

# MINISTÉRIO DA EDUCAÇÃO

## CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA CAMPUS PETRÓPOLIS

CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

DEPARTAMENTO		PLANO DE CURSO DA DISCIPLINA			
ENGENHARIA DE COMPUTAÇÃO		PROGRAMAÇÃO ORIENTADA A OBJETOS			
CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS	
GCOM5037PE	5	2014	1		
CRÉDITOS	AULAS/SEMANA			TOTAL DE AULAS NO SEMESTRE	Algoritmos e Estrutura de Dados I
6	TEÓRICA	PRÁTICA	ESTÁGIO	108	
	3	3	0		

### EMENTA

1. Introdução e histórico linguagem Java.
2. Introdução a orientação a objetos: conceito, classes e objetos, métodos, variáveis de instância e encapsulamento, relacionamento entre objetos.
3. Abstração de Dados.
4. Instruções de controle: condicional, repetição, aninhados, break, continue.
5. Métodos: estáticos, de instância, promoção e coerção de argumentos, controles de acesso.
6. Arrays e ArrayLists.
7. Composições, enumeração, coleta de lixo.
8. Hierarquias: generalização/especialização, agregação/decomposição.
9. Herança simples e múltipla.
10. Polimorfismo.
11. Classes abstratas, modularização, visibilidade.
12. Tratamento de exceções, metaclasses, coleções.
13. Interfaces.
14. Expressões regulares.
15. Arquivos.
16. Interfaces gráficas (GUI).
17. Coleções genéricas. Classes e métodos genéricos.
18. Prática de laboratório em linguagem Java.

### BIBLIOGRAFIA

#### Básica:

1. DEITEL, P.; DEITEL, H. **Java como programar**. 8ª edição. Pearson, 2010.
2. SANTOS, R. **Introdução à Programação Orientada a Objetos usando Java**. 2ª edição. Rio de Janeiro, Elsevier, 2013.
3. BARNES, D. **Programação Orientada a Objetos com Java**. Prentice Hall Brasil. 2004.

#### Complementar:

1. GOODRICH, M.T.; TAMASSIA, R. **Estruturas de Dados & Algoritmos em Java**. 5ª edição. Bookman, 2013.
2. BARNES, D.J.; KÖLLING, M. **Programação orientada a objetos com Java: uma**

**introdução prática usando o BlueJ.** 4ª edição. Pearson, 2009.

3. FURGERI, S. **Programação orientada a objetos - conceitos, técnicas - Série eixos.** 1ª edição. Editora Érica, 2015.
4. COELHO, P. **Programação em Java - Curso Completo.** 5ª edição. FCA, 2016.
5. MENDES, D.R. **Programação Java com ênfase em Orientação a Objetos.** Novatec, 2009.

### OBJETIVOS GERAIS

O objetivo da disciplina é apresentar ao aluno os conceitos de programação orientada a objetos (POO) para que o aluno entenda e seja capaz de aplicá-los de maneira apropriada. Desenvolver aplicações usando a linguagem Java e o paradigma de orientação a objetos de forma eficiente. Desenvolver aplicações com interfaces gráficas com o usuário que sejam ergonômicas e realizem o armazenamento persistente.

### METODOLOGIA

Cada aula consistirá na combinação adequada de:

- Exposição detalhada de cada conceito com aulas expositivas sobre o conteúdo.
- Exemplificação dos conceitos através de solução de problemas que requerem a aplicação adequada de tais conceitos com proposta de problemas e apresentação de solução para os mesmos.
- Exercícios, atividades e estudos de casos, de forma individual ou em grupo.
- Trabalhos práticos de implementação que requerem a utilização dos conceitos de orientação a objetos ensinados.
- Ênfase na implementação como instrumento para garantir a assimilação dos conceitos de orientação a objetos.
- Provas individuais.

### CRITÉRIO DE AVALIAÇÃO

A avaliação será feita através de duas provas teórica, correspondendo a 60% da nota final e dois trabalhos práticos, juntamente com seu relatório correspondendo a 40% dos pontos da disciplina.

### CHEFE DO DEPARTAMENTO

NOME	ASSINATURA
Laura Silva de Assis	

### PROFESSOR RESPONSÁVEL PELA DISCIPLINA

NOME	ASSINATURA
Laura Silva de Assis	

**APROVADO PELO CONSELHO DEPARTAMENTAL EM:**

\_\_\_/\_\_\_/\_\_\_

### CONTEÚDO PROGRAMÁTICO

1. Apresentação da disciplina:
  - 1.1. Programa;
  - 1.2. Avaliação;
  - 1.3. Datas importantes;
  - 1.4. Onde encontrar informação, prazos, carga horária;

- 1.5. Bibliografia.
2. Introdução ao Java:
  - 2.1. Árvores AVL: apresentação, características, motivação, operações de busca, inserção e remoção, análise de complexidade, exercícios.
  - 2.2. Árvores Rubro-Negra: apresentação, características, motivação, operações de busca, inserção e remoção, análise de complexidade, exercícios.
3. Árvores de busca balanceadas:
  - 3.1. Árvores B: apresentação, características, motivação, operações de busca, inserção e remoção, análise de complexidade, exercícios;
  - 3.2. Variações de árvores B: árvores B\* e árvores B+. Características, motivação, operações de busca, inserção e remoção, exercícios.
4. Introdução a teoria de grafos:
  - 4.1. Conceitos básicos, vértices, arestas, definição formal, exemplos, histórico;
  - 4.2. Representação:
    - 4.2.1. Geométrica;
    - 4.2.2. Matemática;
    - 4.2.3. Computacional: matriz de adjacência, matriz de incidência e lista de adjacência;
    - 4.2.4. Exercícios.
  - 4.3. Tipos de grafos, características:
    - 4.3.1. Trivial, simples, completo, regular, ponderado, com atributos;
    - 4.3.2. Subgrafos, complementar, multigrafo, conexo;
    - 4.3.3. Maximal, minimal, planares;
    - 4.3.4. Grafos bipartidos, grafos eulerianos e algoritmos de Fleury e Hierholzer, grafos hamiltonianos.
    - 4.3.5. Exercícios.
  - 4.4. Operações e métricas:
    - 4.4.1. Adição e remoção de vértices e arestas, contração de vértices e arestas;
    - 4.4.2. Desdobramento, inversão, união, soma, produto cartesiano;
    - 4.4.3. Cintura, excentricidade, raio, diâmetro, centro, vizinhança, fechos transitivos.
    - 4.4.4. Exercícios.
5. Algoritmos em grafos:
  - 5.1. Busca em grafos: busca em largura (BFS) e busca em profundidade (DFS), exercícios;
  - 5.2. Árvores: árvores em grafos e seus teoremas, árvore geradora mínima, algoritmos de Prim e Kruskal, exercícios;
  - 5.3. Caminhos mínimos: problemas de caminhos mínimos, algoritmos de Dijkstra, Bellman-Ford e Floyd-Warshall, exercícios;
6. Técnicas de programação:
  - 6.1. Backtracking:
    - 6.1.1. Ideia da força bruta e melhoria com o backtracking, exemplos de problemas comparando as duas técnicas;
    - 6.1.2. Construção da árvore de estados;
    - 6.1.3. Ideia geral do algoritmo de backtracking;
    - 6.1.4. Exemplo de problemas e proposta de solução utilizando backtracking, mostrando pseudo-código e a árvore de estados;
    - 6.1.5. Problemas: ciclo hamiltoniano, mochila, N rainhas, passeio do cavalo, labirinto;
    - 6.1.6. Exercícios
  - 6.2. Divisão e conquista:
    - 6.2.1. Conceitos, definições, etapas;
    - 6.2.2. Algoritmo genérico, vantagens e desvantagens;
    - 6.2.3. Exemplos de problemas mostrando o pseudo-código de como resolvê-los através de divisão e conquista.
    - 6.2.4. Exercícios