230086 - POO - Object Oriented Programming

Degree competences to which the subject contributes

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 the algorithmic concepts studied in 1A term in Java language.

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.

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

ECTS credits: 6

Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: Cruellas Ibarz, Juan Carlos

Others: Fornes De Juan, Jorge
        Macías Lloret, Mario
        Perello Muntan, Jordi
        Bofill Soliguer, Pablo
        Guerrero Zapata, Manel
        Morillo Pozo, Julián David

Transversal:

06 URI N1. EFFECTIVE USE OF INFORMATION 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.

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 the algorithmic concepts studied in 1A term in Java language.

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

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group:</th>
<th>39h</th>
<th>26.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours small group:</td>
<td>26h</td>
<td>17.33%</td>
</tr>
<tr>
<td></td>
<td>Self study:</td>
<td>85h</td>
<td>56.67%</td>
</tr>
</tbody>
</table>
## Content

| Lesson 1. The paradigm of object orientation: classes and objects | Learning time: 3h 30m  
Theory classes: 1h 30m  
Self study : 2h |
|---|---|
| **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. |

| Lesson 2. Classes and Objects in Java | Learning time: 21h  
Theory classes: 6h  
Self study : 15h |
|---|---|
| **Description:** | Syntax class definition in Java.  
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. |

| Lesson 3. Primitive types in Java and relevant classes | Learning time: 7h  
Theory classes: 2h  
Self study : 5h |
|---|---|
| **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. |
## 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.

**Learning 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.

**Learning time:** 11h
- Theory classes: 3h
- Self study: 8h

## Lesson 6. Inheritance and polymorphism

**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.

**Learning time:** 24h
- Theory classes: 8h
- Self study: 16h
<table>
<thead>
<tr>
<th>Lesson 7. Exceptions</th>
<th>Learning time: 11h 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Theory classes: 4h 30m</td>
</tr>
<tr>
<td>content english</td>
<td>Self study: 7h</td>
</tr>
<tr>
<td>Specific objectives:</td>
<td></td>
</tr>
<tr>
<td>The concept of exception. Creating, throwing and catching Java exceptions.</td>
<td></td>
</tr>
<tr>
<td>Exception handling.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 8. Input/ Output in Java</th>
<th>Learning time: 18h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 6h</td>
</tr>
<tr>
<td></td>
<td>Self study: 12h</td>
</tr>
</tbody>
</table>

Description:
Concept of stream.
Hierarchy of Java classes for Input/Output based on streams as exemple of Decorator pattern.
Classes for managing Input/Output of bytes.
Classes for managing Input/Output of characters.
### Planning of activities

| TEST | Hours: 2h  
|      | Theory classes: 2h  |
|LAB SESSION 1 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
|LAB SESSION 2 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
Usage of arrays in Java  |
|LAB SESSION 3 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
Using objects in Java. String and StringBuffer. Using relevant String methods: split (), indexOf (), etc.  |
|LAB SESSION 4 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
Defining and instantiating a class.  |
|LAB SESSION 5 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
Using containers in Java.  |
|LAB SESSION 6 | Hours: 2h  
|      | Laboratory classes: 2h  |
| **Description:**  
Small application with containers, inheritance, polymorphism, exceptions, and input/output (1)  |
### LAB SESSION 7

**Description:**
Small application with containers, inheritance, polymorphism, exceptions, and input/output (2)

**Hours:** 2h  
Laboratory classes: 2h

### LAB SESSION 8

**Description:**
Small application with containers, inheritance, polymorphism, exceptions, and input/output (3)

**Hours:** 2h  
Laboratory classes: 2h

### FINAL EXAM

**Hours:** 3h  
Laboratory classes: 3h

### LAB SESSION: PROJECT WORKING SESSION 1

**Description:**
Working session on deliverable 1 of the project

**Hours:** 2h  
Laboratory classes: 2h

### LAB SESSION: PROJECT WORKING SESSION 2

**Description:**
Working session on deliverable 2 of the project

**Hours:** 2h  
Laboratory classes: 2h

### Qualification system

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

### Regulations for carrying out activities

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

Basic:


Complementary:


Others resources:

Computer material

Java Development Kit
Java Development Kit

https://netbeans.org/
NetBeans