



IN201 Conception et Programmation Orientées Objet TP Observateur : diagrammes établis en classe

Christophe Garion
DMIA – ISAE



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license (CC BY-NC-SA 3.0)

You are free to Share (copy, distribute and transmit) and to Remix (adapt) this work under the following conditions :



Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial – You may not use this work for commercial purposes.

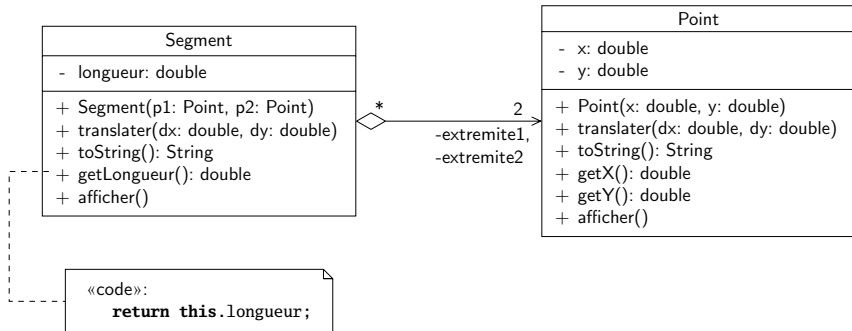


Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

See <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Une classe Segment

On décide de réaliser une classe Segment comme suit :





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
```



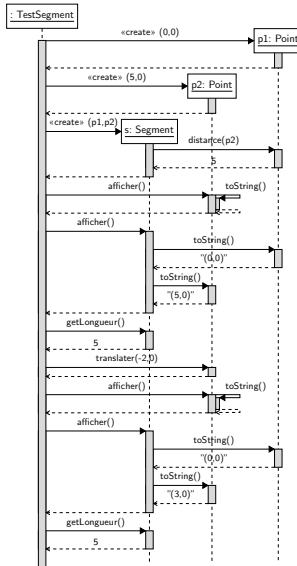

Exercice

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
```



Hypothèse

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

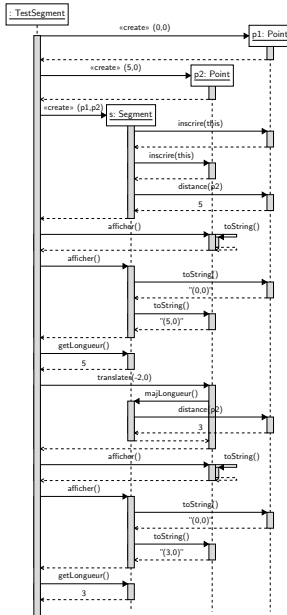
Exercice

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

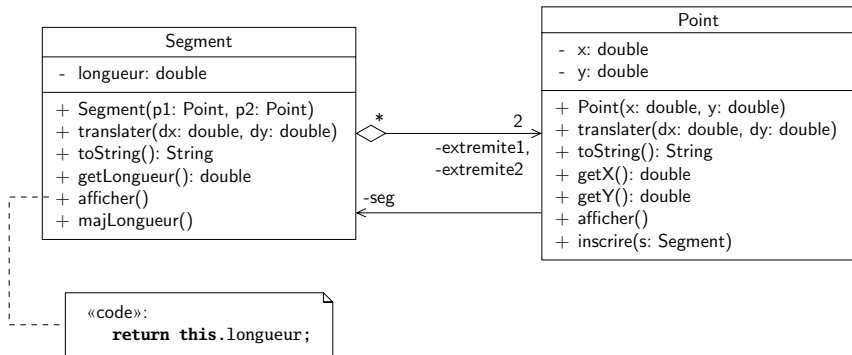
Exercice

Modifier le diagramme de classes initial en conséquence.

Correction des classes : diagramme de séquence



Correction des classes : diagramme de classes

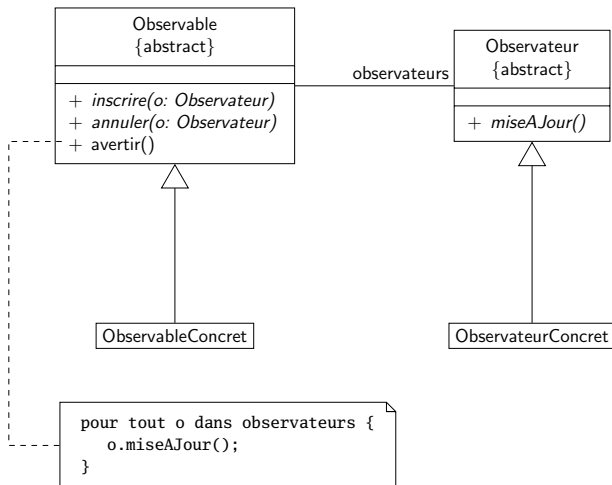




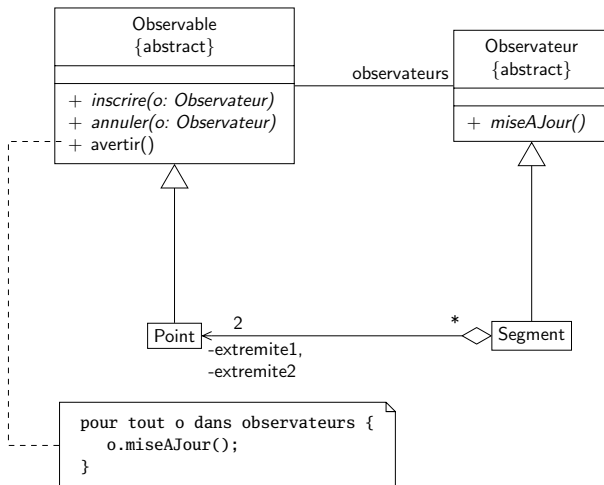
Exercice

- 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



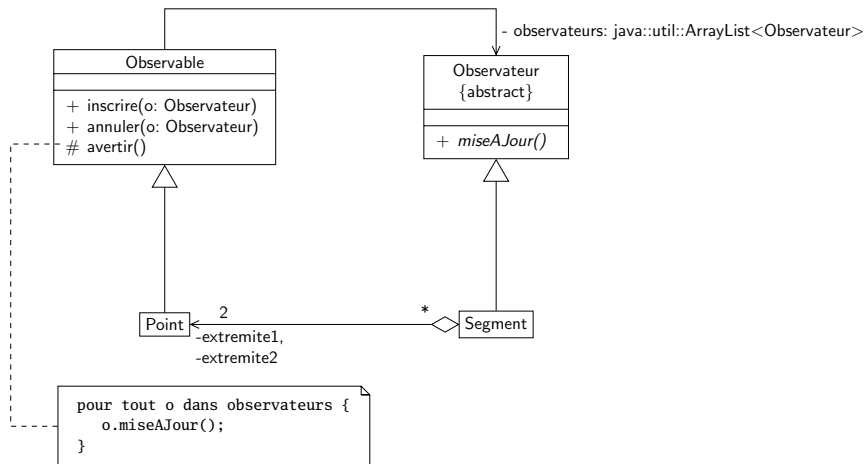


Exercice

Adapter le patron :

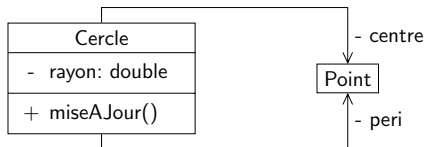
- peut-on utiliser des interfaces ?
- Observable est-elle abstraite ?
- visibilité des méthodes de Observable ?
- méthode `miseAJour` « efficace » ?

Adapter le patron : classe Observable



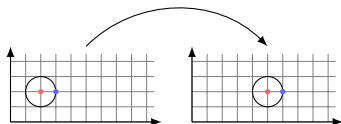
Méthode miseAJour

Supposons que l'on ait une classe Cercle :



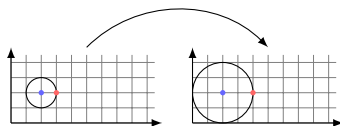
Translation du centre :

`translater(3,0)`



Translation du point périphérique :

`translater(1,0)`



Comment coder `miseAJour` ?

Méthode miseAJour

