

Rapport - Modélisation et Implémentation d'une Application Java

Lien : <https://github.com/FlaZaY29/s2-sae-dev-app-ef5>

1. Introduction

Ce rapport présente la modélisation et l'implémentation d'une application Java pour la gestion de la **CIUP (Cité Internationale Universitaire de Paris)**. L'objectif du projet est de modéliser les différentes entités et leurs relations, puis d'implémenter une solution fonctionnelle en **Java** à partir de diagrammes UML réalisés sous **Visual Paradigm**.

2. Analyse et Modélisation UML

2.1. Diagramme de classes

Nous avons conçu un **diagramme de classes** représentant les entités principales de l'application :

- **CIUP** : Gère les maisons et les services.
- **Maison** : Représente une maison d'étudiants avec ses caractéristiques et sa liste d'étudiants.
- **MaisonClassique** : Représente une maison classique avec un nombre de places limité.
- **MaisonInternationale** : Spécialisation de Maison qui offre des services spécifiques.
- **Étudiant** : Représente un étudiant résident avec ses informations personnelles.
- **Service** : Modélise un service proposé par la CIUP.
- **FactoryCIUP** : Une classe permettant de centraliser la création des objets (Maisons, Étudiants et Services).

2.2. Relations entre les classes

- **CIUP** contient plusieurs **Maisons** et propose différents **Services**.
- Une **Maison** héberge plusieurs **Étudiants**.
- Une **MaisonInternationale** peut avoir des **Services** associés.
- La **FactoryCIUP** est utilisée pour la création des instances des différentes classes.

3. Implémentation en Java

3.1. Structure du projet

Le projet a été développé sous **IDE (Visual Studio Code, Eclipse, IntelliJ)** et **Git**, en respectant une **architecture orientée objet et qualité de développement**. La structure du projet est composée des fichiers suivants :

CIUP/

- **CIUP.java** : Classe principale gérant les maisons et services.
- **Maison.java** : Gère les maisons et leurs étudiants.
- **MaisonClassique.java** : Hérite de Maison et définit des maisons classiques avec des restrictions de places.
- **MaisonInternationale.java** : Spécialisation de la classe Maison avec ajout de services.
- **Etudiant.java** : Contient les informations personnelles des étudiants.
- **Service.java** : Définit les services disponibles.
- **FactoryCIUP.java** : Fabrique des objets pour éviter une instanciation directe.

3.2. Principales fonctionnalités implémentées

- **Gestion des Maisons** : Création, ajout et suppression d'étudiants.
- **Gestion des Étudiants** : Inscription, affichage des informations.
- **Gestion des Services** : Ajout et suppression de services pour les maisons internationales.
- **Utilisation d'une Factory** pour centraliser la création d'objets.

4. Répartition des tâches

Nom	Tâches réalisées
Se Donald	Modélisation UML, Implémentation de la classe Maison, Maison Classique et MaisonInternationale
Urkmez Yavuz	Modélisation UML, Implémentation de la classe FactoryCIUP et CIUP
Zamperlini Flavio	Implémentation de la classe Etudiante et MaisonClassique
Mouhou Maksen	Implémentation de la classe Service, Réalisation du rapport

