Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Architecture Hexagonale

Construire une application respectant l'architecture hexagonale

L'architecture hexagonale, appelée aussi Ports & Adapters, n'est pas un sujet nouveau : elle est définie pour la première fois par Alistair Cockburn en 2005. Robert C. Martin a repris plus récemment, en 2012, les concepts en définissant la clean architecture. Malgré tout, la plupart des projets continuent aujourd'hui d'être construits sur une architecture classique en couches et subissent les défauts identifiés près de 20 ans plus tôt.
Lors de cette formation, vous découvrirez les raisons et arguments qui ont permis l'apparition de ce modèle architectural et de l'engouement récent qu'il suscite.
Un cas pratique simple servira de fil rouge tout au long de la formation pour mettre en œuvre les concepts.

Détails

- Code : AE-HEXA
- Durée : 2 jours (14 heures)

Public
- Architectes
- Développeurs expérimentés
- Leaders Techniques

Pré-requis
- Expérience en programmation Objet si possible Java ou C# (les exemples sont principalement en Java)

Objectifs
- Comprendre les concepts clefs de l'architecture hexagonale et quelques éléments du DDD
- Construire de manière itérative une architecture

Programme

Introduction : C'est quoi l'architecture logicielle ?

L'architecture, au delà des client-services, des monolithes, des POJO/DAO/Services/CRUD… etc
- Architecture en couche DDD (Layered Architecture)
- Architecture hexagonale
- Architecture en oignon
- Clean Architecture

Pourquoi et quand utiliser l'architecture hexagonale ?
- Domain complexe
- Compétence stratégiques
- Adapté pour la POO, utile en FP, étrange en Procédurale
- Microservices & Bounded Context

Architecture hexagonale (AKA Port & Adapters) en détail
- Domaine : agrégats, entités , VO, Domain Events, Services de domaine, etc
- Application
- Infrastructure
- Adapteurs, Port, Anti-corruption layer
- Les dépendances en hexagonale

La pyramide des tests d'une architecture hexagonale
- Test couche domaine : Tests Unitaires
- Test couche application : BDD
- Test couche infrastructure : Tests d'intégrations

La construction
- Construction Itérative
- Retarder les décisions d'architecture grâce au design
- Double boucle BDD TDD
- Infra en dernier : ports puis adapteurs
- Les couches d'anti-corruptions

Un mot sur :
- Event Sourcing, CQRS
- La documentation vivante et les architectures

Conclusion
- Coût de la dette d'architecture
- Scaler un SI

Modalités
- Type d'action :Acquisition des connaissances
- Moyens de la formation :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- Modalités pédagogiques :Exposés – Cas pratiques – Synthèse
- Validation :Exercices de validation – Attestation de stages

Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu