

PLANO DE ENSINO

IDENTIFICAÇÃO

CURSO: Bacharelado em Engenharia de Computação

DISCIPLINA: Programação Orientada a Objetos

CÓDIGO DA DISCIPLINA: 34

PRÉ-REQUISITO(S): Algoritmos e Programação e Laboratório de Algoritmos e Programação

UNIDADE CURRICULAR: Obrigatória [X] Optativa [] Eletiva [] SEMESTRE: 3º

CARGA HORÁRIA

TEÓRICA: 67h.r PRÁTICA: 0h.r EaD: 0h.r

CARGA HORÁRIA SEMANAL: 4h/a CARGA HORÁRIA TOTAL: 67h.r

DOCENTE RESPONSÁVEL:

Ementa

O paradigma de programação orientada a objetos. Classes e objetos. Troca de mensagens, composição e coleções de objetos. Herança. Sobreposição. Encapsulamento. Visibilidade. Interface e polimorfismo. Sobrecarga. Tratamento de exceções. Desenvolvimento de programas orientados a objetos através de uma linguagem de programação.

Objetivos

Geral

- Apresentar a metodologia de desenvolvimento orientada a objetos, mostrando as técnicas e ferramentas para criação de programas usando uma linguagem de programação orientada a objetos.
- Entender a principal diferença entre programas desenvolvidos utilizando a tradicional metodologia de programação estruturada e orientada a objeto.
- Familiarizar-se com os principais conceitos que determinam o entendimento do paradigma orientado a objeto.
- Valorizar a importância da utilização de boas práticas de programação na elaboração de código fonte.

Específicos

- Instalar o pacote de programas necessário para dar início à prática de programação.
- Configurar o ambiente de desenvolvimento para programação.
- Importar bibliotecas para uso em projetos de programação.
- Escrever programas utilizando dos recursos disponíveis para tratamento de erros e exceções.

Conteúdo Programático

1ª Unidade

- Fundamentos da Linguagem Java:

- Histórico e evolução da linguagem Java.
- Arquitetura da tecnologia Java.
- Características da linguagem.
- Produtos e APIs Java.
- Escrevendo, compilando e executando aplicações Java.
- Estado da arte em ambientes de desenvolvimento e execução.
- Abstração, objetos e visão geral de conceitos de POO.
- Utilização de suporte ferramental adequado e configuração do ambiente de trabalho.
- Programação Orientada a Objetos com Java:
 - Classes e criação de objetos.
 - Membros de classe: atributos e métodos (classe e instância).
 - Abstração de dados e encapsulamento.
 - Construtores e suas características.
 - Definindo mensagens e interface de objetos.
 - Sobrecarga e sobreposição de métodos.
 - Ciclo de vida dos objetos (instanciação à destruição).
 - Classes Wrappers (Boolean, Character, Short, Integer, etc.).
 - Estruturação e Manipulação de Objetos em Java.
 - Herança e noções de Polimorfismo.
- Tipos, Literais, Operadores e Controle de Fluxo:
 - Palavras reservadas da linguagem.
 - Constantes e variáveis.
 - Tipos primitivos e de referência.
 - Expressões.
 - Coerção, conversão e promoção de tipos.
 - Operadores: atribuição, aritméticos, relacionais, lógicos e bits.
 - Estruturas de controle de fluxo.
 - Operador '==' versus método equals (Objecto).
 - Enumerações versus Variáveis de Classe.

2ª Unidade

- Reutilização com Herança e Composição de Objetos:
 - Quando usar Herança ou Composição.
 - Técnicas de composição e associação de objetos.
 - Herança: vantagens e desvantagens sobre composição.
 - Polimorfismo com herança e com composição.
 - Upcasting e Downcasting.
 - Boas práticas de programação.
 - Padrões de Projeto (essenciais) e boas práticas de programação.
- Interfaces e Polimorfismo:
 - Fundamentos sobre polimorfismo.
 - Aplicando polimorfismo com Interfaces.
 - Classes abstratas e métodos abstratos.
 - Mecanismo Late binding (vinculação dinâmica).
 - Interfaces e Herança múltipla em Java.

3ª Unidade

- Encapsulamento e Visibilidade:
 - Definindo e refinando encapsulamento.
 - Modificadores de visibilidade: public, protected, default e private.
 - Criação de pacotes em Java.

- Importação de classes.
- Tratamento de Erros e Exceções:
 - Fundamentos acerca de tratamentos de erros e seus tipos.
 - Mecanismos Try-Catch e Finally.
 - Capturando e lançando exceções, finalizando exceções.
 - Exceções padrão em Java.
 - Criando novas exceções.
 - Exceções Runnable.

4ª Unidade

- Entrada e Saída Padrão de Dados em Java:
 - Entrada padrão de dados (classe Console).
 - Saída padrão de dados (System.out).
 - Entrada/Saída de dados GUI (classe JOptionPane).
- Arrays e Strings:
 - Arrays simples e multidimensionais.
 - Ordenação de arrays (classe Arrays).
 - Características e manipulação de Strings e caracteres.
 - Classes String, StringBuilder e StringBuffer.
- Arquivos e Fluxos de Dados em Java:
 - Manipulação de dados em arquivos (pacote java.io).
 - Arquivos (classe File), fluxos de entrada e saída em Java.
 - Leitura e gravação de Objetos e Textos em Java.

Metodologia de Ensino

- Aulas teóricas expositivas, aulas práticas, pesquisas individuais e em grupo, seminários, discussões e listas de exercícios.

Avaliação do Processo de Ensino e Aprendizagem

- Provas escritas, trabalhos práticos, teóricos e listas de exercícios.

Recursos Necessários

- Quadro branco (negro) e pincel atômico.
- Projetor multimídia.
- Ambiente de desenvolvimento integrado (Eclipse, NetBeans etc.).

Bibliografia

Básica

- DEITEL, H.; DEITEL, P. **Java**: como programar. 8. ed. [S.l.]: Pearson Brasil, 2010.
- ECKEL, B. **Thinking in Java**. [S.l.]: Prentice Hall, 2008.
- SIERRA, K. **Use a cabeça!**: Java. 2. ed. [S.l.]: Alta Books, 2009.

Complementar

- HORSTMANN, C. **Java SE for the really impatient**. 1. ed. [S.l.]: Addison-Wesley, 2014.

- HORSTMANN, C., Cornell, G. **Core Java**. Vol 1: fundamentals. 9. ed. [S.I.]: Prentice Hall, 2012. ISBN: 0137081898.
- _____. **Core Java**. Vol 1: advanced features. 9. ed. [S.I.]: Prentice Hall, 2012. ISBN: 0137081898.
- LIANG, D. **Introduction to Java Programming**: comprehensive version. 10. ed. [S.I.]: Prentice-Hall, 2014. ISBN: 0133761312.
- TAHCHIEV, P.; LEME, F.; MASSOL, V.; GREGORY, G. **Junit in Action**. 2. ed. [S.I.]: Manning Publications, 2010.