

# Interfaces Graphiques

420-AZ3-RO – Évaluation 2 - projet

# But et Objectifs

Ce projet a pour but le développement d'une application 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 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.

The screenshot shows a window titled "Table Article" with a standard Windows-style title bar. The interface is divided into several sections:

- Navigation Bar:** Located at the top, it contains four buttons: "Premier", "Suivant", "Précédent", and "Dernier".
- Form Fields:** Below the navigation bar, there are four labeled input fields:
  - "Code Article" with the value "401001"
  - "Designation Article" with the value "Bureau Travail1"
  - "Code Catégorie" with the value "40"
  - "Prix Unitaire" with the value "200.50"
- Footer Bar:** At the bottom, there are five buttons: "Nouveau", "Ajouter", "Annuler", "Modifier", and "Supprimer".
- Instructions:** A line of text at the bottom of the main area reads: "Cette application vous permet de parcourir et modifier les enregistrements de la table Article".

# Option 1 : Travail à réaliser

1. Créez une base de données formée de deux tables articles et catégorie. La table article contient les champs code article, désignation article, code catégorie, prix unitaire. La table catégorie contient les champs code catégorie et désignation catégorie. Remplissez vos deux tables avec quelques enregistrements.
2. Créez un projet dans NetBeans : Az3Eval2VotreNom.
3. Ajoutez une classe VueArticle qui construit l'interface graphique utilisateur permettant le parcours de tous les articles de la table, l'affichage, la modification, la saisie et la suppression d'articles.
4. Ajoutez une classe ModeleVueArticle. Elle doit avoir des méthodes pour : se connecter à une BD, obtenir un objet de type ResultSet de la table article, parcourir le resultSet, modifier ses articles et insérer de nouveaux articles.
5. Ajoutez une classe Article pour contenir des objets de la table article.
6. Utilisez le motif de conception (pattern) Observateur - Observable pour relier les classes VueArticle et ModeleVueArticle
7. Ajoutez une classe ControleurVueArticle pour gérer les interactions de l'utilisateur avec l'application :
  1. parcours des articles par les boutons Premier, Suivant, Précédent et Dernier.
  2. Saisie , modification et suppression d'articles par les boutons Nouveau, Ajouter, Modifier, Supprimer et Annuler.
8. Modifiez la méthode main de votre application pour créer les objets de votre application et afficher l'interface graphique aux utilisateurs.

# 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 boîte 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 boîte 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 boîte `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 boîte `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'utilisateur 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()` ou les requêtes SQL `INSERT INTO`.
- 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 boîte `JOptionPane` sera affichée indiquant le bon déroulement de l'opération et sa nature.
- En cas d'erreur, une boîte `JOptionPane` sera affichée indiquant l'erreur.

## 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 absolue (`position`) de `ResultSet`.

# Détails

## 8- Bouton Modifier

- Elle doit modifier l'enregistrement en utilisant les méthodes de ResultSet: updateXXX(), updateRow() ou les requêtes SQL UPDATE
- une boîte JOptionPane sera affichée indiquant le bon déroulement de l'opération et sa nature.
- En cas d'erreur, une boîte JOptionPane sera affichée indiquant l'erreur.

## 9- Bouton Supprimer

- Elle doit supprimer l'enregistrement en utilisant la méthode deleteRow() du ResultSet ou les requête SQL DELETE FROM
- 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 boîte JOptionPane sera affichée indiquant le bon déroulement de l'opération et sa nature.
- En cas d'erreur, une boîte JOptionPane sera affichée indiquant l'erreur.

# Dates importantes

date début : 13 novembre 2019

date retour : 29 novembre 2019

Travaillez en binôme.

Bonne Réussite !



# Option 2: Travail à réaliser

1. Créez une base de données formée de deux tables articles et catégorie. La table article contient les champs code article, désignation article, code catégorie, prix unitaire. La table catégorie contient les champs code catégorie et désignation catégorie. Remplissez vos deux tables avec quelques enregistrements.
2. Créez un projet dans NetBeans : Az3Eval2VotreNom.
3. Ajoutez une classe VueArticle qui utilise la classe JTable pour afficher un tableau permettant l'affichage de tous les articles de la table, la modification, la saisie et la suppression d'articles.
4. Ajoutez une classe ModeleVueArticle. Elle doit implémenter les méthodes de la classe AbstractTable.
5. Ajoutez une classe ControleurVueArticle pour gérer les interactions de l'utilisateur avec l'application
6. Modifiez la méthode main de votre application pour créer les objets de votre application et afficher l'interface graphique aux utilisateurs.