
Delphine Longuet, Thibaut Balabonski, Robin Pelle, Hadi Zaatiti
longuet@lri.fr, blsk@lri.fr, pelle@lri.fr, hadizaatiti@gmail.com

TD 3 - Premiers pas de test

Semaine du 3 octobre 2016

Exercice 1

Dans un cahier des charges on trouve le texte suivant à propos d'une fonction de calcul d'un impôt sur les véhicules polluants :

« Afin de favoriser l'utilisation de véhicules moins polluants, l'état a instauré une taxe à laquelle sont soumis les citoyens possédant une voiture. Cette taxe doit être déclarée et versée le 15 janvier de chaque année par chaque personne possédant une voiture, et ce pour chaque voiture indépendamment.

Le montant de cette taxe est calculé en fonction de la puissance en cheval-vapeur (ch) de la voiture, du carburant utilisé et du nombre de kilomètres parcourus depuis la dernière déclaration. Le but de cette taxe est de favoriser les comportements les moins polluants : plus le véhicule est polluant, plus le montant de la taxe est élevé. Le calcul du montant de cette taxe se fait de la manière suivante.

Pour les voitures à essence, le montant de base est 200€, il est de 250€ pour les moteurs diesel et 120€ pour les moteurs purement électriques.

La puissance du moteur est également prise en compte : un moteur plus puissant consomme plus et est donc plus polluant. Ainsi, pour une voiture dont la puissance ne dépasse pas 150ch, on applique une première réduction de 20%. Les moteurs diesel étant particulièrement polluants, la puissance du moteur n'entre pas en compte dans le calcul du montant pour ce type de véhicules.

L'administration considère qu'un véhicule parcourt en moyenne 10 000km par an. Chaque propriétaire doit déclarer le nombre de kilomètres parcourus en un an par le véhicule (en arrondissant au millier inférieur) : le montant de base est multiplié par le rapport entre le nombre de kilomètres parcourus et le nombre moyen de kilomètres fixé à 10 000. Cependant, afin d'encourager les propriétaires les plus polluants à changer leurs habitudes, aucune réduction liée au nombre de kilomètres parcourus ne peut s'appliquer ni aux moteurs diesel, ni aux moteurs à essence d'une puissance supérieure à 150ch.

L'administration a décidé de fixer à 40€ le montant minimal de cette taxe, les voitures électriques n'étant pas concernées par ce minimum. »

1. Construire un ensemble de cas de tests pour cette fonction. On les présentera sous la forme d'un tableau avec, pour chaque test : l'objectif du test, des valeurs concrètes pour les paramètres (type du moteur, puissance du véhicule, kilomètres parcourus), et le résultat attendu du test.
2. Donner quelques exemples de tests aux limites à ajouter aux tests précédents.
3. Une révision de cette taxe demande de prendre en compte également l'âge du véhicule, depuis la date de sa première immatriculation. Le montant de la taxe est augmenté de 5% pour les véhicules qui ont entre 2 et 5 ans, de 10% pour ceux qui ont entre 5 et 10 ans, et de 15% pour les véhicules de plus de 10 ans. Pour les véhicules électriques,

cette augmentation est de 4% et 6%, la dernière catégorie étant les véhicules de plus de 5 ans. Donner un ensemble d'objectifs de test pertinent pour le nouveau calcul de cette taxe, sans essayer de couvrir toutes les combinaisons possibles des critères.

Exercice 2

On considère une classe `Tableau` permettant de stocker des entiers dans un tableau trié. Le nombre d'éléments du tableau est stocké dans un attribut `taille` et la longueur du tableau dans un attribut `capacite`. La taille est toujours supérieure ou égale à 0, la capacité toujours strictement positive, et la taille inférieure ou égale à la capacité. En particulier, on ne peut pas appeler le constructeur de la classe avec un argument négatif.

```
public class Tableau {
    private int tab[];
    private int taille;
    private int capacite;

    public boolean supprimer(int val) {
        int i = 0;
        while(i < taille && tab[i] != val) {i++;}
        if(i == taille) {return false;}
        for(int j = i; j < taille-1; j++) {tab[j] = tab[j+1];}
        tab[taille-1] = 0;
        taille = taille-1;
        return true;
    }

    public boolean inserer(int val) {
        int i;
        if(taille == capacite) {return false;}
        for(i = taille; i > 0 && tab[i-1] > val; i--) {tab[i] = tab[i-1];}
        tab[i] = val;
        taille = taille+1;
        return true;
    }
}
```

On cherche à tester les deux méthodes `inserer` et `supprimer` de cette classe, de manière « interne », c'est-à-dire en ayant accès à tous les attributs. Si le tableau n'est pas plein, la méthode `inserer` ajoute l'élément passé en argument au tableau en conservant l'ordre et renvoie `true`. Si le tableau est plein, elle renvoie `false`. La méthode `supprimer` enlève une occurrence de l'élément passé en argument et renvoie `true`. Elle renvoie `false` si l'élément n'était pas présent.

1. Proposer un ensemble de tests pour la fonction `supprimer`.
2. Pour les modifications suivantes du programme, dire si la modification introduit une faute et lorsque c'est possible, donner un cas de test permettant de révéler la faute.
 - (a) On modifie la condition du `if` par `i == taille-1`.
 - (b) On modifie la condition d'arrêt de la boucle `for` par `j < taille`.
 - (c) On modifie l'initialisation de la boucle `for` par `j=i+1`.
 - (d) On change `taille` par `capacite` dans la condition du `while` et du `if`.

- (e) On supprime la ligne `tab[taille-1] = 0`.
- (f) On fait les modifications (d) et (e) ensemble.

3. Pour chacun des tests suivants, donner une implémentation de la fonction `insérer` qui échoue sur ce test, en apportant des modifications simples au code donné.

	tab	taille	capacite	val
Test 1	[3,5,6,8]	4	4	7
Test 2	[2,4,5]	3	8	1

4. On teste maintenant ces méthodes depuis l'extérieur de la classe. De quels observateurs aurait-on besoin idéalement pour pouvoir détecter autant de fautes qu'avec les tests internes ?
5. On ajoute les méthodes suivantes à la classe `Tableau` :

```
int taille()
int capacite()
boolean present(int val)
```

Quels aspects de l'implémentation ne peut-on plus observer ? Peut-on les tester ? Comment ?

6. Pour deux des tests proposés en question 1, construire la suite d'appels de méthodes (le scénario de test) permettant d'exécuter ce test, depuis l'initialisation des objets jusqu'à la vérification du résultat obtenu.