

# Gestion de comptes bancaires

Stéphanie Pok - Zouhour  
Boughammoura

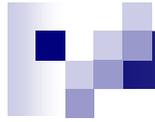


# - SOMMAIRE -

- Cahier des charges

- Solution proposée

- Simulation



# CAHIER DES CHARGES

Stéphanie Pok - Zouhour  
Boughammoura



# Objectifs

- Création d'un programme convivial qui permet la gestion de comptes bancaires
- Ce programme doit être relié à une base de données afin de stocker les informations nécessaires au programme



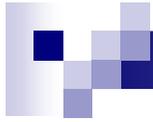
# Catégories d'utilisateurs

- Mode banque : destiné au personnel de la banque
- Mode client : simule un Distributeur Automatique de Billets (DAB)



# Outils de développement

- Le langage utilisé doit être le Java.
- Nous avons choisi MySql comme base de données pour stocker les données.



# **SOLUTION PROPOSEE**

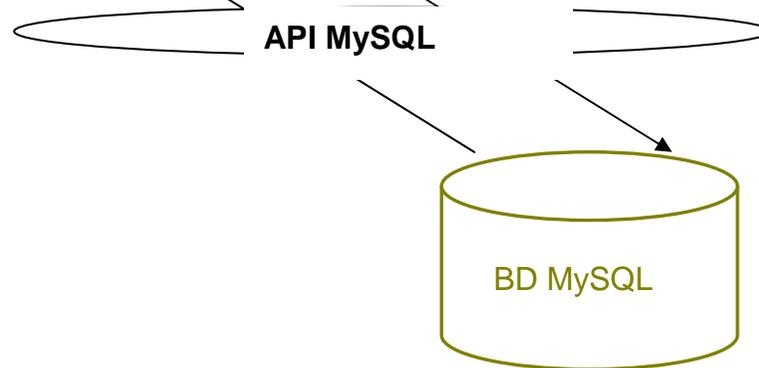
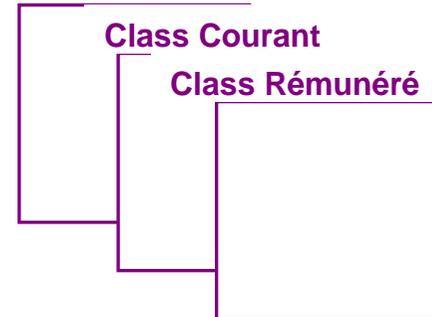
Stéphanie Pok - Zouhour  
Boughammoura

# Structure générale de l'application

Programme principal Java



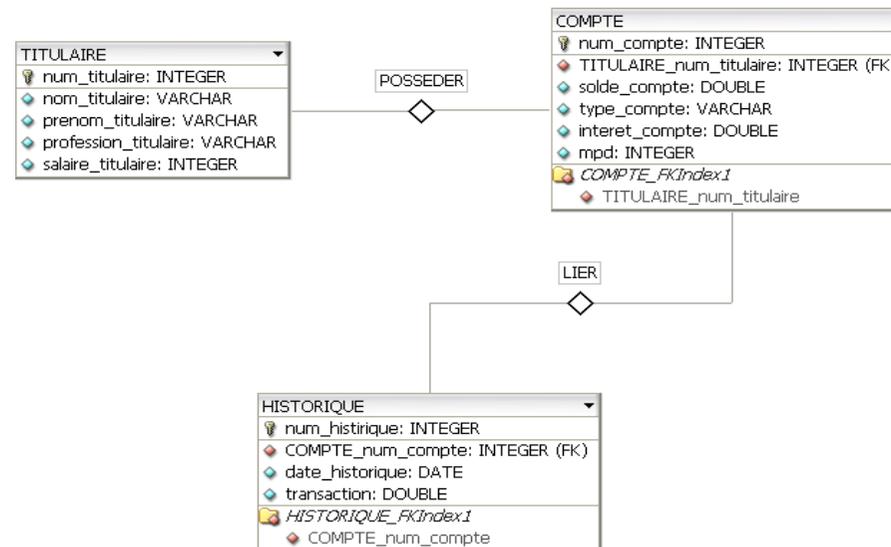
Class Client



Stéphanie Pok - Zouhour  
Boughammoura

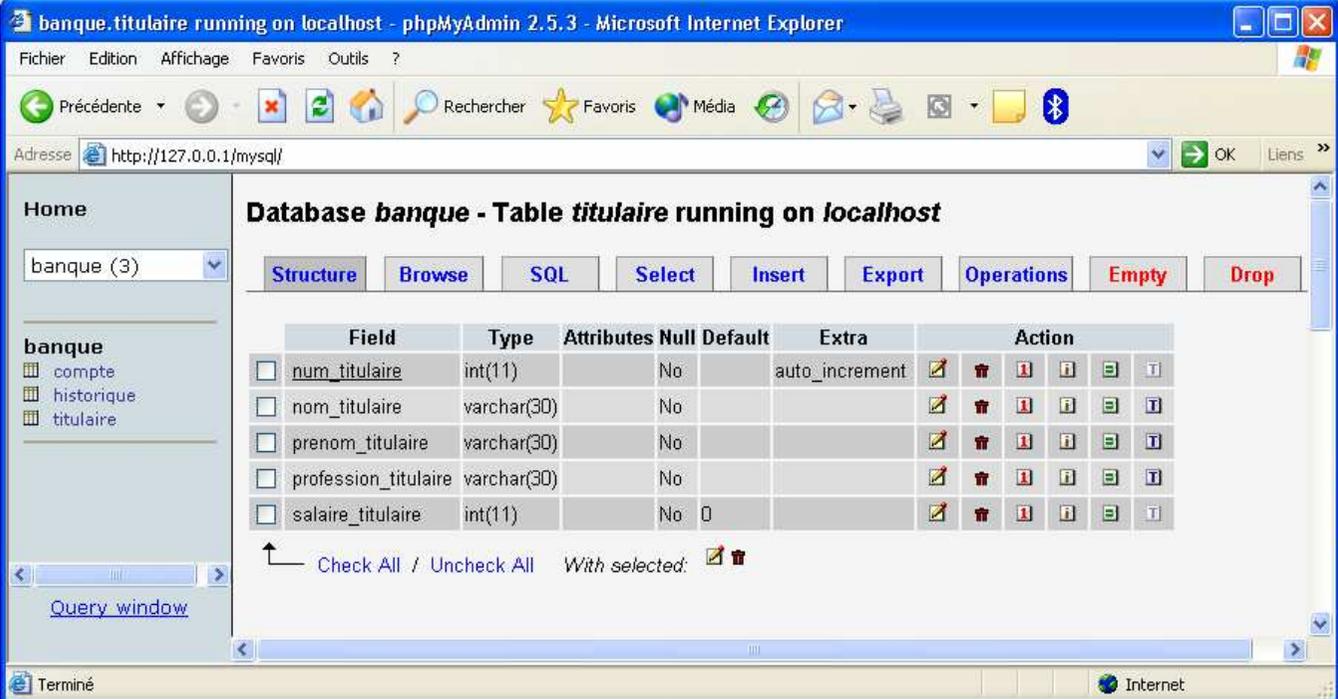
# Modèle Logique de Données

## ■ La base de données MySql : BANQUE



# La base BANQUE (1)

## ■ Table titulaire



The screenshot shows the phpMyAdmin interface for a database named 'banque'. The 'titulaire' table structure is displayed with the following fields:

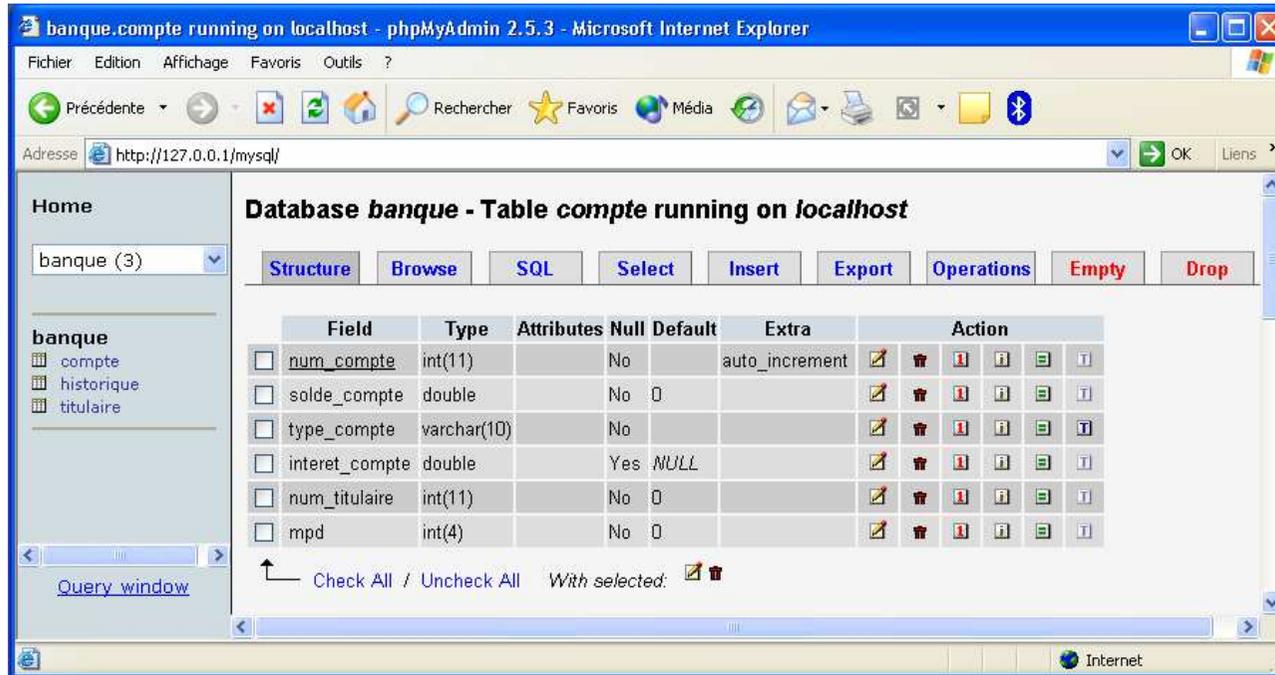
Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> num_titulaire	int(11)		No		auto_increment	
<input type="checkbox"/> nom_titulaire	varchar(30)		No			
<input type="checkbox"/> prenom_titulaire	varchar(30)		No			
<input type="checkbox"/> profession_titulaire	varchar(30)		No			
<input type="checkbox"/> salaire_titulaire	int(11)		No	0		

Below the table, there are controls for 'Check All / Uncheck All' and 'With selected:'.

Stéphanie Pok - Zouhour  
Boughammoura

# La base BANQUE (2)

## ■ Table compte



banque.compte running on localhost - phpMyAdmin 2.5.3 - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Rechercher Favoris Média

Adresse http://127.0.0.1/mysql/ OK Liens >>

Home

banque (3)

banque

- compte
- historique
- titulaire

Query window

Database *banque* - Table *compte* running on localhost

Structure Browse SQL Select Insert Export Operations Empty Drop

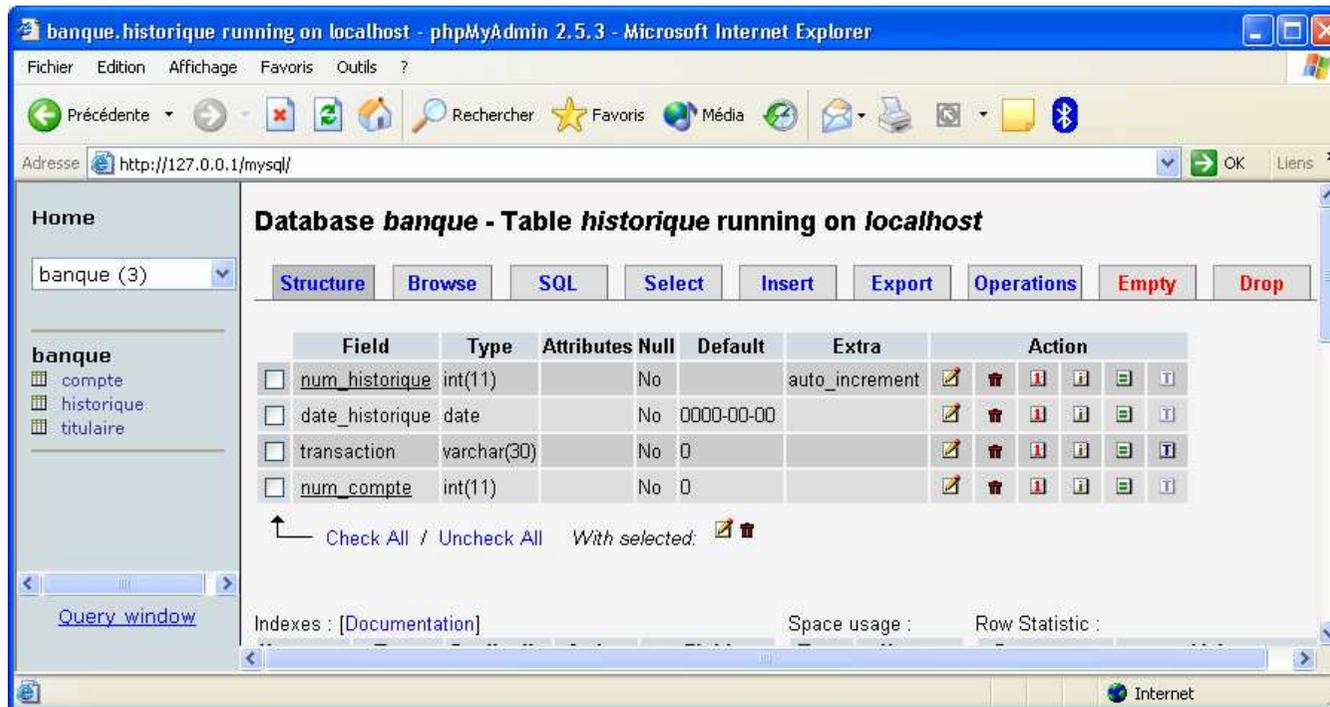
Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> num_compte	int(11)		No		auto_increment	      
<input type="checkbox"/> solde_compte	double		No	0		      
<input type="checkbox"/> type_compte	varchar(10)		No			      
<input type="checkbox"/> interet_compte	double		Yes	NULL		      
<input type="checkbox"/> num_titulaire	int(11)		No	0		      
<input type="checkbox"/> mpd	int(4)		No	0		      

Check All / Uncheck All With selected:  

Internet

# La base BANQUE (3)

## ■ Table historique

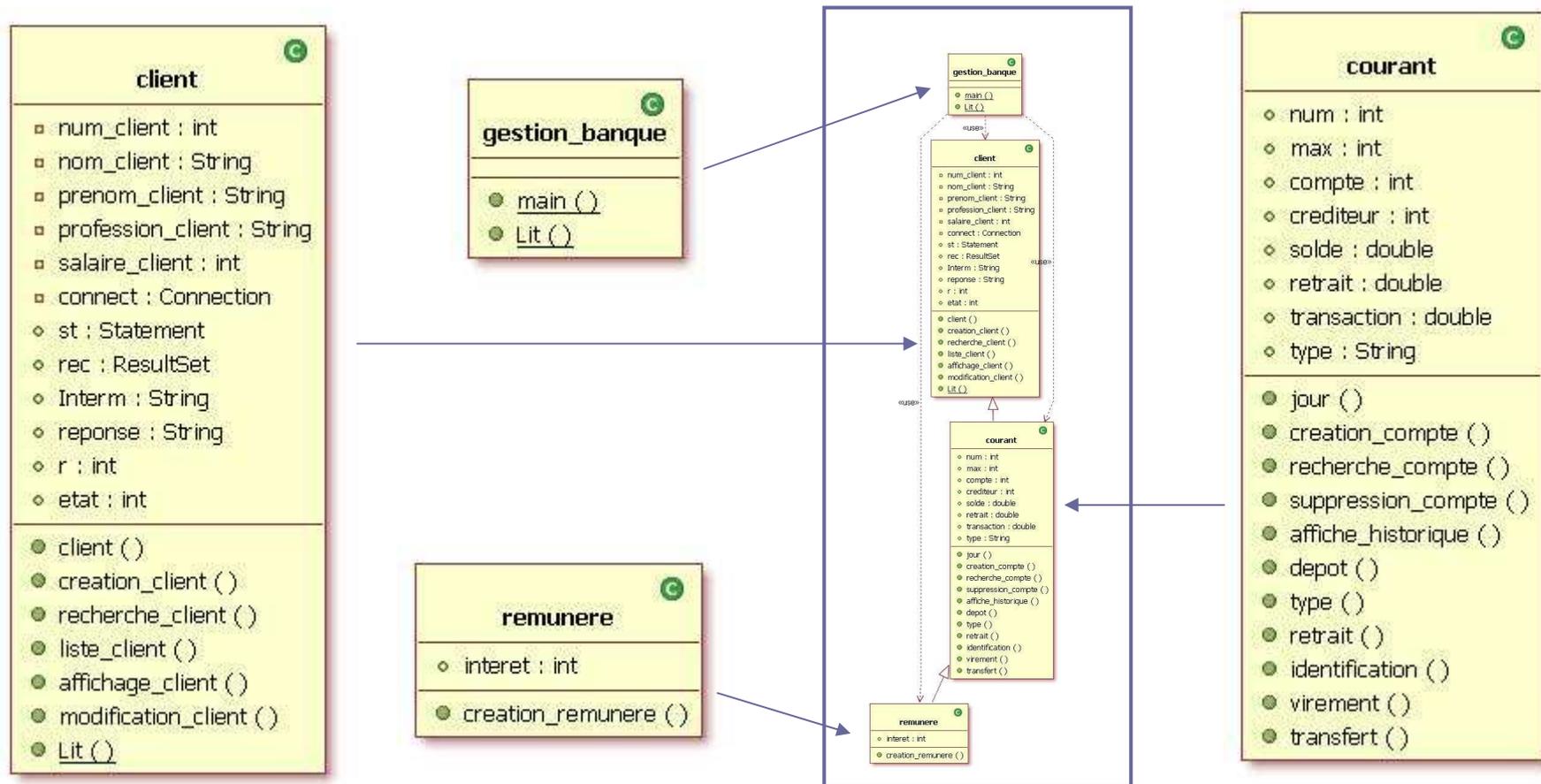


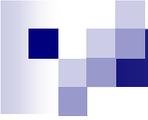
The screenshot shows the phpMyAdmin interface for a database named 'banque'. The 'historique' table is selected, and its structure is displayed. The table has four columns: 'num\_historique' (int(11), auto\_increment), 'date\_historique' (date, 0000-00-00), 'transaction' (varchar(30), 0), and 'num\_compte' (int(11), 0). The interface includes a navigation menu on the left, a toolbar with buttons like 'Structure', 'Browse', 'SQL', 'Select', 'Insert', 'Export', 'Operations', 'Empty', and 'Drop', and a 'Query window' at the bottom.

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> num_historique	int(11)		No		auto_increment	
<input type="checkbox"/> date_historique	date		No	0000-00-00		
<input type="checkbox"/> transaction	varchar(30)		No	0		
<input type="checkbox"/> num_compte	int(11)		No	0		

Stéphanie Pok - Zouhour  
Boughammoura

# Schéma UML





# Connexion à la base

- **JDBC (Java Data Base Connectivity) :**

Spécification de Sun qui définit une API permettant aux programmes Java d'accéder à des bases de données.

- **java.sql.SQLException :**

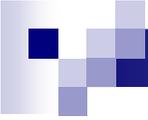
Package Java permettant d'exécuter des requêtes SQL.

- **API (Interface de programme d'application) :**

Interface fonctionnelle qui permet à un programme d'application écrit dans un langage de haut niveau d'utiliser des données ou des fonctions spécifiques du système d'exploitation ou d'un autre programme.

# Connexion à la base de données

```
public client()
{
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
    }
    catch(ClassNotFoundException e)
    {
        System.out.println("Erreur au chargement du driver : " + e.toString());
    }
    try
    {
        connect = DriverManager.getConnection
        ("jdbc:mysql://localhost/banque2?user=root&password=");
        st = connect.createStatement();
    }
    catch (SQLException ex)
    {
        System.out.println("SQLException: " + ex.getMessage());
        System.out.println("SQLState: " + ex.getSQLState());
        System.out.println("VendorError : " + ex.getErrorCode());
    }
}
```



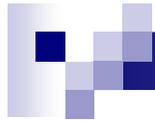
# Notion d'héritage

- Technique de programmation objet qui permet d'utiliser des classes existantes comme base de création d'autres classes.
- Ici, la classe " courant " hérite des propriétés publiques de la classe " client ".



# Notion de polymorphisme

- Le polymorphisme est la capacité, pour un même message de correspondre à plusieurs formes de traitement selon l'objet auquel ce message est adressé. La gestion du polymorphisme est assurée par la machine virtuelle dynamiquement à l'exécution.



# Notion d'exception

- Pour traiter les erreurs, Java utilise un mécanisme qualifié d'exception, qui consiste à effectuer les instructions dans un bloc d'essai (le bloc *try*) qui surveille les instructions. Lors de l'apparition d'une erreur, celle-ci est lancée dans un bloc de traitement d'erreur (le bloc *catch*, appelé *handler d'exception*) sous forme d'un objet appelé **Exception**.



# Fonctionnalités en mode banque (1)

- **Consultation et modifications sur un client**
  - On peut visualiser toutes les informations concernant un client ainsi que la liste et solde des comptes dont il est titulaire.
  - Possibilité de modifier la profession et le salaire du client.
- **Consultation et opérations sur un compte**
  - Dépôt
  - Retrait
  - Virement : débiter le compte du client pour créditer le compte d'un tiers.



## Fonctionnalités en mode banque (2)

### ■ Création d'un nouveau compte

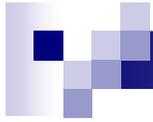
- Compte courant.
- Compte rémunéré : on demande de saisir le pourcentage des intérêts perçus par le client pour son compte.
- Compte sécurisé : avant chaque retrait, un calcul est effectué pour contrôler le solde du compte par rapport au montant demandé.
- Compte rémunéré et sécurisé.

### ■ Suppression d'un compte



# Fonctionnalités en mode DAB

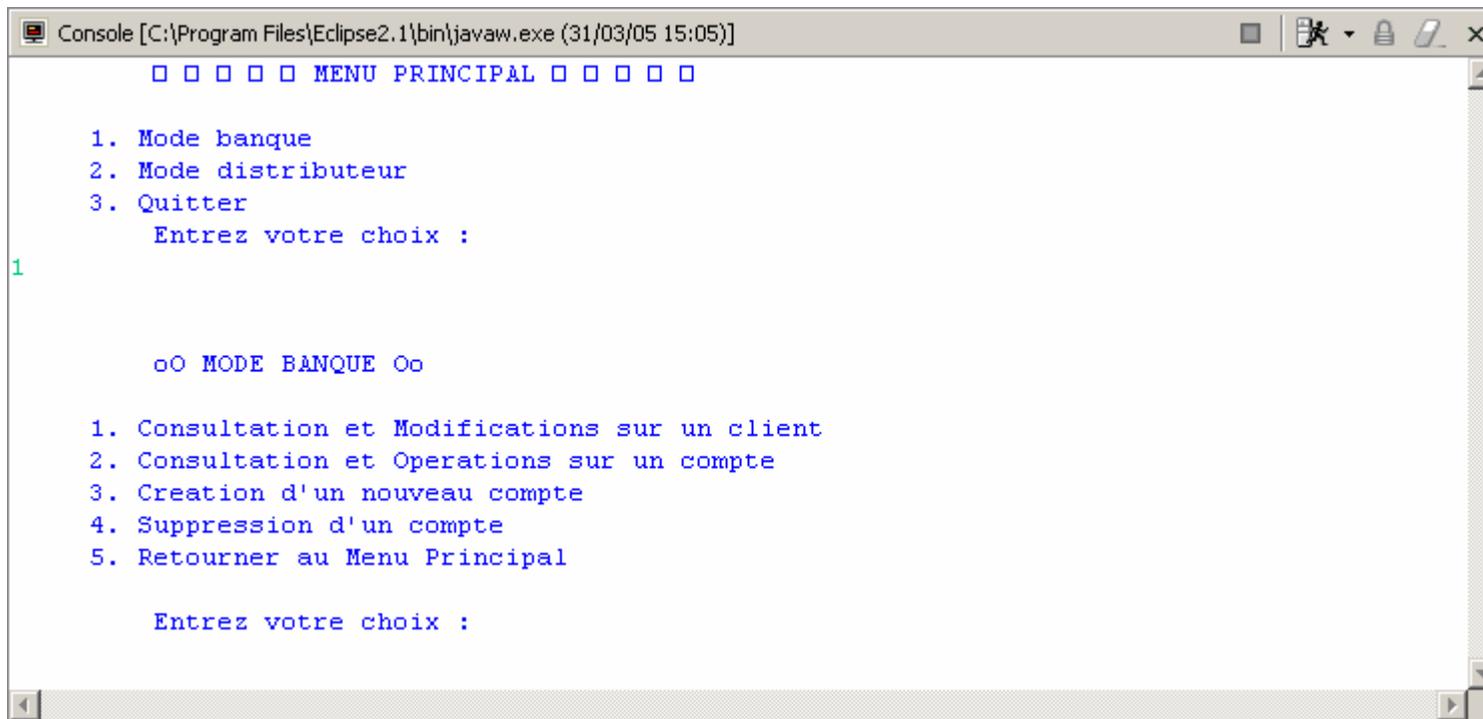
- Saisie du numéro du compte et du code secret de l'utilisateur
  - Au bout de 3 erreurs de saisie, le client est reconduit au menu principal
- Si validation du code secret :
  - Consultation du solde
  - Retrait
  - Dépôt



# **SIMULATION**

Stéphanie Pok - Zouhour  
Boughammoura

# Menu principal

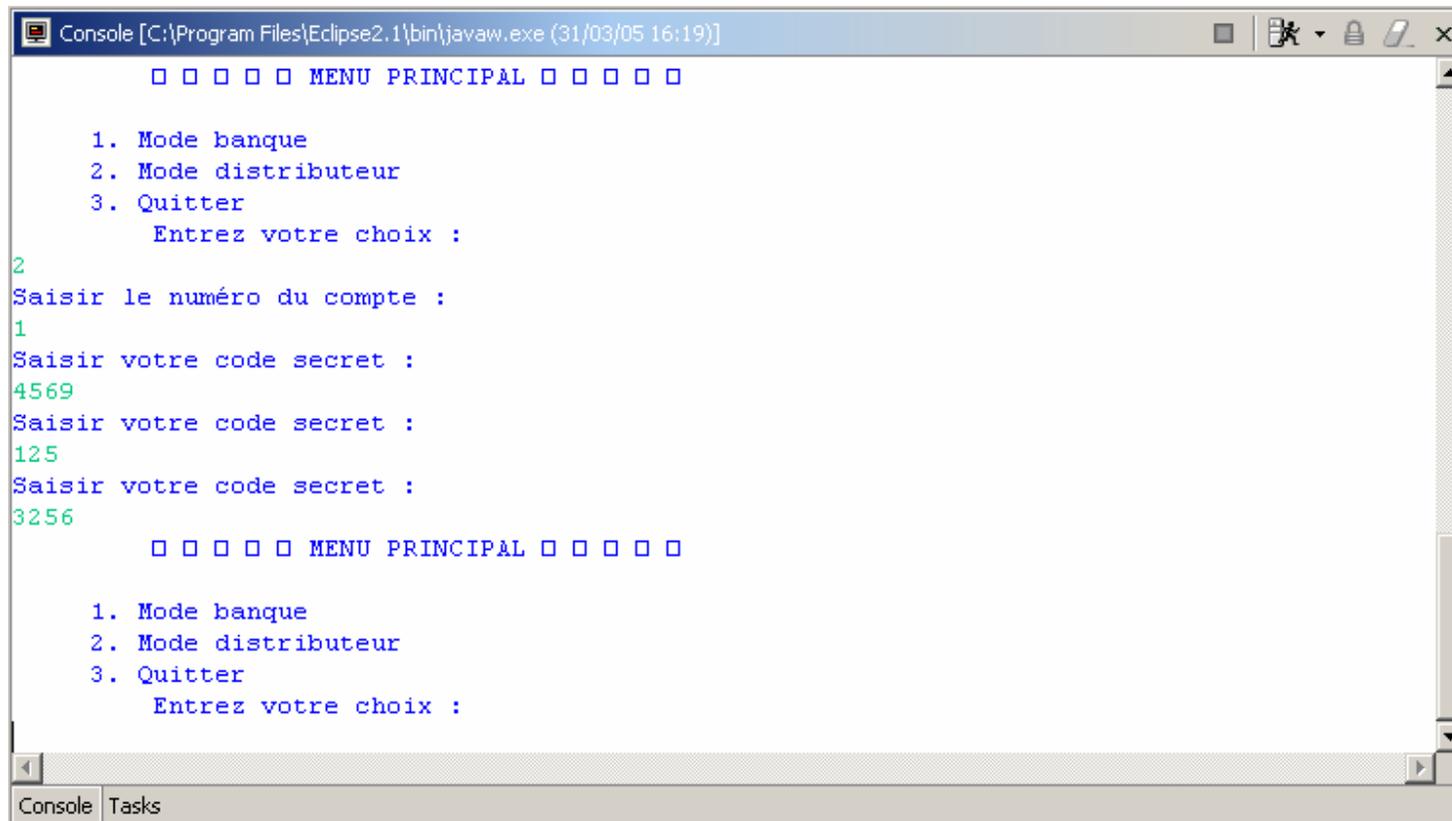


```
Console [C:\Program Files\Eclipse2.1\bin\javaw.exe (31/03/05 15:05)]
  □ □ □ □ □ MENU PRINCIPAL □ □ □ □ □
1. Mode banque
2. Mode distributeur
3. Quitter
  Entrez votre choix :

oO MODE BANQUE Oo
1. Consultation et Modifications sur un client
2. Consultation et Operations sur un compte
3. Creation d'un nouveau compte
4. Suppression d'un compte
5. Retourner au Menu Principal

  Entrez votre choix :
```

# Simulation DAB (1)



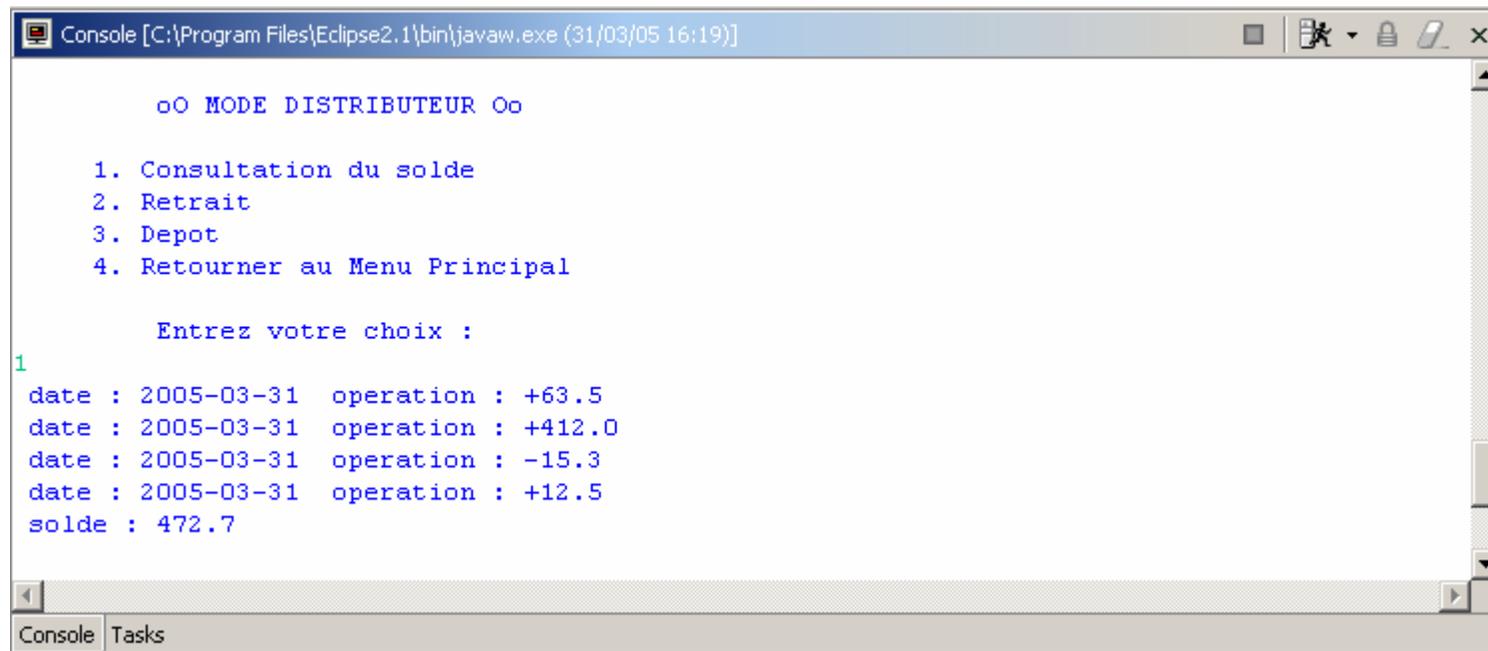
```
Console [C:\Program Files\Eclipse2.1\bin\javaw.exe (31/03/05 16:19)]
    □ □ □ □ MENU PRINCIPAL □ □ □ □

    1. Mode banque
    2. Mode distributeur
    3. Quitter
    Entrez votre choix :
2
Saisir le numéro du compte :
1
Saisir votre code secret :
4569
Saisir votre code secret :
125
Saisir votre code secret :
3256

    □ □ □ □ MENU PRINCIPAL □ □ □ □

    1. Mode banque
    2. Mode distributeur
    3. Quitter
    Entrez votre choix :
```

# Simulation DAB (2)



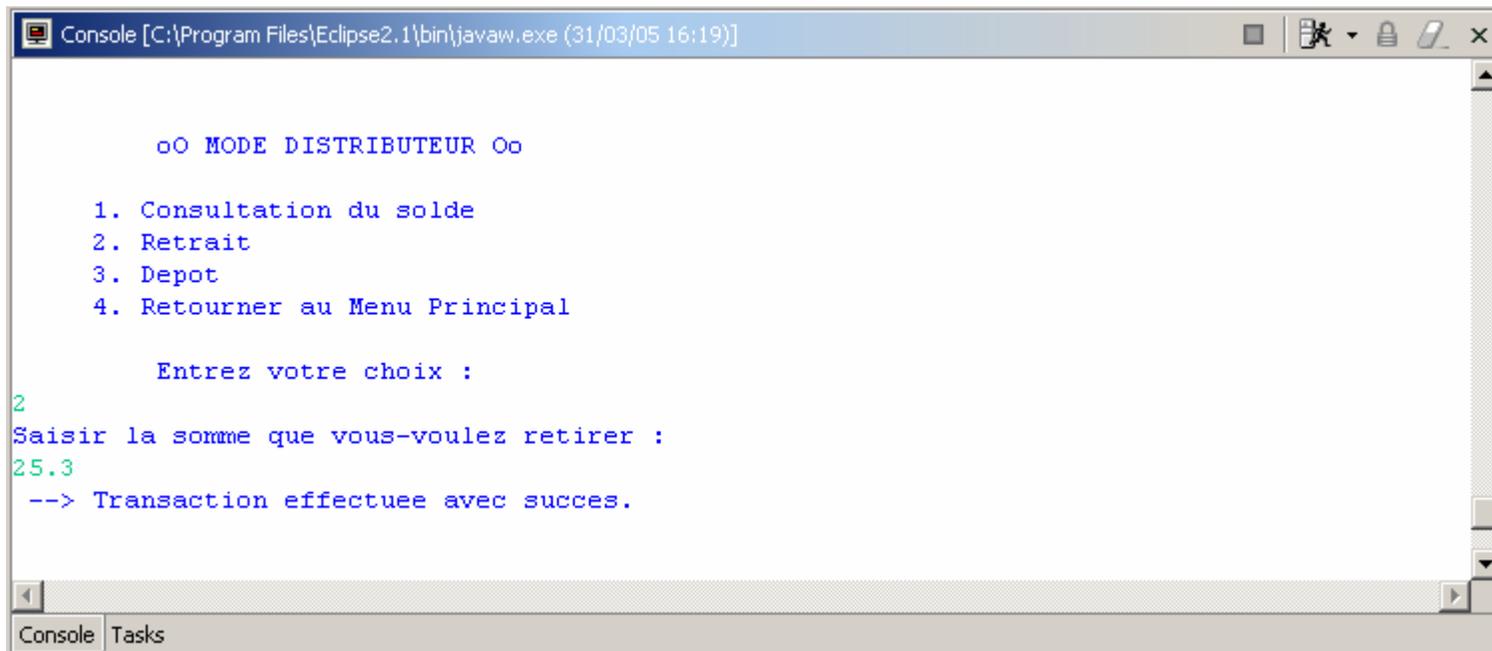
```
Console [C:\Program Files\Eclipse2.1\bin\javaw.exe (31/03/05 16:19)]

    oO MODE DISTRIBUTEUR Oo

1. Consultation du solde
2. Retrait
3. Depot
4. Retourner au Menu Principal

    Entrez votre choix :
1
date : 2005-03-31  operation : +63.5
date : 2005-03-31  operation : +412.0
date : 2005-03-31  operation : -15.3
date : 2005-03-31  operation : +12.5
solde : 472.7
```

# Simulation DAB (3)



```
Console [C:\Program Files\Eclipse2.1\bin\javaw.exe (31/03/05 16:19)]

    oO MODE DISTRIBUTEUR Oo

    1. Consultation du solde
    2. Retrait
    3. Depot
    4. Retourner au Menu Principal

    Entrez votre choix :
2
Saisir la somme que vous-voulez retirer :
25.3
--> Transaction effectuee avec succes.
```