

R2.01 - Développement Orienté Objets

Introduction

Arnaud Lanoix Brauer
Arnaud.Lanoix@univ-nantes.fr



Nantes Université

Département informatique

Développement Orienté Objets – Ressource R2.01

Descriptif détaillé

Quel objectif pour cette ressource ?

L'objectif de cette ressource est d'initier à la programmation objets.

Quels savoirs de référence à étudier ?

- Concepts fondamentaux de la programmation orientée objets
- Application orientée objets des algorithmes sur des structures de données (ex : collections...)
- Illustration de l'exécution d'un algorithme dans un schéma mémoire (pile et tas)
- Lecture d'une conception orientée objet détaillée (ex : diagramme de classes)
- Bases de la modélisation objet pour l'analyse et la conception détaillée (ex : diagramme de classes, diagramme des cas d'utilisation, diagramme de séquences...)

Comment cette ressource fait-elle monter en compétence ?

Cette ressource amène une progression dans les apprentissages critiques des compétences 1 et 2. En effet, la réalisation d'un développement d'application et l'optimisation des applications informatiques passent par la compréhension des paradigmes objets.

Programmation objets

Analyse

Conception objets

Compétence 1

Développer des applications informatiques simples

AC 1 Implémenter des conceptions simples

AC 2 Élaborer des conceptions simples

AC 3 Faire des essais et évaluer leurs résultats en regard des spécifications

Compétence 2

Appréhender et construire des algorithmes

AC 1 Analyser un problème avec méthode (découpage en éléments algorithmiques simples, structure de données...)

AC 3 Expérimenter la notion de compilation et les représentations bas niveau des données

(extraits de la fiche ressource - PN V20)

Objectifs du module

Objectif

S'initier à la **conception et programmation Orientée Objets**, c.à.d.

- appréhender les concepts de la programmation objet : **encapsulation**, **héritage**, **redéfinition**, etc.
- lire (et élaborer) une **conception** orientée objet **UML** (diagrammes de classes, diagrammes de séquences)
- découvrir un **langage de programmation** orienté objet : **Kotlin**
- comprendre les concepts mémoire sous-jacent = **références**
- **passer d'une conception orientée objet à une programmation orientée objet**
- coder (et concevoir) des **algorithmes** dans un contexte Objet



+



Organisation du module

- **Equipe pédagogique :**

- ▶ Responsable du cours : **Arnaud Lanoix Brauer**
- ▶ Chargés de TD +TP : Arnaud Lanoix (TD2 + TP 2.1), Johann Leray (TP 2.2), Christine Jacquin (TD3), Jean-François Berdjugin (TD4), Jean-François Remm (TD1)

- **Volume horaire :**

- ▶ 7 créneaux de CMs
- ▶ 15 créneaux de TDs ("papier" ou "machine")
- ▶ 16 créneaux de TPs "machine"
- ▶ (+ des créneaux de SAE possible)

répartis entre la semaine S3 et la semaine S14

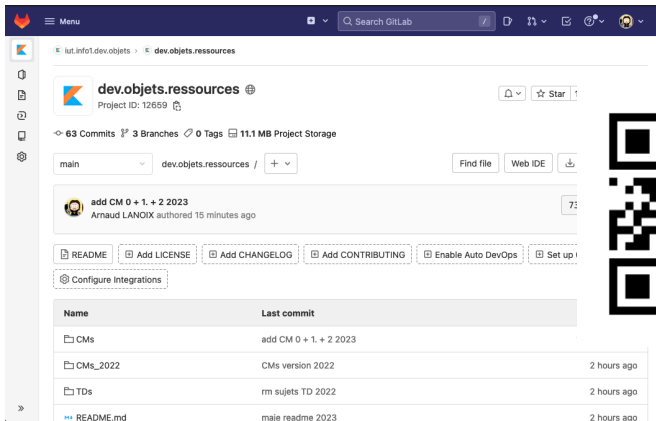
- **Evaluations :**

- ▶ des évaluations sont **planifiées** lors de plusieurs créneaux de TDs
- ▶ des évaluations "**surprises**" sont également au programme

- **SAE associée :**

- ▶ en collaboration avec les ressources *R2.02 IHM, R2.03 Qualité*, etc.
- ▶ encore à planifier sur les semaines S22, S23, S24 et S25

Ressources de R2.01



dev.objets.ressources

Project ID: 12659

63 Commits 3 Branches 0 Tags 11.1 MB Project Storage

main dev.objets.ressources / +

Find file Web IDE

add CM 0 + 1. + 2 2023
Arnaud LANOIX authored 15 minutes ago

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Set up

Configure Integrations

Name	Last commit
CMs	add CM 0 + 1. + 2 2023
CMs_2022	CMs version 2022 2 hours ago
TDs	rm sujets TD 2022 2 hours ago
README.md	maje readme 2023 2 hours ago



<https://gitlab.univ-nantes.fr/iut.info1.dev.objets/dev.objets.ressources>

Quelques remerciements

- à Christine Jacquin, Dalila Tamzalit et Jean-François Remm pour leurs supports de cours/TDs/TPs
- à Geoffrey Challen pour son retour d'expérience sur l'enseignement de Kotlin :
<https://www.geoffreychallen.com/essays/2021-11-21-cs1-in-kotlin>
- aux auteurs du style Kotlin pour LaTeX/Listings :
<https://github.com/cansik/kotlin-latex-listing>
- à Wikipedia, pour plusieurs images et quelques explications,
- aux auteurs de Kotlin (et en particulier à l'entreprise JetBrains) :
<https://kotlinlang.org/>