
Ano Letivo 2021-22

Unidade Curricular PROGRAMAÇÃO ORIENTADA POR OBJETOS

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 140064323

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de Desenvolvimento Sustentável - ODS (Indicar até 3 objetivos) 4,8,10

Línguas de Aprendizagem Português-PT

Modalidade de ensino

Presencial diurno

Docente Responsável

José Luís Valente de Oliveira

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
José Luís Valente de Oliveira	PL; T	T1; PL1	28T; 28PL
Helder Aniceto Amadeu de Sousa Daniel	PL	PL2; PL3	56PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
2º	S2	28T; 28PL	156	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação Imperativa

Laboratório de Programação

Algoritmos e Estruturas de Dados

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

No fim desta disciplina os alunos deverão ser capazes de entender e aplicar os princípios e as técnicas de programação orientada por objectos. Em particular, deverão ser capazes de gerar uma especificação UML recorrendo a padrões de desenho e implementar essa especificação na linguagem JAVA.

Conteúdos programáticos

1. Princípios e conceitos fundamentais da programação orientada por objetos.
 2. Introdução à modelação orientada por objetos e à UML- Unified Modeling Language.
 3. Projeto de classes
 4. Herança, polimorfismo e binding
 5. Interfaces
 6. Gestão de erros e exceções
 7. Classes aninhadas
 8. Padrões de projeto fundamentais: Template Method, Strategy, Iterator, Composite
 9. Classes e métodos genéricos
 10. Java Collection Framework
 11. Streams
 12. Internet networking (java.net)
-

Metodologias de ensino (avaliação incluída)

Método clássico de ensino e aprendizagem para as disciplinas científico-tecnológicas.

Aulas teóricas expositivas com recurso ao quadro e projector de video.

Aulas práticas com tutoriais, problemas e mini-projectos de programação, incluindo actividades que vão desde a modelação, implementação e depuração.

A disciplina adota a modalidade de avaliação por frequência, nos termos do art. 9, no. 1, b) do Regulamento Geral de Avaliação da Universidade do Algarve, de 31 de agosto de 2016, sem exame.

A nota final será obtida pela média ponderada dos trabalhos práticos realizados individualmente ou em grupo. Quando um trabalho for realizado em grupo, a nota do trabalho é convertida numa nota individual no momento da validação do trabalho. A nota do trabalho é, portanto, individual e está dependente do desempenho de cada elemento do grupo.

As validações dos trabalhos são pois individuais e poderão ser colocadas questões de índole teórica e/ou prática sobre qualquer dos tópicos da disciplina.

Bibliografia principal

[Referência principal]

Horstmann, Big Java: Early Objects 7th ed, Wiley, December 2018. ISBN: 978-1-119-49909-1

[Referência complementar sobre padrões de projeto]

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Pattern Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

[Referência complementar sobre UML]

UML Quick Ref: <http://www.uml.org>

Academic Year 2021-22

Course unit OBJECT-ORIENTED PROGRAMMING

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

CNAEF code (3 digits) 481

Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives) 4, 8, 10

Language of instruction Português-PT

Teaching/Learning modality Daytime presential

Coordinating teacher José Luís Valente de Oliveira

Teaching staff	Type	Classes	Hours (*)
José Luís Valente de Oliveira	PL; T	T1; PL1	28T; 28PL
Helder Aniceto Amadeu de Sousa Daniel	PL	PL2; PL3	56PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours	T	TP	PL	TC	S	E	OT	O	Total
	28	0	28	0	0	0	0	0	156

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Imperative Programming

Programming Lab

Algorithms and Data Structures

The students intended learning outcomes (knowledge, skills and competences)

1. Enumerate, describe, and justify the object-oriented concepts, principles, and techniques.
2. Application modeling using UML, emphasizing on architectural aspects (diagrams of classes and objects)
3. To Select and employ fundamental design patterns.
4. To use JAVA as a object-oriented programming language

Syllabus

1. Object-Oriented concepts and principles;
 2. Introduction to Object Oriented Modelling and UML (Unified Modelling Language);
 3. Designing classes
 4. Inheritance, polymorphisms, and binding
 5. Interfaces
 6. Exceptions and Error Handling
 7. Nested classes
 8. Fundamental Design Patterns : Template Method, Strategy, Iterator, Composite
 9. Generics
 10. Java Collection Framework
 11. Stream processing
 12. Internet networking (java.net)
-

Teaching methodologies (including evaluation)

Classic method of teaching and learning for scientific-technological subjects.

Lectures using the board and video projector.

Hands-on lab classes with tutorials, problems, and programming mini-projects, including activities ranging from modeling, implementation, and debugging.

The subject adopts the modality of assessment by frequency, pursuant to art. 9, no. 1, b) of the General Evaluation Regulation of the University of Algarve, of August 31, 2016, without exam.

The final grade will be obtained by the weighted average of practical work carried out individually or in groups. When a work is carried out in a group, the work grade is converted into an individual grade at the time of work validation. The grade of the work is, therefore, individual and depends on the performance of each member of the group.

The validations of the works are therefore individual and theoretical and/or practical questions may be asked about any of the topics of the course.

Main Bibliography

[Main reference]

Horstmann, Big Java: Early Objects 7th ed, Wiley, December 2018. ISBN: 978-1-119-49909-1

[Other optional references]

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Pattern Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

UML Quick Ref: <http://www.uml.org>