

Développement de programme dans un environnement graphique Automne 2023 Cégep Limoilou Département d'informatique Professeur : Martin Simoneau	Formatif 0 Révision
--	--

Nom : _____

Objectifs

- Réviser la programmation OO

À remettre :

- À faire seul.
- Remettre votre projet sur Léa.

À faire

- 1) Application de base:
 - a) Faire le code du garage pour que le scénario de la classe *Application* fonctionne correctement. Programmer toutes les méthodes de la classe **Garage** pour qu'elles fassent ce que décrit leur javadoc. Le scénario dans la méthode *main* de la classe *Application* doit s'exécuter correctement.
- 2) Collection – historique de réparation:
 - a) Créez une *Map* pour contenir l'historique des réparations.
 - i) En clé vous devez placer un objet de type *LocalDateTime* qui indique le moment de la réparation. Il peut être créé avec la méthode *LocalDateTime.now()*.
 - ii) Comme valeur, placez une chaîne de caractères contenant:
 - (1) Un point-virgule suivi de
 - (2) Le NIP du véhicule suivi de
 - (3) chaque type de réparation qui a été faite sur le véhicule (séparées par des espaces).
Exemple : `" : B- moteur transmission"`
 - b) Ajoutez une méthode **afficheHistorique** dans la classe *Garage*. Cette méthode doit:
 - i) Afficher la chaîne de caractères *Historique*
 - ii) Afficher toutes les entrées contenues dans l'historique:
 - (1) Afficher la date de saisie
 - (2) Afficher la chaîne de caractères contenant l'information de réparation
 - iii) Afficher "fin de l'historique:"
Exemple:

Historique :

2022-08-24T14:04:13.452498 : C- carrosserie

2022-08-24T14:04:13.452538 : D- moteur
2022-08-24T14:04:13.448144 : A- moteur transmission
2022-08-24T14:04:13.451258 : B- transmission

fin de l'historique

c) Modifiez le scénario pour appeler la méthode affiche historique après les réparations.

3) Héritage:

a) Nous allons maintenant tenir compte de plusieurs types de véhicules.

i) **Automobile** qui a 4 états:

- (1) Moteur
- (2) Transmission
- (3) Carrosserie
- (4) Habitacle

ii) **Moto** qui a 2 états:

- (1) Moteur
- (2) Transmission

iii) **Camion** qui a 4 états:

- (1) Moteur
- (2) Transmission
- (3) Carrosserie
- (4) Espace de chargement

b) Créez la hiérarchie de classe qui s'impose pour bien réutiliser les attributs (les états) et la méthode *repare* (avec le *super.repare()*)

c) Modifiez le *Garage* pour bien utiliser les nouvelles classes génériques.

d) Changer quelques *automobiles* pour d'autres types de véhicule dans le scénario de la classe *Application*.

4) Fichiers:

a) Créez la méthode **enregistreVehicules** dans la classe garage. Cette méthode doit:

- i) Recevoir le nom du fichier à produire en paramètre.
- ii) Enregistrer dans un fichier texte tous les véhicules
 - (1) du stationnement
 - (2) des espaces de garage

Utilisez la méthode *toString* de chaque véhicule pour mettre dans le fichier

5) Validation

a) Vérifiez si votre code fonctionne toujours lorsque vous ajoutez de nouveaux espaces de stationnements et de garage!

FIN