

Interfaces et classes abstraites

Dans ce projet, nous souhaitons modéliser des graphiques, qui pourront être construits de manière programmée (en Java) puis exportés au format SVG (*Scalable Vector Graphics*). Ces graphiques pourront contenir des cercles, des rectangles et du texte. Un exemple simple est disponible sur le site de MDN :

https://developer.mozilla.org/fr/docs/Web/SVG/Tutorial/Getting_Started



Le code correspondant est le suivant :

```
<svg version="1.1"
  width="300" height="200"
  xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="red" />

  <circle cx="150" cy="100" r="80" fill="green" />

  <text x="150" y="125" font-size="60" text-anchor="middle" fill="white">SVG</text>

</svg>
```

1 Implantation

1. Définir une interface **Figure**, et trois classes qui implante **Figure** : **Rectangle**, **Circle**, et **Text**. La position d'une figure est (0, 0) si elle n'est pas précisée (comme pour **rect** dans l'exemple).

Les attributs de ces classes, et leur types, viendront de l'exemple ci-dessus et de la documentation en ligne de MDN :

<https://developer.mozilla.org/fr/docs/Web/SVG/Element/rect>

2. Ajouter une méthode **toString** à chacune de ces classes.
3. Définir une classe **Graphics**, un graphique possède une taille donnée (largeur et hauteur) et contient une liste de figures.
4. Ajouter une méthode **toString** à la classe **Graphics**.

5. Ajouter de *getter/setter* pour la position à la classe `Rect`, puis ajouter ces méthodes à l'interface `Figure` et les implanter aussi pour `Text` et `Circle`.
6. Ajouter à l'interface `Figure` une méthode `moveTo` qui permet de déplacer une figure du graphique. Implanter cette méthode pour chacune des figures.


```
void moveTo(int x, int y)
```
7. Ajouter une classe abstraite `AbstractFigure` qui implante l'interface `Figure` et modifier votre code de manière à ce que `moveTo` soit implanté au niveau de `AbstractFigure`.
8. (*optionnel*). Ajouter une méthode qui permet de sauver une figure directement dans un fichier au format SVG.

2 A retenir

- Une interface sert pour le typage, elle annonce les méthodes communes aux classes qui l'implament.
- Une classe abstraite peut aussi servir pour le typage, mais elle sert en général à factoriser du code (et on définit aussi l'interface correspondante).
- On ne peut pas créer d'instance d'une interface ni d'une classe abstraite.
- Une classe peut implanter plusieurs interface, mais ne peut étendre qu'une seule classe.

A Diagramme de classes (partiel)

