

Unidade Curricular	Programação Orientada por Objetos	Área Científica	Ciências da Computação
Licenciatura em	Engenharia Informática	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	9119-706-1204-00-23		

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

Nome(s) do(s) docente(s) Paulo Duarte Ferreira Gouveia, Carlos Eduardo Castro Correia, Jose Paulo Machado Da Costa, Nelson Alexandre Perdigao Figueiredo, Sergio Paulo Perdigao do Vale

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. Construir diagramas de classes UML e implementar soluções com base na descrição de problemas
3. Definir classes, objetos, atributos e métodos com a linguagem C++, identificando e definindo os construtores necessários para a correta inicialização das instâncias
4. Reconhecer a necessidade de implementação de construtores de cópia, operadores de atribuição e destrutores como forma de manipular classes com atributos dinâmicos
5. Distinguir a agregação das associações simples e realizar convenientemente a sua implementação em C++ usando, nas relações 1-n, estruturas de dados adequadas para colecionamento dinâmico das entidades
6. Identificar e implementar a herança entre classes e estabelecer hierarquias de classes
7. Compreender o conceito de polimorfismo e definir e aplicar funções virtuais como forma de o implementar
8. Compreender os conceitos de classe abstrata e funções virtuais puras 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. Introdução à linguagem C++. Definição de classes em C++. Implementação de associações. Templates de funções e de classes. Bibliotecas standard do C++. Implementação da herança e de hierarquias de classes. Gestão de memória dinâmica interna a uma classe. Coleções híbridas. Entrada e saída de dados e acesso a ficheiros.

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. Introdução à linguagem C++
 - MS Visual Studio
 - Declarações
 - Constantes
 - Tipos de dados
 - Expressões e operadores
 - Funções
5. Definição de classes em C++
 - Atributos
 - Construtores. Categorias de construtores
 - Métodos
6. Funcionalidades básicas do C++
 - Arrays e apontadores de objetos. Autorreferência nas classes
 - Membros constantes
 - Sobrecarga de operadores (Overloading)
7. Implementação de associações simples e agregação de classes
8. Templates de funções e de classes
9. Bibliotecas standard do C++
 - Classes string e set
 - O template de classes 'Colecao'
10. Implementação de associações
 - Associações e coleções
 - Coleções de cópia e referência
 - Associações 1-N
 - Associações N-N
 - Classes associativas
11. Referências
 - Definição de referências
 - Passagem de parâmetros e retorno
12. Implementação da herança e de hierarquias de classes
 - Acréscimo e substituição de métodos
 - Construtores e herança. Listas de inicialização
 - Tipos de proteção no acesso aos membros
 - Conversão ascendente (Upcast) e descendente (downcast)
 - Polimorfismo e funções virtuais
 - Classes abstratas e funções virtuais puras
13. Gestão de memória dinâmica interna a uma classe

Conteúdo da unidade curricular (versão detalhada)

- Construtor de cópia
- Destrutor
- Operador afetação
- 14. Agregação com apontadores
- 15. Implementação de coleções híbridas
- 16. Operadores de conversão. Membros estáticos
- 17. Declarações 'friend'
- 18. Entrada e saída de dados e acesso a ficheiros

Bibliografia recomendada

1. The C++ Programming Language (4th Edition). Bjarne Stroustrup, Addison-Wesley, 2013
2. C++ Programming: An Object-Oriented Approach. B. Behrouz A. Forouzan and Richard F. Gilberg, McGraw-Hill Education, 2020
3. Fundamentals of C++ Programming. Richard L. Halterman, School of Comp. South. Adv. University US, 2018
4. C++: Guia Moderno de Programação. Henrique Loureiro, FCA – Editora de Informática, 2019
5. Modelação de Dados em UML – uma abordagem por problemas. Borges, T. Dias e J. Cunha, FCA – Editora de Informática, 2015

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)
 - Trabalhos Práticos - 50% (Modelação e implementação em C++ de uma solução no paradigma POO)
 - Exame Final Escrito - 50% (Componente com nota mínima de 7 valores em 20.)
2. Alternativa 2 - (Ordinário, Trabalhador) (Recurso, Especial)
 - Exame Final Escrito - 100%

Língua em que é ministrada

1. Português
2. Inglês

Validação Eletrónica

Paulo Duarte Ferreira Gouveia	Tiago Miguel Ferreira Guimaraes Pedrosa	Luís Manuel Alves	José Carlos Rufino Amaro
08-03-2024	14-03-2024	15-03-2024	24-03-2024