

27024 - Computer Science II

Syllabus Information

Academic year: 2023/24

Subject: 27024 - Computer Science II

Faculty / School: 100 - Facultad de Ciencias

Degree: 453 - Degree in Mathematics

ECTS: 6.0

Year: 4

Semester: First semester

Subject type: Optional

Module:

1. General information

In this subject it is carried out an introduction to the object-oriented programming paradigm (OOP). The basics of the event-driven programming and the use of generics types are also presented.

The approaches and objectives of this module are aligned with the Sustainable Development Goals (SDGs) of the United Nations 2030 Agenda; the learning activities could contribute to some extent to the achievement of the goals 4 (quality education), 5 (gender equality), 8 (decent work and economic growth), and 10 (reducing inequality).

2. Learning results

- Solve problems using the object-oriented programming paradigm.
- Understand and use inheritance and polymorphism in application design.
- Understand and use class diagrams in application development.
- Know the basics of event-driven programming and use it in the development of graphical user interfaces.
- Know the fundamentals of generic programming and its application in the design and implementation of complex data types.

3. Syllabus

1. The object-oriented paradigm (OOP). The Java Virtual Machine: compiling and executing programs.
2. Review of elements of structured programming in the Java language: predefined types; variables and constants; operators and expressions; usual mathematical functions (the Math class). Structured statements: sequential, conditional and iterative. Defining and invoking class methods. Method signatures: overloading methods.
3. Introducing OOP. Objects, classes and references (the null reference). The life cycle of Java objects: the new operator and constructor methods, accessing members and message passing, the Java garbage collector. Array objects.
4. Defining classes. Instance and class members. Writing constructor methods. Access levels: the public interface of a class. Namespaces: Java packages.
5. An introduction to UML class diagrams. Class associations and associative classes. Roles and navigation.
6. Inheritance: concept and types, method overriding. Class hierarchy: the Java Object class. Polymorphism: virtual methods. Modeling (generalization and specialization): abstract classes and methods.
7. The Java type system: Java interfaces. Generic programming in Java. The Java collections framework.
8. Exceptions: rising, handling and specification.
9. Persistence: binary and text streams. Object persistence: the Serializable interface. Access to remote resources: File and URL classes.
10. Event-driven programming: the Java event model. GUIs programming: containers, menus and basic controls.

4. Academic activities

Master classes: 30 hours.

Computer classes: 30 hours.

Project: 27 hours.

Study: 60 hours.

Assessment tests: 3 hours.

5. Assessment system

Any student will be guaranteed the right to pass the subject by a final comprehensive exam. Alternatively, the student will be given the possibility to pass the subject by continuous evaluation as follows:

- 75% of the final mark by solving (design and programming) three problems, which will be posed and done in three of the problem solving sessions.
- 25% by the design and programming of a complete application.