

✱ Escola Superior de Tecnologia de Abrantes

Ano Letivo 2017/2018

Informática e Tecnologias Multimédia

Licenciatura, 1º Ciclo

Plano: Despacho n.º 12419/2016 - 14/10/2016

Ficha da Unidade Curricular: Linguagens de Programação

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, TP:30.0; PL:45.0; OT:5.0;

Ano|Semestre: 1|S2; Ramo: Tronco comum;

Tipo: Obrigatória; Interação: Presencial; Código: 81439

Área Científica: Informática

Docente Responsável

Sandra Maria Gonçalves Vilas Boas Jardim

Docente e horas de contacto

Fernando Sérgio Hortas Rodrigues - Equiparado Assistente 1º Triénio: TP: 30 + PL: 45

Sandra Maria Gonçalves Vilas Boas Jardim - Professora Adjunta: OT: 5

Objetivos de Aprendizagem

1. Adquirir conhecimentos sobre o paradigma da orientação por objetos, tais como hereditariedade, abstração, encapsulamento e polimorfismo.
2. Adquirir conhecimentos sobre linguagem de programação C# e da *Framework .NET*.
3. Adquirir conhecimentos sobre utilização do ADO.NET para acesso a bases de dados.

Objetivos de Aprendizagem (detalhado)

1. Descrever as estruturas de dados e algoritmos mais comuns, assim como as suas vantagens, limitações e aplicações.
2. Utilizar estruturas de dados na resolução de problemas concretos.
3. Conceber, desenvolver, e testar código para a resolução de problemas de médio e grande porte.

Conteúdos Programáticos

C#: Framework .NET; Tipos; Expressões; Classes; Métodos; Construtores; Hereditariedade; Polimorfismo; Coleções; Interfaces; Exceções; Eventos; Delegações; Sobreposição de Métodos; *Streams* de ficheiros; ADO.NET; GUI.

Conteúdos Programáticos (detalhado)

1. Conceitos gerais
 - 1.1. Introdução à .NET e à framework .NET
 - 1.2. Exploração do ambiente de trabalho Visual Studio .NET
 - 1.3. Criação de um Projeto de uma Aplicação Windows
2. Fundamentos da linguagem C#
 - 2.1. Perceber os fundamentos da linguagem C#
 - 2.2. Os vários tipos predefinidos do C#
 - 2.3. Expressões
 - 2.4. Instruções Condicionais
 - 2.5. Instruções Iterativas



3. Criar Objetos em C#
 - 3.1. Value Types e Reference Types
 - 3.2. Boxing e Unboxing
 - 3.3. Classes e Objetos
 - 3.3.1. Sintaxe e Semântica
 - 3.3.2. Definição de Classes
 - 3.3.3. Criação e Inicialização de Objetos
 - 3.4. Namespaces
 - 3.5. Access Modifiers (Scope)
 - 3.6. Métodos
 - 3.6.1. Sintaxe e Semântica
 - 3.6.2. Parâmetros, variáveis locais e scope
 - 3.6.3. Passagem de parâmetros por valor e por referência
 - 3.6.4. Overloading
 - 3.7. Construtores
 - 3.7.1. Sintaxe e Semântica
 - 3.7.2. Overload de Construtores
 - 3.7.3. Inicialização de Construtores
 - 3.8. Inicialização de Classes
 - 3.9. Variáveis readonly
 - 3.10. Membros de Classe Estáticos
4. Técnicas de Programação Orientadas a Objetos em C#
 - 4.1. Programação Estruturada vs. Orientada por Objetos
 - 4.2. Encapsulamento
 - 4.3. Propriedades e Propriedades Automáticas
 - 4.4. Hereditariedade
 - 4.4.1. Classes Derivadas
 - 4.4.2. Acesso a Construtores da Classe Base
 - 4.4.3. Classes Seladas
 - 4.4.4. Métodos Virtuais
 - 4.4.5. Classes Abstratas
 - 4.4.6. Propriedades Abstratas
 - 4.5. Polimorfismo
5. Programar em C#
 - 5.1. Instrução *foreach*
 - 5.2. Indexação de Objetos (*Indexers/Propriedades Indexadas*)
 - 5.3. Interfaces
 - 5.4. Operadores *is* e *as*
 - 5.5. Coleções
 - 5.5.1. Implementação de Tipos Enumeráveis
 - 5.5.1.1. A interface *IEnumerable*
 - 5.5.1.2. Implementação explícita de *IEnumerator*
 - 5.5.1.3. Iteradores *Yield*
 - 5.5.1.4. Iteradores Nomeados (*Named Iterators*)
 - 5.5.2. *ArrayList*
 - 5.5.3. *Queue*
 - 5.5.4. *Stack*
 - 5.5.5. *Hash Table*
 - 5.6. Exceções (*Exception Handling*)
 - 5.7. Eventos e Delegações (*Delegates and Events*)



6. Implementação de Aplicações Baseadas em .NET, em C#
 - 6.1. A Biblioteca de Classes da .NET Framework
 - 6.2. Sobreposição de Métodos (*Overriding Methods*) na biblioteca System.Object
 - 6.3. Formatação de Cadeias de Caracteres e Números
 - 6.4. *Streams* e ficheiros

7. ADO.NET
 - 7.1. ADO.NET *Data providers*
 - 7.2. *Disconnected layer* do ADO.NET
 - 7.3. *DataSets*
 - 7.4. Ligação de Objetos *DataTable* à Interface do Utilizador *DataGrid*
 - 7.5. *Data Adapters*
 - 7.6. Geração de código automático com o *DataSet Designer*
 - 7.7. Desacoplamento do código gerado automaticamente, da camada de apresentação (*interface do utilizador*)
 - 7.8. Projetos do tipo *Class Library*
 - 7.9. Utilização dos *Strongly typed types* gerados automaticamente

Metodologias de avaliação

Época Normal de Exame e Época de Recurso

Trabalho Prático (100%)

O Trabalho Prático (TP) de avaliação, terá discussão se o docente assim solicitar. Se o aluno faltar à discussão do TP, ficará automaticamente reprovado nessa época de avaliação com a nota de zero (0) valores.

Software utilizado em aula

Visual Studio. SQL Server

Estágio

Não aplicável.

Bibliografia recomendada

Principal:

- Watson, K. et al. (2012). *Beginning Visual C# 2012 Programming*. Wrox
- Troelsen A. (2007), *Pro C# 2008 and the .NET 3.5 Platform*. Apress

Secundária:

- Marques, P. e Pedroso, H. e Figueira, R. (2011). *C# 4.0*. Lisboa: FCA
- Benjamin Perkins, Jacob Vibe Hammer, Jon D. Reid (2015). *Beginning C# 6 Programming with Visual Studio 2015*. Wrox

Coerência dos conteúdos programáticos com os objetivos

Aprendizagem inicial da linguagem de programação orientada a objeto c#, que cumpre dois objetivos: Em primeiro lugar a aprendizagem dos conceitos de orientação a objetos, presentes nas modernas linguagens de programação e em segundo lugar a aprendizagem e a prática da linguagem de programação, necessária para desenvolvimento de aplicações de arquitetura cliente-servidor, com ligação a base de dados.

Metodologias de ensino

Aulas teóricas expositivas para apresentação dos conteúdos programáticos. Aulas práticas laboratoriais para resolução de problemas e consolidação de conhecimentos utilizando o computador.

Coerência das metodologias de ensino com os objetivos

Os conceitos teóricos são lecionados com recurso a exemplos simples e ilustrativos. Na componente prática os alunos efetuam exercícios e tutoriais com exemplos mais complexos e completos, para uma consolidação profunda dos conceitos transmitidos.

Língua de ensino

Português

Pré requisitos

Considera-se importante que os alunos detenham os conhecimentos transmitidos na unidade curricular de Programação e Algoritmia.

Docente Responsável

Sandra
Jardim

Assinado de forma
digital por Sandra
Jardim
Dados: 2018.06.19
17:35:23 +01'00'

Diretor de Curso, Comissão de Curso

Sandra
Jardim

Assinado de forma
digital por Sandra
Jardim
Dados: 2018.06.19
17:35:42 +01'00'

Conselho Técnico-Científico

