HE2B ESI

Nom : _____

Prénom : _____

Groupe : _____

Identifiant : _____

/ 20

Haute École de Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

27 avril 2017

Développement – Q2

Interrogation 1

Yahtzee

- *Vous réaliserez votre travail avec* `Netbeans` *et le remettrez à votre professeur via* `git` *sur un* repository *créé pour l'occasion.*
- *Vous disposez de toutes vos notes ainsi que de l'aide en ligne.*
- *Vous avez deux heures de temps.*
- *Il ne suffit pas que votre code compile. Testez-le pour identifier d'éventuelles erreurs à l'exécution.*
- *La cotation tiendra compte aussi du style de programmation que vous avez acquis.*
- *Nous ne demanderons jamais d'écrire la Javadoc. Il* ***faut*** *l'écrire.*
- *Vous écrirez la Javadoc en anglais.*
- *Durant ce travail* g12345 *représente (évidemment) votre login*

1 Description du programme (0 point)

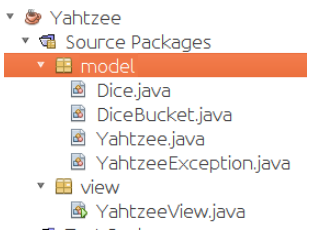
Nous allons développer une version simplifiée du jeu de dés Yahtzee. Le but est d'enchaîner les combinaisons à l'aide de cinq dés pour remporter un maximum de points.

À chaque lancé de dé, le joueur choisi quels dès mettre de coté afin de maximiser son score. Une fois les dès sélectionnés mis de coté, le joueur peut relancer les dès encore en jeu. Une partie de Yahtzee se termine :

— si on a effectué plus de trois lancés ;

— si il n'y a plus de dès à mettre de coté.

Afin de programmer ce jeu, nous allons créer un projet dans `NetBeans` appelé `Yahtzee`. Ce projet sera constitué du package `model` qui contient quatre classes décrites dans la suite de ce document ainsi que du package `view` qui contient toutes les interactions avec l'utilisateur.



Vous trouverez ci-dessous un exemple d'éxécution du jeu.

Version du document : 25 avril 2017

```
----- ----- ----- ----- ----- ----- -----
        Tour numero 1 sur 3
----- ----- ----- ----- ----- ----- -----
Le score est actuellement de 0

dés en jeu      :       2       1       2       5       6
dés mis de coté :
Souhaitez-vous mettre un dè de coté (o/n)?
o
Entrez le numéro du dé que vous souhaitez conserver
Numéro entre 1 et 5
5
dés en jeu      :       2       1       2       5
dés mis de coté :       6
Souhaitez-vous mettre un dè de coté (o/n)?
n
----- ----- ----- ----- ----- ----- -----
        Tour numero 2 sur 3
----- ----- ----- ----- ----- ----- -----
Le score est actuellement de 6

dés en jeu      :       4       1       5       4
dés mis de coté :       6
Souhaitez-vous mettre un dè de coté (o/n)?
n
----- ----- ----- ----- ----- ----- -----
        Tour numero 3 sur 3
----- ----- ----- ----- ----- ----- -----
Le score est actuellement de 6

dés en jeu      :       6       4       3       1
dés mis de coté :       6
Souhaitez-vous mettre un dè de coté (o/n)?
o
Entrez le numéro du dé que vous souhaitez conserver
Numéro entre 1 et 4
1
dés en jeu      :       4       3       1
dés mis de coté :       6       6
Souhaitez-vous mettre un dè de coté (o/n)?
n
----- ----- ----- ----- ----- ----- -----
        Partie terminée
----- ----- ----- ----- ----- ----- -----
Votre score final est de 20
dés en jeu      :
dés mis de coté :       6       6       4       3       1
----- ----- ----- ----- ----- ----- -----
```

2 Classe `Dice` (2 points)

La classe `Dice` correspond aux dés du jeu. Elle possède comme unique attribut un entier `value` contenant la valeur du dé (de 1 à 6).

Cette classe bénéficie au minimum des méthodes ci-dessous [1] :

- un constructeur qui initalise la valeur du dé à 6 ;
- un accesseur mais aucun mutateur ;
- `roll` : qui lance le dé et choisi une valeur aléatoire comprise entre 1 et 6.

Une fois cette classe terminée, faites un **commit** de votre projet.

3 Classe `DiceBucket` (3 points)

La classe `DiceBucket` correspond à l'ensemble des dés du jeu. Il possède les attributs suivants :

- `actives` : la liste des dès encore en jeu ;

1. Pensez à développer des méthodes additionnelles afin d'améliorer la lisibilité et la réutilisabilité du code

— `asides` : la liste des dès mis de cotés pour être conservés.

Cette classe bénéficie au minimum des méthodes ci-dessous :

— un constructeur qui initalise les listes de dès et place les 5 dès dans la liste des dés actifs ;

— `roll` : qui lance tous les dès encore en jeu ;

— `putAside(index)` : qui met de coté le dè d'indice donné en paramètre ;

— `putAllAside` : qui met de coté tous les dès encore en jeu ;

— `getNbAsides(int value)` : qui compte le nombre de dès mis de cotés avec une certaine valeur ;

— `getNbActives` : qui retourne le nombre de dés encore en jeu ;

— `isEmpty` : qui indique si il reste des dès encore en jeu.

Une fois cette classe terminée, faites un **commit** de votre projet.

4 Classe `Yahtzee` (6 points)

La classe `Yahtzee` représente le jeu avec ses règles. Cette classe possède les attributs suivants :

— `bucket` de type `DiceBucket` : l'ensemble des dès du jeu ;

— `nbMoves` : le nombre de lancés de dès réalisé.

Cette classe bénéficie au minimum des méthodes ci-dessous :

— un constructeur qui initialise l'ensemble des dès et initialise le compteur des lancés à 0 ;

— `isOver` : une méthode qui détermine si la partie est terminée ;

— `roll` : lance les dès encore en jeu ;

— `putAsides(int index)` : qui met de coté le dè d'indice donné en paramètre ;

— `putAllAside` : qui met de coté tous les dès encore en jeu ;

— `getNbMoves()` : qui retourne le nombre de lancé effectué ;

— `getScore` : qui comptabilise le score du joueur. Chaque dè rapporte sa valeur en point.

Une fois cette classe terminée, faites un **commit** de votre projet.

5 Gestion des Exceptions et tests unitaires (3 points)

Écrivez une classe `YahtzeeException` qui hérite de `Exception`. Modifiez la méthode `putAsides(int index)` de la classe `Yahtzee` afin qu'elle renvoie une `YahtzeeException` si l'indice donné en paramètre est incorrect. Ensuite créez un test unitaire JUnit pour vérifier que l'exception est bien lancée quand il faut. Une fois ces modifications terminées, faites un commit de votre projet.

6 Classe `YahtzeeView` (6 points)

Cette classe du package `view` gère l'affichage du jeu. Elle contient le point d'entrée du programme. Ce `main` :

1. instancie un jeu ;
2. boucle tant que le jeu n'est pas terminé ;
   (a) affiche l'état du jeu ;
   (b) lance les dès encore en jeu ;
   (c) tant que le joueur souhaite conserver un dè ;
      i. demande au joueur quel dè il souhaite conserver ;
      ii. met le dè demandé de coté ;
3. mets les dès restant de cotés.

Une fois cette classe terminée, faites un **commit** de votre projet.

7 Finalisation (0 point)

Déposez votre projet complet sur le *repository git* créé par votre votre professeur **pour cette interrogation**. Il sera de la forme
https ://git.esi-bru.be/javl-XXX-20162017/i-12345.git.