Fractales

1. Comment compiler le projet

gradlew build

2. Comment lancer les deux versions du programme

- **Graphique** :

gradlew run --args="-G"

- **Terminal** :

gradlew run --args="-C ..." où ... = options telles que :

✗ **Pour Julia et Mandelbrot** :

* nécessaires : options -cst (forme x;y)(pour Julia), -p, -r (forme x1;x2;y1;y2) , -t (J/M)
* facultatives : options -fi, -fo (forme x2+c), -col, -it, -rad

ex *: "-C -cst 0.285;0.01 -p 0.01 -r -15;1;-1;1 -fi Julia -fo x2+c -col 3 -t J"*

✗ **Pour Sierpiński :**

* nécessaires : option -o, -t (S)
* facultatives : options -fi, -col

ex *: "-C -o 4 -fi Tapis "*

3. Les principaux choix techniques

* Nous avons utilisé Apache Commons Cli pour gérer les arguments de notre ligne de commande. Nous sommes partis du principe qu'une seule ligne serait nécessaire : soit on lance le mode graphique, soit on entre directement ce que l'on souhaite obtenir comme fractale.

* Nous avons permis à l'utilisateur de rentrer sa fonction facilement : 3x5+2x+c. De plus, nous avons fait en sorte d'avoir un large éventail de fonctions possibles : toutes les fonctions polynomiales sont possibles, mais également des fonctions composées de cos/sin/sinh (sous certaines formes cependant) avec de plus une division sur la constante (4/c, mais aussi bien x/c ou encore $1/c^2$).

* Nous avons créé une classe abstraite pour les Fractales qui est étendue par Julia, Mandelbrot et Sierpiński. Chacune va donc redéfinir les fonctions nécessaires.

* Concernant la création d'une fractale, nous avons opté pour un Builder (ce qui permet d'ajouter nos éléments voulus en ayant déjà des paramètres par défauts).

* L'utilisation d'une Fonction de Première classe pour nos fonctions (qui va étendre Fonction) semblait intelligente.

* Nous avons implémenté un redimensionnement de l'image générée en mode graphique afin de permettre à l'utilisateur de voir sa fractale entière malgré la restriction de taille de son écran

4. Rapport sur performance des versions parallèles

Implémentation de ForkJoinPool sur nos 3 types de Fractales.
Voyons les différences :

✗ Pour Julia et Mandelbrot

Nous obtenons le même temps avec et sans programmation concurrente (environ 3 secondes).
Cependant, nous avons remarqué que la programmation concurrente semble plus lente que celle sans, lors de gros calculs (grand rectangle/pas).
Au sein même de la programmation concurrente, diviser plus de fois prends un peu plus de temps (cela reste quand même proportionnel au temps initial)

✗ Pour Sierpiński

Nous obtenons le même temps avec et sans programmation concurrente (temps variable de quelques secondes selon l'ordre voulu).

5. Références aux sources

https://fr.wikipedia.org/wiki/Fractale
https://fr.wikipedia.org/wiki/Ensemble_de_Julia
https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot
https://youtu.be/Y4ICbYtBGzA
https://fr.wikipedia.org/wiki/Tapis_de_Sierpi%C5%84ski
https://fr.wikipedia.org/wiki/Liste_de_fractales_par_dimension_de_Hausdorff (Nous avons essayé d'implémenter d'autres types de fractales, mais nous ne savions pas comment procéder pour obtenir un rendu correct (notamment pour le flocon de Koch et le triangle de Sierpiński))