Projet de
Programmation en Java

Dessine-moi un mouton

''Comment devenir un Petit Prince

Java''

Saint Exupéry aurait peut-être programmé une animation en Java...

L'objectif est de créer un programme fonctionnel en Java utilisant les concepts de la P.O.O. Le principe d'encapsulation ne doit en aucun cas pouvoir être violé.
L'application devra réaliser une grande fresque (patchwork) murale réalisée à partir de plusieurs dessins géométriques.
Les notions mathématiques à utiliser sont de niveau lycée et sont disponibles sur de nombreux sites.
Le choix des données des classes n'est volontairement pas précisé.

- Les formes géométriques élémentaires disponibles sont les lignes, les polygones, les cercles, et les ellipses.
- Une image est composée de formes géométriques.
  L'itération sur les formes géométriques d'une image est à mettre en œuvre.
  Aucun doublon de forme géométrique ne peut exister dans une image.
- Un dessin est composé d'images.
  L'itération sur les images d'un dessin est à mettre en œuvre.
  Aucun doublon d'image ne peut exister dans un dessin.
- Il faut bien réfléchir aux relations entre les différentes classes du problème
- Les calculs du périmètre et de l'aire des formes géométriques, des images et des dessins sont requis.

  L'aire d'un polygone quelconque peut être calculée par triangulation.
  Le périmètre (l'aire) d'une image est la somme des périmètres (des aires) de ses composants.

  Il en est de même pour un dessin.
- Les formes géométriques peuvent subir les transformations suivantes : homothétie, translation, rotation, symétrie centrale, symétrie axiale. Une classe matrice peut être utile.

  Appliquer une transformation :
  - sur une image revient à appliquer la même transformation sur ses composants
  - sur un dessin revient à appliquer la même transformation sur ses composants.
- La copie d'un dessin est obtenue par copie de toutes ses images.

- Le tri des formes d'une image selon leur périmètre.
- Le tri des images d'un dessin selon leur aire

Afin d'assurer l'absence de doublons, il peut être judicieux de choisir une classe *Collection* assurant cette fonctionnalité.
Penser à mettre en place une gestion d'exceptions lorsque cela s'avère nécessaire. L'utilisation à bon escient du mot-clé final est très vivement recommandée.
Les conventions de codage Java sont à respecter scrupuleusement et les sources présentées lisiblement.

La mise en œuvre des points suivants est demandée :

- une application console créant la fresque et réalisant une batterie de tests les plus exhaustifs possibles.
  Plus précisément, l'ensemble des services proposé sera testé et leurs résultats affichés (méthodes de calcul de périmètre et d'aire, de transformations, de tri et de copie).
  La fresque devra être affichée sous une forme textuelle la plus explicite possible.

Rendu du projet

Le projet est à réaliser par binôme.
Le rendu est sous la forme NOM1_NOM2.zip, archive contenant les fichiers ***dessin.jar, rapport.pdf*** et ***une vidéo de présentation (5 minutes maximum)***.

Le fichier ***dessin.jar*** doit contenir les sources Java et la Javadoc.
Une documentation des sources doit être générée automatiquement (utiliser les annotations adéquates en java afin de fournir une Javadoc).

Les projets sont à déposer sur l'espace de rendu de *Campus Efrei*, ceux rendus uniquement par mail seront pénalisés.
Merci de votre sens des responsabilités.

Afin de bien mettre en évidence les relations entre les classes de votre projet, vous pouvez utiliser un logiciel qui vous générera un diagramme de classes à partir de vos sources.
L'idéal est d'utiliser un logiciel libre permettant la rétro-conception.

StarUML : https://www.projet-plume.org/mots-cles-proposes-par-lauteur/retro-ingenierie BOUML : https://www.projet-plume.org/fiche/bouml

Un rapport doit fournir votre analyse et votre conception.

Les éléments suivants doivent notamment y figurer :

- une introduction exposant clairement les objectifs, limites, choix du projet
- un mode d'emploi de l'application (en annexe)

- des schémas décrivant l'architecture fonctionnelle
- un diagramme de classes (réalisé si possible avec un logiciel UML, ou une photographie d'un diagramme réalisé sur papier, …).
- une explication en français et/ou pseudo-code et/ou langage d'implémentation choisi des principales fonctions (code à fournir et à commenter)
- une conclusion résumant le travail effectué et ouvrant des perspectives
- une bibliographie utilisée
- une table des matières ou un sommaire

Les classes conçues doivent être détaillées aussi bien leur interface que leur implémentation. Une bonne idée est de réaliser le rapport en même temps que le développement.

Des revues notées sont à prévoir lors des séances de suivi.

Date limite du rendu : le 24/11/2021 avant minuit.

Barème :

- Application : 13 points
- Rapport : 3 points
- Présentation : 4 points