

# Développement Orienté Objets

## Introduction

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

IUT de Nantes  
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.
- découvrir un **langage de programmation** orienté objet : **Kotlin**
- comprendre les concepts mémoire sous-jacent
- lire (et élaborer) une **conception** orientée objet : **UML (diagrammes de classes, diagrammes de séquences)**
- 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 : Arnaud Lanoix (1 groupe), Christine Jacquin (1), Jean-François Berdjugin (1), Jean-François Remm (1)
- ▶ Chargés de TP : Arnaud Lanoix (1), Christine Jacquin (1), Jean-François Berdjugin (2), Jean-François Remm (2), Ali Benjilany (2)

- **Volume horaire :**

- ▶ 7 créneaux de CMs
- ▶ 18 créneaux de TDs ("papier" ou "machine")
- ▶ 24 créneaux de TP "machine"

répartis entre la semaine S4 et la semaine S19

- **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.
- ▶ à planifier sur les semaines S22, S23, S24 et S25 ?

# Planning (prévisionnel)

Sem	lundi	CM	TD1 papier	TP	TD2 machine	evaluation TD	cr /sem
4	24/01	1		2			3
5	31/01	1	1	2	1		5
6	07/02	vacances d'hiver					
7	14/02						
8	21/02	1	1	2		1	5
9	28/02	1	1	2	1		5
10	07/03	1	1	2	1		5
11	14/03	1		2		1	4
12	21/03	1		2	1		4
13	28/03		1	2	1		4
14	04/04		1	2		1	4
15	11/04	vacances de printemps					
16	18/04						
17	25/04			2			2
18	02/05			2		2	4
19	09/05			2	2		4
20	16/05						
21	23/05						
22	30/05	SAE ?					
23	06/06						
24	13/06						
25	20/06						

# 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/>