2015-2016

Évaluation de calcul mental

Projet de Conduite de Tests

**Etudiants :**

COLMART Benjamin

COQUELET Jean

RENAULT Kévin

WESCHLER Tom

Projet de Conduite de Tests :   
**Évaluation de Calcul Mental**

# Présentation de l’Application

## Objectif de l’application

Afin de réaliser les tests de ce projet, il était nécessaire de construire une application Java que nous pourrions donc tester au fur et à mesure de son développement. Nous avons donc entreprit diverses modélisations et discussions afin d’aboutir à une architecture et un fonctionnement bien définie de l’application.

L’objectif de l’application est de réaliser et évaluer des tests de calculs mentaux. Les calculs mentaux proposés devront être de plusieurs niveaux (facile, moyen et difficile). Plusieurs calculs composent un test, et celui-ci est associé à un nom d’utilisateur/de joueur. Nous proposerons 3 scénarios de test (facile, moyen et difficile).

Aussi, les scores réalisés seront évalués en fonction de ces 3 niveaux. Une valeur numérique sera d’abord déterminée dans le programme, qui sera transformé en notation américaine afin d’attribuer une lettre au test réalisé. Les notes iront de F- à A+ voir même S+ pour les utilisateurs répondant en moins d’une demi-seconde.

Les résultats du test serviront ensuite à créer un objet de la classe « Score ». Celui-ci sera ensuite ajouté aux listes de scores séparées selon les niveaux de test. On pourra mettre en place différentes méthodes de tri des listes afin d’afficher les scores réalisés sur l’application.

Dû aux temps imparti sur ce projet et au contexte dans lequel il est réalisé (nombreux autres projets en cours), nous n’irons pas jusqu’à l’implémentation de méthode de gestion des scores…etc. Cependant, il sera possible d’afficher les contenus des différentes listes triés par valeurs par exemple.

L’interface qui sera réalisé devra donc être en mesure de :

* Proposer de réaliser un test
  + De choisir un niveau
  + De définir un nom d’utilisateur (qui sera proposé par défaut ensuite)
* Donner le résultat de ce test à l’utilisateur (qui sera automatiquement enregistré)
* Permettre à l’utilisateur de visionner les différents scores réalisés dans les différents niveaux de jeu

## Architecture de l’application

L’application est composée de 5 classes dédiés à son fonctionnement sans compter l’IHM. On dénombre cependant 3 classes principales qui sont :

* Évaluation, qui est la classe tirant parti de toutes les autres. Elles possèdent cependant ses propres attributs et méthodes qui lui servent à évaluer chacun des tests qui sont réalisés
* Calcul, la classe permettant de générer les différents calculs et résultats.
* Scores, la classe qui permet de sauvegarder les différentes valeurs des tests réalisés par les utilisateurs

Voici un schéma de représentation des dépendances fonctionnelles des classes de l’application. Nous allons vous présenter ces classes plus en détails.



Figure 1 - Architecture et dépendance fonctionnelle des classes

### La classe Scores

La classe « Scores » est en résumé, composé de plusieurs listes d’objet de la classe « Score » (au singulier). Elle permet la gestion des résultats des différents tests. On implémente donc des méthodes de sauvegarde et de chargement via une sérialisation des objets de l’application.

Aussi, la sauvegarde des scores réalisés serait vaine si nous n’implémentions pas de méthodes afin de les trier et les visualiser.

### La classe Calcul

Cette classe fonctionne comme un générateur de calcul. Elle possède de nombreux attributs statiques. On peut ainsi choisir de ne l’utiliser qu’une fois pour chaque niveau, ou une fois pour chaque calcul. Dans les attributs statiques on distingue notamment le Random Seed, qui permet de générer des calculs différents à chaque génération aléatoire, mais aussi d’autres attributs comme les tableaux des limites des nombre qui n’ont besoin d’être définis qu’une seule fois.

Ces tableaux servent à fixer des limites hautes et basses pour les nombres générés en fonction du niveau du calcul ET de l’opération à réaliser (ex : addition de 1 à 100 au niveau moyen, de 50 à 150 au niveau difficile, et multiplication de 1 à 10 au niveau moyen, de 5 à 12 au niveau difficile).

On doit prendre bien soin de borner correctement les calculs qui seront à réalisés afin de rester dans le thème du calcul **mental**.

### La classe Évaluation

Comme illustré en Figure 1, la classe « Evaluation » s’appuie sur toutes les autres classes afin d’organiser les tests. Le chronomètre de la classe Chrono sert à évaluer les temps de chaque calcul de l’utilisateur, ainsi qu’un chrono. Un tableau de calculs est créé à l’aide de la classe « Calcul ». Enfin, les résultats des tests sont des objets de la classe « Score ». Cet objet Score sera alors renvoyé par la classe Evaluation, pour permettre sa sauvegarde dans la classe « Scores ».