

---

## **TP 06 Tob**

Héritage comme spécialisation

Julien Blanchon

08-02-2020

---

This is the abstract

## Table des matières

|  |          |
|--|----------|
| <b>Exercice 1 : Formaliser le schéma</b>                         | <b>4</b> |
| <b>Exercice 2 : Écrire la classe X et adapter l'application</b>  | <b>4</b> |
| <b>Exercice 3 : Construire le schéma en utilisant les listes</b> | <b>5</b> |
| <b>Exercice 4 : Définir un groupe</b>                            | <b>5</b> |

---

## Liste des tableaux

## Table des figures

## Exercice 1 : Formaliser le schéma

Dans la question 3.2 du TP 5, nous avons défini le schéma comme plusieurs objets (points, points nommés et segments) qui sont référencés par des variables différentes. Il serait plus logique et pratique d'avoir une seule variable qui représente le schéma (on l'appellera naturellement `schema`). Comme un schéma est constitué d'un nombre variable d'éléments, on peut le représenter par un tableau. Si nous appelons `X` le type des éléments de ce tableau, nous pouvons alors écrire le code du listing 1

1. Indiquer à quelles conditions sur `X` les lignes suivantes compilent.

```
1 class MaClasse{
2   this.function{}
3 }
```

**Listing 1** – test caption

```
class MaClass{}
```

Il faut que `X` soit un tableau dont les éléments ne sont pas privée pour pouvoir `Point.afficher`  
...

2. Quel code sera exécuté pour `x.afficher()` et `x.translater(4, -3)` ?

`Point.afficher()` et `Point.translater(double, double)`

3. Indiquer les autres éléments à définir sur `X` ? Justifier la réponse.

∅

4. Donner un nom plus significatif à `X`.

`SchemaTab` serait plus significatif pour `X`.

## Exercice 2 : Écrire la classe `X` et adapter l'application

1. Est-ce que l'on sait écrire le code des méthodes `afficher` ou `translater` de `X` ?

Oui si tout les éléments de `X` possède les méthodes `afficher` et `translater`. Ce qui est la cas car `X` peut contenir soit un `Point`, soit un `PointNomme`, soit un `Segment`. Il serait judicieux de faire un interface: `ObjectGeometrique`.

- 
2. Peut-on créer des instances de X ?

Oui ?

3. Quels constructeurs définir sur X ?

?

4. Quand ces constructeurs seront-ils appelés ?

Lors de la création de X ?

5. Écrire le code de la classe X.
6. Lister et effectuer les modifications à apporter aux autres classes de l'application.

## Exercice 3 : Construire le schéma en utilisant les listes

Au lieu d'utiliser un tableau comme dans l'exercice 2, on veut utiliser l'interface List et sa réalisation ArrayList du paquetage java.util (en particulier la méthode add et la structure de contrôle foreach).

1. Indiquer les avantages et inconvénients des listes par rapport aux tableaux.
2. Construire le schéma en utilisant une liste.

## Exercice 4 : Définir un groupe

Dans un éditeur de schémas mathématiques, il serait pratique de pouvoir grouper plusieurs X pour les manipuler comme un seul et leur appliquer à tous, en une seule fois, la même opération (translater, afficher, etc.).

1. Sachant que la classe X est abstraite, la classe Groupe est-elle abstraite ou concrète ?
2. Écrire la classe Groupe et l'utiliser (ExempleSchemaGroupe).
3. On souhaite pouvoir mettre un groupe dans un groupe. Par exemple, on souhaite grouper les trois segments, puis ce groupe et le barycentre. Comment faire ?