

## LABORATOIRE 7

*ArrayList, interfaces Comparable et Comparator, GregorianCalendar et Collections*

### La classe à développer

NOM de la classe:	Tache
Appliquer l' <b>ENCAPSULATION</b> :	Oui
Faire le schéma en <b>UML</b> de la classe:	Non
<b>LES ATTRIBUTS DE LA CLASSE</b> Les variables d'instances de la classe:	<ol style="list-style-type: none"><li>1- Nom de la tâche à accomplir (String)</li><li>2- Date de début de la tâche, faire une composition avec la classe GregorianCalendar. (Voir <a href="#">PDF GregorianCalendar</a>)</li><li>3- Date de fin de la tâche, faire une composition avec la classe GregorianCalendar.</li><li>4- La durée en nombre d'heure(s) de la tâche (int)</li><li>5- Le coût de la tâche (double),</li><li>6- Remarques ou commentaires au sujet de la tâche (String).</li></ol>
<b>LES OPÉRATIONS</b> Les services ou méthodes publiques:	<ol style="list-style-type: none"><li>1- Faire la méthode equals ( ).</li><li>2- Méthode hashCode ( ) avec l'assistant.</li><li>3- <u>Faire la méthode compareTo ( ) sur l'attribut date de début de tâche.</u></li><li>4- <u>Utilise l'interface que tu désires pour trier les tâches sur l'attribut nom de celle-ci.</u></li><li>5- <u>Utilise l'interface que tu désires pour trier les tâches selon leur coût.</u></li><li>6- Faire une méthode toString() qui permet d'afficher clairement l'état de l'objet.</li></ol>
<b>LES OPÉRATIONS</b> Les constructeurs:	<ol style="list-style-type: none"><li>1- Un constructeur par défaut,</li><li>2- Un constructeur paramétré avec un paramètre pour chacune des variables d'instances de la classe.</li></ol>
<b>LES OPÉRATIONS</b> Les méthodes d'accès (get et set):	<ol style="list-style-type: none"><li>1- Prévoir un get ( ) et un set ( ) pour chacune des variables d'instances de la classe.</li></ol>

### Classe TestTache

Créez un tableau statique qui permet de déclarer et d'initialiser cinq tâches de votre choix. (**Voir PDF – saisie de données -- tableau à ArrayList**).

Créez un ArrayList en y ajoutant toutes les informations sur les tâches du tableau qu'on vient de créer.

À partir du tableau dynamique, l'utilisateur aura la possibilité d'effectuer les opérations qu'il désire sur les objets du tableau, à l'aide d'un menu ayant les choix suivants:

- 1) Ajouter un objet dans l'ArrayList. Il faut ensuite saisir toutes les infos de la tâche et l'ajouter à la fin.
- 2) Supprimer un objet de l'ArrayList à partir du nom de la tâche. Ne rien faire si la tâche n'est pas trouvée.
- 3) Modifier un objet de l'ArrayList à partir du nom de la tâche.  
Il s'agit en fait de rechercher l'objet désiré, y conserver sa position, le supprimer, saisir à nouveau toutes les informations pour le nouvel objet, puis l'insérer à la même position dans l'ArrayList.
- 4) Rechercher un objet de l'ArrayList à partir du nom de la tâche.  
Indiquer si l'objet est présent ou non dans l'ArrayList.
- 5) Visualiser un objet de l'ArrayList à partir du nom de la tâche.  
Afficher toutes les informations de l'objet.
- 6) Trier tous les objets de l'ArrayList par rapport à leur date de début.
- 7) Trier tous les objets de l'ArrayList par rapport au nom des tâches.
- 8) Trier tous les objets de l'ArrayList par rapport au coût des tâches.
- 9) Visualiser tous les objets de l'ArrayList.
- 10) Indiquer la taille de l'ArrayList, soit le nombre d'objets.
- 11) Quitter le programme.

**SVP** indiquer clairement en console les informations affichées et assurez-vous que l'on puisse distinguer facilement les objets entre eux lorsque vous faites afficher l'ArrayList (ajuster le format de sortie dans toString()).

Menu principal des tâches

Sélectionnez votre choix parmi les options suivantes:

Ajouter

Supprimer

Modifier

Rechercher

Visualiser un objet

Trierdate

Trier/nom

Trier/cout

Visualiser tous les objet

Visualiser le nombre d'objet

Quitter

## JEU DE CARTES

*Créez un programme qui permet d'obtenir toutes les cartes d'un jeu de cartes.*

### **La classe à développer**

<b>NOM</b> de la classe:	<b>Carte</b>
Appliquer l' <b>ENCAPSULATION</b> :	Oui
Faire le schéma en <b>UML</b> de la classe:	Non
<b>LES ATTRIBUTS DE LA CLASSE</b> Les variables d'instances de la classe:	1- La « couleur » de la carte (Pique, Trèfle, Carreau, Cœur) 2- La « valeur » de la carte (entier de 1 à 13 : 1 = As, 13 = Roi). Cet attribut vous sera utile pour comparer les cartes entre elles. 3- La « face » de la carte (de As, Deux, ..., Dame, Valet, Roi) [ * La valeur de cet attribut dépend de la valeur de la carte. Vous pouvez l'exclure des paramètres du constructeur paramétré et « setter » sa valeur à partir de l'attribut valeur. N'oubliez pas d'ajuster la face si vous changer la valeur d'une carte.]  <b>Afin de vous aider, déclarer 2 tableaux de String qui seront public, final et static : un pour les 4 couleurs possibles, et un pour les 13 faces possibles.</b>
<b>LES OPÉRATIONS</b> Les méthodes d'accès (get et set):	1- Prévoir un get ( ) et un set ( ) pour chacune des variables d'instances de la classe.
<b>LES OPÉRATIONS</b> Les services ou méthodes publiques:	1- Faire la méthode equals( ) basée sur la face et la couleur. 2- Méthode hashCode( ) avec l'assistant. 3- Faire la méthode compareTo( ) sur les attributs <u>valeur puis couleur de la carte</u> . Suggestion : pour départager la couleur, vous pouvez utiliser l'indice de celle-ci dans le tableau statique de la classe. 4- Faire une méthode toString( ).
<b>LES OPÉRATIONS</b> Les constructeurs:	1- Un constructeur par défaut qui initialise la carte à l'As de Pique, 2- Un constructeur paramétré avec un paramètre pour chacune des 2 variables d'instances de la classe (couleur et valeur). Si la valeur est invalide, assignez « 1 » (ainsi que la face associée) et si la couleur est invalide, assignez « Pique ».

```
public class TestJeuDeCartes {
```

```
    public static void main(String[] args) {
```

*À l'aide des 2 tableaux statiques publics de la classe, créer un tableau de la classe Carte pouvant contenir 52 cartes.*

*Remplir le tableau de cartes en y insérant toutes les cartes d'un jeu.*

*Créer un ArrayList en y ajoutant toutes les données du tableau qu'on vient de créer soit, toutes les cartes d'un jeu de cartes.*

**Bien suivre l'ordre d'exécution** des commandes suivantes.

Te servir de la [classe Collections](#) pour effectuer les commandes suivantes.

- 1) *Afficher toutes les cartes du jeu de cartes.*
- 2) *Brasser toutes les cartes du jeu (les mélanger afin qu'elles soient dans un ordre aléatoire).*
- 3) *Afficher toutes les cartes du jeu de cartes.*
- 4) *Interchanger les positions 2 et 4 des cartes.*
- 5) *Afficher toutes les cartes du jeu de cartes.*
- 6) *Trier les cartes.*
- 7) *Afficher toutes les cartes du jeu de cartes.*
- 8) *Faites une rotation avant des cartes.*
- 9) *Afficher toutes les cartes du jeu de cartes.*
- 10) *Inverser l'ordre de toutes les cartes du jeu.*
- 11) *Afficher toutes les cartes du jeu de cartes.*
- 12) *Inverser l'ordre de toutes les cartes du jeu mais, te servir de la commande de tri.*
- 13) *Afficher toutes les cartes du jeu de cartes.*

}

}