

Mission 4 : Ajout fonctionnel

Gestion des villes

(de 1,5 à 2 journées)

Difficulté : **

Table des matières

[Ressources](#)

[Activités et compétences du référentiel mises en œuvre](#)

[Résultats attendus](#)

[Travail à faire](#)

[Préparation de la mission](#)

[Ajout fonctionnel](#)

[Présentation](#)

[Les scénarios](#)

Ressources

- L'application initiale

Activités et compétences du référentiel mises en œuvre

P1 : Production de services

- A1.1.1 Analyse du cahier des charges d'un service à produire
 - C1.1.1.2 Identifier les fonctionnalités attendues du service à produire
- A1.4.1 Participation à un projet
 - C1.4.1.2 Rendre compte de son activité

P4 - Conception et maintenance de solutions applicatives

- A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative
 - C4.1.2.2 Maquetter un élément de la solution applicative
- A4.1.3 Conception ou adaptation d'une base de données
 - C4.1.3.2 Implémenter le schéma de données dans un SGBD
 - C4.1.3.4 Manipuler les données liées à la solution applicative à travers un langage de requête
- A4.1.7 Développement, utilisation ou adaptation de composants logiciels
 - C4.1.7.1 Développer les éléments d'une solution
 - C4.1.7.4 Utiliser des composants d'accès aux données

Résultats attendus

- Le projet
- Le compte-rendu
 - votre démarche globale de solution
 - les lignes de la table VILLE correspondant à un même code postal : 44310
 - la ligne de la table VILLE correspondant à PARIS,
 - le schéma relationnel de la table VILLE,
 - le diagramme de classes relatif aux classes Ville et DaoVille,
 - la maquette ou une copie d'écran de la vue
 - avec la justification des contrôles graphiques utilisés pour le code postal et les villes,
 - avec le choix des événements pour valider un code postal et choisir une ville,
 - Le code des classes de test TestVille et TestDaoVille avec leurs résultats,
 - bilan : temps passé, ce que vous avez appris, les problèmes que vous avez pu résoudre par internet ...

Travail à faire

1. Préparation de la mission

- Faites une copie du projet projet_1SLAM_S2_2019_M1 en projet_1SLAM_S2_2019_M4
- Dans ce projet, renommez `VueAdresse_M1.java` en `VueAdresse_M4.java` et changez le titre de la `JFrame` en « Gestion des adresses Mission 4 ».

2. Ajout fonctionnel

1. Présentation

Votre chef de projet souhaite que vous conceviez une interface qui permet de choisir une ville suivant le code postal ou le début de son code postal.

L'objectif est ensuite d'utiliser cette interface pour la saisie des villes des adresses. Il n'y aura ainsi aucune erreur de saisie dans les villes et celles-ci seront toujours au même format (majuscules, tirets ...).

Votre chef de projet a récupéré pour vous la liste des villes de France sur Internet, au format SQL, sur le site <http://sql.sh/736-base-donnees-villes-francaises>.

Il vous est fourni 2 fichiers :

- `villes_france-structure-Oracle.sql`, contenant l'ordre de création de la structure de la table `VILLES_FRANCE`,
- `villes_france-donnees-Oracle.sql`, contenant les ordres de peuplement de cette table.

Cependant, cette table est assez imposante, et toutes les données ne sont pas pertinentes pour le projet. Votre chef de projet vous fournit un script de création d'une table `VILLE` à partir de la table `VILLES_FRANCE`, qui ne contient plus que quelques colonnes :

- `Create_Ville.sql`.

Vous devez utiliser l'ensemble de ces scripts pour créer la table VILLE.

Vous pourrez remarquer qu'un code postal peut correspondre à plusieurs villes et inversement qu'une ville peut posséder plusieurs codes postaux. Dans ce dernier cas, vous considérerez que l'ensemble de ces codes constituent un seul code postal.

Dans la description de votre démarche de solution, vous préciserez bien les impacts à tous les niveaux de cet ajout fonctionnel.

Conseil : pour créer l'interface graphique, vous utiliserez la classe `JDialog` et non la classe `JFrame`.

Un choix pertinent des contrôles graphiques utilisés est attendu.

Le code devra respecter les normes de développement habituelles. Vous devrez en particulier utiliser une classe Ecouteur interne à la vue pour la gestion des événements.

2. Les scénarios

Les scénarios suivants décrivent le comportement de l'interface.

Scénario nominal :

1 l'utilisateur saisit le code postal 44310 et valide

2 le système affiche une liste de villes (triée par nom) correspondant à ce code postal, en l'occurrence :

LA LIMOUZINIÈRE

SAINT-COLOMBAN

SAINT-LUMINE-DE-COUTAIS

SAINT-PHILBERT-DE-GRAND-LIEU

3 l'utilisateur choisit une ville.

4 le système affiche le code postal de la ville

5 l'utilisateur quitte l'interface

Scénario alternatif 1 :

1 l'utilisateur saisit le début du code postal 443 et valide

2 le système affiche une liste de villes (triée par nom) dont le code postal commence par ces chiffres, en l'occurrence :

ARTHON-EN-RETZ

BELIGNE

BOUGUENNAIS

...

LA LIMOUZINIÈRE

SAINT-COLOMBAN

SAINT-LUMINE-DE-COUTAIS

SAINT-PHILBERT-DE-GRAND-LIEU

3 retour au 3 du scénario nominal

Scénario erreur1 :

- 1 l'utilisateur ne saisit rien et valide
- 2 le système n'affiche rien
- 3 l'utilisateur quitte l'interface

Scénario erreur2 :

- 1 l'utilisateur ne saisit rien et valide
- 2 le système n'affiche rien
- 3 retour 1 scénario nominal