

Evaluation 1 - partie pratique

Le Blackjack

Travail à accomplir (temps à disposition : 120 minutes)

Le jeu du Blackjack est un jeu de hasard basé sur des cartes à jouer. Le principe est d'obtenir le score s'approchant le plus de 21 sans dépasser cette valeur. A tour de rôle, chaque joueur tire une carte et on additionne à son score la valeur représentée par la carte.

Voici les règles pour cette version simplifiée du Blackjack :

- Les cartes sont représentées par un nombre allant de 2 à 14. Les nombres 11, 12, 13 et 14 représentent respectivement : le valet, la dame, le roi et l'as.
- Afin de calculer les scores, pour les cartes de 2 à 10, on utilisera les valeurs nominales (2,3,4,...) tandis que pour les cartes habillées, la valeur sera toujours de 11 (y compris pour l'as, contrairement à la règle normale qui veut que l'on puisse choisir entre 1 ou 11 pour la valeur de cette carte).
- La partie se déroule entre trois joueurs. Elle se termine lorsqu'un joueur obtient le score de 21 (dans ce cas il est déclaré vainqueur) ou lorsqu'un joueur dépasse le score de 21 (dans ce cas, le vainqueur est le joueur dont le score s'approche le plus de 21 sans dépasser cette valeur).



Voici les étapes à réaliser pour implémentez ce programme :

1. Créez un nouveau projet Java dans NetBeans que vous nommerez **403_NOM_E1** (exemple 403_ROUILLER_E1).
2. Créez les constantes **MIN** et **MAX** qui contiendront des entiers avec les valeurs 2 et 14.
3. Déclarez et implémentez une méthode nommée **tirerUneCarte()** qui retournera une valeur aléatoire comprise entre **MIN** et **MAX**.
4. Déclarez et implémentez une méthode nommée **determinerIndexGagnant()** qui prendra en paramètre un tableau d'entiers nommé **scores** et qui retournera un entier. Cette méthode nous permettra de déterminer l'index dans le tableau de la valeur qui se rapproche le plus de 21. En d'autres termes, il faut déterminer la position de la valeur maximale sans prendre en considération les valeurs dépassant 21. Exemple : si le tableau passé en paramètre est le suivant :

0	1	2
22	19	17

Dans cet exemple, La valeur retournée par la méthode sera « 1 », ce qui correspond à la position de la valeur qui se rapproche le plus de 21 sans la dépasser.

5. Déclarez et implémentez une méthode nommée `donnerNomCarte()` qui prendra en paramètre un entier nommé `carte` et qui retournera un String. Cette méthode nous permettra de donner le nom d'une carte en fonction de sa valeur.

- 5.1. Si la valeur se situe entre 2 et 10 (y compris), on utilisera directement la valeur nominale comme nom de carte. Pour cela, vous devrez convertir un entier en String, voici, sous forme d'exemple, l'instruction qui vous permettra de réaliser cette opération :

```
int valeur = 12 ;  
String resultat = String.valueOf(valeur);
```

- 5.2. Si la valeur est supérieure à 10, déterminez le nom de la carte à l'aide d'un switch : 11 → « valet », 12 → « dame », 13 → « roi » et 14 → « as ». Dans le switch, si la valeur ne correspond à aucune des valeurs mentionnées précédemment (11, 12, 13 ou 14), retournez « indéterminé ».

6. Dans la méthode main, réalisez les opérations suivantes :

- 6.1. Déclarez et créez un tableau de String nommé `joueurs` dans lequel vous stockerez les valeurs suivantes : « Jacques », « Pierre » et « Paul ».

- 6.2. Déclarez et créez un tableau d'entiers nommé `scores` qui sera de la même taille que le tableau `joueurs`.

- 6.3. Déclarez une variable de type booléenne nommée `jeuTermine` à laquelle vous affecterez la valeur `false`.

- 6.4. Tant que le jeu n'est pas terminé (la variable `jeuTermine` a la valeur `false`) :

- 6.4.1. Pour chaque joueur :

- 6.4.1.1. Appelez la méthode `tirerUneCarte()` et stockez le résultat dans une nouvelle variable que vous nommerez `carte`.

- 6.4.1.2. Si la valeur de `carte` est supérieure à 10, ajoutez 11 au score du joueur. Le score est contenu dans la cellule du tableau `scores` dont la position est identique à celle du joueur dans le tableau `joueurs` (le joueur 0 a son score dans la cellule 0 du tableau `scores`, le joueur 1 a son score dans la cellule 1 du tableau `scores`, etc...)

- 6.4.1.3. Sinon, ajoutez la valeur de la variable `carte` au score du joueur.

- 6.4.1.4. Appelez la méthode `donnerNomCarte()` et stockez le résultat dans une nouvelle variable que vous nommerez `nomCarte`.

- 6.4.1.5. Affichez à la console le nom du joueur et le nom de la carte qu'il a tirée. Exemple :

```
Jacques a tiré la carte roi
```

- 6.4.1.6. Affichez à la console le nouveau score du joueur (ajoutez trois espaces au préalable pour faciliter la lecture → voir exemple de sortie à la console à la fin du document) :

```
Nouveau score : 15
```

- 6.4.1.7. Si le score du joueur est désormais égal à 21, indiquez que le jeu est terminé (avec la variable `jeuTermine`) et affichez que le joueur en question a gagné. Exemple :

```
Paul a gagné !
```

- 6.4.1.8. Sinon, si le score du joueur est désormais plus grand que 21, indiquez également que le jeu est terminé (avec la variable `jeuTermine`) et appelez la méthode `determinerIndexGagnant()`

afin de connaître la position qui désigne le joueur gagnant dans le tableau **joueurs**. Affichez ensuite que ce joueur a gagné de manière analogue au point précédent.

6.4.1.9. Dans les deux cas précédents, il faut quitter la boucle qui parcourt les joueurs car le jeu est immédiatement terminé.

Exemple d’affichage souhaité

```
Pierre a tiré la carte 10  
Nouveau score : 10  
Paul a tiré la carte 3  
Nouveau score : 3  
Jacques a tiré la carte 9  
Nouveau score : 9  
Pierre a tiré la carte 8  
Nouveau score : 18  
Paul a tiré la carte as  
Nouveau score : 14  
Jacques a tiré la carte 6  
Nouveau score : 15  
Pierre a tiré la carte dame  
Nouveau score : 29  
Jacques a gagné !
```

Restitution

Lorsque vous avez terminé, faites-signer au professeur pour lui remettre votre travail.

N'oubliez pas de formater votre code !

