

# 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.

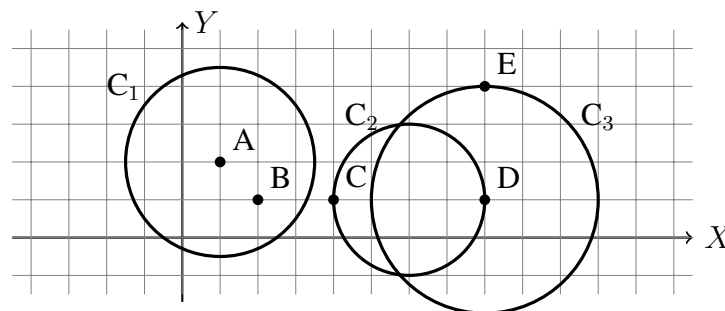


FIGURE 1 – Quelques exemples de cercles

## Spécification

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

- ~~E<sub>1</sub>~~ On peut traduire un cercle en précisant un déplacement suivant l'axe des X et un déplacement suivant l'axe des Y.
- ~~E<sub>2</sub>~~ On peut obtenir le centre d'un cercle.
- ~~E<sub>3</sub>~~ On peut obtenir le rayon d'un cercle.
- E<sub>4</sub> On peut obtenir le diamètre d'un cercle.

- ~~E<sub>5</sub>~~ 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  $C_1$  et le point  $C$  à l'extérieur. Le point  $E$  est à l'intérieur du cercle  $C_3$ .
- ~~E<sub>6</sub>~~ Un cercle est un Mesurable2D (interface Mesurable2D). À ce titre, 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  $\pi R^2$ .
- ~~E<sub>7</sub>~~ Cette exigence a été supprimée.
- E<sub>8</sub> Le cercle possède une couleur qui est utilisée pour dessiner sa circonférence.
- ~~E<sub>9</sub>~~ On peut obtenir la couleur d'un cercle.
- ~~E<sub>10</sub>~~ On peut changer la couleur d'un cercle.
- ~~E<sub>11</sub>~~ 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  $2,5$ .
- E<sub>12</sub> 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  $D$ .
- E<sub>13</sub> On peut construire un cercle à partir de deux points diamétralement opposés et de sa couleur.
- E<sub>14</sub> 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.
- E<sub>15</sub> 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)`.
- E<sub>16</sub> On peut changer le rayon du cercle.
- E<sub>17</sub> On peut changer le diamètre du cercle.
- E<sub>18</sub> 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...`).
- E<sub>19</sub> La classe `SujetCercleTest` est une classe de test qui précise les exigences données ici. La classe `Cercle` doit donc réussir ces tests.

## Contraintes

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

- C<sub>1</sub> Ce travail est un travail individuel. *Toute tricherie sera sanctionnée par la note minimale.*

- C<sub>2</sub> Le programme doit fonctionner sans aucune modification sur les machines Unix des salles de TP du bâtiment C de l'ENSEEIH.
- C<sub>3</sub> Les principes vus en cours, TD et TP doivent être respectés lors de la réalisation de ce travail.
- C<sub>4</sub> Il est nécessaire de partir des classes fournies sur le répertoire SVN (voir page du module).
- C<sub>5</sub> Il est interdit de modifier les classes fournies, qu'elles soient de test ou non.
- C<sub>6</sub> Les exigences E<sub>12</sub>, E<sub>13</sub> et E<sub>14</sub> 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.
- C<sub>7</sub> Les tests de SujetCercleTest (et CercleTest) ne sont pas exhaustifs. Vous pouvez ajouter de *nouveaux*<sup>1</sup> tests dans la classe ComplementsCercleTest (nom à respecter), toujours en utilisant JUnit 4.
- C<sub>8</sub> Les commentaires de documentation doivent être fournis au format javadoc.
- C<sub>9</sub> On utilisera le mot-clé assert de Java pour exprimer toutes les préconditions des méthodes.
- C<sub>10</sub> ~~Cette contrainte a été supprimée.~~
- C<sub>11</sub> On ne stockera pas d'informations redondantes.
- C<sub>12</sub> On définira dans la classe Cercle une constante appelée PI qui sera initialisée à la valeur de PI donnée dans la classe Math.
- C<sub>13</sub> Les lettres accentuées ne doivent pas être utilisées dans les identifiants.

## Livrables

Les livrables sont :

- L<sub>1</sub> le fichier Cercle.java,
- L<sub>2</sub> le fichier CercleTest.java,
- L<sub>3</sub> le fichier ComplementsCercleTest.java,
- L<sub>4</sub> le fichier LISEZ-MOI.txt complété (réponse aux questions et explications utiles à la compréhension des livrables rendus).

**Attention :** Les consignes et les dates à respecter sont disponibles sur la page du module Technologie Objet (Moodle).

---

1. Il ne faut pas reprendre les tests de SujetCercleTest ou CercleTest !