



Analyste Programmeur en Automatisation, Robotique et Informatique Industrielle TS ARII

Module MF 1.4

Analyser un projet informatique

La Programmation Orientée Objet

Patrick MONASSIER – année 2019-2020

Module 1.4 – Analyser un projet informatique

Compétences

Modéliser selon les modalités d'analyse orientées objet.

Objectifs

Décrire un traitement séquentiel

Définir un système de données.

Contenu

- Algorithmes
 - Type de données
 - Structure de contrôle
 - Procédures et fonctions
- Méthode d'analyse UML
- Analyse MERISE.
 - Outil d'interface graphique
 - Langage SQL

Les principaux chapitres qui seront développés pendant le cours :

Introduction générale

Types de programmation

Les algorithmes

La Programmation Orientée Objet – POO

Les Bases de Données - SQL

La méthode d'analyse UML

L'analyse MERISE

Le langage SQL

de nombreux travaux pratiques et utilisations de logiciels

Sommaire du diaporama

La programmation orientée objets **Introduction et Principes**

Principes de la programmation objet
Caractérisation d'un objet
Communication entre objets
Synchronisation
Création d'objets
Suppression d'objets

Application et initialisation

Encapsulation
Responsabilité
Cohésion
Couplage

Exemples à suivre

Exemples et travaux dirigés

La programmation orientée objet

- ✓ *Principes : qu'est ce qu'on objet, à quoi ça sert*
- ✓ *Application objet : des objets qui communiquent entre eux*
- ✓ *Conception : encapsulation, cohérence*
- ✓ *Exemple - Space Invader*

Un objet est caractérisé par :

- ✓ Son identité exemple : #1
- ✓ Ses données : chaque objet a sa petite base de données propres
- ✓ Les traitements qu'il sait réaliser et qu'il propose aux autres objets

Exemple :

Une bibliothèque : identité #1

Les données : pile de livres

Traitements : ajouterlivre, emprunterlivre

Communication

Les objets communiquent entre eux par des échanges de messages
Le message le plus échangé est la demande de réalisation de traitement

Demande de réalisation de traitement

Pour envoyer une demande de réalisation de traitement, un objet doit :

- ✓ Connaitre l'Id de l'objet qui va réaliser le traitement
- ✓ Lui envoyer un message avec le nom du traitement et les paramètres nécessaires
- ✓ Recevoir la réponse

Exemple : #2 emprunte livre (Java) à #1

Synchrone par défaut

- ✓ Les communications entre objets sont principalement synchrones
- ✓ L'objet appelant attend la réponse de l'appelé avant de pouvoir faire autre chose

Exemple :

#2 -> #1 (standard)

#2 -> #3 -> #4 (chaine d'échange de messages – à éviter)

Création d'objet et référence

- ✓ Chaque objet peut créer d'autres objets
- ✓ Le créateur possède une référence vers les objets créés
- ✓ Il est possible de les transmettre via échange de message

Exemple :

#1 créateur de -> #2 #3 #4

#5 -> #3 ? Demande getRef à #1 – car #5 n'a pas en mémoire l'Id #3

Traitement getRef

#5 envoie getRef à #1 qui envoie id #3

#5 garde en mémoire Id #3 . Il peut alors communiquer avec #3

Suppression d'objet

Soit on peut supprimer les objets et alors il faut faire attention aux références qui pointent vers des objets supprimés (C++)

Soit on ne peut pas supprimer les objets mais ceux-ci sont mis aux oubliettes si aucun autre objet ne les référence (Java)

APPLICATION = ENSEMBLE D'OBJETS

- ✓ Une application objet est un ensemble d'objets qui communiquent pour rendre un service global à son utilisateur
- ✓ L'utilisateur est extérieur : il utilise les objets contenus dans l'application
- ✓ Tout langage de programmation objet propose un moyen pour capter les interactions de l'utilisateur (souris, clavier ...) - Bien souvent, via des échange de messages

INITIALISATION

- ✓ On passe par un objet particulier qui peut capter des messages venant d'évènements utilisateurs extérieurs
- ✓ Au démarrage, le « main » construit tous les objets nécessaires au démarrage de l'application

ENCAPSULATION

- ✓ L'objet protège ses données afin de préserver son intégrité
- ✓ Seul l'objet peut modifier ses propres données
- ✓ Seul l'objet sait comment il gère ses données

Pas possible pour un autre objet de modifier les données de l'objet considéré !

RESPONSABILITE

- ✓ L'objet est responsable des traitements qu'il sait faire
- ✓ Il a donc toutes les données nécessaires pour le faire
- ✓ *Les données peuvent être cherchées à l'extérieur (autre objet)*

COHESION

- ✓ Les traitements et les données d'un objet forment un tout cohérent
- ✓ Préférer plusieurs petits objets à un gros

COUPLAGE

- ✓ Même si la communication inter-objet est incontournable, il faut essayer de minimiser les communications (surtout les chaînes de communication)

Analyste Programmeur en Automatisation, Robotique et
Informatique Industrielle
TS ARII

Module MF 1.4

Analyser un projet informatique

La Programmation Orientée Objet

Fin de Présentation