



PLANO DE ENSINO

IDENTIFICAÇÃO

CURSO: Bacharelado em Engenharia de Computação

DISCIPLINA: Laboratório de Programação Orientada a Objetos

CÓDIGO DA DISCIPLINA: 35

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

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

CARGA HORÁRIA

TEÓRICA: 0h.r | PRÁTICA: 67h.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

- Desenvolver soluções para os problemas propostos, visando à obtenção dos resultados por computador. Escrever programas, utilizando uma linguagem de programação orientada a objetos. Utilizar eficientemente um Ambiente de Desenvolvimento Integrado.

Específicos

- Instalar o pacote de programas necessário para realizar as práticas de programação.
- Configurar o ambiente de desenvolvimento para programação.
- Importar bibliotecas para uso em projetos de programação.
- Criar classes e elaborar testes de unidade.
- Escrever programas utilizando os recursos disponíveis para tratamento de erros e exceções.
- Explorar os diversos recursos de um Ambiente de Desenvolvimento Integrado (ADI).
- Realizar testes de unidade e validar o software.

Conteúdo Programático

1ª Unidade

- Ambiente de Desenvolvimento Integrado:
 - Escrever, compilar e executar aplicações Java.
 - Estado da arte em ambientes de desenvolvimento e execução.
 - Criar um Projeto e um espaço de trabalho (workspace).
 - Configurar a execução de um programa.
- 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 Encapsuladoras (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.
- Encapsulamento e Visibilidade:
 - Definindo e refinando encapsulamento.
 - Modificadores de visibilidade: public, protected, default e private.
 - Criação de pacotes em Java.
 - Importação de classes.

3ª Unidade

- 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.
- 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 expositivas e dialogadas utilizando recursos audiovisuais.
- Aulas práticas.
- Pesquisas à Internet e elaboração de trabalhos em dupla sobre temas em evidência ou não cobertos pela disciplina.

Avaliação do Processo de Ensino e Aprendizagem

- Prova prática, listas de exercícios, desenvolvimento e apresentação de projetos.

Recursos Necessários

- Laboratório de Informática com suporte de ferramental para desenvolvimento de softwares.
- Quadro branco (negro) e pincel atômico (giz).
- Projetor multimídia.
- Ambiente de desenvolvimento integrado (Eclipse, NetBeans etc.).

Bibliografia

Básica

- DEITEL, H.; DEITEL, P. **Java**: como programar. 8. ed. [S.I.]: Pearson Brasil, 2010.
- SANTOS, R. **Introdução à programação orientada a objetos usando Java**. Rio de Janeiro : Elsevier, 2003. ISBN:9788535212068.
- SIERRA, K. **Use a cabeça!**: Java. 2. ed. [S.I.]: Alta Books, 2009.

Complementar

- LARMAN, C.. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007. ISBN: 9788560031528.
- CAVALCANTI, V. M. B., RODRIGUES, N. N. **Estruturas de dados lineares básica : Abordagem prática, com implementações em C e Java**. João Pessoa : IFPB, 2015. ISBN: 9788563406613.
- Goodrich, M. T. **Estruturas de dados e algoritmos em Java**. 5. ed. Porto Alegre: Bookman, 2013. ISBN: 9788582600184.
- BASHAM, B., et al. **Use a cabeça : Servlets & JSP**. 1Rio de Janeiro : Alta Books, 2005. ISBN: 9788576082941.
- HALL, M., P.; BROWN, L. **Core Servlets e JavaServer Pages: tecnologias Core**. Rio de Janeiro: Moderna, 2005, ISBN: 8573934328.