

Caracterização da Unidade Curricular / Characterization of the Curricular Unit

Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU): Programação Orientada a Objetos / Object Oriented Programming

Área científica da UC / CU Scientific Area: Ciências Informáticas / Computer Science

Semestre / Semester: 3º

Número de créditos ECTS / Number of ECTS credits: 6

Carga horária por tipologia de horas / Workload by type of hours: TP: 45; OT: 6; O: 9

Carga letiva semanal / Weekly letive charge: 3h

Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Pretende-se que os alunos tomem contacto com as técnicas de programação orientada aos objetos, aplicando essas técnicas no desenvolvimento de programas em linguagem Java. No final do semestre os alunos devem estar habilitados a desenvolver aplicações nas linguagens Java dominando os conceitos fundamentais da programação orientada aos objetos e da programação procedimental.

Intended learning outcomes (knowledge, skills and competences to be developed by the students):

It is intended that students make contact with object-oriented programming techniques, applying these techniques in the development of Java language programs. At the end of the semester, students should be able to develop applications in Java languages, mastering the fundamental concepts of object-oriented programming and procedural programming.

Conteúdos programáticos:

1. Introdução a conceitos de programação orientada aos objetos

- 1.1. Noção de orientação por objetos
- 1.2. Tipos Abstratos de Dados
- 1.3. Objetos, classes e métodos

2. Objetos, classes e métodos

- 2.1. Variáveis de instância e variáveis de classe
- 2.2. Métodos de instância e métodos de classe
- 2.3. Construtores
- 2.4. Encapsulamento e níveis de acesso
- 2.5. Sobrecarga de métodos
- 2.6. Composição na definição de classes

3. Herança

- 3.1. Mecanismo de herança
- 3.2. Sobreposição de métodos
- 3.3. Subclasses e construtores em herança
- 3.4. Procura dinâmica de métodos
- 3.5. Polimorfismo
- 3.6. Classes abstratas em Java

4. Interfaces em Java

- 4.1. Declaração de interfaces
- 4.2. Herança de interfaces
- 4.3. Interfaces pré-definidas
- 4.4. Classes abstratas e interfaces

5. Tratamento de exceções em Java

- 5.1. Lançamento de exceções
- 5.2. Tratamento de exceções
- 5.3. Classes de exceção

Syllabus:

1. Introduction to concepts of object-oriented programming

- 1.1. Notion of object orientation
- 1.2. Abstract Types of Data
- 1.3. Objects, classes and methods

2. Objects, classes and methods

- 2.1. Instance variables and class variables
- 2.2. Instance methods and class methods
- 2.3. Builders
- 2.4. Encapsulation and access levels
- 2.5. Method overload
- 2.6. Composition in class definition

3. Heritage

- 3.1. Inheritance mechanism
- 3.2. Overlapping methods
- 3.3. Subclasses and constructors in inheritance

Sem Validade
Administrativa

- 3.4. *Dynamic method search*
- 3.5. *Polymorphism*
- 3.6. *Abstract classes in Java*

4. Java interfaces

- 4.1. Interface declaration
- 4.2. Interface inheritance
- 4.3. Predefined interfaces
- 4.4. Abstract classes and interfaces

5. Exception handling in Java

- 5.1. Throwing exceptions
- 5.2. Exception handling
- 5.3. Exception classes

