

## TP n° 6

### Quizz et plus

**Exercice 1** Implémenter en entier le jeu Quizz vu en TD. Il est conseillé de suivre l'ordre du TD et de tester chaque nouvelle classe/interface.

En l'occurrence, il faut écrire :

- La classe Quizz.
- Les différents types (interfaces, classes abstraites ou non) de questions.
- Les classes et méthodes de test.

**Exercice 2** La classe Quizz pose les questions de sa liste dans l'ordre. Ajouter une classe BanqueAleatoire qui possède une grande quantité de questions et en pose au hasard (jamais deux fois la même) jusqu'à atteindre une note maximale que l'on peut obtenir (par exemple, 20).

Il ne faut bien évidemment pas copier-coller la gestion d'entrée-sortie de la classe Quizz vers la nouvelle classe BanqueAleatoire mais externaliser celle-ci dans une nouvelle classe EntreeSortie.

**Exercice 3** Écrire une classe FabriqueQuizzInteractive qui permet à l'utilisateur de créer un Quizz de façon interactive (ajout d'une question de la banque, fonction recherche dans la banque, suppression d'une question du Quizz, déplacement d'une question, etc.).

**Exercice 4** Programmer comptage des points :

1. Proposer une manière d'ajouter la gestion des points.
2. Une même question posée à des élèves de L1 ou de L2 ne vaudra pas le même nombre de points. Comment peut-on faire cela sans créer plusieurs fois la même question. (Dans le cas des QCM, il est de plus usuel d'attribuer des points négatifs en cas de mauvaise réponse).

**Exercice 5** Permettre des questions appelant plusieurs réponses et dont le score est fonction du nombre de réponses données, comme dans les exemples suivants :

- *Donner le nom de 5 planètes du système solaire.*
- *Donner 10 nombres premiers.*

L'implémentation doit être suffisamment générale pour prendre en compte comme cas particuliers les cas suivants :

- 1 seule réponse
- plusieurs réponses, tous les points si une bonne réponse est donnée
- x points par bonne réponse, y points en moins par mauvaise réponse
- tous les points si toutes les bonnes réponses sont données, 0 sinon

Pour cela, écrire un nouveau type capable d'englober tous les cas ci-dessus, et réécrire le type `Question` pour qu'il dérive de celui-ci (ainsi tous les types de question déjà écrits héritent aussi du nouveau type).

Écrivez de nouvelles classes dérivées pour les cas particuliers que vous n'avez pas encore programmés.