TD/TP #3 - Patron de conception Décorateur

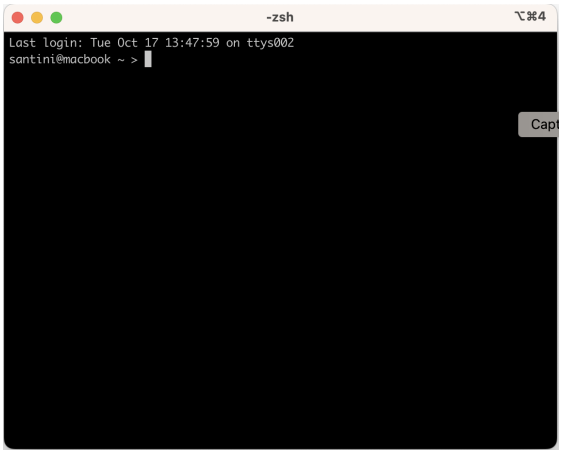
Guillaume Santini

22 janvier 2024

1 Exercice Principal : Mise en forme de textes

1.1 Description du sujet

L'objectif de cet exercice est de nous doter de classes permettant d'effectuer des opérations de mise en forme des textes affichés lors de l'exécution d'un programme dans une console de type bash. Par souci de simplicité nous formerons l'hypothèse que le programme Java produisant les affichages est exécuté dans une fenêtre de shell dont les couleurs d'affichages et de fond sont respectivement blanche et noire.



Les textes seront toujours manipulés au niveau des *lignes*.
Dans un premier temps nous considérerons deux types de textes :

- un paragraphe est constitué d'une ligne de texte libre,
- une ligne d'une liste à puce est constitué d'un texte libre qui est affiché avec un préfixe de la forme `"[-] "`.

Les mises en forme que l'on souhaite pouvoir appliquer aux textes sont :

- la coloration du fond du texte,
- la coloration des caractères du texte,
- l'encadrement du texte.

Ces mises en formes *doivent* pouvoir être cumulées.

La coloration utilisera les caractères spéciaux Linux (*cf.* https://doc.ubuntu-fr.org/ls_couleur). À titre d'exemple :

```
System.out.println("\033[34m Bonjour \033[0m")
```

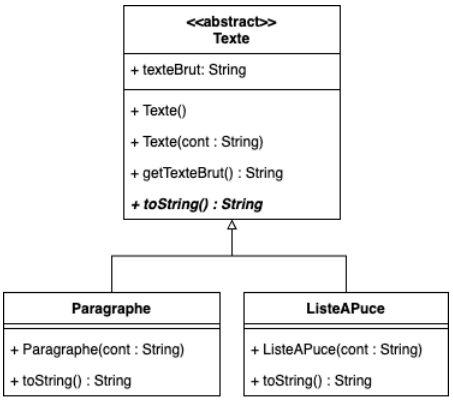
affiche la chaîne de caractères `"Bonjour"` en bleu à l'écran.
Les codes couleurs pour la coloration du texte et du fond sont respectivement codées par les énumérations `CouleurDeTexte` et `CouleurDeFond`.
La mise en forme des textes sera assurée automatiquement par la méthode `toString()`.

1.2 Modélisation UML

Exercice 1 : Modélisation des entités de base : les `Textes`

Proposez un diagramme de classe permettant de représenter les deux types de textes de type *paragraphe* et *liste à puce*.

Correction :

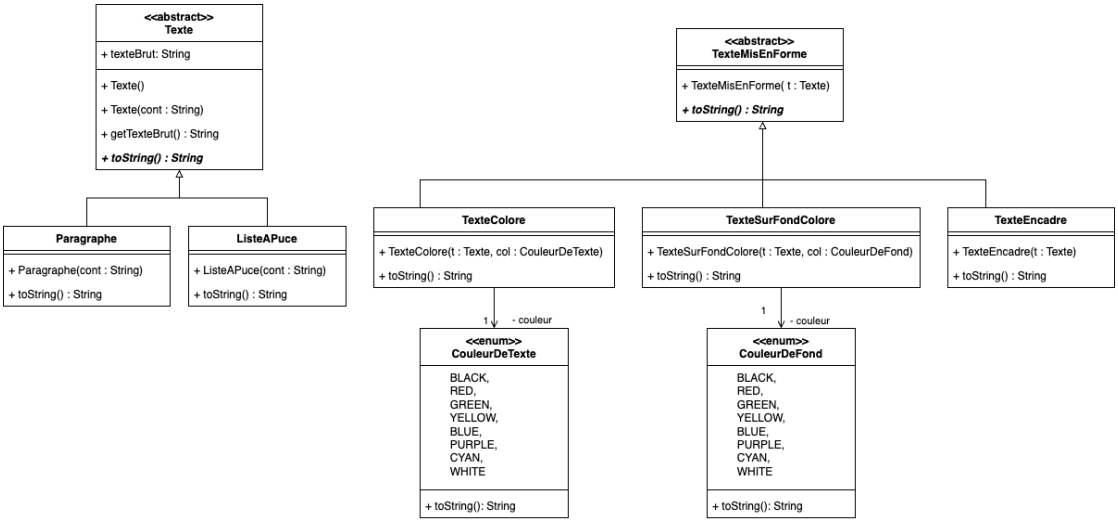


◇

Exercice 2 : Modélisation des entités de base : les `TexteMisEnFormes`

Enrichissez le diagramme pour créer des textes mis en forme (texte coloré, texte sur fond coloré et texte encadré). Lorsque la couleur est nécessaire à la définition des instances vous penserez à intégrer les liens d'association avec les énumérations `CouleurDeTexte` et `CouleurDeFond`.

Correction :

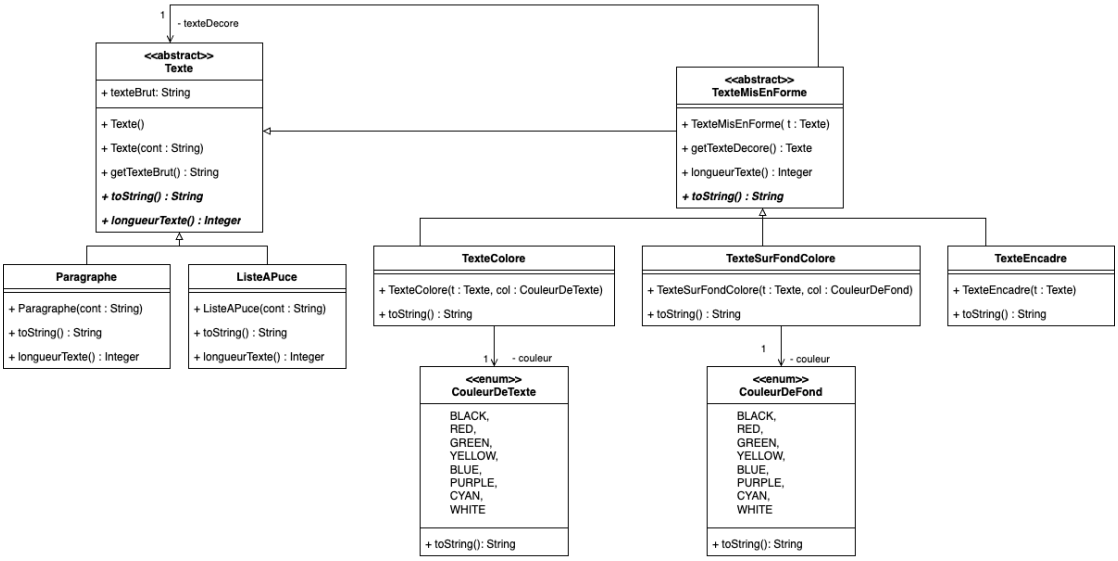


◇

Exercice 3 : Modélisation des textes décorés

Complétez le modèle afin de pouvoir mettre en forme les textes de type *paragraphe* et *liste à puce* avec une ou plusieurs mises en forme.

Correction :



◇

1.3 Implémentation Java

Exercice 4 : Récupération des énumérations

Récupérez le code des deux énumérations `CouleurDeTexte` et `CouleurDeFond` à partir du lien suivant : https://www.lipn.univ-paris13.fr/~santini/Patrons_conception/seance3/src.tgz Faites un projet ´Eclipse (ou autre IDE de votre choix) et ajouter les énumérations.

Exercice 5 : Implémentation du modèle

Traduisez la modélisation obtenue en code Java.

Attention : *Il est impératif de tester chaque comportement un par un dans une ou plusieurs classes de test indépendantes.*

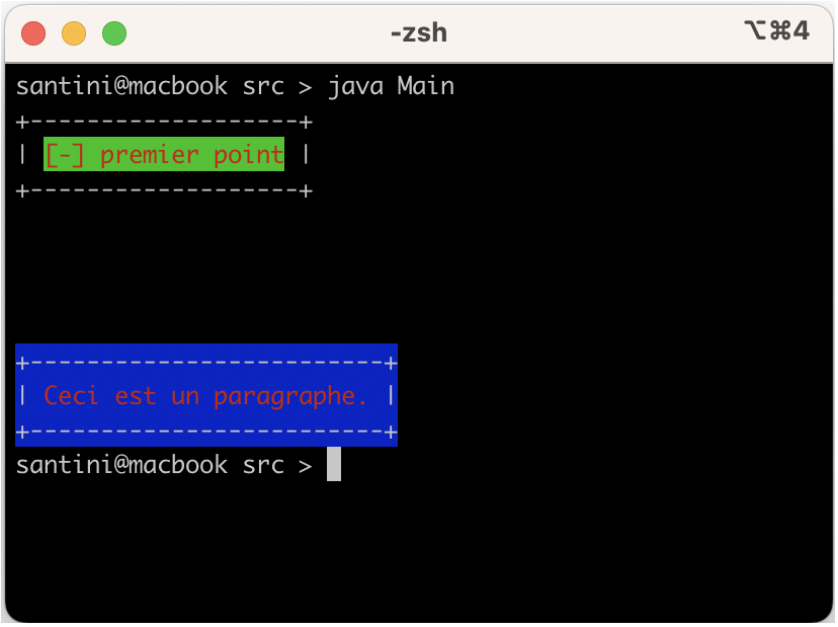
Correction :

https://www.lipn.univ-paris13.fr/~santini/Patrons_conception/seance3/texte_mis_en_forme.tgz

◇

Exercice 6 : Exécution du modèle

Proposez un programme `Main.main()` principal qui produise les affichages suivants :



2 Exercice Optionnel : Voitures toutes options

Nous souhaitons calculer le montant de voitures dont le prix dépend des options d'équipements.

2.1 Modélisation UML

Une voiture est caractérisée par un nom de modèle, une masse (Double) et un prix (Double). Le prix de celle-ci dépend de son niveau de finition et d'un bonus/malus écologique.

Le niveau de finition est défini par les options ajoutées au modèle de base.

Le bonus/malus écologique est déterminé par le type de motorisation. Il faut à ce sujet noter qu'un même véhicule peut être doté de plusieurs motorisations. Un véhicule hybride pourra avoir un moteur thermique et un moteur électriques par exemple.

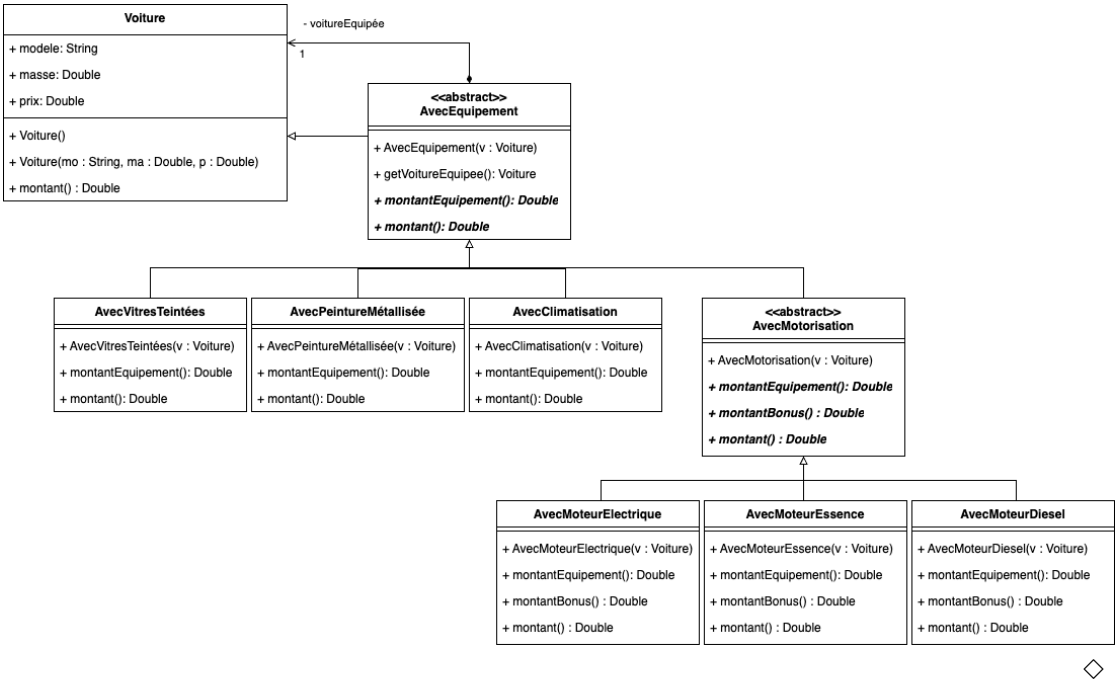
Pour information on donne ci-dessous le montant des équipements et des bonus/malus écologiques associés à chaque motorisation.

Options	Montant Equipement	Montant Bonus
Vitre teintée	2000€	
Couleur métallisée	1000€	
Climatisation	3000€	
diesel	5000€	-5000€
essence	4000€	0€
électrique	9000€	3000€

Exercice 7 : Modélisation des entités de base

Proposez un diagramme de classe permettant de représenter toutes les voitures pouvant être produites.

Correction :



2.2 Implémentation Java

Exercice 8 : Implémentation du modèle

Traduisez la modélisation obtenue en code Java.

Attention : *Il est impératif de tester chaque comportement un par un dans une ou plusieurs classes de test indépendantes.*

Correction :

https://www.lipn.univ-paris13.fr/~santini/Patrons_conception/seance3/voiture_equipee.tgz

Exercice 9 : Exécution du modèle

Proposez un programme `Main.main()` principal qui affiche le montant d'une voiture respectant les spécifications suivantes :

modéle C4
masse 1400kg,
montant 22000€
motorisation hybride essence/electrique,
option climatisation.

Remarque : Le montant obtenu doit être de 35000€.