**Programmation avancée – TP1**

Exercice 1 :

* 1. D’après l’analyse UML de la classe point :
* 4 attributs au total :
  + X et y de type Double
  + Couleur de type Color
  + La fonction distance qui prend en paramètre un objet de classe Point et qui retourne un Double
* 5 commandes :
  + Translater() qui prend en paramètre deux point dx, dy de type Double
  + Afficher() pour afficher le point actuel dans le terminal
  + setX() avec vx de type double en paramètre pour modifier x
  + setY() avec vy de type double en paramètre pour modifier y
  + setCouleur() avec nc de type color en paramètre pour modifier couleur
* Le constructeur de Point qui se défini par :
  + Le paramètre vx de type double -> abscisse
  + Le paramètre vy de type double -> ordonnée
  1. Une entorse dans le code est que les attributs sont initialisés en fin du code, alors que la convention exige de les mettre au début, avant le constructeur.

Exercice 2 :

2.1) Exemple pour l’execution de la ligne   
Une image contenant diagramme, ligne, texte, croquis

Description générée automatiquement

- je réserve une place mémoire pour la poignée p1 de Point

- je créé le point et je l’attache à p1

- j’affiche p1 avec la méthode static afficher() donc initialiser avant l’execution

- je reserve une place mémoire pour poignée p2 de Point et j’attache un Point à p2

- j’affiche p1 avec la méthode static afficher() donc initialiser avant l’execution

…

2.2) Compiler ExempleComprendre.java créer un ExempleComprendre.class mais aussi un Point.class, tiré de Point.java, car Point est utilisé dans ExempleComprendre.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

2.3) Après compilation, je lance l’execution de ExempleComprendre

Une image contenant texte, capture d’écran

Description générée automatiquement

2.4) Après vérification de l’exécution de ExempleComprendre, tout correspond avec le listing effectué au préalable. Tout marche bien.

2.5) Execution du ExempleComprendreTutor sur le site pythonTutor/java :  
Une image contenant texte, Police, nombre, logiciel

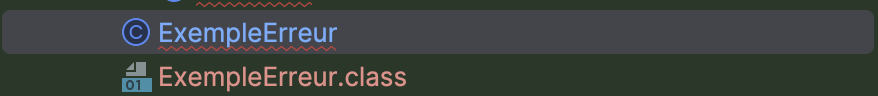
Description générée automatiquement

2.6) La compilation de ExempleErreur.java donne une erreur :  
Une image contenant texte, capture d’écran, Police

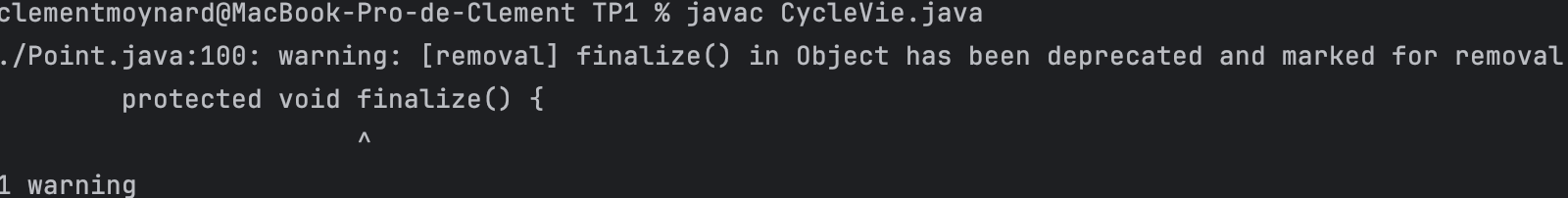
Description générée automatiquement

Car le constructeur de la classe Point spécifie qu’il attend deux paramètres de type Double et ceci ne sont pas fournis quand j’instancie l’objet Point qui est relié à p1  
Je corrige : *Point p1 = new Point(2.0, 4.0) ;*

*Et je peux enlever les setters d’en dessous, car mon Point est déjà initialisé*

Après correction la compilation marche bien et ExempleErreur.class se créé.  


2.7) Je décommente les fonctions pour afficher la création des Point et la déstruction de ces derniers dans Point.java

Je compile CycleVie.java pour créer CycleVie.class et mettre à jour Point.class : 



J’execute CycleVie :  
Une image contenant texte, capture d’écran, livre, menu

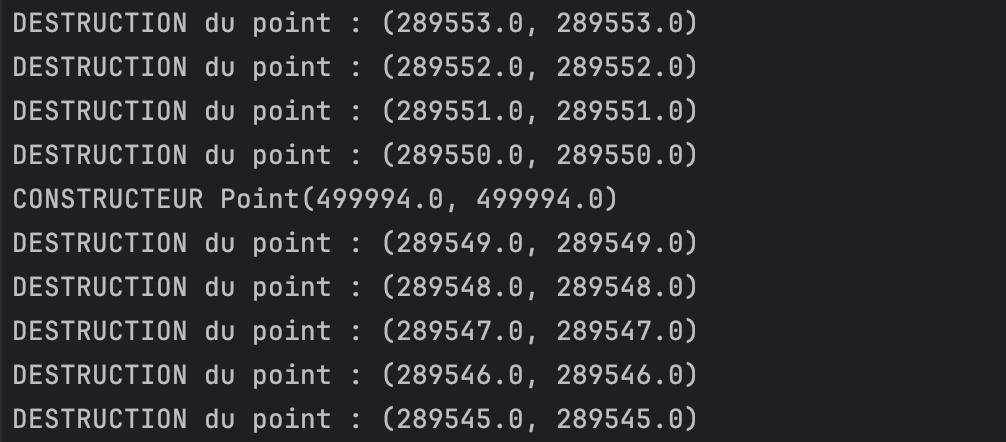
Description générée automatiquement Pour 1000 itérations de Point, aucun n’est détruit

Une image contenant texte, Police, capture d’écran

Description générée automatiquement Pareil pour 5000 itérations

Une image contenant texte, Police, capture d’écran

Description générée automatiquement Pareil pour 50 000 itérations

 Pour 500 000 itérations, la mémoire sature et libère de l’espace en supprimant d’anciens Point.

Exercice 3 :

3.1) Je lance la commande javadoc -d Doc \*.java et cela me produit la documentation de toutes les classes dans mon dossier TP1 :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Je peux donc consulter la documentation de classes :  
Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement

3.2) Grâce à la documentation je peux choisir la documentation de chaque classe, je choisis donc celle de Segment :  
- la méthode translater() prends deux paramètres dx et dy, tous deux de type Double, dx permet un déplacement sur l’axe des X et dy permet un déplacement sur l’axe des Y

- la méthode afficher() affiche le Segment sous la forme   
[extremite1-extremite2]

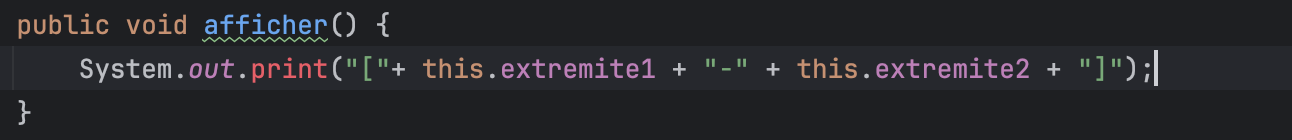
3.3)  
 Une image contenant texte, capture d’écran, Site web, Page web

Description générée automatiquement

Exercice 4 :

4.1) Diagramme UML d’analyse de Segment :  
Une image contenant texte, écriture manuscrite, Police, ligne

Description générée automatiquement

4.2) 



Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Exercice 5 :