Révision des tables de multiplications

Lors d'un TP précédent vous avec implémenté un questionnaire proposant plusieurs opérations de multiplications pour une table de multiplication tirée au hasard, de façon procédurale avec des méthodes de classe (méthodes ***static***) :

Exemple : Si la table tirée au hasard est la table de 4, alors vous aurez, pour 5 opérations proposées :

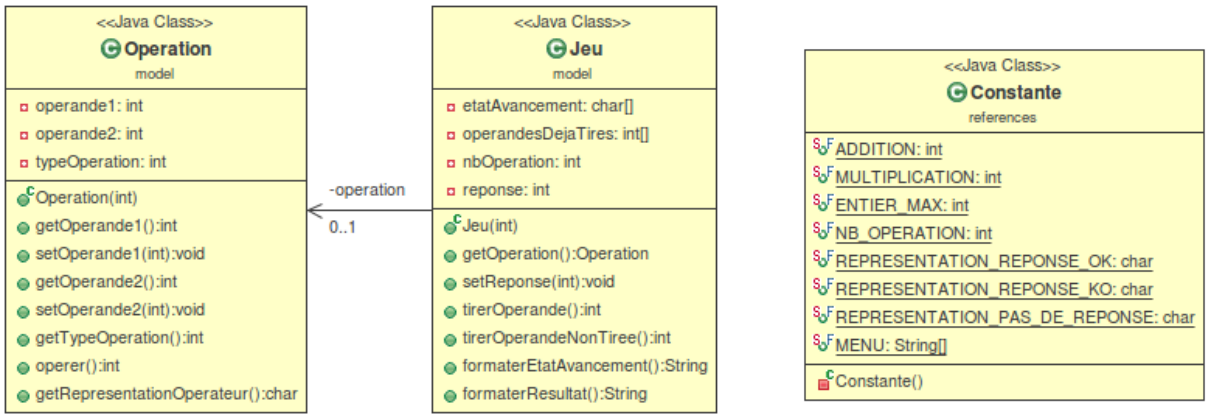
```
4 x 4 = 16
Bonne réponse
-....
4 x 10 = 4
Mauvaise réponse, 4 x 10 = 40
-x...
4 x 8 = 32
Bonne réponse
-x-..
4 x 9 = 36
Bonne réponse
-x--.
4 x 9 = 35
Mauvaise réponse, 4 x 9 = 36
-x--x
Vous avez 3 bonnes réponses sur 5
```

Réponse de l'utilisateur

Etat d'avancement du jeu :

- \- Bonne réponse
- x Mauvaise réponse
- . pas de réponse

Il s'agit maintenant de coder les classes suivantes afin de proposer le même type d'opération. Votre application devra, outre les opérations de multiplications, proposer également les opérations d'additions. Ce schéma est un diagramme de classe représentant les attributs et les méthodes des classes.



Vous élargirez la fonctionnalité en proposant, en plus des opérations d'additions.

Classe ***Operation***	Représente une opération sur 2 opérandes. L'attribut ***typeOperation*** est un entier pouvant contenir la constante ***ADDITION*** ou ***MULTIPLICATION***. La méthode ***operer()*** retourne le résultat de l'opération des 2 opérandes. La méthode ***getRepresentationOperateur()***, retourne le symbole graphique de l'opération (**+** ou **x**).
Classe ***Jeu***	L'attribut ***etatAvancement*** représente l'état d'avancement du jeu : `--x..` L'attribut ***operandesDejaTirees*** stocke la liste des opérandes déjà proposées, pour éviter de proposer plusieurs fois la même opération. C'est un attribut technique, non accessible. L'attribut ***nbOperation*** compte le nombre d'opérations proposées. La méthode ***tirerOperande()*** retourne un entier tiré aléatoirement. La méthode ***tirerOperandeNonTiree()*** retourne un entier qui n'a pas été déjà proposé. La méthode ***formaterEtatAvancement()*** retourne une chaine de caractères indiquant si la réponse est bonne ou mauvaise et l'état d'état d'avancement graphique. La méthode ***formaterResultat()*** retourne une chaine de caractères indiquant le score.
Classe ***Constante***	Classe utilitaire regroupant les constantes de l'application. La constante ADDITION prend la valeur 0 et MULTIPLICATION la valeur 1. La constante MENU contient les options pour le choix d'opération.

Puis, dans une classe Lanceur, vous coderez la méthode main proposant le déroulé suivant :

```
1. addition
2. multiplication
CHOIX :
1

5 additions sur la table de 7
7 + 6 = 13
Bonne réponse
-....

7 + 8 = 15
Bonne réponse
--...

7 + 5 = 11
Mauvaise réponse, 7 + 5 = 12
--x..

7 + 3 = 10
Bonne réponse
--x-.

7 + 9 = 16
Bonne réponse
--x--

Score : 4 / 5
```

Menu du choix d'opération.

Opérations proposées.

Etat d'avancement.

Vous disposez de la bibliothèque CLRUtilConsole.jar contenant notamment la classe utilitaire ***LectureConsole***.