Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Subject introduction

Software Architecture

Juan Carlos Cruellas

What this subject is about?

- About accepting that software engineering is not only about coding (in fact coding is only the last but one step, and the less payed job).
- About learning some basic ideas on what software architecture is.
- About learning some basic ideas on how to build software systems with a good architecture. This includes:
  - Learning basic techniques of object-oriented analysis and design (OOAD).
  - Learning some popular software design patterns and identifying the worthiness of using them during the design of a program.

What this subject is about?

- About applying OOAD and design patterns in the analysis of moderate/big problems and the design of moderate/big programs with an acceptable quality level.
- About **becoming able to code** a moderate/big program with an acceptable quality level implementing OO design.
- About learning basic techniques for automatic program testing.

What this subject IS NOT about?

- Not a subject for learning the principles of OO programming paradigm.
- Not a subject for learning an OO programming language.
- Not a subject on algorithms.
- Not a subject for learning software coding (although you will indeed have to code).
- Not a subject for learning popular tools for automating project management like Maven, Gradle, etc. (although you will use some)
- Not a subject for learning tools for software versioning and revision control, like Git, Subversion, Mercurial, etc. (although you can use some)

Topics covered

- Introduction. Keywords.
  - quality software, bad software symptoms, requirements, requirements change, classes coupling, class coherence, class responsibilities.
- Object Oriented analysis.
  - Requirements list
  - Use cases
  - Conceptual model
- Object Oriented design.
  - Design Class Diagram.
  - Designing static aspects: assigning responsibilities to classes through GRASP patterns.
  - Designing dynamic aspects: from use cases to sequence diagrams.

Topics covered

- Advanced Object Oriented design: software design patterns:
  - Factory method and Abstract Factory
  - Visitor
  - Observer
  - Decorator
  - Strategy
- Automated testing.
  - Types of tests
  - JUnit for unit tests
  - Mockito for integration tests

Prior skills

- English language.
  - No need to be native, but at least having consolidated conversation and technical level.
- **Solid conceptual bases of object oriented programming:**
  - **Object, class, attributes, methods, encapsulation, exceptions.**
  - **Inheritance and polymorphism**.
  - **Without them, you will not be able to properly follow the subject.**
- Unified Modelling Language (UML).
  - Class diagrams.

Prior skills

- Experience with Java language will be valuable (selected language for lecturing the subject; examples will be given in Java).
  - Syntax: classes, abstract classes, interfaces, static methods and attributes, flow control, inheritance definition, exceptions management.
  - Java containers: sets, lists, maps.
  - Input / Output classes.
- However knowledge/experience with other OO language different than Java (like Python) will also be useful.

Basic bibliography

Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, third edition. Addison-Wesley Professional, 2004.

Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. Head First Design Patterns. O'Reilly Media Inc., 2004.

Robert C. Martin. Clean Architecture. Prentice Hall. 2018.

Complementary bibliography (highly advisable for professional life)

E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

Bibliography for Java language

Bruce Eckel. Thinking in Java. Prentice Hall.

Free electronic versions available until its version 3 (inclusive).

Joshua Bloch. Effective Java. 2nd or 3rd edition. O'Reilly Media, 2017.