

La classe Cercle

Objectifs :

- Écrire complètement une classe Java ;
- Revoir (et appliquer !) les concepts objets ;
- Savoir écrire la documentation d'une classe ;
- Savoir tester une classe avec JUnit.

Exercice 1 : Écrire une classe Cercle

Un cercle est une courbe plane fermée constituée des points situés à égale distance d'un point nommé *centre*. La valeur de cette distance est appelée *rayon* du cercle. On appelle diamètre la distance entre deux points diamétralement opposés. La valeur du diamètre est donc le double de la valeur du rayon (source : Wikipédia). La figure 1 présente trois cercles.

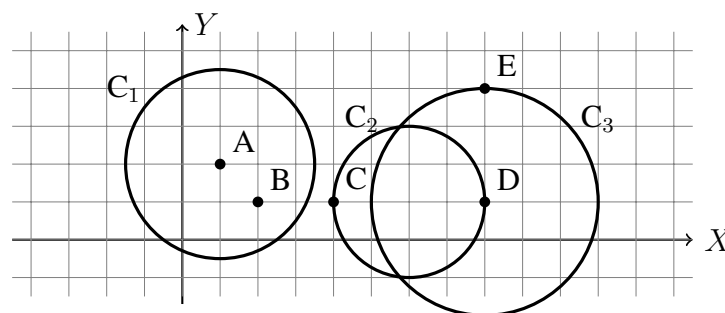


FIGURE 1 – Quelques exemples de cercles

Spécification

On demande d'écrire une classe Cercle respectant les exigences suivantes.

- E₁ On peut translater un cercle en précisant un déplacement suivant l'axe des X et un déplacement suivant l'axe des Y.
- E₂ On peut obtenir le centre d'un cercle.
- E₃ On peut obtenir le rayon d'un cercle.
- E₄ On peut obtenir le diamètre d'un cercle.
- E₅ On peut savoir si un point est à l'intérieur (au sens large) d'un cercle. Par exemple, les points A et B sont à l'intérieur du cercle C₁ et le point C à l'extérieur. Le point E est à l'intérieur du cercle C₃.

- E₆ On peut obtenir son périmètre et son aire (en fait, il s'agit de l'aire de la surface délimitée par le cercle). Le périmètre d'un cercle est donnée par la formule $2\pi R$ où R représente le rayon du cercle. L'aire est πR^2 .
- E₇ Non applicable.
- E₈ Le cercle possède une couleur.
- E₉ On peut obtenir la couleur d'un cercle.
- E₁₀ On peut changer la couleur d'un cercle.
- E₁₁ On peut construire un cercle à partir d'un point qui désigne son centre et d'un réel correspondant à la valeur de son rayon. Sa couleur est considérée comme étant le bleu. Par exemple, le cercle C_1 est construit à partir du point A de coordonnées $(1, 2)$ et du rayon $2,5$.
- E₁₂ On peut construire un cercle à partir de deux points diamétralement opposés. Sa couleur est considérée comme étant le bleu. Par exemple, le cercle C_2 est construit à partir des deux points C et D .
- E₁₃ On peut construire un cercle à partir de deux points diamétralement opposés et de sa couleur.
- E₁₄ Une méthode de classe `creerCercle(Point, Point)` permet de créer un cercle à partir de deux points, le premier correspond au centre du cercle et le deuxième est un point du cercle (de sa circonférence). Ces deux points forment donc un rayon du cercle. Par exemple, le cercle C_3 est construit à partir des points D (centre) et E (circonférence). Le cercle est bleu.
- E₁₅ Lorsqu'un cercle est affiché sur le terminal, il est affiché sous la forme suivante `Cr@(a, b)` où r est la valeur du rayon et (a, b) le centre du cercle, par exemple `C2.5@(1.0, 2.0)`.
- E₁₆ On peut changer le rayon du cercle.
- E₁₇ On peut changer le diamètre du cercle.
- E₁₈ On ne doit pas pouvoir changer les caractéristiques d'un cercle sans passer par les opérations de modification que la classe propose (`translater`, `setRayon`, `setDiametre`, `setCouleur...`).
- E₁₉ La classe `SujetCercleTest` est une classe de test qui permet de compléter les exigences données ici. La classe `Cercle` doit donc réussir ces tests.

Contraintes

Les contraintes de réalisation à respecter *impérativement* sont les suivantes :

- C₁ Non applicable.
- C₂ Non applicable.
- C₃ Les principes énoncés en cours doivent être respectés lors de la réalisation de ce travail.
- C₄ Il est nécessaire de partir des classes fournies (voir site du module).
- C₅ Il est interdit de modifier les classes fournies, qu'elles soient de test ou non.
- C₆ Les exigences E₁₂, E₁₃ et E₁₄ ne sont pas testées par `SujetCercleTest`. En conséquence, vous devez écrire une classe `CercleTest` (nom à respecter impérativement) qui teste ces exigences et uniquement celles-ci. Elle devra s'appuyer sur `JUnit 4`.

- C₇ Les tests de `SujetCercleTest` (et `CercleTest`) ne sont pas exhaustifs. Vous pouvez ajouter de nouveaux tests dans la classe `ComplementsCercleTest` (nom à respecter), toujours en utilisant JUnit 4.
- C₈ Les commentaires de documentation doivent être fournis au format javadoc.
- C₉ On utilisera le mot-clé `assert` de Java pour exprimer toutes les préconditions des méthodes.
- C₁₀ Non applicable.
- C₁₁ On ne stockera pas d'informations redondantes.
- C₁₂ On définira dans la classe `Cercle` une constante appelée `PI` qui sera initialisée à la valeur de `PI` donnée dans la classe `Math`.
- C₁₃ Les lettres accentuées ne doivent pas être utilisées dans les identifiants.