmexamples/tempconverter/temperatureConverter.class
  ihmexamples/tempconverter/textInterface.class
  Resources.properties
  Resources_en_US.properties
  Resources_fr_FR.properties

```




# Manifeste

- Manifeste décrit les caractéristiques du .jar
- **MANIFEST.MF**
  - META-INF/MANIFEST.MF
- Entrées sur la forme « **header: value** »
  - Manifest-Version:** 1.0
  - Main-Class:** ClassePrincipale
  - Class-Path:** dir-name/lib1.jar dir-name/lib2.jar
  - Name:** aPackage
  - Specification-Title:** PackageTitle
  - Specification-Version:** 1.2
  - Implementation-Title:** myappli.mypackage
  - Implementation-Vendor:** MyCompany

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
41



# Exemple


**META-INF\MANIFEST.MF**

Manifest-Version: 1.0  
 Main-Class: ihmexamples.tempconverter.converterswing.TempConverterSwingGUIv2  
 Name: ihmexamples/tempconverter  
 Specification-Title: TemperatureConverter  
 Specification-Version: 1.2  
 Specification-Vendor: Kirsch, co  
 Implementation-Title: ihmexamples.tempconverter  
 Implementation-Version: build57  
 Implementation-Vendor: Kirsch, co

Attention à tous les détails !!  
 Pas d'espace en trop, pas de tabulation

---

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
42




## Exemple

```
jar mcf META-INF\MANIFEST.MF ihmexamples.jar ihmexamples Resources*.properties
```

- JarTest
  - ihmexamples
    - tempconverter
      - converterawt
      - converterswing
  - META-INF

11/02/2009
Manuele Kirsch Pinheiro - CRI/UP1 -  
mkirschpin@univ-paris1.fr
43




## Exemple

```
jar mcf META-INF\MANIFEST.MF ihmexamples.jar ihmexamples Resources*.properties
```

- JarTest

```
C:\Users\kirsch\Documents\NetBeansProjects\JarTest>jar tvf ihmexamples.jar
0 Tue Feb 10 23:44:58 CET 2009 META-INF/
407 Tue Feb 10 23:44:58 CET 2009 META-INF\MANIFEST.MF
0 Tue Feb 10 21:43:28 CET 2009 ihmexamples/
990 Tue Jan 01 00:00:00 CET 1980 ihmexamples/tempconverter/CelsiusConverter.class
0 Tue Feb 10 21:43:28 CET 2009 ihmexamples/tempconverter/converterawt/
968 Fri Jan 16 17:44:46 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$1.class
995 Fri Jan 16 17:44:46 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2.class
995 Fri Jan 16 17:44:46 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$3.class
750 Fri Jan 16 17:44:46 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$4.class
4504 Fri Jan 16 17:44:46 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI.class
982 Fri Jan 16 00:26:48 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2$1.class
1009 Fri Jan 16 00:26:48 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2$2.class
1009 Fri Jan 16 00:26:48 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2$3.class
760 Fri Jan 16 00:26:48 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2$4.class
4236 Fri Jan 16 00:26:48 CET 2009 ihmexamples/tempconverter/converterawt/TempConverterAwTGUI$2.class
0 Tue Feb 10 21:43:28 CET 2009 ihmexamples/tempconverter/converterswing/
1008 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2$1.class
1008 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2$2.class
1035 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2$3.class
1035 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2$4.class
778 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2$5.class
4401 Fri Feb 06 20:31:18 CET 2009 ihmexamples/tempconverter/converterswing/TempConverterSwingGUI$2.class
759 Tue Feb 10 21:42:54 CET 2009 ihmexamples/tempconverter/TemperatureConverter.class
1275 Fri Feb 06 21:03:50 CET 2009 ihmexamples/tempconverter/TextInterface.class
134 Tue Feb 10 11:21:30 CET 2009 Resources.properties
123 Tue Feb 10 10:39:20 CET 2009 Resources_en_US.properties
135 Tue Feb 10 10:39:40 CET 2009 Resources_fr_FR.properties
```

mkirschpin@univ-paris1.fr

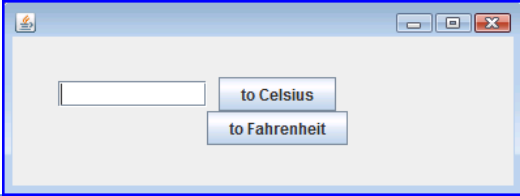
 UNIVERSITÉ PARIS 1

## Jar exécutable


- Exécution d'un jar
  - Header « **Main-Class** » dans le **MANIFEST.MF**

```
java -jar ihmexamples.jar
```

```
ihmexamples.temconverter.converterswing.TempConverterSwingGUIv2
```



11/02/2009 Manuele Kirsch Pinheiro - CRI/UP1 - mkirschpin@univ-paris1.fr 45

 UNIVERSITÉ PARIS 1

## Exercices



## Exercices

- 1) Créer une base de données Dictionnaire
- 2) Implémenter une application permettant l'insertion des nouveaux mots dans le dictionnaire
  - Modèle MVC
    - Vue : interface utilisateur indépendante
    - Contrôle : gestion interaction, écouteur modèle
    - Modèle : *facade* pour l'accès à la BD, événements newWord
- 3) Jeu de scrabble (optionnel) :
  - Intégration de la BD au jeu