

## Rapport de projet : BILOMO

### I. Structure du projet

Notre projet est séparé en plusieurs parties distinctes. Nous avons implémenté :

#### 1 interface :

`GestionStock` : Contient tout ce que seuls les ouvriers et les chefs « stock » peuvent faire. Possède une classe de tests. On choisit une interface car les ouvriers et les chefs « stock » héritent déjà d'une classe.

#### 10 classes :

`Personnel` : Classe abstraite car on n'instancie jamais un personnel en tant que tel. Représente l'ensemble du personnel, ouvrier ou chef, dans tout ce qu'ils ont en commun. Possède une classe de tests.

`ChefEquipe` : Classe abstraite car on n'instancie jamais un chef d'équipe en tant que tel, il est soit chef « stock », soit chef « brico ». Classe fille de `Personnel`. Représente l'ensemble des chefs (chefs « brico » ou chefs « stock »).

`ChefStock` : Classe fille de `ChefEquipe`. Implémente `GestionStock`. Représente les chefs « stock ».

`ChefBrico` : Classe fille de `ChefEquipe`. Représente les chefs « brico ».

`Ouvrier` : Classe fille de `Personnel`. Implémente `GestionStock`. Représente les ouvriers.

`Tuple` : Permet de manipuler des couples de valeurs de types différents.

`LotDePiecesDetachees` : Représente les lots de pièces détachées.

`Meuble` : Représente les meubles.

`Entrepot` : Contient toutes les méthodes liées à la gestion de l'entrepôt, en terme de stock et de personnel. Possède une classe de tests.

`Main` : Contient le déroulement de la simulation par pas de temps. On propose à l'utilisateur le mode console ou le mode lecture de fichier. Il peut ensuite choisir d'effectuer la simulation parmi deux stratégies aux objectifs différents.

#### 2 énumérations :

`Metier` : L'entrepôt ne comportant que trois types d'employés, on les définit dans cette énumération.

`PiecesMaison` : L'entrepôt ne comportant qu'une liste fixée de pièces de la maison auxquelles les meubles construits appartiennent, on les définit dans cette énumération.

Nous avons créé un package **entrepot** contenant `Entrepot` et `Main`, un package **meuble** contenant `Meuble` et `LotDePiecesDetachees`, et un package **personne** contenant `Personnel`, `ChefEquipe`, `ChefStock`, `ChefBrico`, `Ouvrier`, `GestionStock`, `Tuple`, `Metier` et `PiecesMaison`.

Les structures de données que nous avons majoritairement utilisées sont les `HashMap` et les `ArrayList`. Les `HashMap` nous permette de stocker le personnel, les chefs, les lots, et les meubles. Chaque entrée de ces dictionnaires a pour clé l'identifiant de l'objet qui y est stocké, rendant ainsi l'accès plus aisé. En ce qui concerne les `ArrayList`, nous les avons choisies car leur taille variable les rend plus pratique à utiliser que les tableaux statiques. Pour plus de détails sur la structure de notre projet, vous trouverez un diagramme de classes à la fin de ce rapport.

## **II. Stratégies de rangement et de recrutement**

Dans ce projet, nous avons implémenté deux stratégies possibles de rangement et gestion du personnel.

### **a) Stratégie 1 : objectif simplicité**

Dans cette première stratégie, nous imaginons que le personnel effectue toutes ses tâches en allant au plus facile. Afin de stocker un lot, on trouve le premier emplacement directement disponible dans l'entrepôt. Lorsqu'un lot doit être stocké ou un meuble monté, la personne mobilisée pour accomplir le travail est la première qu'on a trouvée inactive. Lorsqu'on cherche à monter un meuble, on récupère les premiers lots trouvés qui correspondent aux pièces recherchées.

En matière de gestion du personnel, on recherche également la simplicité. On ne recrute que si l'entrepôt n'est pas en déficit. On recrute un employé uniquement si on constate que la totalité du personnel était actif durant le pas de temps. On recrute un ouvrier si on lui trouve une place dans l'équipe d'un chef, sinon on recrute un chef. Si la trésorerie est négative, on licencie le premier personnel qu'on trouve inactif.

L'avantage de cette stratégie est le gain de temps : on ne réorganise jamais les lots et on ne sélectionne pas particulièrement les ressources.

Le désavantage majeur est qu'en prenant toujours les premières ressources disponibles, on ne maximise pas nécessairement les gains de l'entrepôt. De plus, on perd de la place dans l'entrepôt car on ne cherche pas à réorganiser les lots, et trouver un emplacement disponible est de plus en plus coûteux en temps car on cherche le premier emplacement disponible. Les recrutements/licenciements sont également faits sans prêter attention aux capacités de l'employé géré, ce qui pourrait ne pas être avantageux selon les besoins de l'entrepôt.

### **b) Stratégie 2 : objectif optimisation**

Dans cette deuxième stratégie, on cherche à optimiser le rangement des lots dans l'entrepôt afin de pouvoir accueillir un maximum de nouveaux lots à l'avenir. Pour cela, on stocke dans un premier temps chaque nouveau lot au premier emplacement trouvé. Puis à la fin de chaque pas de temps, on réorganise les lots dans l'entrepôt : si toute une rangée est complète, on ne touche pas aux lots qui y sont car on considère qu'on a déjà optimisé l'espace occupé dans cette rangée. Si la rangée n'est pas complète, on met les lots qu'elle contient de côté pour remanier leur position dans l'entrepôt. Une fois qu'on a mis de côté tous ces lots à réorganiser, on les replace en commençant par les plus gros. On cherche à trouver le dernier emplacement libre pour chacun de ces gros lots, puis on réitère l'opération avec des lots plus petits pour compléter les espaces vides laissés dans les rangées par les plus gros lots, jusqu'à avoir remplacé tous les lots de façon plus compacte et plus organisée (inspirée par la [parabole des gros cailloux](#), pour l'anecdote). Pour chaque déplacement de lot, un employé est mobilisé. Cependant, si malgré la réorganisation un lot n'a pas changé de place, on ne mobilise aucun employé pour lui. Pour toutes les opérations nécessitant un employé, on mobilise toujours le premier qu'on trouve inactif, mais on garde en plus une trace du nombre de fois où chaque employé a été actif au cours du pas de temps. Lors de la construction d'un meuble, on récupère les lots de pièces détachées recherchés ayant le prix par unité de volume le plus élevé, pas simplement les premiers disponibles.

En matière de gestion du personnel, on cherche à recruter et licencier de façon plus astucieuse que dans la stratégie 1. Pour le recrutement, on considère plusieurs possibilités. Si on ne trouve aucun employé apte à stocker disponible, on recrute un ouvrier si une équipe est prête à l'accueillir, sinon un chef « stock ». Sinon, de la même façon, si on ne trouve aucun employé apte à monter un meuble disponible, on recrute un ouvrier si une équipe est prête à l'accueillir, sinon un chef « brico ». Le licenciement est envisagé dans deux cas. Dans un premier cas, on licencie lorsque la trésorerie est négative. Dans un deuxième cas, on

compare tous les 5 pas de temps si l'employé actuellement le moins actif est le même que 5 pas de temps auparavant, auquel cas on le licencie.

L'avantage majeur de cette stratégie est la sélection faite sur chaque ressource afin de rendre le fonctionnement de l'entrepôt rentable. Les lots sont régulièrement réorganisés dans l'entrepôt, ce qui permet de gagner de la place pour accueillir davantage de lots. Pour la construction d'un meuble, les lots sont sélectionnés afin de correspondre aux pièces recherchées et maximiser les gains de l'entrepôt. On rend également l'entrepôt plus efficace en recrutant un employé ayant un profil adapté, et en licenciant l'employé le moins actif donc le moins rentable.

Cette stratégie peut cependant s'avérer coûteuse en personnel et en temps à cause de la réorganisation fréquente du stock. Toutefois, cette réorganisation est conçue pour ne pas stagner, étant donnée qu'on ne mobilise pas de personnel pour un lot dès lors qu'il a trouvé sa place au sein d'une rangée complète, ou n'a pas changé d'emplacement.

### III. Difficultés d'implémentation

Au cours du projet, nous sommes passées par quelques étapes qui nous ont demandé davantage de réflexion.

La réorganisation efficace des lots, gardant une mémoire des lots déjà bien stockés de pas de temps en pas de temps, a notamment soulevé de nombreuses interrogations.

Lors des tests unitaires de la classe Entrepot, nous avons également dû bien définir le cadre de nos tests afin de ne pas obtenir des résultats aberrants ou des tests inter-dépendants. En effet, beaucoup de variables communes aux tests devaient être définies et réinitialisées avec soin.

Enfin, le recrutement de la stratégie 2 a été délicat à implémenter car ce qu'on attendait de la méthode permettait difficilement d'obtenir une structure de code à la fois lisible et sans trop de répétition de code.

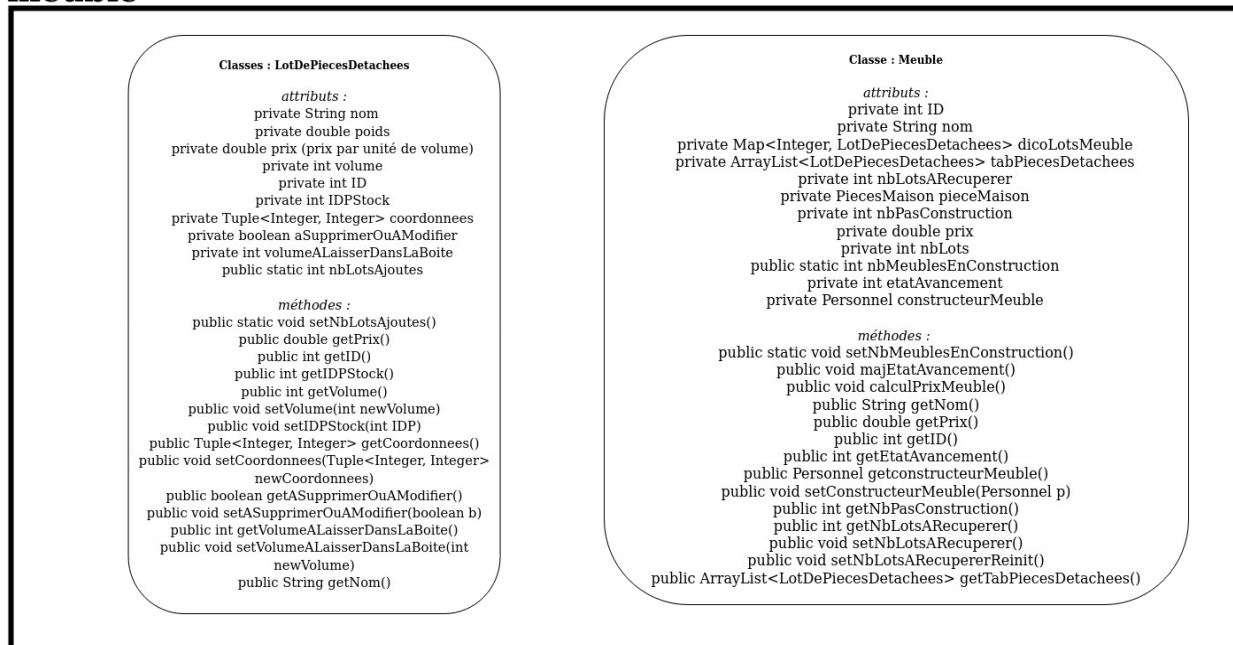
### IV. Bonus

Le nom BILOMO est inspiré de l'enseigne Miliboo, dont il est une anagramme. Miliboo est une boutique de meubles.

### V. Diagramme de classes de notre projet

#### package meuble

#### meuble



package entrepot

entrepot



package personne

personne

