COLLÈGE ROSEMONT

# Technique de développement en environnement graphique 1
# Évaluation 2 - TP

Abderrazak Sahraoui

# But et Objectifs

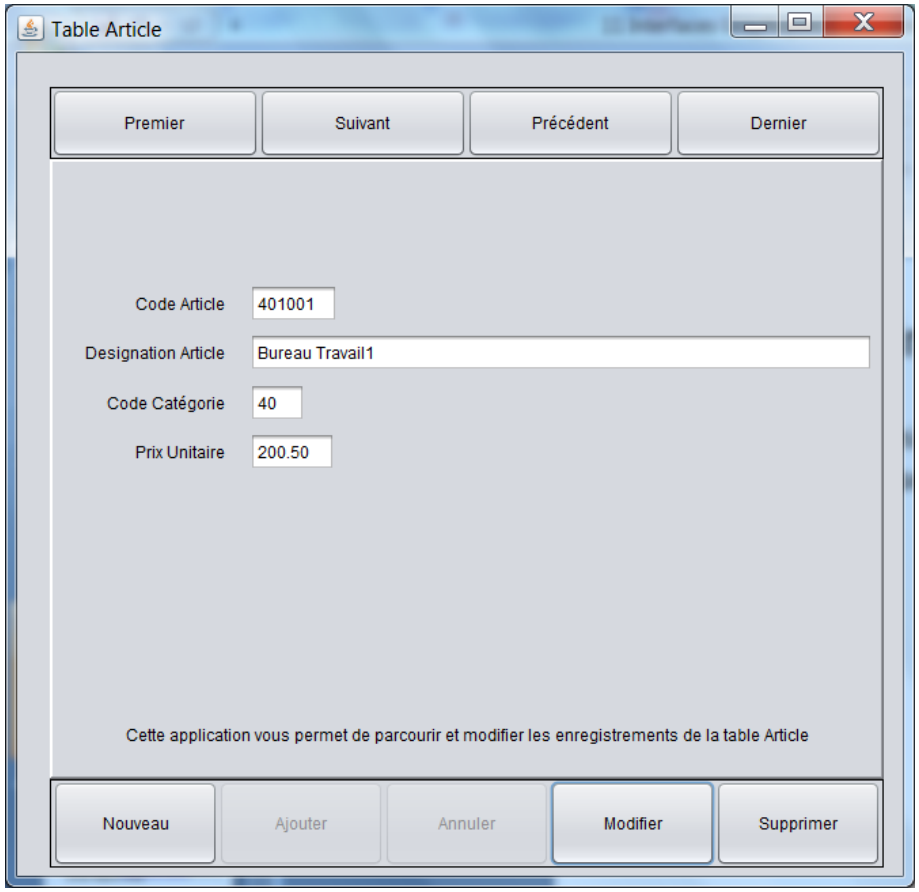
Ce projet a pour but le développement d'un package de programmes Java pour le parcours et la modification d'une table de base données via une interface graphique.

Ce projet vise à mettre en pratique des concepts de programmation Java en interface graphique et base de données. Il couvrira en particulier les apprentissages suivants :

- Utilisation des composants graphiques Swing pour concevoir un formulaire ;
- Utilisation des interfaces JDBC pour se connecter à une base données et travailler avec ses tables ;
- Utilisation de l'environnement NetBeans pour la génération du programme du formulaire ;
- Utilisation de Java DB pour la base de données ;
- Gestion des événements permettant le parcours des enregistrements de la table , l'insertion, la modification et la suppression d'enregistrements ;
- Utilisation d'une architecture MVC.

# Spécimen Interface Graphique

L'interface graphique ci-dessous pourrait servir de spécimen à utiliser dans le développement de l'application voulue.



# Travail à réaliser

Dans ce projet, vous allez procéder au développement d'un package Java qui met en œuvre le développement de la fenêtre d'interface de la page précédente et la programmation des événements de cette interface en utilisant les objets et les méthodes de l'interface JDBC:

1. Créez un projet dans NetBeans .
2. Générez l'interface graphique du formulaire par NetBeans.
3. Modifiez le programme obtenu pour associer à l'interface graphique un contrôleur et un model
4. Programmez l'interface pour gérer les événements des boutons de parcours : Premier, Suivant, Précédent et Dernier.
5. Programmez l'interface pour gérer les événements des boutons de changement: Nouveau, Ajouter, Modifier, Supprimer et Annuler.
6. Développez une classe pour le modèle. Celui-ci doit gérer un objet de type ResultSet permettant de récupérer les tuples d'une table stockée dans une base de donnée. Le modèle doit pouvoir réaliser le parcours des différents tuples, la modification et l'ajout de nouveaux tuples ainsi que la suppression de tuples.
7. Développez une classe pour le contrôleur.
8. Développez une classe utilitaire pour obtenir une connexion de base de données.
9. Développez une classe principale pour lancer l'application établir la connexion avec la base de données magasin et sa table article (La base doit être créée au préalable sur le serveur Java DB).

# Détails

1- Bouton Suivant

- L'application devrait afficher l'enregistrement suivant en utilisant la méthode next() de ResultSet.
- S'il n'y pas de suivant, l'application devrait afficher une boite de message de JOptionPane indiquant que l'enregistrement courant est le dernier de la table.

2- Bouton Précédent

- L'application devrait afficher l'enregistrement précédent en utilisant la méthode previous() de ResultSet
- S'il n'y pas de précédent, l'application devrait afficher une boite de message de JOptionPane indiquant que l'enregistrement courant est le premier de la table.

3- Bouton Premier

- L'application devrait afficher l'enregistrement premier en utilisant la méthode first() de ResultSet
- Si l'on est déjà sur le premier, une boite JOptionPane sera affichée indiquant que l'enregistrement courant est le premier de la table.

4- Bouton Dernier

- L'application devrait afficher l'enregistrement dernier en utilisant l'application last () de ResultSet
- Si l'on est déjà sur le dernier, une boite JOptionPane sera affichée indiquant que l'enregistrement courant est le dernier de la table.

# Détails

5- Bouton Nouveau

- L'application doit vider les champs du formulaire, désactiver les boutons de parcours et les boutons nouveau, modifier et supprimer, et activer les boutons ajouter et annuler.
- L'application doit sauvegarder la position de l'enregistrement courant pour y retourner et la réafficher à l'écran si l'usager décide de suspendre et annuler l'ajout. Elle utilisera à cet effet la méthode getRow().

6- Bouton  Ajouter

- Elle doit insérer l'enregistrement en utilisant les méthodes de ResultSet:: moveToInsert(), updateXXX(), insertRow()
- L'application doit relancer une nouvelle requête de sélection d'enregistrement article, récupérer le nouveau ResultSet, afficher l'enregistrement premier en utilisant la méthode next() de ResultSet
- L'application doit désactiver les boutons annuler et ajouter, et activer les boutons de parcours et les boutons nouveau, modifier et supprimer
- une boite JOptionPane sera affichée indiquant le bon déroulement de l'opération et sa nature.

7- Bouton Annuler

- L'application doit désactiver les boutons annuler et ajouter, et activer les boutons de parcours et les boutons nouveau, modifier et supprimer
- Elle doit également afficher l'enregistrement qui était affiché avant l'utilisation du bouton Nouveau. Elle utilisera à cet effet la méthode absolute (position) de ResultSet.

# Détails

8- Bouton Modifier

- Elle doit modifier l'enregistrement en utilisant les méthodes de ResultSet: updateXXX(), updateRow()
- une boite JOptionPane sera affichée indiquant le bon déroulement de l'opération et sa nature.

9- Bouton Supprimer

- Elle doit supprimer l'enregistrement en utilisant la méthode deleteRow() du ResultSet
- L'application doit relancer une nouvelle requête de sélection d'enregistrement article, récupérer le nouveau ResultSet, afficher l'enregistrement premier en utilisant la méthode next() de ResultSet
- une boite JOptionPane sera affichée indiquant le bon déroulement de l'opération et sa nature.

# Dates importantes

date début : 27 mars 2018

date retour : 17 avril 2018

# Références

https://java.developpez.com/faq/jdbc/?page=Les-resultats-moins-ResultSet