

---

# Mission 1 : découverte de l'application, amélioration de l'ergonomie (pas plus d'une demi-journée)

Difficulté : \*\*

## Table des matières

[Ressources](#)

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

[Résultats attendus](#)

[Travail à faire](#)

[Préparation de la mission](#)

[Documentation Java](#)

[Amélioration de l'application](#)

[Annexe](#)

## Ressources

- L'application
- Le script de création de la base de données.

## 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
- 1.4.1 Participation à un projet
  - C1.4.1.2 Rendre compte de son activité

### P4 - Conception et maintenance de solutions applicatives

- 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
  - C4.1.7.3 Analyser et modifier le code d'un composant logiciel
  - C4.1.7.4 Utiliser des composants d'accès aux données
- A4.2.1 Analyse et correction d'un dysfonctionnement, d'un problème de qualité de service ou de sécurité
  - C4.2.1.2 Repérer les composants à l'origine du dysfonctionnement
  - C4.2.1.3 Concevoir les mises à jour à effectuer
  - C4.2.1.4 Réaliser les mises à jour

## Résultats attendus

- Le projet
- Le compte-rendu doit contenir en plus
  - la constitution des sous-équipes avec la répartition des missions,
  - une copie d'écran commentée montrant que vous avez accès à la documentation (après SHIFT F1 sur le nom d'une classe),
  - une copie d'écran commentée montrant que la documentation est en local,
  - les lignes de code ajoutées ou modifiées, avec un commentaire,
  - une copie d'écran commentée de l'interface obtenue,
  - bilan : temps passé, les problèmes que vous avez pu résoudre par internet ...
  - aucun rapport de test fonctionnel n'est demandé.

## Travail à faire

### 1. Préparation de la mission

- Constituez les équipes et les sous-équipes.
- Affectez les missions aux sous-équipes.
- Créez la base de données sous Oracle Database :
  - en tant qu'administrateur, créez un nouveau schéma (utilisateur) LABARTE, mot de passe "secret"
  - en tant que LABARTE, exécutez le script "Create\_Adresse.sql"
- Ouvrez le projet "projet\_1SLAM\_S2\_2019\_initial" et testez l'application :
  - exécutez les classes `TestAdresse` et `TestDaoAdresse`
  - exécutez la classe `VueAdresse_init.java`
- Faites une copie du projet en `projet_1SLAM_S2_2019_M1`
- Dans ce projet, renommez `VueAdresse_init.java` en `VueAdresse_M1.java` et changez le titre de la `JFrame` en « Gestion des adresses Mission 1 ».
- Étudiez le code, en localisant (nom du paquetage et de la classe) les classes suivantes :
  - classe qui lance l'application
  - classe(s) d'interface(s) graphique(s)
  - classe(s) écouteur(s) d'événements sur l'interface graphique
  - classe(s) métier(s)
  - classe de connexion à la BDD
  - classe(s) qui permet(tent) d'exécuter les requêtes SQL sur la table ADRESSE et créer des objets Adresse
  - classe(s) qui contiennent les paramètres de connexion à la BDD (adresse du serveur, login et mot de passe de l'utilisateur de la BDD)

## 2. Documentation Java

Vérifiez que vous avez accès à une documentation java en local. Pour information, la documentation java se trouve dans le répertoire `/home/btssio/Bibliothèques` de la machine virtuelle.

## 3. Amélioration de l'application

Nous souhaitons d'une part améliorer le code et l'ergonomie de l'application. La classe concernée est la classe `VueAdresse_M1.java`

- Supprimez les imports inutiles.
- Insérez des commentaires dans le code de la classe `VueAdresse_M1` uniquement :
  - au niveau du cartouche
  - au niveau de la déclaration des attributs
  - au niveau des méthodes

Vous devez observer les règles concernant les commentaires énoncées dans le document `1SIO_S14_NormesDePresentationDuCodeSource_v2015-2016.pdf`.

- Comparez les 3 portions de code suivantes : ligne 54 à 64, 410 à 420, 538 à 548. Elles sont identiques. Créez une méthode privée nommée `afficherPremiereAdresse()` qui contient ce code et remplacez aux 3 endroits, les lignes de code identiques par un appel à cette méthode.
- Rajoutez un bouton "Quitter" qui permet de sortir de l'application.
- Faites en sorte que l'appui sur la touche "Entrée" lorsque l'on est dans le champ de saisie de l'identifiant, déclenche la validation, mais ceci uniquement en mode « Recherche » et non en mode « Ajout ».
- En mode « Recherche », si l'on saisit un id qui n'est pas un entier, alors il y a un message d'erreur qui s'affiche. Faites le même contrôle de saisie en mode « Ajout ».

## Annexe

Écran de l'application existante.



The screenshot shows a window titled "Adresse" with a light gray background. It contains four text input fields stacked vertically, each with a label to its left: "Id : 1", "Rue : 221B Baker Street", "Cdp : NW1 6XE", and "Ville : LONDON". Below the input fields, there are six buttons arranged in two rows. The top row contains "Ajouter", "Supprimer", "Modifier", and "Rechercher". The bottom row contains "Annuler" and "Valider". The "Ajouter" button is highlighted with a blue border.