



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

Faculdade de Ciências Exatas e Tecnologia

Ciência da Computação

PLANO DE ENSINO DE DISCIPLINA

1º / 2020

Disciplina	LED – Laboratório de Estruturas Dinâmicas		
Código	028188	Período	3º.
Carga Horária Semanal	04 aulas		
Carga Horária Semestral	72 aulas		
Professores	Julio Arakaki e Wladimir Roberto Esposito		

EMENTA

Alocação de estruturas dinâmicas. Análise do ciclo de vida de variáveis dinâmicas e ponteiros. Análise da realização de tipos de dados abstratos e da organização da memória de dados

OBJETIVOS

Gerais

Capacitar o estudante para especificar tipos de dados abstratos e avaliar diferentes modelos de implementação com estruturas de dados dinâmicas.

Específicos

Conduzir gradativamente o estudante a:

- Projetar o ciclo de vida de variáveis dinâmicas estruturadas.
- Utilizar tipos de dados abstratos para especificar estruturas de dados.
- Utilizar parâmetros-referência para manipular estruturas dinâmicas.
- Distinguir variáveis globais, variáveis locais, variáveis estáticas e variáveis dinâmicas.
- Implementar tipos de dados abstratos por meio de estruturas de dados dinâmicas.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

Faculdade de Ciências Exatas e Tecnologia

Ciência da Computação

PROCEDIMENTOS DE ENSINO

O método de ensino consiste em aulas práticas (4 aulas semanais em laboratório de informática). Nas aulas em laboratório serão realizados os experimentos de programação para melhorar o entendimento e a fixação dos conceitos sobre estruturas dinâmicas.

A cada duas ou três semanas de curso, serão propostos exercícios extra-aula que devem constituir desafios ao aluno para utilizar conhecimentos já adquiridos na resolução de novos problemas. A avaliação continuada propiciada pelos exercícios extra-aula deve favorecer uma realimentação do processo de aprendizagem.

INSTRUMENTOS E CRITÉRIOS DE AVALIAÇÃO

Para ser aprovado nesta disciplina o aluno deverá atingir pelo menos 75% de presença em 18 semanas de aula e média final igual ou superior a 5.0 (cinco).

$$MF = (N1 + N2) / 2. (0.8 + 0.04 A) \text{ com } Ni = (a \cdot Pi + b \cdot Ai) / (a + b) \text{ com } a, b \{1, 2, 3\}$$

Pi: nota da Prova do bimestre i (i: 1, 2)

Ai: nota de Atividades do bimestre i (i: 1, 2)

A **Média Final (MF)**, é obtida fazendo-se:

$$a = 1$$

$$b = 1$$

$$A = 5$$

Ai é a média aritmética das atividades de implementação realizadas durante o bimestre i, i=1,2

Não haverá, em hipótese alguma, substituição ou reposição para atividade perdida por não comparecimento à aula na qual foi aplicada a atividade.

A Prova Substitutiva, se efetuada, substitui P1 ou P2, sempre favorecendo a melhor média MF.

Com isso, a média final é calculada da seguinte forma:

$$MF = (N1 + N2) / 2. \text{ com } Ni = (Pi + Ai) / 2, i = 1,2$$

RECURSOS NECESSÁRIOS

Laboratório equipado com microcomputadores e programas que configurem um ambiente mínimo para o desenvolvimento de programas em Linguagem Java.

Software: ambiente para desenvolvimento de programas em Linguagem Java (IDE). Por exemplo: BlueJ, Eclipse e NetBeans.

Projektor para apresentação de material de apoio às aulas.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

Faculdade de Ciências Exatas e Tecnologia

Ciência da Computação

CONTEÚDO PROGRAMÁTICO

Principais construções em linguagem OO: objetos, classes e estruturas dinâmicas

PRÉ-REQUISITOS - TÓPICOS

Conteúdos básicos apresentados nas disciplinas LIC e LP

BIBLIOGRAFIA

BASICA

- Barnes,D. e M.Köll, **Programação Orientada a Objetos com Java: Uma Introdução Prática Usando Bluej**, 4.ed., Pearson-Prentice Hall, 2009.
- Goodrich,M.T. e R.Tamassia, **Estruturas de Dados e Algoritmos em Java**, Bookman, 2013.
- Watt,D.A. e W.Findlay, **Programming Language Design Concepts**, Wiley, 2004.

COMPLEMENTAR

- Arnold,,K., J.Gosling e D.Holmes, **A linguagem de Programação Java**, Bookman, 2007.
- Blaha,M. e J.Rumbaugh, **Modelagem e Projetos Baseados em Objetos com UML 2**, Elsevier, 2006.
- Deitel,P. e H.Deitel, **Java – Como Programar**, 8.ed., Prentice Hall, 2010.
- Eckel,B., **Thinking in Java**, 4.ed., Prentice Hall, 2006.
- Fowler,M., **UML Essencial: Um Breve Guia para a Linguagem-padrão de Modelagem de Objetos**, Bookman, 2005.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

Faculdade de Ciências Exatas e Tecnologia

Ciência da Computação

CRONOGRAMA

Semana	Tópicos
1	Apresentação da disciplina, plano de ensino, critério de avaliação, revisões básicas.
	Introdução ao Java: compilador, máquina virtual, byte-codes, compilação sem IDE, parâmetros para o main. Exercício: Calculadora de linha de comando
2	Ferramenta de desenvolvimento (IDE): BlueJ, Eclipse
	Exemplo de herança simples: NomePessoa e Texto. Calculadora simples no BlueJ Atividade: implementar operações avançadas na calculadora: raiz quadrada, seno, inverte, fatorial.
3	Entrega de atividade (avaliação individual)
	Alocação dinâmica, em C e em Java (new, “Garbage Collector”)
4	Classes e Objetos: sintaxe, atributos, construtores, acessos
	Exercício: calculadora OO
5	Métodos estáticos, entrada de dados em java: Scanner, JOptionPane
	Saída de dados: System.out, JOptionPane
6	Conceito de interface
	Exemplo: cadastro de alunos
7	Como fazer um “Vetor dinâmico”?
	Atividade: implementação de um cadastro de alunos com vetor dinâmico
8	Tratamento de exceções
	Exercícios: tratamento de exceções
9	GUI em java: swing e javax
	Atividade: cadastro de alunos com GUI, tratamento de exceções, interface,
10	Avaliação P1
	Correção e vista da P1
11	Lista Ligada, Pilha, Fila (implementação – teoria ED)
	Exercícios: uso de lista ligada no lugar do vetor dinâmico (uso de interface)
12	Lista Ligada, Pilha, Fila (uso de bibliotecas)
	Exercícios
13	Lista Duplamente ligada, Lista Duplamente ligada circular
	Exercícios
14	Projeto Final: implementar um simulador do Josephus
	Projeto Final: implementação
15	Projeto Final: implementação
	Projeto Final: implementação
16	Avaliação P2 , entrega do projeto final
	Correção e vista da P2
17	Avaliação PS
	Vista da prova PS
18	Semana do TCC
	Semana do TCC