

Unidade Curricular	Programação Orientada por Objetos	Área Científica	Ciências da Computação
Licenciatura em	Informática de Gestão	Escola	Escola Superior de Tecnologia e de Gestão de Bragança
Ano Letivo	2023/2024	Ano Curricular	1
Tipo	Semestral	Semestre	2
Horas totais de trabalho	162	Horas de Contacto	T - - TP 60 PL - TC - S - E - OT - O -
Nível	1-1	Créditos ECTS	6.0
Código	9186-709-1204-00-23		

T - Ensino Teórico; TP - Teórico Prático; PL - Prático e Laboratorial; TC - Trabalho de Campo; S - Seminário; E - Estágio; OT - Orientação Tutórica; O - Outra

Nome(s) do(s) docente(s) Luís Manuel Alves, Maria João Tinoco Varanda Pereira

Resultados da aprendizagem e competências

No fim da unidade curricular o aluno deve ser capaz de:

1. Identificar os princípios que regem a programação orientada por objetos;
2. Modelar a solução de problemas construindo diagramas de classes e de comunicação UML;
3. Definir classes, interfaces, objetos, atributos e métodos na linguagem Java, identificando e definindo os construtores necessários para a correta inicialização das instâncias;
4. Compreender e implementar a clonagem e comparação de objetos em Java;
5. Distinguir a agregação das associações simples e realizar convenientemente a sua implementação em Java;
6. Identificar e implementar a herança entre classes e estabelecer hierarquias de classes;
7. Compreender e implementar o conceito de polimorfismo;
8. Compreender os conceitos de interfaces e classes abstratas como forma de impor funcionalidades nas classes derivadas.

Pré-requisitos

Antes da unidade curricular o aluno deve ser capaz de:

1. Ser capaz de efetuar raciocínios lógicos para a resolução de problemas;
2. Criar programas utilizando o paradigma procedimental.

Conteúdo da unidade curricular

Definição de Programação Orientada por Objetos. Princípios da Programação Orientada por Objetos. Conceitos de Modelação Orientada por Objetos. Definição de Classes em Java. Clonagem e Comparação de Objetos Java. Hierarquia de Classes e Herança. Classes Abstratas e Polimorfismo. Framework de Coleções do Java (JCF).

Conteúdo da unidade curricular (versão detalhada)

1. Definição da Programação Orientada por Objetos:
 - Motivação;
 - Conceitos básicos.
2. Princípios da Programação Orientada por Objetos:
 - Encapsulamento;
 - Herança;
 - Polimorfismo.
3. Conceitos de Modelação Orientada para Objetos:
 - Diagramas de classes em UML;
 - Diagramas de comunicação em UML;
 - Associações entre classes: simples, agregação e composição;
 - Herança: sobreposição e acréscimo de características, classes abstratas, herança múltipla.
4. Definição de classes em Java:
 - Definição de classe;
 - Criação de classes;
 - Classes simples, objetos e comportamento;
 - Métodos e variáveis de classe;
 - Classes não instanciáveis;
 - Classes Wrapper.
5. Clonagem e comparação de objetos Java:
 - Composição na definição de classes;
 - Os métodos get(), set() e encapsulamento;
 - Encapsulamento e clone;
 - Os métodos toString(), equals() e clone();
6. Hierarquia de classes e Herança:
 - Hierarquia de classes;
 - Mecanismos de herança;
 - Criação de classes via herança;
 - Compatibilidade entre classes e subclasses;
 - Programação genérica via polimorfismo;
 - Implementação da herança e polimorfismo.
7. Classes abstratas e polimorfismo:
 - Introdução às classes abstratas;
 - Utilização de classes abstratas;
 - Classes abstratas e variáveis de instância;
8. Framework de Coleções do Java (JCF):
 - Tipos genéricos;
 - As interfaces Iterable e Iterator;
 - A arquitetura de classes e interfaces da JCF;
 - As interfaces List, Set, Map e Queue;
 - Tipos de ordenação das coleções concretas da JCF;
 - As interfaces Comparable e Comparator.

Bibliografia recomendada

1. F. Mário Martins, "Java – Poo + Construções Funcionais", Tecnologias de Informação, FCA, 2017.
2. Pedro Coelho, "Programação em Java", Curso Completo, FCA, 2016.
3. J. L. M. Borges, J. F. Cunha e T. G. Dia, "Modelação de Dados em UML - Uma abordagem por problemas", FCA, 2015
4. Herbert Schildt, "Java the Complete Reference", 11th Ed., McGraw-Hill Education, 2019.
5. Paul Deitel, Harvey Deitel, "Java How to Program, Late Objects", 11th Edition, Pearson, 2020.

Métodos de ensino e de aprendizagem

Exposição teórica de conceitos acompanhada pela discussão prática de exemplos modelo. Aplicação dos conceitos através da resolução de pequenos exercícios práticos que exemplifiquem a sua utilização. Resolução de um problema que permita a aplicação global de todos os conceitos adquiridos. Execução de um projeto final.

Alternativas de avaliação

1. Alternativa 1 - (Ordinário, Trabalhador) (Final)
 - Prova Intercalar Escrita - 25%
 - Trabalhos Práticos - 25%
 - Exame Final Escrito - 50% (Componente com nota mínima de 7 valores em 20.)
2. Alternativa 2 - (Ordinário, Trabalhador) (Recurso, Especial)
 - Exame Final Escrito - 100% (Inclui exercícios suplementares destinados a substituir a Prova Intercalar e o Trabalho Prático.)

Língua em que é ministrada

Português, com apoio em inglês para alunos estrangeiros

Validação Eletrónica

Luis Manuel Alves, Maria João Tinoco Varanda Pereira	Tiago Miguel Ferreira Guimaraes Pedrosa	José Carlos Rufino Amaro	Nuno Adriano Baptista Ribeiro
29-02-2024	14-03-2024	16-03-2024	17-04-2024

Este documento só tem validade académica depois de autenticado, em todas as suas folhas, com o selo a óleo da Instituição.