

Jeu - Calcul mental

Préambule

La notation sera faite comme suit :

- $\frac{1}{4}$ **Comportement** : pendant les sessions (savoir être, respect de la vie de classe, etc)
- $\frac{1}{2}$ **Connaissances & compétences** : Montrer que vous avez appris en POO, Java / EE
- $\frac{1}{4}$ **Production** : Livrer quelque chose qui fonctionne même partiellement. Une attention particulière sera donnée à la qualité (respect des normes, architecture et qualité des livrables) comme une livraison faite à un client.

La livraison sera faite au moyen d'un lien Git ou d'une archive unique (zip), aux prénoms et noms des personnes du groupe (2 à 3 personnes max par groupe), envoyés par mail « sega.sylla@epsi.fr » au plus tard le **20/10/2019**.

Le jeu

Le but est de développer une application web d'entraînement au « calcul mental ». Elle proposera les cas d'utilisations suivants :

1. Les utilisateurs devront s'identifier afin d'accéder au jeu avec un couple de login et password ; Ils devront alors arriver sur une page d'accueil présentant les 10 meilleurs scores (tous joueurs confondus).
2. Sur la page d'accueil, l'outil devra proposer une action pour démarrer une nouvelle partie ; Ensuite 10 calculs successifs aléatoires seront proposés auxquels l'utilisateur devra répondre ;
3. Le score (nb de bonnes réponses) pour chaque partie sera enregistré en base de données. Et il faudrait être capable de retrouver facilement le meilleur score de chaque joueur sans avoir à parcourir son historique de jeu ;
4. Le classement des meilleurs scores est alors affiché (tous joueurs confondus) avant de proposer un lien vers l'accueil de l'application ;
5. L'utilisateur devra pouvoir faire autant de parties qu'il le souhaitera.

Générateur de calcul - Expression's computing

Les calculs à effectuer seront générés automatiquement. Ils pourront contenir plusieurs opérateurs et plusieurs opérandes. On considère une expression post-fixée sous forme d'une chaîne de caractères pouvant comprendre :

- les opérations binaires : +, -, *, /
- les opérateurs unaires **inverse** et **racine carrée** notés *inv* et *rac*

Exemples : 35 7 + 8 * rac et 4.25 8 - 3 rac *

On peut évaluer une telle expression à l'aide d'une pile. Pour chaque composant de la chaîne :

- S'il correspond à un nombre, on empile sa valeur
- Si c'est un opérateur binaire, on dépile deux valeurs, on effectue l'opération et on empile le résultat
- Si c'est un opérateur unaire, on dépile une valeur, on effectue l'opération et on empile le résultat
- Si l'expression est générée est correcte (chaîne est bien formée), à la fin du traitement il reste une valeur dans la pile, c'est le résultat de l'évaluation.

Pour cela, vous pourrez définir à minima :

- Une classe Operateur ; Cette classe comprendra des méthodes permettant de déterminer si un opérateur est binaire ou unaire et des méthodes de calculs ;
- Une classe Expression où seront générés aléatoirement les expressions et le calcul de leurs évaluations à l'aide de l'implémentation « *java.util.Stack<E>* »

Des points bonus au TP seront attribués aux solutions les plus innovantes.

En plus d'un vrai développement objet bien structuré, une attention particulière sera portée au fait d'avoir une application bien découpée avec les responsabilités attendues dans les différentes couches.

La base de données

Vous êtes libres d'utiliser n'importe quelle base de données relationnelle vue en cours en appuyant sur l'API JDBC

Travail à faire

- Analyser et concevoir le modèle objets de cette application ;
- Concevoir une base de données pour stocker les informations ;
- Développez une application Java EE en implémentant les différentes classes et les méthodes demandées ; vous utiliserez les technologies Servlet, JSP (JSTL et EL) ainsi que les Filters vues en cours ;