

## II.1102 – Algorithmic et programming

Module title : Algorithmic et programming  
Module code : II.1102  
Person in charge : Hedi YAZID  
ECTS : 4  
Workload : 150h, including 52h face-to-face  
Teamwork : Yes (group project)  
Keywords : Algorithmic, Java, Graphical Interface, Games

### Presentation

This module introduces basic algorithmic concepts and the fundamental principles of object-oriented programming using the Java language. It familiarizes students with the programming concepts of a compiled language, as well as with the use of external libraries, particularly for graphical interfaces. It also requires students to develop their ability to model programs using UML diagrams, and to analyze the algorithmic complexity of a program to produce efficient code.

### Academic Objectives

- Design a tested and documented application in Java, from its modeling using a UML diagram, to its delivery (source code or .jar file).
- Be able to break down and model algorithmic problems, then analyze solutions (via complexity) and produce an optimized solution.

#### *Prerequisites*

- None

#### *Content/program*

- Recommended IDE: IntelliJ IDEA. Alternatives: VS Code, Eclipse, Vim, ...
- Build manager: maven or gradle
- Graphical interface: JavaFX. Swing authorized
- Git for file versioning.

### Academic methods

#### *Learning methods*

Sessions are divided into two types: "Lectures + Tutorials" and "Project".

- In the first part, essential concepts are introduced with slides and examples. In the second part, students put these notions into practice by solving exercises.
- "Project" sessions enable students to work on their projects (individual and group). These sessions can be used to present concepts that are important to the realization of an application, or to help individual students and groups overcome difficulties they may be encountering.

#### *Assessment*

- The course and the ability to produce Java code are assessed at mid-semester with a written exam.
- Programming skills are assessed via graded practical work.

- During the first two-thirds of the semester, each student is required to complete an individual project (e.g. a role-playing game with console and GUI).
- At the same time, a group project (board game) is to be carried out by the students and presented during the last session of the year.
- A final written exam containing questions on coursework, code, algorithms, and UML modeling.

#### *Working Language*

- English

#### **Bibliography, Webography, Other resources**

- For bases of algorithmic, « Algorithms » from Sedgewick and Wayne
- For design patterns (Object oriented programming), "Design Patterns: Elements of Reusable Object-Oriented Software" from Gamma et al.
- Official documentations of Java or JavaFX.
- "Learn JavaFX 17" from Sharan and Späth (graphical interfaces programming with JavaFX).
- "Effective Java" from Joshua Bloch (provides a list of best practices for producing high-quality Java code).