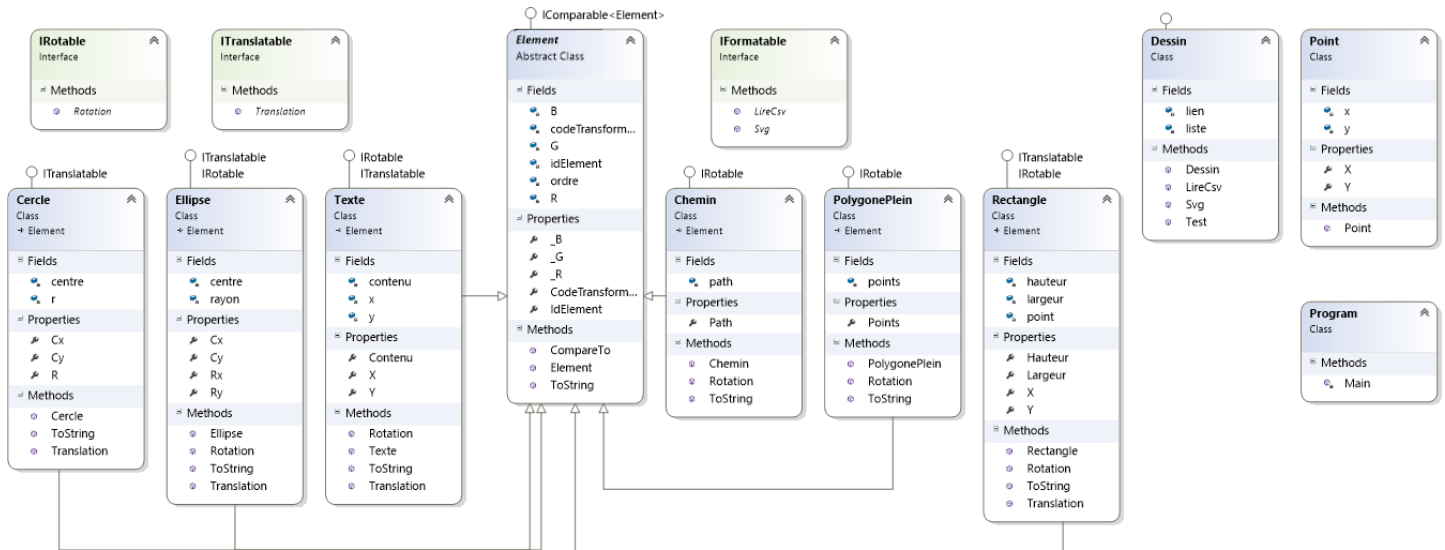


## TD final POO

### Exercice 1 :

- Proposez une modélisation objet de l'ensemble des données.



- A quelle classe doit-on attacher la lecture csv ? l'écriture svg ?

La lecture csv et l'écriture svg issue de l'interface IFormatable doivent être attachées à la classe Dessin.

- Comment assurer que les éléments soient écrits dans le bon ordre dans le fichier svg ?

Afin d'assurer que les éléments soient écrits dans le bon ordre, on peut vérifier le code svg en ouvrant le fichier svg dans un éditeur de code. Ainsi, on aura accès au code svg et on pourra en fonction des erreurs modifier la méthode Svg.

### Exercice 2 : Ecrire, sur papier, l'algorithme de lecture et de traitement d'un fichier csv.

Je crée un objet de type StreamReader à partir du lien me permettant d'accéder à mon fichier csv.

Je parcours ensuite mon fichier ligne par ligne à partir de la méthode Readline().

Tant qu'il y a des lignes dans le fichier : on retrouve 2 cas.

- 1<sup>er</sup> cas : Soit le premier élément de la ligne est un élément (de type cercle, rectangle etc...). Dans ce cas, on instancie un objet en fonction du nom de l'élément qu'on retrouve dans la ligne. Les classes associées ont déjà été créées avec leur constructeur et ceci en fonction du nombre d'attributs que l'on trouve.
- 2<sup>ème</sup> cas : Soit le premier élément de la ligne est une transformation de type Translation ou Rotation. Dans ce cas, selon le type de transformation, on déclenche la méthode associée. Cette méthode a pour but de travailler sur le dernier attribut codeTransformation d'un objet

de type `Elément`. En effet, cette méthode va introduire un contenu de type `string` en adéquation avec le `code.svg`. Ce dernier attribut sera utilisé lors de l'écriture du fichier `.svg`.

Je ferme le fichier et je retourne ma liste.

**Exercice 3 : Codez chacune des classes et réalisez la conversion des fichiers csv vers svg.**

Voir le programme.

**Exercice 4 : Proposez la gestion d'erreurs par exceptions.**

Nous avons géré les erreurs liées à la lecture du fichier csv.

**Exercice 5 : Quelle(s) classe(s) est(sont) susceptible(s) d'être abstraite(s) ?**

La classe `Element` est une classe abstraite. En effet, on n'utilise pas explicitement des objets de type `Element`. Par contre, les classes qui héritent de `Element` (`Cercle`, `Rectangle`, `Texte`, ...) utilisent les attributs de `Element`. (D'où, `Element` classe abstraite).

**Remarques supplémentaires :**

- Attribut `codeTransformation` dans la classe `Element`

Dans chaque classe héritant de la classe `Element`, nous avons ajouté un attribut supplémentaire de type `string` : `codeTransformation`. Cet attribut sera initialisé comme étant vide. Lors de la lecture du fichier csv, si l'on trouve une transformation de type translation ou rotation, alors la méthode translation ou la méthode rotation (se trouvant dans la classe `Dessin`) concaténera un bout de code svg de type transform dans l'attribut `codeTransformation`. Ainsi, si lors de l'écriture, l'attribut `codeTransformation` de la classe d'une figure n'est pas vide alors on l'ajoutera dans le code svg en l'associant avec une balise transform dans le fichier `.svg`.

Ainsi, l'ajout d'une transformation dans un fichier `.svg` se fera de manière simple.

- Utilisation des interfaces `ITranslatable` et `IRotable` :

Ces interfaces contiennent chacune une méthode (Translation et Rotation). L'objectif est d'attribuer les transformations aux classes héritant de la classe `Element`. Ceci permettra d'interdire des transformations à ceux qui ne devraient y avoir accès.

- Utilisation de l'interface `Comparable` :

L'interface par défaut `Comparable` nous permet d'utiliser ou redéfinir la méthode `CompareTo()` qui nous permet d'ordonner les éléments par leur attribut ordre (attribut qui donne l'ordre d'affichage des différents éléments).

- Méthode test de la classe Dessin pour l'ajout ou non d'un élément selon le nombre d'éléments par ligne (fichier csv) :

Cette méthode va vérifier s'il y a bien le nombre exact d'éléments sur une ligne en fonction de la classe (Cercle, Rectangle, ...). S'il y a le nombre exact, il y a ajout des informations dans la liste sinon on passe sans prendre en compte la ligne.