

Projet P00 en Java

Promotion immobilière

Décembre 2022

```
38         nbRoom = 25;  
39         break;  
40     case 5 :  
41         nbRoom = 30;  
42         break;  
43     default:  
44         System.out.println("Error, number of stars can be 1 to 5");  
45         break;  
46     }  
47 }  
48  
1 usage  AlexisFa  
48 public Hotel() { rentals = new HashMap<Integer, Occupant>(); }  
0 usages AlexisFa  
51 public Hotel(String address, float livingSpace, int stars, Owner owner) {  
52     super(address, livingSpace, owner);  
53     rentals = new HashMap<Integer, Occupant>();  
54     nbStars = stars;  
55     nbSpa = 0; // first set the special attributes to 0 and change them after depending on stars number  
56     nbPool = 0;  
57     nbSuite = 0;  
58     switch (nbStars){  
59         case 1 : nbRoom = 10;break;  
60         case 2 : nbRoom = 15;break;  
61         case 3 : nbRoom = 20;break;  
62         case 4 : // nbSpa will be set after the constructor in the main  
63             nbPool = 1;  
64             nbRoom = 25;  
65             break;  
66         case 5 :  
67             nbRoom = 30;  
68             break;  
69         default:
```

Table des matières

- 1. Introduction**
- 2. Diagramme des classes**
- 3. Présentation fonctionnelle du projet**
 - 3.1. Fonctionnalités réalisées parmi celles proposées dans le projet**
 - 3.2. Fonctionnalités ajoutées en plus de ce qui a été proposé**
- 4. Présentation technique du projet**
 - 4.1. Description des principaux algorithmes réalisés**
 - 4.2. Justification des choix des structures de données employées**
 - 4.3. Explication des difficultés (techniques)**
- 5. Présentation des résultats**
 - 5.1. Tests réalisés et résultats obtenus**
- 6. Conclusion**
 - 6.1. Apprentissages sur le plan technique**
 - 6.2. Organisation du travail et gestion du temps**

1. Introduction

Pour ce projet, nous devons concevoir l'architecture orientée objet d'une résidence dans une île. La résidence comportera plusieurs types de bâtiments dont des hôtels, des maisons, des immeubles et des magasins.

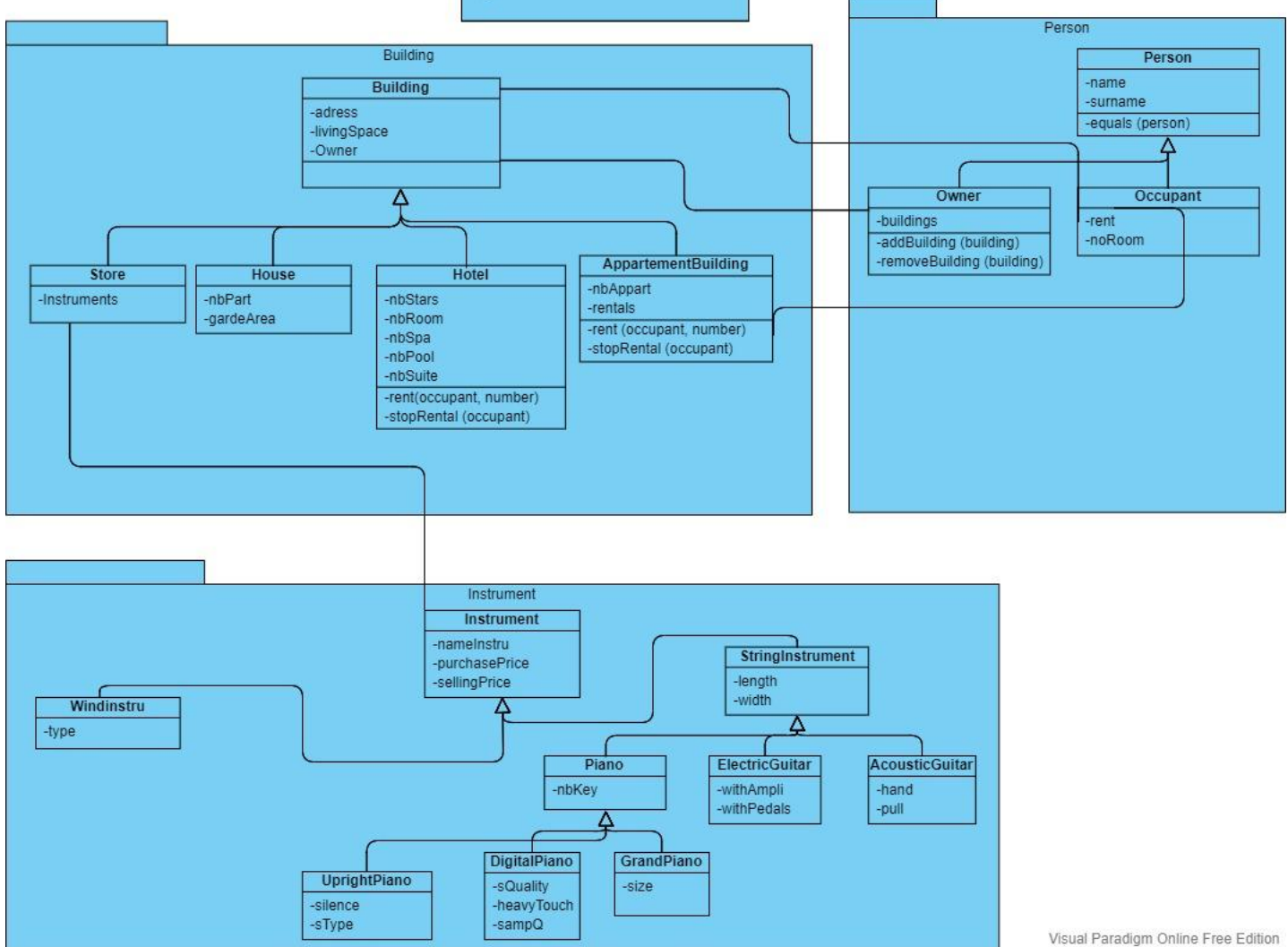
L'un des buts de ce projet est de mettre en avant nos compétences en codage acquis durant nos cours de Programmation Objet Orienté Java. De plus, il permet d'améliorer ces compétences techniques à partir d'un groupe de projet qui permet de réunir les capacités de chaque personne du groupe afin de mener à bien le projet.

Ce projet a également pour but de faire toutes les étapes d'un projet de A à Z afin de s'habituer pour quand nous serons dans le monde du travail. Il nous a permis de travailler en indépendance à partir de nos cours de Java mais également grâce à nos recherches personnelles.

2. Diagramme des classes

Visual Paradigm Online Free Edition

Les methodes get et set are not in the diagram

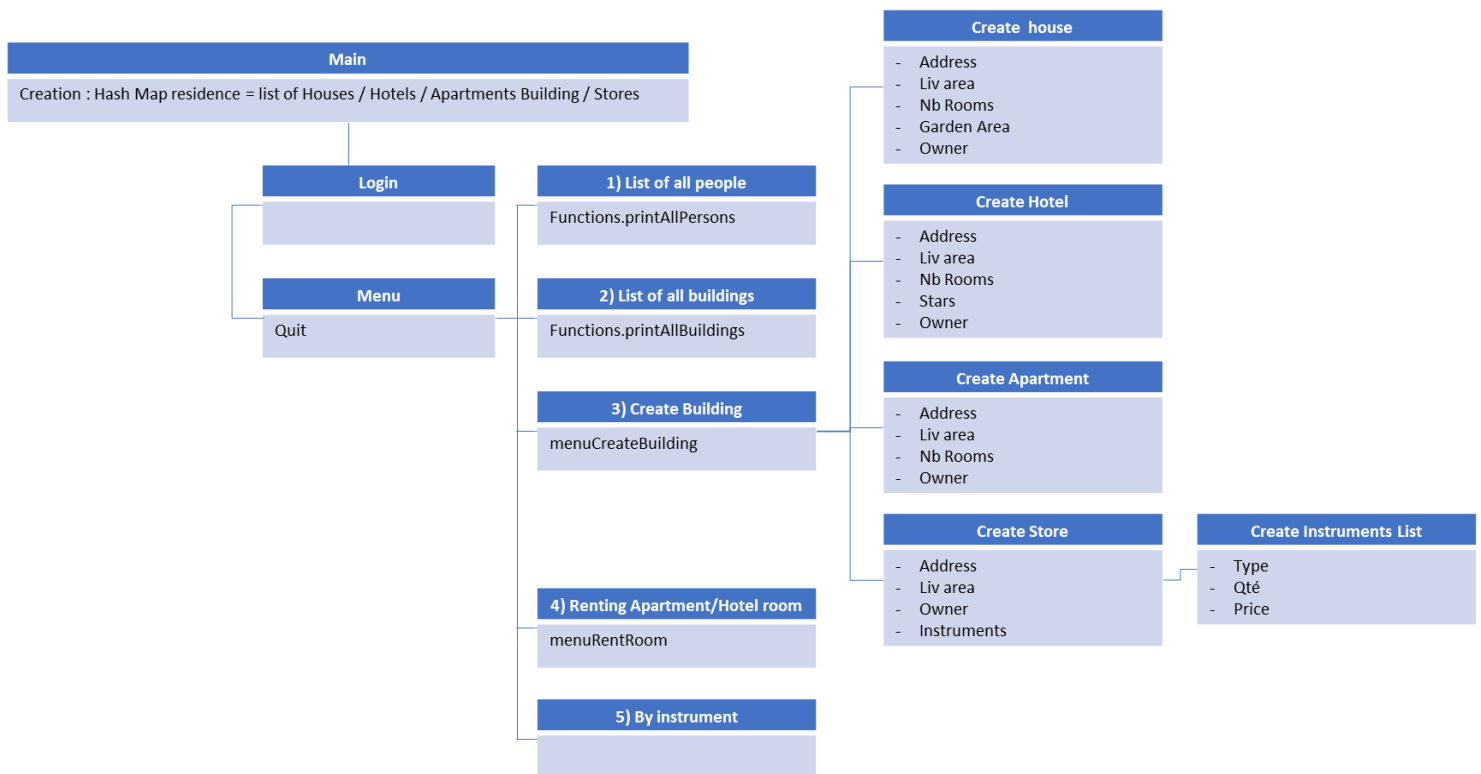


Visual Paradigm Online Free Edition

3. Présentation fonctionnelle du projet

3.1. Fonctionnalités réalisées parmi celles proposées dans le projet

Toutes les fonctionnalités demandées dans l'énoncé ont été réalisées, à savoir :
L'affichage détaillé des descriptions d'infrastructure et d'instruments musicaux. Les associations d'un propriétaire ou du locataire à un bâtiment, appartement, chambre ...
Le listage des biens d'un propriétaire, des appartements loués, des chambres et des suites disponibles. Ainsi que le calcul de l'impôt local en fonction des bâtiments.



3.2. Fonctionnalités ajoutées en plus de ce qui a été proposé

En plus des fonctionnalités proposées nous avons ajouté 4 autres fonctionnalités telles que la possibilité de donner un bâtiment (si le propriétaire souhaite donner un logement une liste de personne enregistrée lui sera donnée). Il est possible de faire l'achat d'instruments. Le propriétaire à la possibilité de stopper la location d'un logement. De plus, un système de logging (possibilité de s'enregistrer, le nom entré va être enregistré afin de ne pas perdre les données, à chaque fois que l'on rentrera le même nom, les données correspondantes seront disponibles).

4. Présentation technique du projet

4.1. Description des principaux algorithmes réalisés

a. Menu divisées en différentes fonctions

Nous avons structuré l'application en plusieurs menus dans lesquels navigue l'utilisateur et qui le guide dans les actions qu'il peut faire et les champs qu'il doit remplir pour les effectuer. Ces menus sont affichés avec toutes les actions disponibles, des chiffres associés à chaque action et l'utilisateur entre dans la console les chiffres correspondants aux actions voulues. De ce fait, il y a beaucoup de sous menus qu'on a divisé en sous fonctions pour plus de lisibilité. Ainsi chaque fois qu'il sélectionne une action, un scanner prend ce qu'il a entré et avec des switch, on lance la sous fonction adéquate.

b. La location

La location est l'un des gros algorithmes que nous avons fait. Il consiste en une fonction rent qui associe, dans la classe du bâtiment la chambre à l'occupant. Mais qui modifie aussi la classe de l'occupant car il met dans ses attributs, la chambre qu'il loue et le bâtiment dans laquelle elle se trouve.

4.2. Justification des choix des structures de données employées

a. Hashmap et ArrayList

On a choisi d'utiliser des Hashmap :

- Pour stocker les bâtiments de la résidence. Avec des index de type String qui indique les types de bâtiment et l'objet qui est un tableau de chaque type de bâtiment. Ainsi on a accès aux tableaux des bâtiments avec leur type.
- Pour stocker la population de la résidence. En effet, l'index de type Person sera lié à un tableau de dimension 2 avec un Owner et un Occupent. Ces objets sont reliés à la même personne mais le type change en fonction de l'action qu'elle souhaite effectuer. (louer/créer un bâtiment)
- Pour stocker les locations dans les bâtiments. L'index est le numéro de la chambre ou de l'appartement et l'objet est de type Building qui correspond au bâtiment dans lequel se trouve la chambre/appartement.

Dans les autres cas, il nous semblait plus simple d'utiliser des ArrayList. Notamment pour les listes d'instruments disponibles dans les magasins ou appartenant à une personne ou encore la liste des bâtiments appartenant à un propriétaire.

b. Enum

On a également choisi d'utiliser des Enum pour certains attributs qui ne prenaient que certaines valeurs. C'est le cas pour la taille du Grand Piano ou encore le type des instruments à vent.

c. Les classe rooms/flats

Nous n'avons pas créé de classes Room ou Flat pour les chambres ou les appartements car cela nous semblait plus simple de seulement stocker un nombre de chambre/appartement.

et d'assigner des numéros de 1 au nombre de chambre. La location se fait donc sur les numéros de chambre et dans le hashmap qui stocke toutes les chambres louées.

4.3. Explication des difficultés (techniques)

a. Système de location

Nous avons mis un peu de temps pour choisir un système de location efficace. D'abord, nous avons fait une classe pour stocker le bâtiment, l'occupant et le numéro de la chambre. Mais ensuite, nous nous sommes rendu compte qu'il était plus simple de seulement stocker ces variables dans un hashmap dans le bâtiment ce qui nous permettait d'agir sur ces variables dans une fonction directement dans la classe bâtiment.

b. Gestion du temps

Nous avons eu du mal à bien répartir les tâches dans le temps avec tous les projets/évaluations que nous avons eu à rendre dans un temps très court ce qui nous a valu de devoir faire une croix sur une potentielle interface graphique qui aurait grandement embelli le projet.

5. Présentation des résultats

Dans la finalité, nous avons une application qui permet de faire toutes les fonctionnalités de l'énoncé avec certaines possibilités en plus. Nous pouvons faire une résidence avec autant de bâtiments que veut l'utilisateur. Les bâtiments ont tous un propriétaire qui peut être modifié (le propriétaire peut "léguer" son bâtiment). Les membres de la résidence peuvent tous louer une chambre ou un appartement parmi les appartements et hôtels disponibles, et tout cela avec des tests pour vérifier qu'une chambre ne peut pas être louée par deux personnes en même temps ou qu'un occupant ne peut pas louer deux locations simultanément.

6. Conclusion

6.1. Apprentissages sur le plan technique

Ce projet nous a permis d'apprendre beaucoup sur l'organisation en groupe et la programmation sur de grands projets avec de nombreuses classes qui interagissent entre elles. Nous devons donc beaucoup communiquer entre nous pour que les fonctions ajoutées par quelqu'un fonctionnent avec le travail d'un autre.

Cela nous a aussi permis d'apprendre à utiliser github et son gestionnaire de conflit qui nous a donné des difficultés.

6.2. Organisation du travail et gestion du temps

Lors du commencement de notre projet, le 12 novembre, nous nous sommes immédiatement répartis les tâches afin de commencer et d'avancer de manière efficace.

Le rendu étant prévu pour le 19 décembre, nous avons instauré une réunion hebdomadaire afin de suivre l'évolution de chacun et présenter à tous notre avancement.

Nous avons, dans un premier temps, discuté de notre compréhension du sujet afin de cibler correctement les éléments primordiaux et aller à l'essentiel dans nos tâches respectives.

Au fil des semaines, nous avons tenu un cahier de bord, accessible à tous en ligne, afin que chacun puisse suivre en temps réel la progression des différentes missions. En plus de cela, nous avons également partagé un projet GitHub afin de pouvoir tous travailler sur le même code.

Le but était également de permettre à tout le monde de s'appuyer sur le travail des uns et des autres afin de réaliser plus vite et plus précisément ses propres tâches.

Lors de nos réunions, nous avons pris conscience d'une force majeure de notre groupe de travail ; nos différents parcours académiques.

En effet, venant de filières différentes, de formation éloignés, nous avons eu l'occasion d'être confrontés à différentes méthodes de travail, de gestion des difficultés et de communication. Dans un premier temps, certains ont pu être déstabilisés, mais nous avons tous réalisé à quel point notre diversité était un atout pour nos projets.

Enfin, certains avaient eu l'occasion d'assister à des cours de gestion de projets, ce qui a été très bénéfique pour notre travail.

Pour conclure, nos qualités individuelles, notre communication et le travail approfondi de chacun a permis la réalisation d'un projet qui nous rend fier.

Pour ce qui est de la répartition du pourcentage du travail, nous l'avons récapitulé dans un graphique.

