

## **Sommaire :**

### **1. Introduction à la programmation orientée aspect (POA)**

- Motivation de l'AOP : séparation des préoccupations
- Limites de la programmation orientée objet (POO)
- Concepts fondamentaux de l'AOP : aspects, joint points, pointcuts, advice

### **2. Concepts de base d'AspectJ**

- Syntaxe de base
- Types de conseils : before, after, around
- Définition des pointcuts
- Aspects et classes Java

### **3. Programmation avec AspectJ**

- Écriture d'aspects simples
- Manipulation des joint points
- Cas d'utilisation : logging, sécurité, transactions
- Bonnes pratiques

### **4. Intégration avec Java-J2EE**

- AOP et architectures d'entreprise
- Intégration dans des applications J2EE (EJB, servlets)
- Compatibilité avec les frameworks : Struts, Spring, Hibernate

### **5. Conception orientée aspect**

- Approche méthodologique
- Patterns orientés aspect
- Refactoring vers l'AOP

### **6. Outils et environnement**

- Compilateurs et tisseurs (weavers)
- IDE compatibles : Eclipse avec AJDT
- Outils de test et de débogage

### **7. Études de cas**

- Implémentation d'aspects dans une application de gestion
- Sécurité transversale dans une architecture distribuée
- Gestion des exceptions et des performances

### **8. Perspectives et évolutions**

- Liens entre AOP, MDA, SOA
- Vers une meilleure modularité logicielle

- Limitations et critiques de l'AOP

## **Annexes**

- Références sur AspectJ
- Installation et configuration de l'environnement
- Lexique des termes