
Programmation défensive et tests unitaires

Notions importantes

- boîte noire
- boîte blanche
- programmation par contrat
- précondition
- postcondition
- invariant
- assertion
- (vers les) tests unitaires

Exercices

Exercice 1. On se place dans le cadre du *jeu de memory* (description de wikipedia reproduite ci-dessous).

Le jeu se compose de paires de cartes portant des illustrations identiques. L'ensemble des cartes est mélangé, puis étalé face contre table. À son tour, chaque joueur retourne deux cartes de son choix. S'il découvre deux cartes identiques, il les ramasse et les conserve, ce qui lui permet de rejouer. Si les cartes ne sont pas identiques, il les retourne faces cachées à leur emplacement de départ.

Le jeu se termine quand toutes les paires de cartes ont été découvertes et ramassées. Le gagnant est le joueur qui possède le plus de paires.

1. On se place dans le cadre d'une modélisation du M de MVC, en particulier les aspects gestion de la sauvegarde de l'état du jeu. Réfléchissez rapidement aux classes possibles et à leur rôle (Si vous ne trouvez pas de nom clair pour une classe c'est qu'elle essaye de faire trop de choses).
2. Quelles sont les préconditions et postconditions de la méthode correspondant à l'action de retourner 2 cartes d'un joueur? (on suppose que cette méthode est très bête elle ne fait que retourner les cartes. Une autre méthode se chargera de vérifier si les cartes sont égales et de faire un déplacement, probablement dans une autre classe que celle des cartes).
3. Même question pour cacher 2 cartes
4. Quelles propriétés sont invariantes par l'une et l'autre de ces méthodes?
5. En supposant qu'on ait une classe Carte, quel invariant envisagez vous pour cette classe?

Exercice 2. On considère un algorithme de tri opérant sur un tableau d'entiers.

1. Quelle est la post-condition de l'algorithme ?
2. On suppose que l'algorithme utilisé est le suivant.

```
procédure tri_selection(tableau t)
  n := longueur(t)
  pour i de 0 à n - 2
    min := i
    pour j de i + 1 à n - 1
      si t[j] < t[min], alors min := j
    fin pour
    si min != i, alors échanger t[i] et t[min]
  fin pour
fin procédure
```

Proposez un invariant pour la boucle externe (celle de paramètre i).

Exercice 3. On considère l'algorithme de recherche binaire d'un entier dans un tableau.

1. Expliquez en français l'idée de l'algorithme.
2. Donnez le pseudo-code
3. Quelle est la précondition de cet algorithme ?
4. Quelle propriété est vraie à chaque étape de l'algorithme ? (dans une boucle si vous donnez un algo avec une boucle, dans un appel récursif si vous utilisez cette alternative).