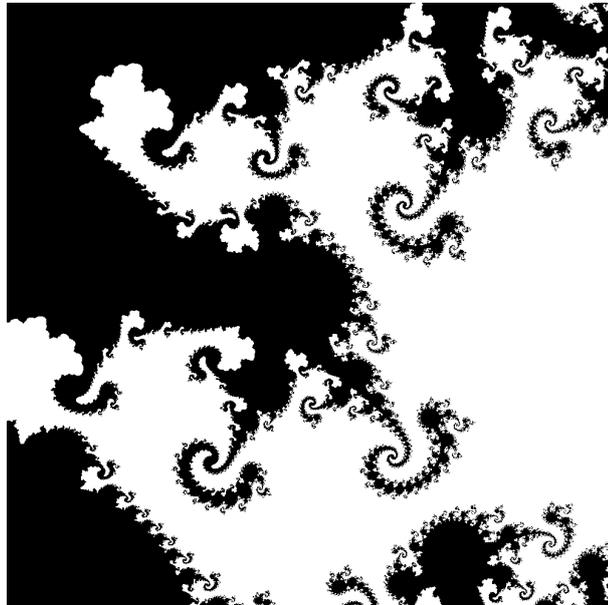


Le but de ce sujet est de créer des images de fractales comme celle-ci :



Vous devez disposer d'une classe permettant de manipuler des nombres complexes.

Dans un premier temps, nous nous intéressons aux fractales de Julia. Il s'agit de l'ensemble des nombres complexes z_0 tels que la suite :

$$z_{n+1} = z_n^2 + c, c \in \mathbb{C}$$

ne diverge pas en module.

Exercice 1. Pour calculer l'appartenance d'un nombre complexe à l'ensemble fractal, on fixe un nombre maximum d'itérations ainsi qu'un module maximum. On calcule ensuite les valeurs successives de la suite. Si le module du complexe reste inférieur au module maximum `rmax` après le nombre maximum d'itérations `nbMaxIterations`, on considère qu'il appartient à l'ensemble fractal.

Ecrire une classe `Fractal` avec :

1. un constructeur permettant de fixer les valeurs `c`, `rmax` et `nbMaxIterations`,
2. une méthode `boolean belongsTo(double x, double y)` qui teste si le complexe de partie réelle `x` et de partie imaginaire `y` appartient à l'ensemble fractal.

Exercice 2. Pour créer une image à partir d'un ensemble fractal défini dans tout le plan complexe, il faut savoir quelle partie du plan complexe on souhaite visualiser. Si on suppose que cette fenêtre de visualisation est carrée, elle peut être décrite par son centre et son demi-côté. On doit alors mettre en correspondance cette zone et l'image binaire de l'ensemble fractal sur cette zone. Cette image peut être vue comme un tableau 2D contenant des 0 et des 1. La taille de l'image (nombre de pixels de son côté) peut être soit une constante, soit un paramètre de construction.

Ecrire une classe `FractalImage` qui permette de créer une image à partir d'un ensemble fractal. Les méthodes de cette classe sont :

1. `int getValue(int i, int j)` qui renvoie la valeur du pixel (i, j) : 1 dans le cas où le point correspondant appartient à l'ensemble fractal, 0 sinon ;
2. `int getSize()` qui renvoie la taille de l'image.

Exercice 3. Pour obtenir un fichier image à partir d'une image fractale, une solution simple est de créer un fichier au format PBM ascii. Un tel fichier permet de stocker une image binaire. Il contient les informations suivantes :

P1

```
<largeur image> <hauteur image>  
<valeur1> <valeur2> <valeur3> ...
```

Les valeurs sont 0 (blanc) ou 1 (noir). Elles sont lues en parcourant l'image de gauche à droite et de haut en bas.

1. Ecrire un programme de test qui crée l'ensemble de Julia avec les paramètres suivants :
nbMaxIterations=100, rmax=50, c=-0.8 + 0.156i.
2. Ecrire une méthode qui permette de sauvegarder une image d'un ensemble fractal dans un fichier PBM. Vous pourrez vous inspirer de l'exemple suivant

```
import java.io.*;  
  
public class TestFile{  
  
    public static void lineFile(String fileName, int n) {  
  
        try{  
            FileWriter file = new FileWriter (fileName);  
            for(int i = 0; i < n; ++i)  
                file.write ("ligne_" + i + "\n");  
            file.close();  
        }  
        catch (java.io.IOException e){  
            System.out.println("erreur_creation_fichier_image");  
        }  
    }  
  
    public static void main(String[] args){  
        lineFile("exemple.txt", 20);  
    }  
}
```

3. Tester en zoomant sur une zone du plan complexe : la taille de la fenêtre de visualisation diminue alors que la taille de l'image reste la même.

Exercice 4. On souhaite maintenant créer des images de l'ensemble fractal de Mandelbrot. La suite qui le définit est :

$$z_{n+1} = z_n^2 + z_0$$

Comment organiser le code de façon à éviter la duplication ?