Mini-projet Java (séance du 01/04/2022)

Exercice 1

1.1) @Override est présent car nous voulons redéfinir une méthode déjà définie par défaut. Ce n'est pas obligatoire de l'avoir, mais c'est une bonne idée, car cela fait comprendre au compilateur que nous voulons bien redéfinir une méthode existante.

1.2) Les attributs ne doivent pas être déclarés publics car ils pourraient être modifier par d'autres classes sans contrôle préalable. Ils doivent être mis en private et accessible via des getters/setters qui permettent de réguler leur modification.

1.3) Oui c'est possible, cependant si nous ne redéfinissons pas la méthode equals dans la classe comme c'est le cas ici, equals se contentera de regarder si la référence vers les objets sont les même, c'est-à-dire c1 == c2. Donc même si la valeur des attributs sont correctes, il nous renverra faux si ce sont deux objets instanciés séparément. Pour obtenir le bon résultat, il faut redéfinir la méthode equals en comparant chaque attribut et en renvoyant le booléen correspondant.

Exercice 2

2.1) voir Git

2.2) Le résultat devrait être « alpha=0.85, epsilon=0.001, indice=100, mode=CREUSE »

2.3) On peut déduire que CLIClassique.configuration de gère pas la conversion des nombres flottants en integer

2.4) Oui on peut le déduire car on parseInt une string, donc le compilateur n'émet pas de réserve, c'est bien une string qui est attendue en paramètre de parseInt.

2.5) voir Git

2.6)

Exercice 3

Voir Git

Exercice 4

Voir Git