Course guides
230086 - POO - Object Oriented Programming

Last modified: 29/04/2020

Unit in charge: Barcelona School of Telecommunications Engineering
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING (Syllabus 2015). (Compulsory subject).

Academic year: 2020 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: Cruellas Ibarz, Juan Carlos

Others: Abadal Cavalle, Sergi
Gil Gómez, María Luisa
Guerrero Zapata, Manel
Macías Lloret, Mario
Perello Muntan, Jordi
Rodriguez Luna, Eva

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Transversal:
06 URI N1. EFFECTIVE USE OF INFORMATI0N RESOURCES - Level 1. Identifying information needs. Using collections, premises and services that are available for designing and executing simple searches that are suited to the topic.

TEACHING METHODOLOGY

LEARNING OBJECTIVES OF THE SUBJECT

To understand what the object-oriented paradigm is. To understand its most remarkable concepts: class, object, encapsulation, agregation, inheritance and polimorphism.

To know how to apply sort and search algorithms studied in 1A term in Java language, and study additional algorithms. Estudy algorithms for advanced structures (graphs).

To know how to generate a Java program from a design given by UML class diagrams and the specification of the methods of the involved classes.

STUDY LOAD

Type	Hours	Percentage
Self study	85,0	56.67
Hours large group	39,0	26.00
Hours small group	26,0	17.33

Total learning time: 150 h

CONTENTS

Lesson 1. The paradigm of object orientation: classes and objects

Description:
Presentation of the paradigm of object orientation. Object and class concepts. Attributes and methods.

Specific objectives:
Formalize the concept of an object as an entity that gathers data (attributes) and functions (methods). Formalize the concept of class.

Full-or-part-time: 3h 30m
Theory classes: 1h 30m
Self study : 2h

Lesson 2. Classes and Objects in Java

Description:
Syntax class definition in Java.
Defining attributes. Access modifiers: public, private.
Definition of methods. Special methods: constructors, methods for accessing attributes (set and get).
Creating Objects: new operator. Dynamic Memory Management
Attributes and static methods. Its purpose and use.
Constant and final modifiers.

Full-or-part-time: 21h
Theory classes: 6h
Self study : 15h

Lesson 3. Primitive types in Java and relevant classes

Description:
Java primitive types: integer types, real types, type character (difference from C), boolean, byte.
Classes corresponding to the primitive types: Integer, Double, Float, Character, Boolean, Byte.
String Class. Concept and relevant methods.
From numerical values to their textual representation as Strings and vice versa.

Full-or-part-time: 7h
Theory classes: 2h
Self study : 5h

Lesson 4. Containers

Description:
Concept of container object.
Container types: sequences, dictionaries and sets.
Implementation of sequences in Java: ArrayList and LinkedList. Most relevant methods.
Iterators. Concept and its role in the path sequence.
Implementation of dictionaries in Java: HashMap. Most relevant methods.
Getting partial views of a dictionary: methods keySet (), values ??()
Implementation of sets in Java: HashSet. Most relevant methods.
Identification of the type of container that should be used in each case.

Full-or-part-time: 28h
Theory classes: 8h
Self study : 20h

(ENG) Tema 5. Introduction to UML. Class diagram

Description:
The UML as a language of graphical representation of a program.
The class diagram.
Relations between classes, associations and dependencies.
Complete definition of an association name, directionality, cardinality.
Implementation of associations in Java code.

Full-or-part-time: 11h
Theory classes: 3h
Self study : 8h

Lesson 6. Inheritance and polimorfism

Description:
Concept of inheritance. Superclasses and derived classes (subclasses). Inheritance of attributes and methods. Constructors in subclasses.
The inheritance as a relationship between classes. Representation of inheritance in class diagrams. The object instance of a subclass are also objects instance of the superclass.
The class Object class as the root class in classes hierarchy of Java. The toString () method of Object.
Concept of polymorphism. Polymorphic methods.
The Interface in Java. The interface as simulator of multiple inheritance. The interface as a mechanism that allows one object to present various types of behavior.
Sorting and searching in polimorfic arrays.

Full-or-part-time: 24h
Theory classes: 8h
Self study : 16h

Lesson 7. Exceptions

Description:
content english

Specific objectives:
The concept of exception. Creating, throwing and catching Java exceptions.
Exception handling.

Full-or-part-time: 11h 30m
Theory classes: 4h 30m
Self study : 7h

Lesson 8. Input/Output in Java

Description:
Concept of stream.
Hierarchy of Java classes for Input/Output based on streams.
Classes for managing Input/Output of characters to/from files.

Full-or-part-time: 4h
Theory classes: 1h 30m
Self study : 2h 30m

Lesson 9.Graphs and algorithms for their manipulation

Description:
Main relevant concetps of graphs. Most relevant algorithms for managing graphs.

Specific objectives:
Concept of graph Elements: vertex and edges.
Estudy and consolidate the following elements below:
Graph's representation: adjacency list, and adjacency matrix.
Java implementation of a graph.
Traversal algorithms: "breath-first search" and "depth-first search".
Finding paths in a graph: "single-source shortest path problem" and "all-pairs shortest path problem"

Related activities:
Problems solving and one lab session.

Full-or-part-time: 12h
Theory classes: 4h
Self study : 8h

ACTIVITIES

TEST

Full-or-part-time: 2h
Theory classes: 2h

LAB SESSION 1: IDE, DEBUGGING AND BASIC ALGORITHMS

Description:
Introduction to NetBeans IDE. Creating projects. Editing code. Debugging. Basic algorithms implementation

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 2: ARRAYS AND SORTING ALGORITHMS.

Description:
Usage of arrays in Java

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 3: SEARCH ALGORITHMS

Description:
Using objects in Java. String and StringBuffer. Using relevant String methods: split (), indexOf (), etc.

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 4: IMPLEMENTATION OF CLASSES AND USE OF OBJECTS IN JAVA

Description:
Defining and instantiating a class.

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 5: ALGORITHMS FOR USING LISTS, DICTIONARIES AND SETS IN JAVA.

Description:
Using containers in Java.

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 6: BUILDING A SMALL APPLICATION; THE MAZE (1)

Description:
First step in building a small application with containers, inheritance, polimorphism, exceptions, and input/output (1)

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 7: INHERITANCE AND POLYMORPHISM, THE MAZE (2)

Description:
Second version of a small application incorporating inheritance and polimorphism (2)

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 8: EXCEPTIONS AND INPUT/OUTPUT, THE MAZE (3)

Description:
Implementation of a small application incorporating exceptions, and input/output (3)

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION 9: ALGORITHMS FOR GRAPHS, THE MAZE (4)

Full-or-part-time: 2h
Laboratory classes: 2h

FINAL EXAM

Full-or-part-time: 3h
Laboratory classes: 3h

LAB SESSION: PROJECT WORKING SESSION 1

Description:
Working session on deliverable 1 of the project

Full-or-part-time: 2h
Laboratory classes: 2h

LAB SESSION: PROJECT WORKING SESSION 2

Description:
Working session on deliverable 2 of the project

Full-or-part-time: 2h
Laboratory classes: 2h

GRADING SYSTEM

- Lab project and lab test 20% to 25%
- Short exam 20% to 25%
- Final exam 55%

EXAMINATION RULES.

In order to pass, it is mandatory to deliver the lab project and to do the lab test

BIBLIOGRAPHY

Basic:
- Eckel, Bruce. Piensa en Java. 4a ed. Madrid: Prentice Hall, 2007. ISBN 9788489660342.
- García de Jalón, J. [i 7 més]. Aprenda Java como si estuviera en primero [on line]. San Sebastián: Escuela Superior de Ingenieros Industriales. Universidad de Navarra, 2000 [Consultation: 07/07/2020]. Available on: http://ocw.uc3m.es/cursos-archivados/programacion-java/manuales/java2-U-Navarra.pdf/view.

Complementary:
- The JAVA tutorials [on line]. Redwood Shores, CA: Oracle, 2019 [Consultation: 07/07/2020]. Available on: https://docs.oracle.com/javase/tutorial/.
- JAVA platform: API specification [on line]. Standard Edition 8. Redwood Shores, CA: Oracle, 2020 [Consultation: 07/07/2020]. Available on: https://docs.oracle.com/javase/8/docs/api/.

RESOURCES

Computer material:
- Java Development Kit. Java Development Kit
- https://netbeans.org/. NetBeans