

Rapport du Projet : Gestion Simplifiée des Stages

1. Contexte

L'objectif de ce projet est de développer une application Java simple pour une entreprise souhaitant gérer efficacement les informations relatives à ses stagiaires et à leurs stages. Cette application doit inclure une interface graphique (GUI) développée avec Swing pour faciliter l'interaction avec les données.

2. Objectifs du Projet

- Concevoir une application modulaire et orientée objet.
- Intégrer une interface graphique intuitive avec Swing.
- Manipuler des collections dynamiques pour stocker et gérer les données.
- Implémenter des concepts fondamentaux de la programmation orientée objet (héritage, polymorphisme, interfaces).
- Assurer la validation des données pour éviter les doublons et les erreurs.

3. Fonctionnalités

3.1. Gestion des Stagiaires

- Ajouter un stagiaire (Nom, Prénom, Niveau d'étude).
- Supprimer un stagiaire par son nom.
- Afficher la liste de tous les stagiaires.

3.2. Gestion des Stages

- Ajouter un stage (Titre, Durée en jours).
- Associer un ou plusieurs stagiaires à un stage.
- Afficher la liste des stages avec les stagiaires associés.

3.3. Recherche et Statistiques

- Rechercher un stagiaire par son nom ou son niveau d'étude.
- Calculer le nombre total de stages.
- Afficher le nombre de stagiaires associés à chaque stage.

4. Architecture du Projet

4.1. Modèle Objet

- **Classe Personne (Classe Mère) :**
 - Attributs : nom, prenom.
 - Méthode abstraite : afficherInfos().
- **Classe Stagiaire (Héritage de Personne) :**
 - Attribut supplémentaire : niveauEtude.

- Implémentation de `afficherInfos()`.
- **Classe Stage :**
 - Attributs : titre, duree, stagiaires (liste des stagiaires associés).
 - Méthodes : ajouter des stagiaires, afficher les informations du stage.

4.2. Interface Gestion

- Méthodes définies :
 - `ajouter()`, `supprimer()`, `afficher()`.

4.3. Collections

- Les stagiaires et les stages sont stockés dans des **ArrayList** pour permettre une gestion dynamique.

5. Interface Graphique (Swing)

5.1. Structure de l'Interface

- **Fenêtre principale (JFrame):**
 - Titre : Gestion des Stages.
 - Composants principaux :
 - Boutons (JButton) pour accéder aux fonctionnalités.
 - Zones de texte (JTextField) pour saisir les données.
 - Tables (JTable) pour afficher les listes de stagiaires et de stages.

5.2. Fonctionnalités de l'Interface

- **Ajouter Stagiaire :**
 - Une fenêtre pop-up avec des champs pour saisir le nom, prénom, et niveau d'étude.
- **Afficher les Stagiaires :**
 - Une table pour afficher les stagiaires enregistrés.
- **Ajouter un Stage :**
 - Une fenêtre pop-up pour saisir le titre et la durée.
- **Associer des Stagiaires à un Stage :**
 - Une fenêtre permettant de sélectionner un stagiaire et un stage dans des listes déroulantes.

5.3. Navigation

Un menu intuitif est affiché avec des boutons bien définis pour chaque fonctionnalité. Les utilisateurs peuvent naviguer facilement entre les différentes options.

6. Contraintes Techniques

- **Héritage :** Une classe mère `Personne` est utilisée, et `Stagiaire` en hérite.
- **Polymorphisme :** Les informations des stagiaires sont affichées en utilisant la méthode abstraite `afficherInfos()` de manière polymorphique.

- **Interface Gestion** : Les méthodes génériques pour ajouter, supprimer et afficher des données sont définies dans une interface commune.
- **Validation des données** :
 - Les doublons sont évités grâce à des vérifications lors de l'ajout.
 - Les entrées utilisateur sont validées avant d'être enregistrées.

7. Exemple d'Interface Utilisateur

Fenêtre principale

- Options disponibles sous forme de boutons :
 - Ajouter un stagiaire.
 - Supprimer un stagiaire.
 - Afficher tous les stagiaires.
 - Ajouter un stage.
 - Associer un stagiaire à un stage.
 - Afficher tous les stages.

Fenêtre pour Ajouter un Stagiaire

- Champs à remplir : Nom, Prénom, Niveau d'étude.
- Bouton : Valider.

Fenêtre pour Ajouter un Stage

- Champs à remplir : Titre, Durée (en jours).
- Bouton : Valider.

8. Résultats Attendus

- Une application fonctionnelle avec une interface utilisateur conviviale.
- Gestion efficace des données des stagiaires et des stages via une approche modulaire.
- Simplification des tâches de recherche, ajout, et suppression grâce à une gestion en mémoire dynamique.

9. Améliorations Futures

- Implémenter une persistance des données en utilisant une base de données ou des fichiers.
- Ajouter des fonctionnalités avancées comme :
 - La génération de rapports PDF pour les stages et stagiaires.
 - Une recherche multicritères.
 - Une gestion des droits utilisateurs pour sécuriser l'accès à l'application.