

Réutilisation et Composants

TD-TP Programmation par Aspects (AOP) - Introduction à AspectJ

Pour utiliser **AspectJ** vous devez en premier lieu installer le plugin “AJDT”, Par exemple AJDT 2.2.3, dans votre Eclipse.

Voir <https://www.eclipse.org/ajdt/gettingstarted.php>, ou directement dans le menu Eclipse “Help-Install new software”, puis *AspectJ*.

1. Pour la prise en main, lisez en premier lieu le tutoriel <http://www.eclipse.org/ajdt/demos/>. Regardez la première vidéo (Hello World). Il y a de petites modifications entre ce tutoriel et la dernière implantation que vous avez.

Pour faire du AspectJ vous allez devoir créer un projet *AspectJ*, que vous trouverez en “File>new>Other>AspectJProject”.

2. Au sein d’un tel projet, programmez l’exemple donné en cours (<https://www.lirmm.fr/~dony/notesCours/aspects.s.pdf> en créant en premier lieu des classes normales *Figure*, *Line*, etc. Il faut qu’une des classes soit la classe principale et possède un *main*.

Créez ensuite des *aspects* en faisant clic droit sur le projet puis : “new>other>aspectJ>aspect”. Essayez les exemples donnés en cours.

3. Utilisez à nouveau la vidéo précédente pour comprendre l’aide à la visualisation des programmes fournie par Eclipse.
4. Faites le lien avec le pattern *Visitor*. Considérez vos classes *ElementDeStockage* du TP correspondant et dotez les d’un mécanisme de visite sans modifier leur code, grâce à un aspect. Si vous ne voyez pas comment faire, vous trouverez la solution dans la documentation AspectJ.

5. Lancez vous plus avant avec <http://www.eclipse.org/aspectj/doc/released/progguide/index.html>

6. Faire un pont entre java-beans et *aspects* :

<http://www.eclipse.org/aspectj/doc/released/progguide/examples-production.html>. On y voit comment programmer les propriétés liées des composants *JavaBeans* de façon plus modulaire, avec des *aspects*.

Vous trouverez le code correspondant dans File>new>other>aspectJ>AspectJ Examples

7. Faites le lien entre annotations (Java) et *aspect*.

<http://www.eclipse.org/aspectj/doc/released/adk15notebook/annotations.html>

8. Faites le lien entre méta-programmation et Aspects, en améliorant l’aspect *MemoObject* donné dans le cours.

9. Vous pouvez faire de l’AOP avec le framework *SPRING*, essayez. Voir <https://docs.spring.io/spring-framework/docs/2.5.x/reference/aop.html>.

Autre piste : <https://dzone.com/articles/implementing-a-method-trace-infrastructure-with-sp>

10. challenge : Il est très simple de faire un programme de Trace avec AspectJ (afficher un message avant et après l’exécution d’une méthode), pourrait-on faire la même avec une annotation Java ?

A vous de jouer pour faire vos tests et acquérir de l’expertise dans les différentes mises en oeuvre de AOP.