

Héritage

1 Point2D

Ajout d'une classe fille

Reprenez votre projet qui traite des Point2D.

Créez une nouvelle classe qui va hériter de Point2D, aura comme nom fr.dessin.Point3D ainsi que les contraintes suivantes :

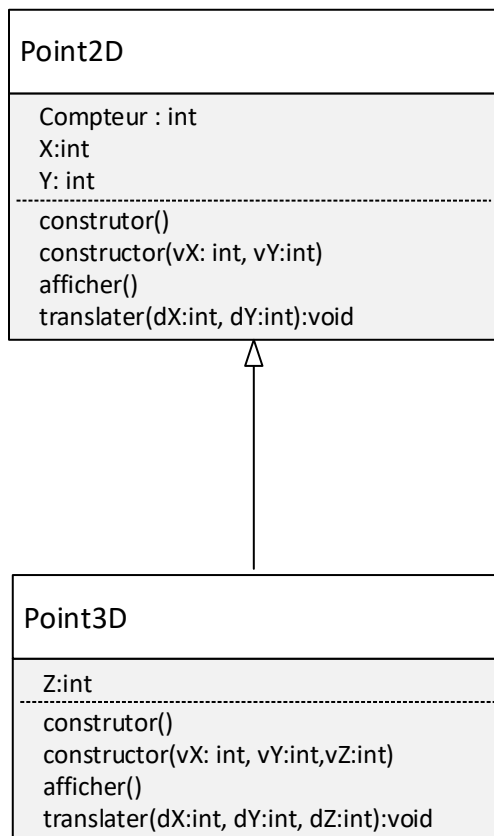
- un nouvel attribut de type entier: z
- deux constructeurs, un sans paramètre, un qui prendra des valeurs d'initialisations pour x, y, z
- une propriété qui permette de fixer et récupérer la valeur de z
- Redéfinissez la méthode affichage afin qu'elle affiche aussi la valeur de z. Par exemple :

```
[4, 34, 45]
```

- Réalisez une nouvelle méthode translater afin qu'elle translate le Point3D sur ses trois axes.

En C# l'héritage de classe est unique \Leftrightarrow une classe ne peut avoir qu'un seul parent direct.

Ici, Point3D appartient à la famille des Point2D ainsi qu'à la famille de System.Object



Dans ce cas de figure, Point3D possède une méthode *translater(dX, dY)* et une méthode *translater(dX, dY, dZ)*

Test

Reprenez votre classe Run/Program, ajoutez des instances de Point3D, utilisez ses méthodes et vérifiez que votre nouvelle classe fonctionne.

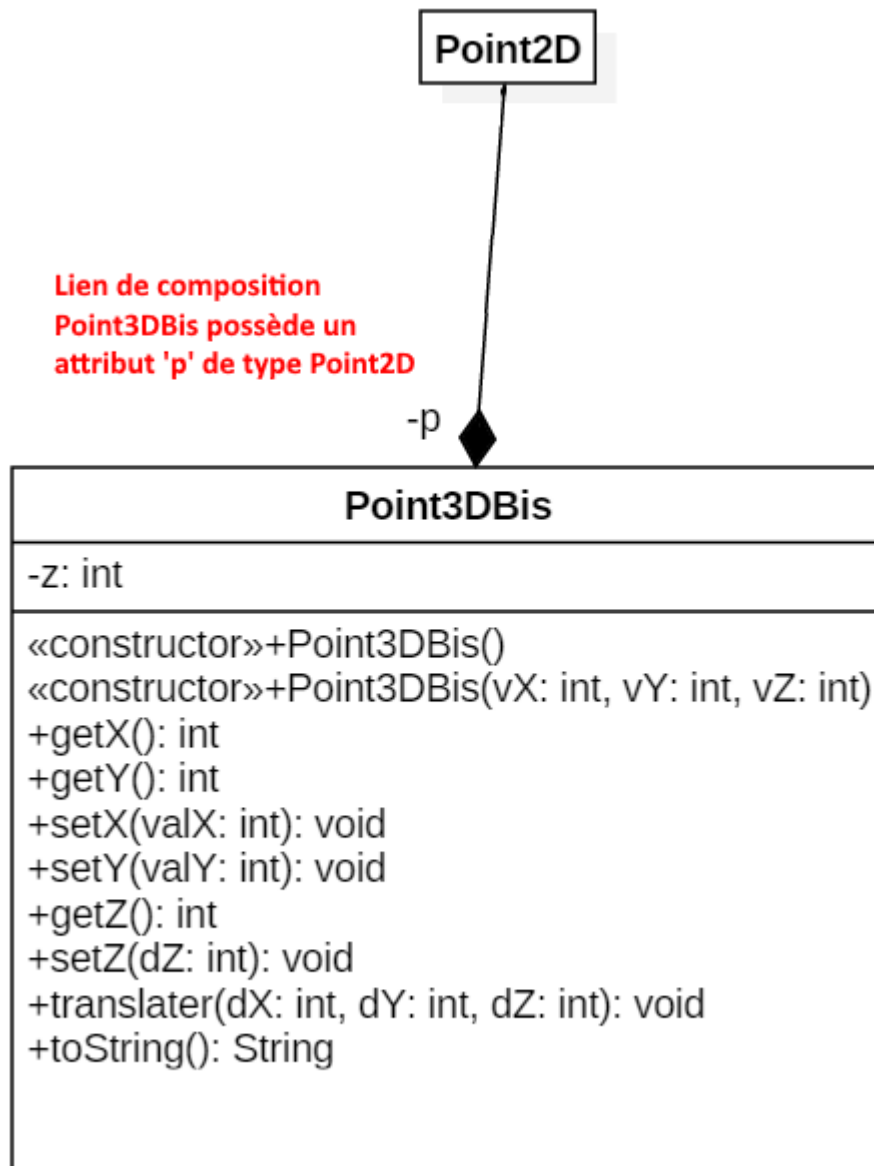
Par composition

Si vous pensez que l'héritage n'est pas une bonne chose entre Point2D et Point3D, vous auriez pu faire usage de la composition.

Dans ce cas-là, Point3D n'hérite plus de Point2D mais possède un attribut de type Point2D. De cet manière, Point3D n'appartient pas à la famille des Point2D.

Ici Point3D n'appartient pas à la famille de Point2D, il n'appartient qu'à la famille de [java.lang.Object](#)

En composition, les méthodes getX, getY, setX, setY délèguent leur comportement à l'instance de Point2D qui fait partie de Point3D.



Dans ce cas de figure, Point3D ne possède pas de méthode *translater*(dX, dY)