

Rapport de projet

1. Présentation de la structure du code

Le projet implémente un calculateur d'empreinte carbone. Il est contenu dans 2 packages : *consoCarbone* et *users*.

Le premier package contient les classes détaillant les différentes consommations possibles, le second contient les classes utilisant les premières pour créer un utilisateur avec plusieurs postes de consommation et les gérer ou pour créer une population dans le but d'observer l'influence de politiques environnementales.

Dans le premier package se trouvent les classes *BienConsoTest*, *LogementTest*, *Alimentation*, *BienConso*, *CE*, *ConsoCarbone*, *HabitudesAli*, *Logement*, *ServicesPublics*, *Taille* et *Transport*.

Les "classes" *CE* et *Taille* sont des énumérations utilisées respectivement par *Logement* et *Transport* pour définir des attributs.

Les classes *BienConsoTest* et *LogementTest* contiennent des tests unitaires pour respectivement *BienConso* et *Logement*.

Les classes *Alimentation*, *BienConso*, *Logement*, *ServicesPublics* et *Transport* représentent chacune un type de consommation différent et héritent directement de *ConsoCarbone*. *ConsoCarbone* est une classe abstraite utilisée comme base pour les consommations. Chaque classe contient un constructeur non paramétré, un constructeur paramétré, un constructeur interactif, un constructeur pour la lecture à partir d'un fichier, des getters pour chaque attribut, des setters pour les attributs modifiables, une méthode pour calculer l'impact de la consommation en TCO2eq, une méthode *modif()* pour modifier de façon interactive un attribut choisi, une méthode *conseil()* pour donner des conseils si la consommation a un impact élevé, une méthode *toString()* pour afficher l'objet et une méthode donnant la consommation moyenne pour ce poste de consommation.

La classe *HabitudesAli* hérite d'*Alimentation* et spécifie des habitudes alimentaires en plus des consommations alimentaires générales.

Dans le second package *users* se trouvent les classes *Main*, *User* et *Population*.

La classe *Main* contient le cadre du calculateur et permet deux types d'actions.

Soit de créer un utilisateur (à la main ou à partir d'un fichier) avec la méthode *debut()* puis de donner l'empreinte carbone de l'utilisateur, le détail de l'empreinte, d'avoir des conseils, de modifier l'utilisateur ou de sauvegarder l'utilisateur dans un fichier avec la méthode *actions()*.

Soit de créer une population à partir d'un utilisateur type (à la main ou à partir d'un fichier) avec la méthode *popu()* puis de rajouter des utilisateurs à cette population ou d'appliquer des politiques environnementales qui changent l'empreinte carbone de la population avec la méthode *actions()*.

Pour créer un utilisateur à partir d'un fichier il faut que le fichier soit de la forme :

Taux de repas à base de bœuf (entre 0 et 1)	0
Taux de repas végétarien (entre 0 et 1)	1
Produits de saison ? (Y/n)	Y
Produits locaux ? (Y/n)	Y
Kg de nourriture gaspillée par an	23
Courses en vrac ? (Y/n)	n
Courses dans des e-commerces ? (Y/n)	n
Dépenses annuelles	2500
Nombre d'appartements	1
Superficie	67
Classe énergétique	D
[Répéter autant de fois qu'il y a de logements]	2
Nombre de voitures	P
Taille de la voiture	1000
Nb de km parcourus par an	5
Durée de conservation du véhicule	4000
Emissions nécessaires à la fabrication de la voiture	G
[Répéter autant de fois qu'il y a de voitures]	400
	10
	300

Par exemple le fichier suivant crée un utilisateur avec 1 logement et 2 voitures :

La classe *User* contient les méthodes permettant d'exécuter toutes les actions proposées dans *Main* ainsi que la structure d'un utilisateur.

La classe *Population* contient la structure d'une population et les méthodes qui permettent de créer et modifier une population.

2. Extensions

Comme extensions, nous avons implémenté une classe *Population* pour permettre de faire des études statistiques et d'observer l'effet de trois politiques environnementales sur l'empreinte carbone d'une population donnée. La première politique environnementale consiste à réduire la consommation carbone des services publics de 0,5 TCO₂eq. La deuxième consiste à forcer les habitants à rénover leur logement(s) si leur classe énergétique est en dessous de D. La troisième consiste à interdire l'achat de plus d'une voiture par la même personne.

De plus nous avons implémenté une option permettant de sauvegarder un utilisateur dans un fichier texte de la forme compatible avec notre calculateur. Cette option est proposée dans la fonction *actions()* en même temps que la modification d'un utilisateur, du détail de l'utilisateur, ou des conseils de consommation.

3. Difficultés d'implémentations

La première difficulté que nous avons rencontrée était de comprendre puis d'implémenter l'héritage et les interfaces pour le projet. Ce ne sont pas des difficultés très techniques mais au début du projet il nous a fallu un moment pour comprendre ces notions.

Une des difficultés du projet a été l'implémentation de l'interactivité du calculateur. En effet l'implémentation se fait à la fin du projet et demande de penser à toutes les actions qui devraient être possibles à chaque étape. De plus, cela demande de changer la structure du code et de son exécution. Cela a mis en valeur de nombreuses faiblesses du code et a pris du temps à corriger.

La création des tests JUnits a été une partie compliquée du projet. L'implémentation de ces tests n'est pas très intuitive et a demandé beaucoup de travail.

Le niveau général de difficulté du projet était correct selon nous. Nous avons dû réfléchir et faire des erreurs de nombreuses fois mais nous avons réussi à compléter presque tout ce que nous avons voulu implémenter.