

Survol de l'approche orientée-objets

- Définitions
- Origines
- Spécificités
- Différences entre approche fonctionnelle et approche orientée-objets

Définitions de l'approche orientée-objets

- Une méthode de modélisation basée sur une représentation abstraite des entités du monde réel
- Une méthode qui regroupe les données et les traitements sur ces données au sein d'une entité unique : l'objet
- Une méthode qui exprime les fonctionnalités sous la forme de collaborations entre les objets

Origines de l'approche orientée-objets

- Des racines dans le monde de la simulation (Simula déjà en 1967)
- Des racines dans la quête d'une nouvelle façon d'interagir avec les ordinateurs : commencée à Stanford (années 60), continuée au PARC (années 70) et concrétisée par le Macintosh (1984)
- Des racines dans l'architecture des ordinateurs, les systèmes de gestion de base de données, les travaux en intelligence artificielle

Evolution de l'approche orientée-objets

- Evolution identique à l'approche fonctionnelle
- Programmation fonctionnelle, conception fonctionnelle, analyse fonctionnelle
- Programmation objets, conception objets, analyse objets
- Une analyse fonctionnelle peut être suivie d'une conception objets, mais ceci implique de traduire une représentation dans une autre
- Le tout objets ne demande pas de traduction entre phases

Différences entre approche fonctionnelle et approche orientée-objets

- La décomposition fonctionnelle induit une architecture du logiciel calquée sur les fonctions
- La décomposition objets repose sur les entités du domaine, pas sur les fonctions
- L'approche orientée-objets rend l'architecture des logiciels indépendante des fonctionnalités
- L'approche orientée-objets rend les logiciels plus adaptables et plus extensibles

Spécificités de l'approche orientée-objets

- Une modélisation basée sur les entités du monde réel
- Une méthode qui s'adresse autant à l'individu que à l'équipe
- Un cadre pour le développement de composants réutilisables
- Un support pour un cycle de vie plus souple

Une modélisation basée sur les entités du monde réel

- Les entités du monde réels sont stables dans le temps
- Les fonctions évoluent dans le temps
- Les objets du domaine peuvent être considérés comme des invariants du domaine

Une méthode qui s'adresse autant à l'individu que à l'équipe

- L'approche orientée-objets est adaptée aux besoins du programmeur individuel et aux besoins de l'équipe
- L'approche orientée-objets facilite la transition entre le détaillé et le général et réciproquement
- L'approche orientée-objets permet d'intégrer la structure de l'équipe de développement avec la structure du logiciel
- L'approche orientée-objets permet de délimiter des espaces de responsabilité

Un cadre pour le développement de composants réutilisables

- Le terme *composant réutilisable* est mal choisi
- La réutilisation peut être vue sous deux aspects
 - hier pour aujourd'hui
 - aujourd'hui pour demain
- Les logiciels ne sont pas réutilisables par simple effet de bord de l'approche orientée-objets
- La réutilisation a un coût, elle doit être vue comme un investissement

Un support pour un cycle de vie plus souple

- Prendre en compte l'évolution des besoins
- Réduire le risque plus tôt dans le cycle de développement
- Augmenter le nombre de points de mesures

L'approche orientée-objet et l'industrie

- Les années 90 correspondent au décollage de l'OO dans l'industrie
- Conforme au décalage habituel de 20 ans entre la recherche et l'industrie
- Généralisation de l'intérêt (d'abord logiciel technique et maintenant également gestion)

Exemples d'utilisation de l'OO

- Interfaces utilisateurs
- Systèmes client - serveur
- Logiciels embarqués
- Télécommunications
- Bases de données
- Compilation
-