TP5 BPO
String/StringBuilder/ArrayList/Paquet de cartes

Exercice 1 - artEoz et les chaînes de caractères

Utiliser **artEoz** pour comprendre la différence de représentation mémoire entre **String** et **StringBuilder**. Vous pouvez utiliser et modifier les exemples directement proposés par **artEoz** (onglet "Un exemple ?").

Exercice 2 - artEoz et les listes

La documentation des classes de la bibliothèque se trouve à l'adresse :

http://docs.oracle.com/javase/11/docs/api/

Retrouver la documentation de la classe **ArrayList**.

1. Dans **artEoz, SANS** ajouter **geometrie.jar** en ressources personnelles, exécuter le code ci-dessous (pas de copier-coller) :

```
ArrayList<Point> liste = new ArrayList<>(5) ;
liste.add(new Point(0,0));
liste.add(new Point(1,1)) ;
```

2. Quelle est la capacité de la liste créée ? Quelle est sa taille ?

3. Modifier le code, étudier l'évolution de la liste au fur et à mesure :

   a. ajouter un **Point** à la position 1 de la liste : `liste.add(1, new Point(10,10)) ;`
   b. ajouter un **Point** en fin de liste : `liste.add(new Point(3,3)) ;`

   Quelle est la capacité de la liste ? Quelle est sa taille ?

4. Supprimer le **Point** en position 2 et étudier l'évolution de la liste.

5. Que fait l'instruction suivante ? `Point p = liste.get(2) ;`

6. Les cases d'indices 3 et 4 sont libres (initialisées avec **null**). Ajouter un **Point** à l'indice 4 ; que peut-on en conclure ?

7. Ajouter successivement 3 points dans la liste. Quelle est la capacité de la liste ? Quelle est sa taille ? Demander la visualisation des objets morts. Que peut-on en conclure ?

Exercice 3 - Carte et PaquetDeCartes

Le diagramme de classes complet est à la fin du sujet, mais il faut réaliser les questions dans l'ordre.

Partie 1

Créer un nouveau répertoire **cartes** ; placer dans ce répertoire le code source de la classe **Carte**, fourni sur Arche.

Écrire la définition de l'énumération **Couleur** dans le package **cartes** (pour mémoire, la définition des énumérations a été faite dans le CM *Construire une classe).*

Dans les deux classes, ajouter les **assert** dans les fonctions qui en ont besoin, pour vérifier la validité des paramètres.

Ces classes n'ont pas besoin d'être testées, car elles ne contiennent que des getter/setter.

Partie 2

L'objectif est de définir et tester la classe **PaquetDeCartes** dans le package **cartes**, tout en suivant une bonne méthode de travail pour être efficace et rapide.

Comme le nombre de cartes d'un paquet peut varier au cours d'un jeu, on choisit de mémoriser les cartes dans une **ArrayList**.

Écrire pour commencer :

- la définition du champ,
- le constructeur,
- la fonction **void ajouter(Carte… cartes)**,
- la fonction **int getNombreDeCartes()**.

> **Complément de Cours :**
>
> - la fonction **void ajouter(Carte… cartes)** admet un nombre quelconque de cartes en paramètre. Par exemple, on peut écrire
>
> ```
> PaquetDeCartes pdc = new PaquetDeCartes();
> pdc.ajouter(carte1, carte2);
> pdc.ajouter(carte3, carte4, carte5);
> ```
>
> Dans le corps de la fonction **ajouter**, le paramètre **cartes** est considéré comme un tableau de **Carte**.

Partie 3

Il faut maintenant s'intéresser au test de ce qui a été écrit dans la classe **PaquetDeCartes**. Pour pouvoir faire différents cas de tests, on a besoin de créer un paquet vide, un paquet à 2 cartes, un paquet à 5 cartes, *etc.*

À cette fin, on définit le singleton **FabriqueCartes** (*cf.* CM Bibliothèque). On y définit par exemple, les fonctions :

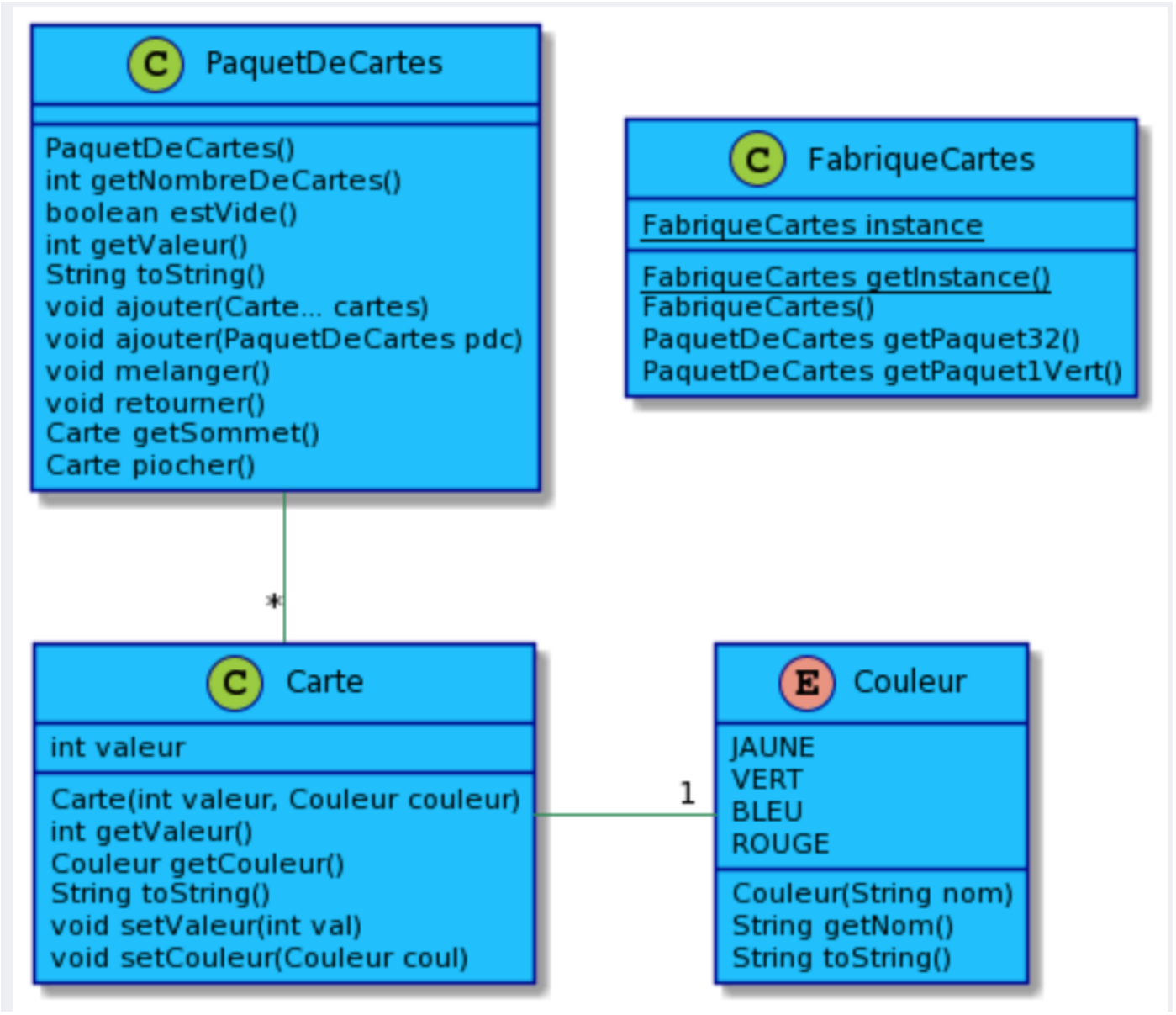
PaquetDeCartes getPaquetVide()
PaquetDeCartes getPaquet1Vert()
PaquetDeCartes getPaquet5Vert()

qui crée trois sortes de paquets différents.

Construire un test de la classe **PaquetDeCartes**, avec, pour l'instant le test de la fonction **getNombreDeCartes()**. Utiliser les fonctions de la fabrique pour envisager différents cas de tests.

Partie 4

Tout est maintenant en place pour continuer à écrire et tester les fonctions de **PaquetDeCartes**, l'une après l'autre bien sûr.


PaquetDeCartes
PaquetDeCartes()
int getNombreDeCartes()
boolean estVide()
int getValeur()
String toString()
void ajouter(Carte... cartes)
void ajouter(PaquetDeCartes pdc)
void melanger()
void retourner()
Carte getSommet()
Carte piocher()
FabriqueCartes
FabriqueCartes instance
FabriqueCartes getInstance()
FabriqueCartes()
PaquetDeCartes getPaquet32()
PaquetDeCartes getPaquet1Vert()
*
Carte
int valeur
Carte(int valeur, Couleur couleur)
int getValeur()
Couleur getCouleur()
String toString()
void setValeur(int val)
void setCouleur(Couleur coul)
1
Couleur
JAUNE
VERT
BLEU
ROUGE
Couleur(String nom)
String getNom()
String toString()