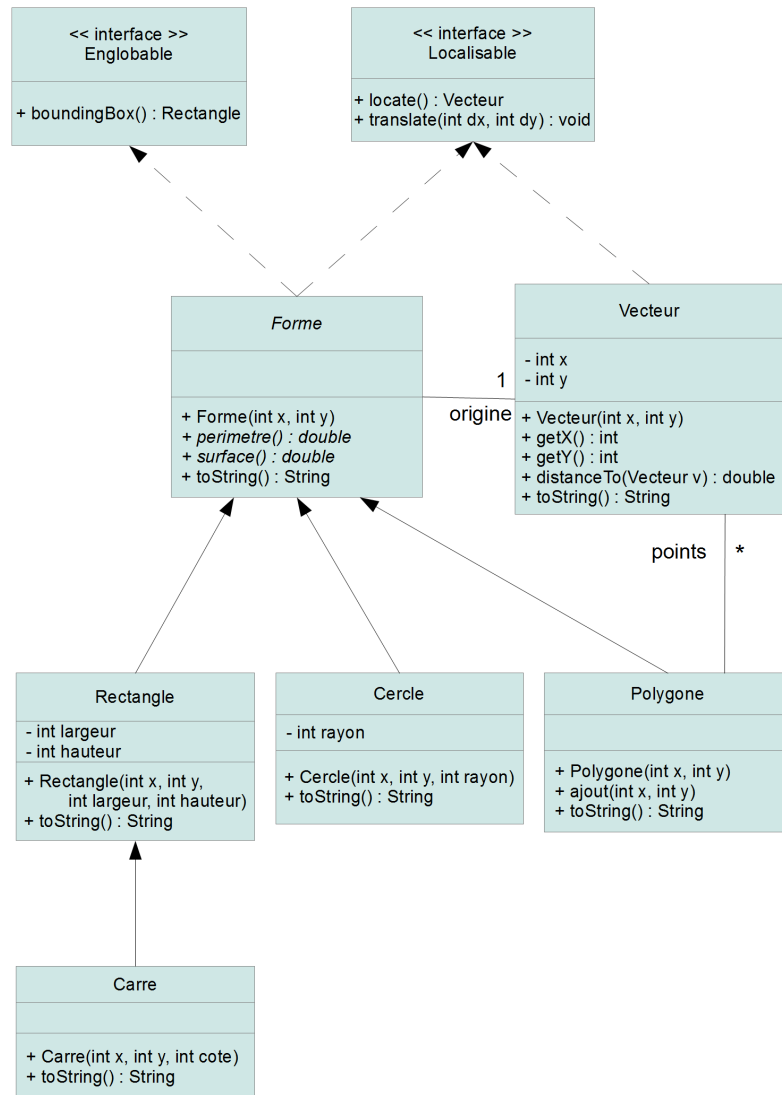


# TD 10 : FORMES



Créez un ensemble de classes et d'interfaces pour représenter des formes selon le diagramme des classes précédent. Ce diagramme ne contient pas forcément toutes les surcharges de méthodes pour vous laisser déduire des cas pour lesquels la surcharge est pertinente (en particulier pour les méthodes `boundingBox`, `locate`, `translate`, `perimetre` et `surface`).

## Notes :

- La méthode `boundingBox` retourne le rectangle englobant de la forme
- En géométrie, un point est assimilable à un vecteur, d'où le nom de la classe de notre diagramme.
- Toutes les formes ont une origine. Dans le cas d'un quadrilatère il s'agit du coin haut-gauche. Dans le cas d'un cercle il s'agit de son centre. Dans le cas d'un polygone quelconque il s'agit du premier point donné.

- La méthode ajout de la classe Polygone permet d'ajouter un sommet. On considère que les côtés du polygone rejoignent les sommets dans l'ordre où ils sont donnés, un dernier côté reliant le dernier sommet au premier.
- La surface d'un polygone quelconque est une formule très complexe à programmer et ne présente pas particulièrement d'intérêt, vous pouvez retourner -1 à la place pour signaler que l'on ne sait pas la calculer.

**Question subsidiaire :** de quelle(s) classe(s) peut-on stocker des instances dans les variables suivantes :

- Localisable toto ;
- Englobable titi ;
- Forme tata ;
- Vecteur tutu ;