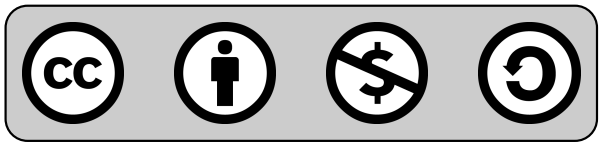
IN201 Conception et Programmation Orientées Objet

TP Observateur : diagrammes établis en classe

Christophe Garion

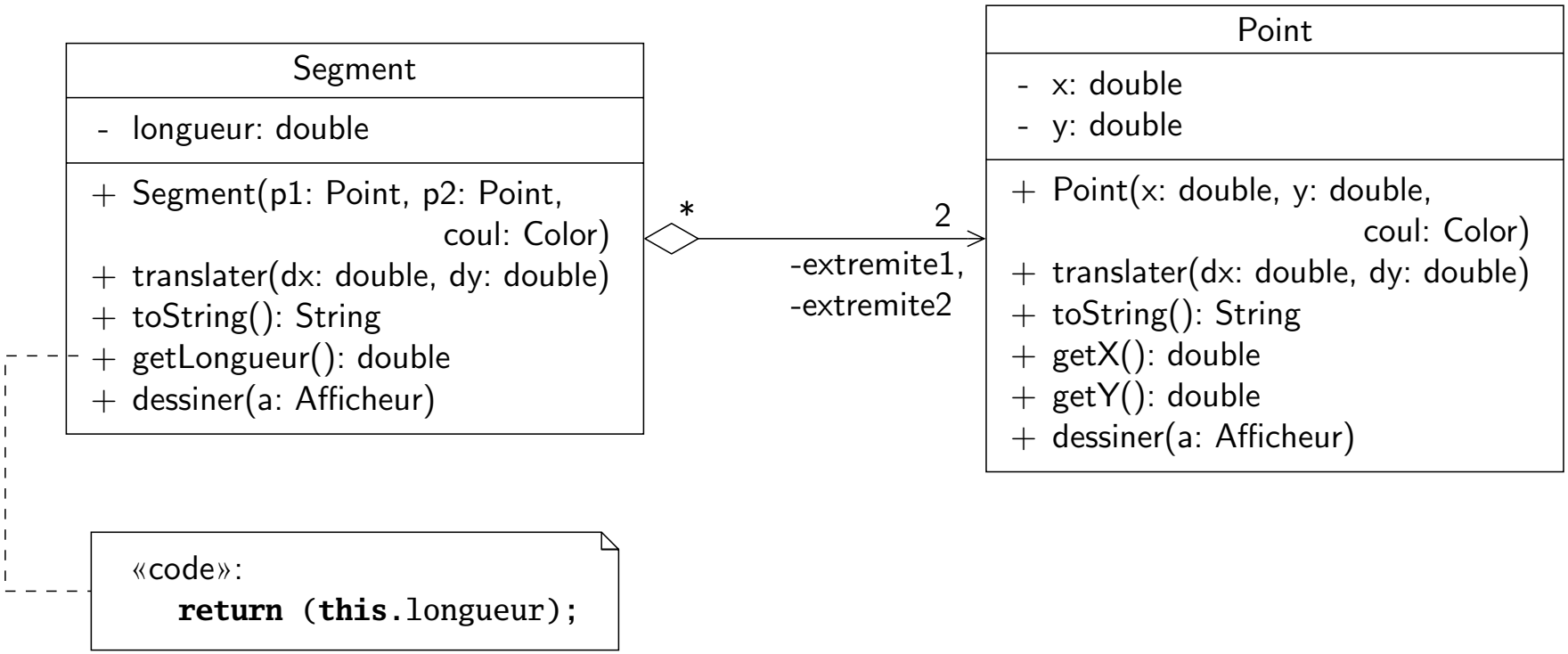
DMIA – ISAE

License CC BY-NC-SA 3.0



Une nouvelle implantation de Segment

On choisit maintenant d'implanter la longueur d'un segment comme un attribut de la classe Segment.



Préparation de la validation

Exercice

Indiquer ce qui devrait apparaître à l'écran lors de l'exécution du programme de test.

Exercice

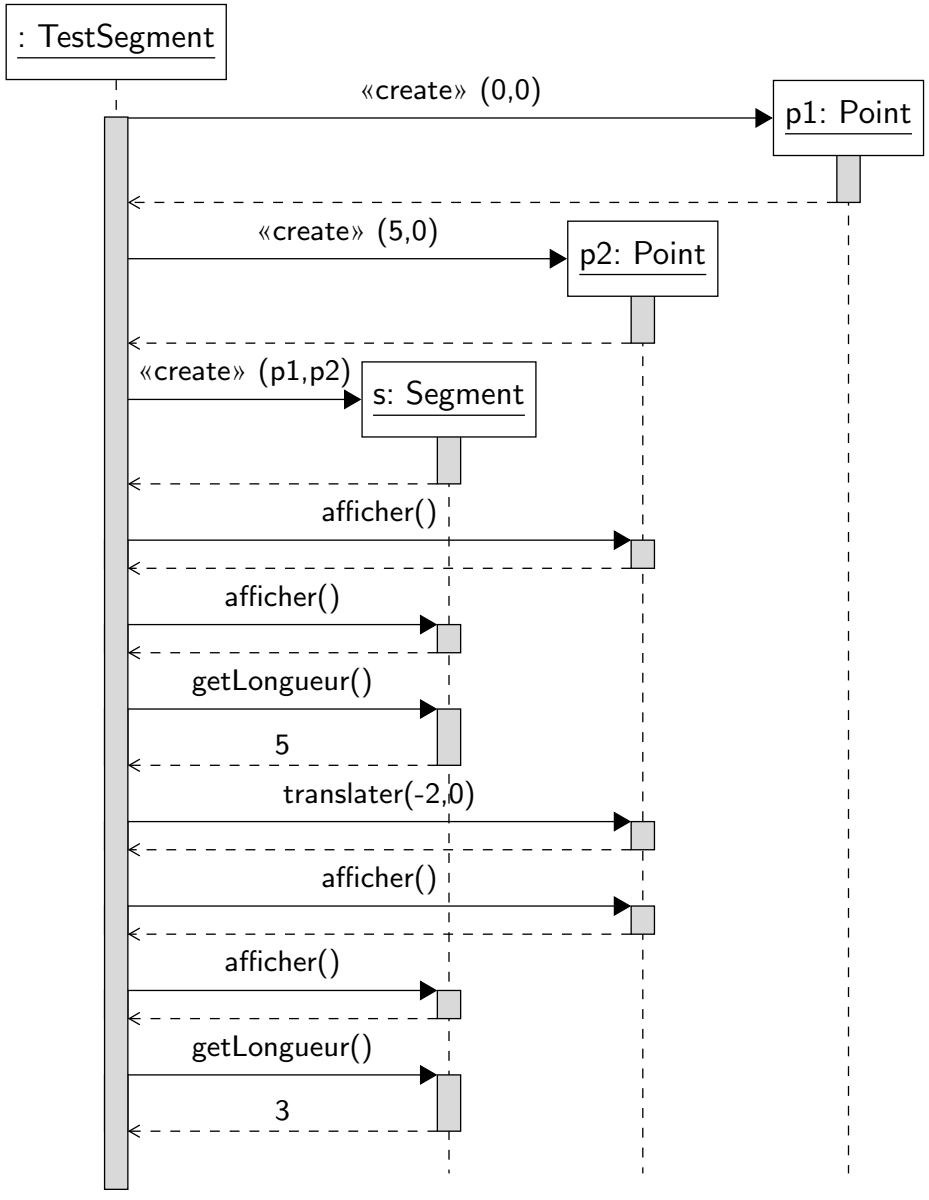
Écrire un diagramme de séquence représentant le scénario.

Préparation de la validation : sortie console

Shell

```
p2 = (5.0,0.0)

s = [(0.0,0.0);(5.0,0.0)]

longueur de s = 5.0

p2 = (3.0,0.0)

s = [(0.0,0.0);(3.0,0.0)]

longueur de s = 3.0
```

Préparation de la validation : diagramme de séquence



Première implantation de Segment

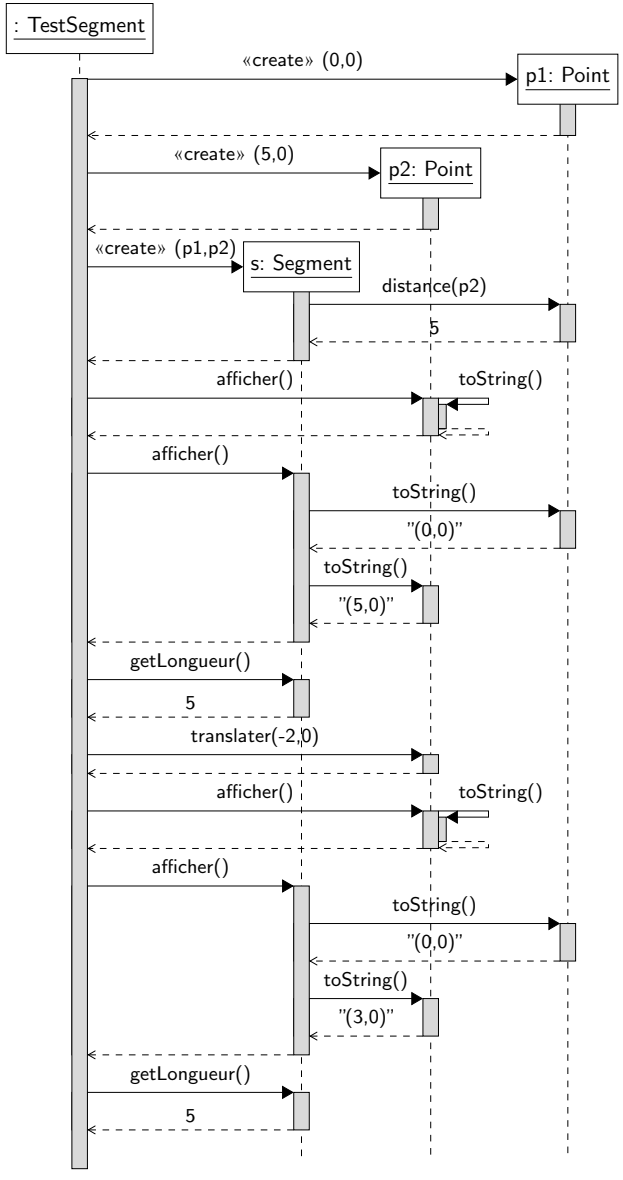
Exercice

En utilisant les sources fournies, compléter le diagramme de séquence précédent.

Exercice

Indiquer ce qui devrait apparaître à l'écran lors de l'exécution du programme de test en utilisant les sources fournies.

Première implantation : diagramme de séquence



Première implantation : sortie console

Shell

```
p2 = (5.0,0.0)

s = [(0.0,0.0);(5.0,0.0)]

longueur de s = 5.0

p2 = (3.0,0.0)

s = [(0.0,0.0);(3.0,0.0)]

longueur de s = 5.0
```

Correction des classes

Hypothèse

- la relation entre `Segment` et `Point` reste inchangée
- l'attribut `longueur` et la méthode `getLongueur()` de `Segment` restent inchangés

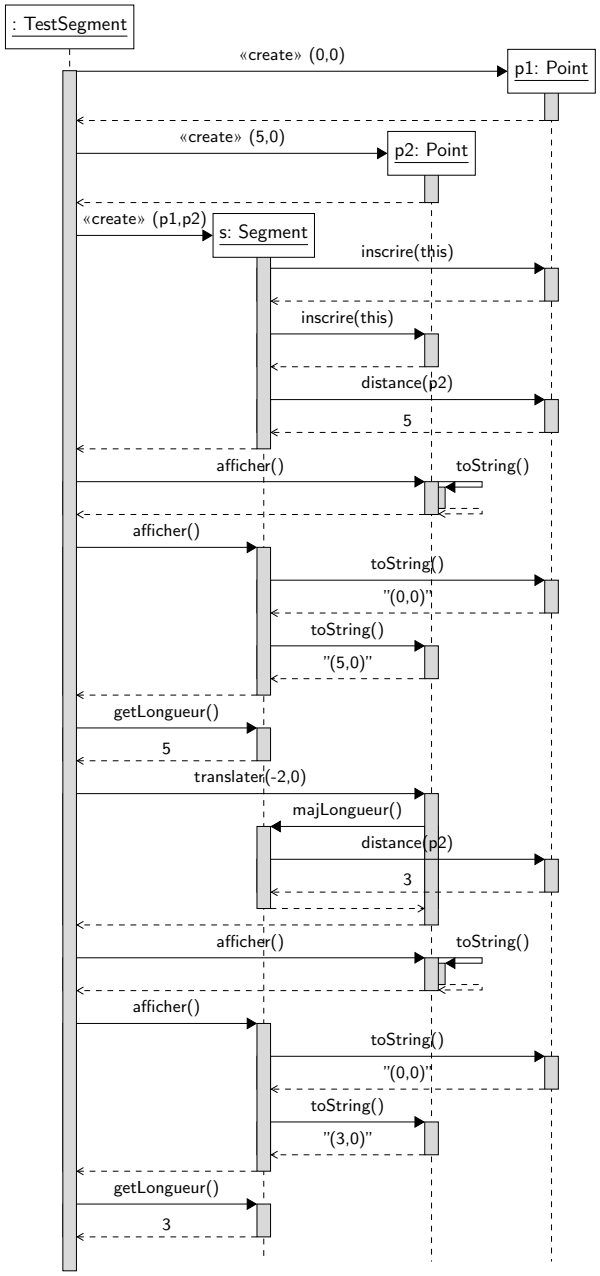
Exercice

Indiquer les modifications à apporter en complétant le diagramme de séquence précédent.

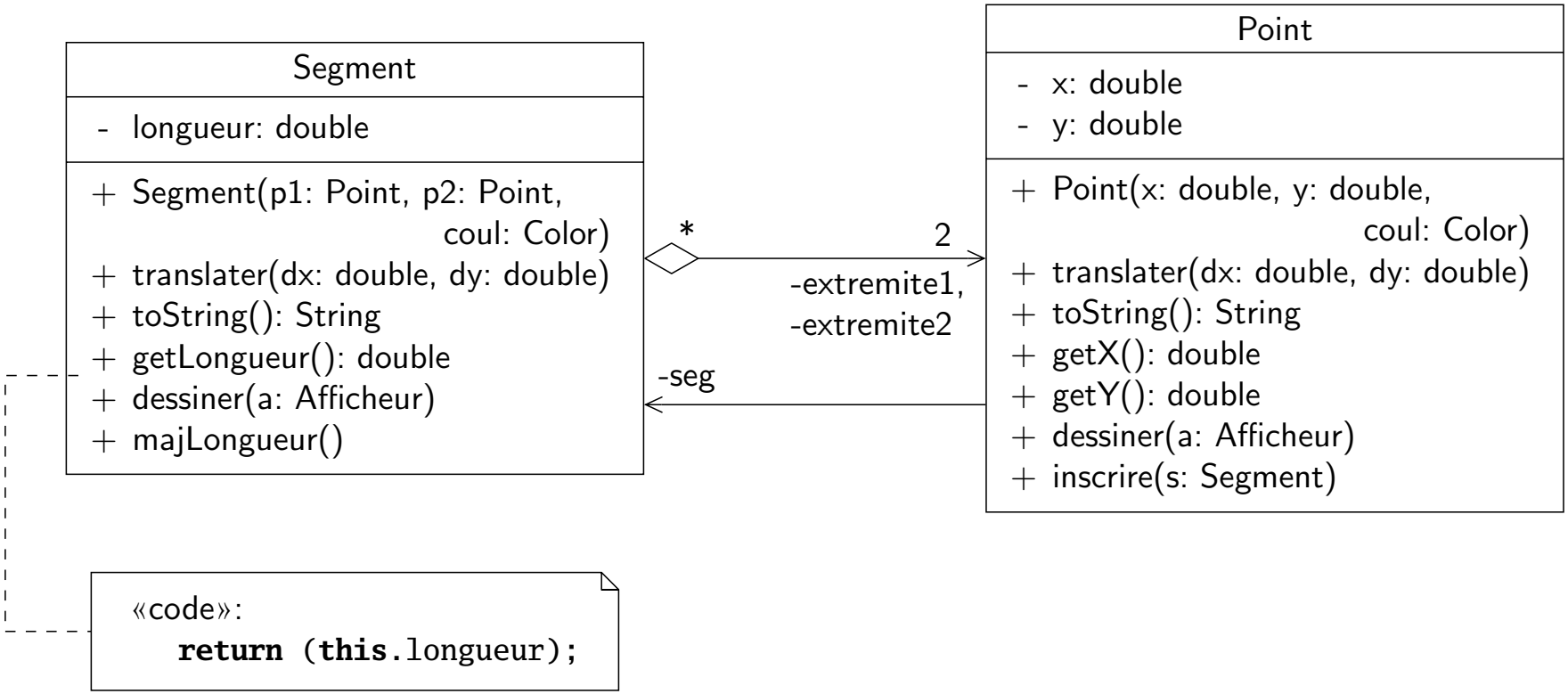
Exercice

Modifier le diagramme de classes initial en conséquence.

Correction des classes : diagramme de séquence



Correction des classes : diagramme de classes

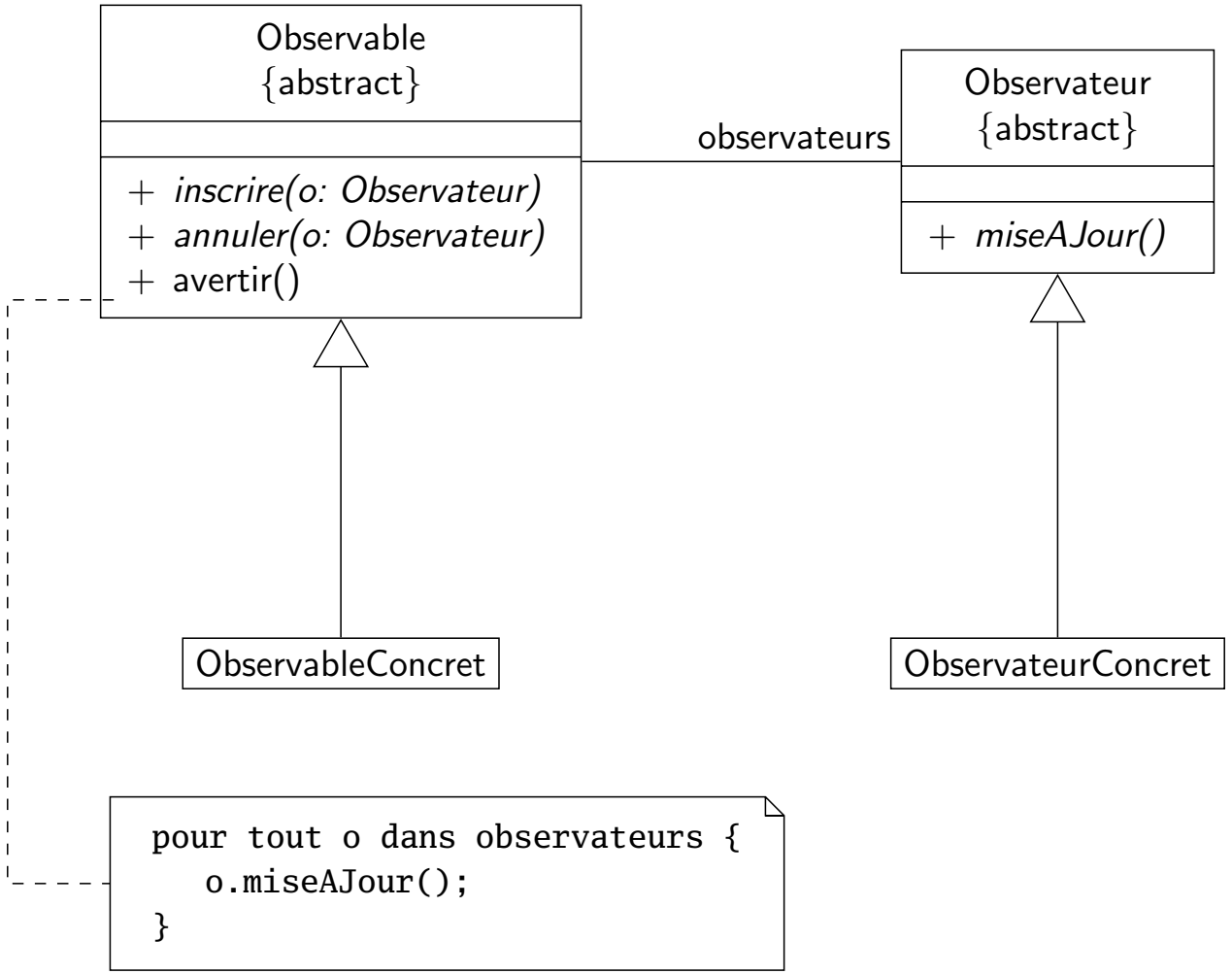


Dernières mises au point

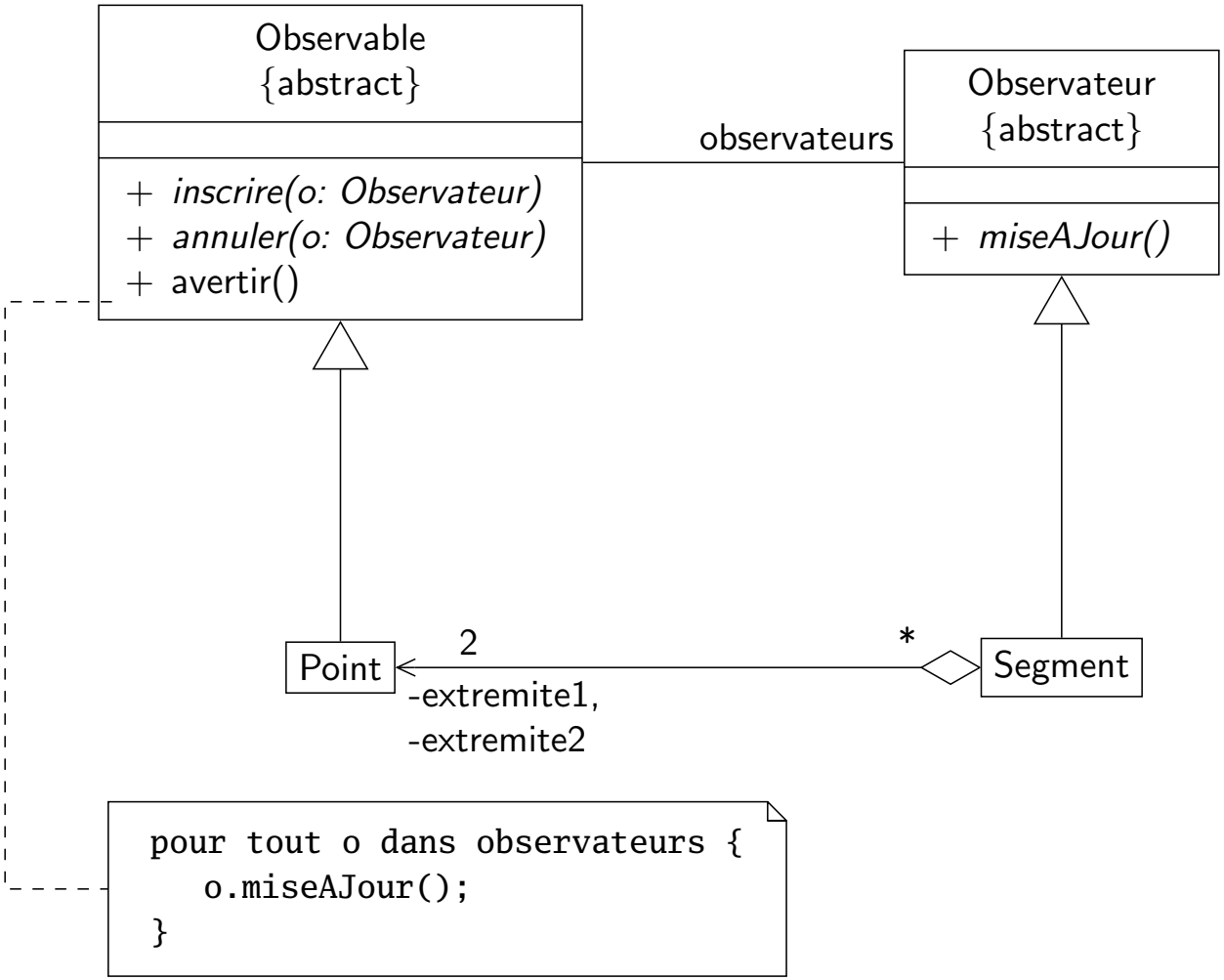
Exercice

- un point peut-il être extrémité de plusieurs segments ?
- la solution peut-elle adaptée à un cas plus général ?

Le patron de conception observateur



Observateur pour notre problème



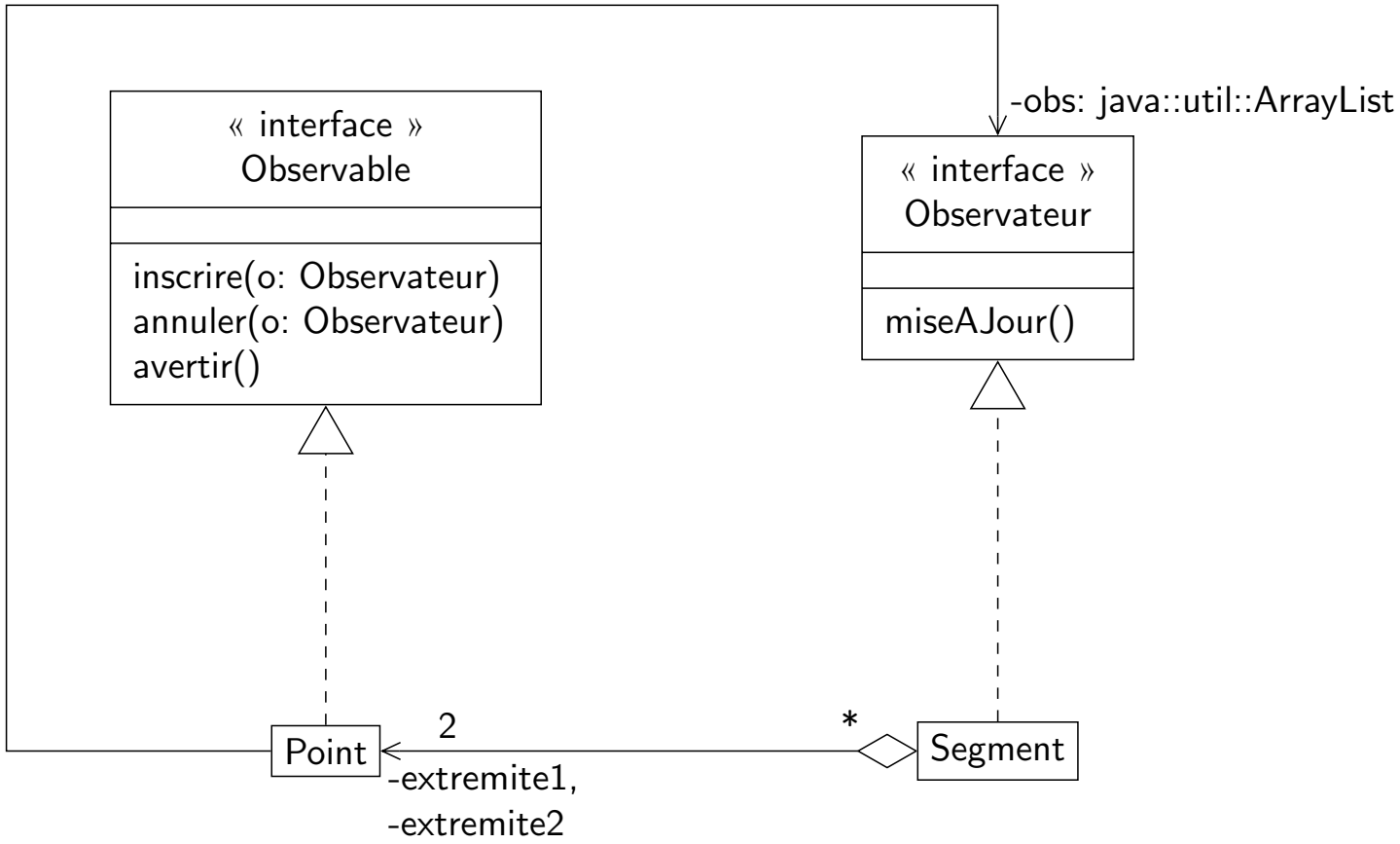
Adapter le patron à l'implantation fournie

Exercice

Adapter le patron pour l'implantation fournie :

- `Point` et `Segment` héritent de `Figure`
- réutilisabilité de la solution

Adapter le patron : héritage de `Figure`



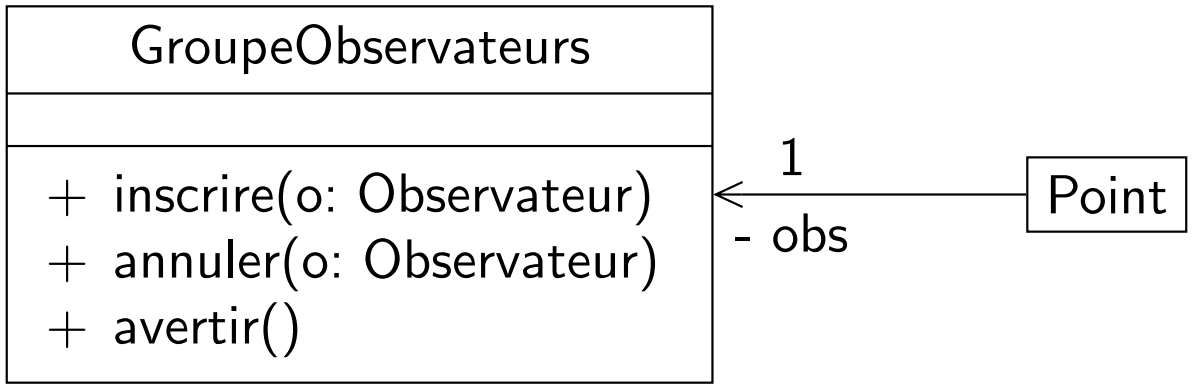
Adapter le patron : réutilisabilité

Remarque

Le code des méthodes `inscrire`, `annuler`, `avertir` est générique, on va le réécrire à chaque fois...

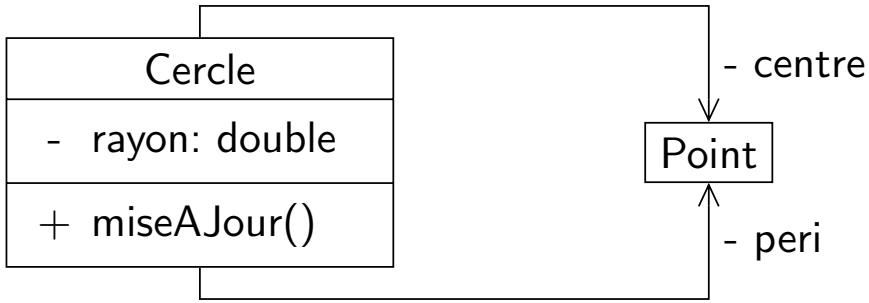
Mais on ne peut pas utiliser une classe abstraite ici...

➥ on délègue le service

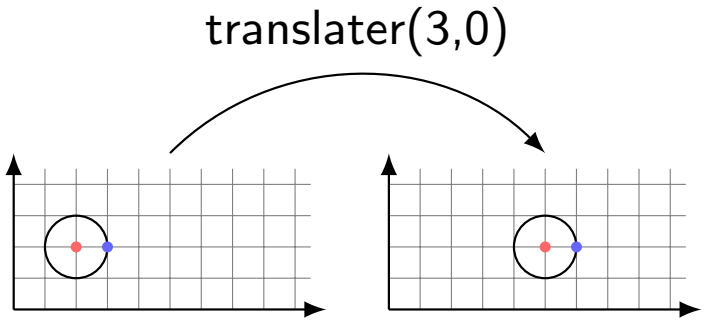


Adapter le patron : réutilisabilité

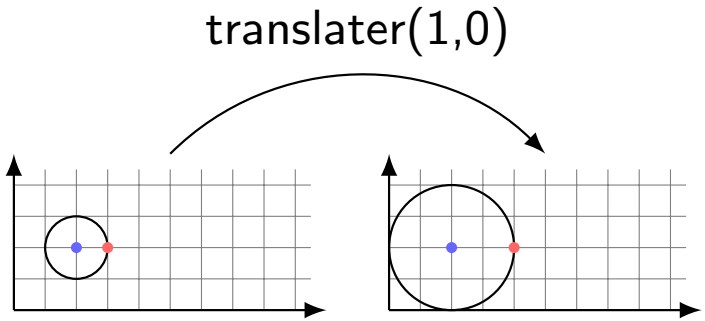
Supposons que l'on ait une classe `Cercle` :



Translation du centre :

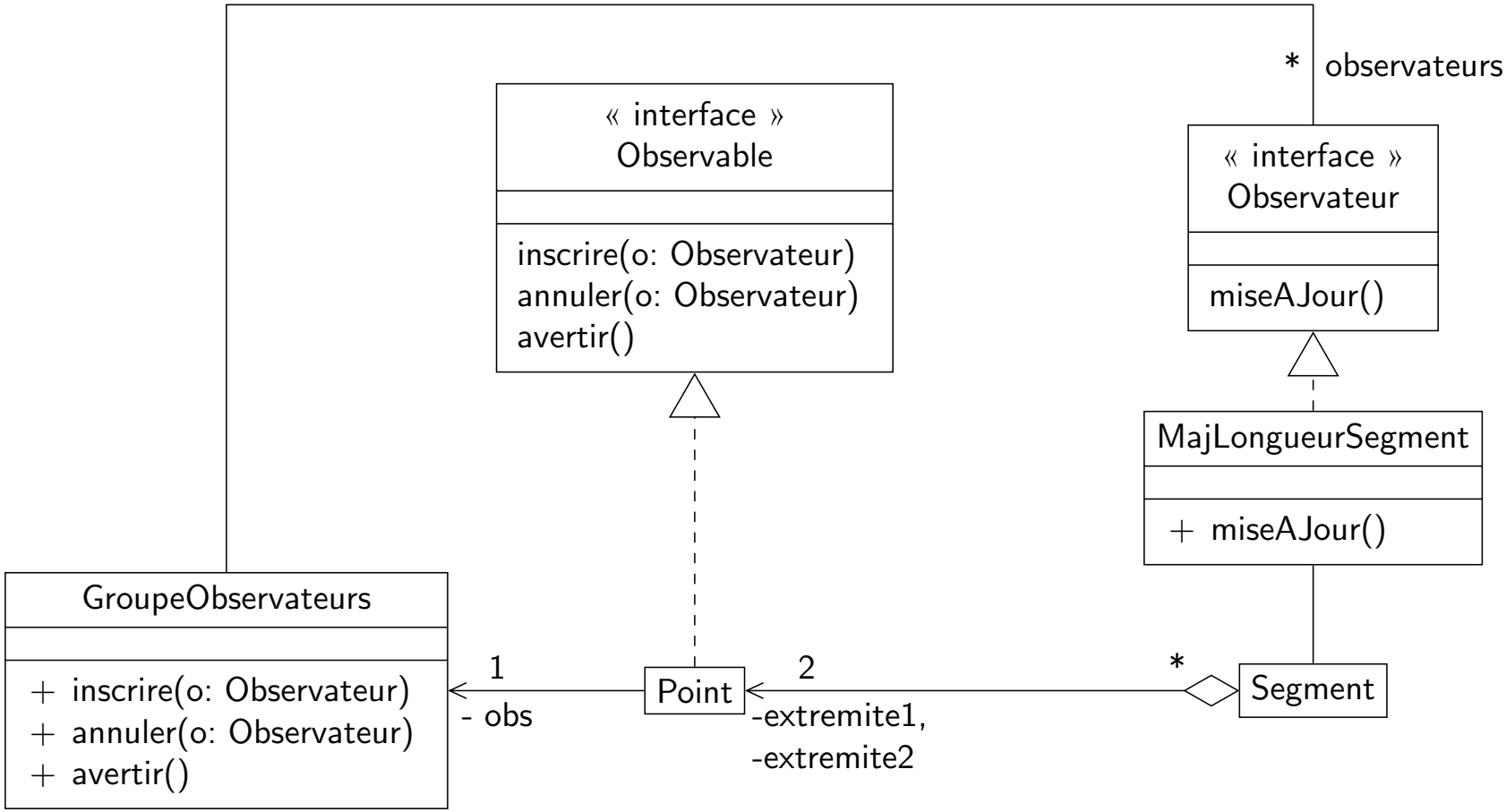


Translation de la périphérie :



Comment coder `miseAJour` ?

Adapter le patron : réutilisabilité



Implantation en TP

Exercice

Implanter la solution « simple » dans un premier temps, puis essayer de coder `GroupeObservateurs` et `MajLongueurSegment`.