INF2010 - Structures de données et algorithmes

Automne 2020

Travail Pratique 1

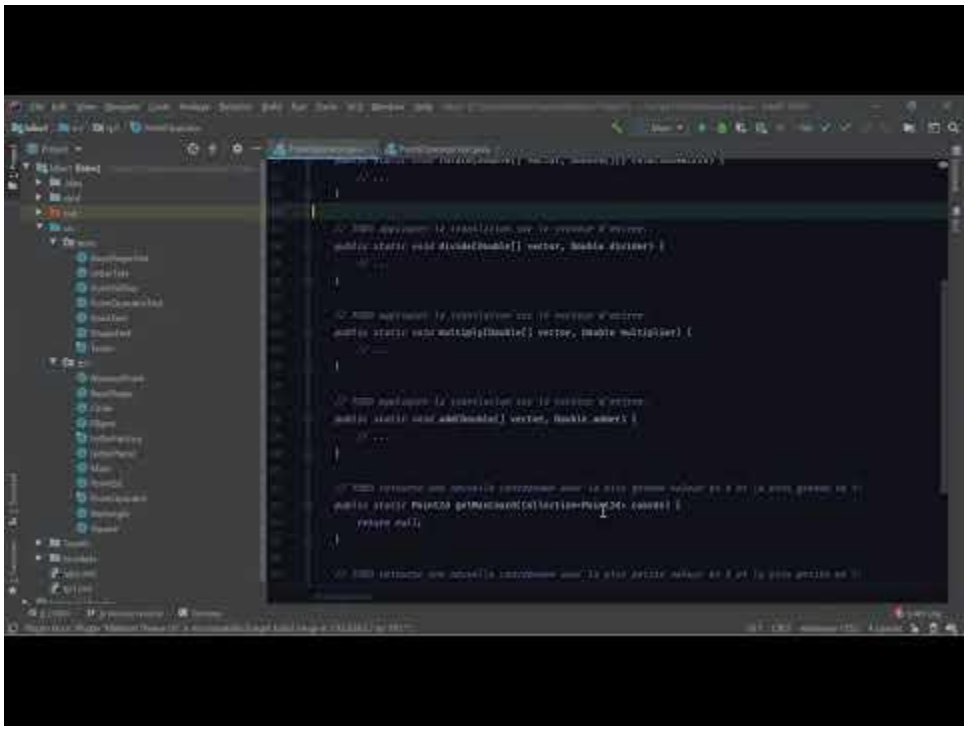
Java

Objectifs

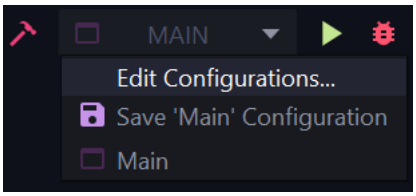
- Apprendre les bases de la programmation avec Java
- Utiliser le polymorphisme, l'agrégation et la composition.
- Utiliser des patrons de conception comme le Fluent
- Compléter les TODOs.

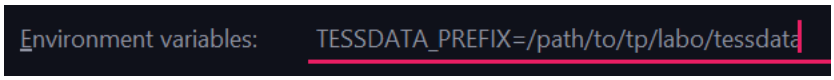
Je vous recommande de prendre le IDE Intellij, vous pouvez simplement choisir d'ouvrir un projet avec le fichier dézippé du labo.

Voici une explication sur comment lancer le projet :



Il est possible que vous deviez changez ceci si vous avez des erreurs avec Tesseract (la librairie utilisez pour identifier les lettres) :





Problème général

Nous allons créer le célèbre “Ca va bien aller“, mais de façon graphique et ce, sans l’aide d’aucune librairie spécialisée. Le but est simple, écrire des pixels à l’écran dans un certain ordre. Le résultat final devrait ressembler à ceci :

On remarque que c'est loin d'être de l'art, mais on comprend le principe! On vous recommande de comprendre le comportement des fichiers non corrigés aussi.

Partie 1 : Point

Un point est une structure de données représentant un vecteur. On s'intéresse généralement plus à la version 2d et 3d de celui-ci. Il est très utile de pouvoir modifier facilement des points dans l'espace, nous allons devoir créer quelques méthodes comme « rotate » et « translate ». Vous allez devoir modifier les fichiers :

- PointOperator
    - Toutes les opérations doivent supporter des vecteurs et des matrices pouvant aller jusqu'à N dimensions (pas seulement 2!)
    - Quelques liens qui pourraient vous être utiles :
        - https://en.wikipedia.org/wiki/Rotation_matrix
        - http://demonstrations.wolfram.com/Understanding2DTranslation/, etc.
- Point2d
    - Il serait bien de comprendre le comportement d'AbstractPoint
    - X et Y valant 1 et 0 représentent des indices

Partie 2 : Forme

Une forme est une composition de plusieurs points, nous allons seulement nous intéresser aux formes en deux dimensions. Une forme peut être quelconque (la lettre 'r'), comme elle peut être prédéfinie (un carré). Vous allez devoir modifier les fichiers :

- BaseShape
    - Toutes les formes sont centrées à l'origine.
    - Il faut remplir l'aire totale, vous pouvez inclure le périmètre, mais l'important est que les formes se ressemblent.
- Rectangle
    - 2x1 pourrait être: (-1;-0.5) (-1;0.5) (0;-0.5) (0;0.5) (1;-0.5) (1;0.5). Ici on a utilisé une précision d'un pixel (le facteur d'incrémentation). Ceci ne crée

effectivement pas 2 points (2x1), mais notre afficheur graphique s'occupera de convertir le tout en pixels.

- Ellipse
    - Il faut trouver la formule qui remplit l'intérieur de l'ellipse

Partie 3 : Lettre

Une lettre est une agrégation de formes simples pour créer une forme plus complexe. Nous utilisons le patron créateur pour générer une série de lettres, pas toutes car ce serait vraiment long et fastidieux. Vous allez devoir modifier les fichiers :

- LetterFactory
    - Les lettres sont des BaseShape alors elles doivent être centrées aussi.
    - Le résultat doit ressembler le plus possible à l'image ici haut.
    - Nous utilisons une librairie open source pour tester si vos lettres ressemblent bien aux lettres demandées, la librairie étant un peu capricieuse, vous aurez tout vos points pour la moitié des lettres bien identifiées. Un chargé de laboratoire s'assurera de vérifier toutes les lettres.
    - Le labo fonctionne très bien sur les ordis de l'école et dans 99% des cas lors des sessions précédentes, cependant, il se peut qu'une des librairies utilisées ne soient pas supportées sur votre plateforme de développement à la maison. https://linuxhint.com/install-tesseract-ocr-linux/

Dates de remise :

Vous avez deux semaines pour compléter les labos :

- Pour le groupe 1 : **4 octobre à 23h59**
- Pour le groupe 2 : **27 septembre à 23h59**
- Pour le groupe 3 : **1 octobre à 23h59**
- Pour le groupe 4 : **24 septembre à 23h59**

Barème de correction

Tests automatisés	/55
Style	/5
Total	/60

Un point de style est accordé. Ceci est plus à correction négative si le code est difficilement lisible.
Correction automatique : Nous allons utiliser de la correction automatique sur votre code, bien que ceci permet d'attraper certaines erreurs, l'entièreté de votre code sera révisée par un chargé de laboratoire. Il peut donc y avoir des différences entre vos résultats de laboratoire et celui de la correction finale. Vous pouvez ajouter d'autres cas de tests si vous le voulez.

Où les étudiants ont perdu des points les sessions précédentes :

- Votre « Ca va bien aller » devrait ressembler le plus possible (évidemment les couleurs ne sont pas pareils, elles sont aléatoires), mais pas d'espace en trop entre les lettres ou de l'espace sur les contours.
- Copier-Coller c'est non (souvent). Posez-vous la question si vous avez déjà codé une partie et essayez de la réutiliser.

Instructions pour la remise

Veuillez envoyer seulement :

- Point2d
- PointOperator
- BaseShape
- Rectangle
- Ellipse
- LetterFactory

Dans une archive de type *.zip qui portera le nom inf2010_lab1_MatriculeX_MatriculeY (de sorte que MatriculeX < MatriculeY). Les travaux en retard seront pénalisés de 20 % par jour de retard. Aucun travail ne sera accepté après 4 jours de retard.

ATTENTION! ATTENTION! ATTENTION! Pour ceux qui voudraient déposer leur laboratoire sur **GitHub**, assurez-vous que vos répertoires soient en mode **privé** afin d'éviter la copie et l'utilisation non autorisée de vos travaux.