

Nom : _____

Prénom : _____

Identifiant : _____ Groupe : _____ Enseignant : _____

/20



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

mars 2024
DEV2
Equipe dev2

DEV2 – Développement II

Examen de 1^{re} session

Bilan – dev2

Consignes

1. Le bilan dure 1h30 et se fait sur une machine du réseau pédagogique.
2. Vous n'avez pas accès à l'Internet
3. Vous pouvez utiliser vos fichiers disponibles localement et vos notes papier.
4. Avant de commencer, clonez le dépôt (sur Gitesi) qui a été créé pour vous.
5. Tous les fichiers devront être placés dans le clone, créé ci-dessus.
6. Lorsque vous avez fini, créez un commit contenant vos fichiers. Depuis le clone :
 - ▷ `git status`
 - ▷ `git add .`
 - ▷ `git commit -m "remise examen"` (La première commande sert à vérifier que tout est là, et rien d'autre.)
7. Poussez le commit vers le dépôt (sur Gitesi) avec `git push`

- 1 À l'intérieur du dossier cloné, commencez par créer un projet nommé gxxxxxx où xxxxx représente votre matricule. Créez ensuite un package nommé **CardGame**. Dans ce package, créez trois classes :

/1

1. la première nommée **Main**;
2. la seconde nommée **Card** (qui représente une carte);
3. la dernière nommée **Player** (qui représente un joueur ou une joueuse).

1 Card

- 2 Créez un attribut entier pour la classe **Card**, cet attribut est nommé **value**. Créez un constructeur avec un paramètre entier. Le constructeur vérifie que l'entier reçu en paramètre est bien entre 1 et 10, sinon il lance une exception. Il attribue ensuite cet entier à l'attribut **value**.

/2

- 3 Dans la classe **Card**, faites une surcharge du constructeur en créant un constructeur sans

/2



argument. Ce constructeur va générer un entier aléatoire entre 1 et 10 et l'attribuer à l'attribut `value`.

- 4 Créez un accesseur pour l'attribut de la classe `Card`. /1

2 Player

- 5 Dans la classe `Player`, créez un attribut nommé `hand` qui est une liste de `Card` et qui représente la main du joueur. Créez ensuite un constructeur sans paramètre qui crée deux cartes de valeur aléatoire et les ajoutes à la liste `hand`. /2

- 6 Dans la classe `Player`, créez une méthode `sumValue` qui retourne la somme des valeurs des cartes de la main du joueur. /2
Dans cette même classe, créez ensuite une méthode `display` qui affiche les cartes de la main du joueur (c'est-à-dire la valeur des cartes), ainsi que la somme des valeurs des cartes de la main du joueur.

- 7 Dans la classe `Player`, créez une méthode `addCards` qui prend en paramètre un entier `n`. Cette méthode va créer `n` cartes de valeur aléatoire et les ajouter à la liste `hand`. Cette méthode lance une exception si l'argument `n` est négatif. /2

3 Tests

- 8 Générez une classe de test pour la classe `Player`. Créez les tests suivants : /4
- ▷ créez un test pour le constructeur de la classe `Player`, il vérifie que la main du joueur contient bien deux cartes ;
 - ▷ créez un test pour la méthode `sumValue` ;
 - ▷ créez un test pour la méthode `addCards`, il vérifie que la taille de la main à bien augmentée de `n` cartes ;
 - ▷ créez un test pour la méthode `addCards`, il vérifie que une exception est bien lancée si l'argument `n` est négatif.

Au besoin, créez les accesseurs des attributs des différentes classes.

4 Main

- 9 Vous allez implémenter une version extrêmement simplifiée et légèrement modifiée du Blackjack. Deux joueurs vont pouvoir piocher le nombre de cartes qu'ils désirent. Le gagnant sera celui dont la somme des valeurs des cartes est la plus proche de 21. /4
Dans la classe `Main` créez la méthode `main`. Dans cette méthode, effectuez les opérations suivantes :
- ▷ créez deux joueurs ;
 - ▷ affichez les cartes du joueur 1 et la somme des valeurs de celle-ci ;
 - ▷ demandez lui combien de cartes il veut piocher. Lui ajouter ce nombre de cartes à sa main ;
 - ▷ réaffichez toutes ses cartes et la somme des valeurs de celles-ci ;

- ▷ répétez ces trois dernières étapes pour le joueur 2;
- ▷ affichez le gagnant, c'est-à-dire le joueur qui a le score le plus proche de 21.