

PROGRAMA DE DISCIPLINA

Disciplina: LINGUAGENS DE PROGRAMAÇÃO

Código: 1028

Carga Horária: 68

Ano Letivo: 2007

Curso: CIÊNCIA DA COMPUTAÇÃO

1. EMENTA

- Estudo de paradigmas e avaliação de linguagens de programação.

2. OBJETIVOS

- Estudar as características peculiares de algumas linguagens de programação.
- Realizar estudo comparativo de linguagens de programação.

3. PROGRAMA

1. Aspectos preliminares

- 1.1. Linguagens naturais X linguagens de programação
- 1.2. Representação e abstração em linguagens de programação
- 1.3. Elementos de uma linguagem de programação
 - 1.3.1. Classes de palavras e metalinguagem
 - 1.3.2. Métodos formais para descrever a sintaxe (gramáticas de atributos)
 - 1.3.3. Semântica estática e dinâmica (*lambda calculus*)
- 1.4. Projeto de linguagens de programação
 - 1.4.1. Propriedades desejáveis em uma linguagem de programação
 - 1.4.2. Influências sobre o projeto de uma linguagem
 - 1.4.3. Métodos de implementação
 - 1.4.4. Ambientes de programação
- 1.5. Evolução das principais linguagens de programação
- 1.6. Paradigmas de linguagens de programação

2. O paradigma de programação orientada a objeto

- 2.1. Questões de projeto referentes às linguagens orientadas a objeto
- 2.2. Visão geral da linguagem *Smalltalk*
- 2.3. Exemplo de programas em *Smalltalk*
- 2.4. Recursos em grande escala em *Smalltalk*
- 2.5. Avaliação da linguagem *Smalltalk*
- 2.6. Suporte para programação orientada a objeto em C++, *Java*, *Ada 95* e *Eiffel*
- 2.7. O Modelo de objetos de *JavaScript*
- 2.8. Implementação de construções orientadas a objeto

- 3.O paradigma de programação lógica
 - 3.1.Uma breve introdução ao cálculo de predicados
 - 3.2.Uma visão geral de programação lógica
 - 3.3.Os elementos básicos do *Prolog*
 - 3.4.Aplicações da programação lógica

- 4.O paradigma de programação funcional
 - 4.1.Fundamentos das linguagens de programação funcionais
 - 4.2.Uma introdução à linguagem *Scheme*
 - 4.3.Outras linguagens funcionais (*Common LISP, ML e Haskell*)
 - 4.4.Aplicações de linguagens funcionais
 - 4.5.Uma comparação entre as linguagens funcionais e imperativas

- 5.Objetos, nomes e vinculações
 - 5.1.O problema com os nomes e variáveis
 - 5.2.O conceito de vinculação
 - 5.3.Vinculando um nome a uma constante
 - 5.4.Uma revisão sobre alocação e vinculação
 - 5.5.Tipos de objetos
 - 5.6.Colocando um valor em um objeto de armazenamento
 - 5.7.O modelo de armazenamento: administrando objetos de armazenamento
 - 5.8.Verificação de tipos, tipificação forte e compatibilidade de tipos
 - 5.9.O escopo de um nome e ambientes de referência
 - 5.10.Implicações para os compiladores/interpretadores

- 6.Expressões e instruções de atribuição
 - 6.1.Expressões aritméticas
 - 6.2.Operadores sobrecarregados
 - 6.3.*Casts*, conversões e coerções de tipos
 - 6.4.Expressões relacionais e booleanas
 - 6.5.Avaliação em circuito-curto
 - 6.6.Instruções de atribuição
 - 6.7.Atribuição de modo misto

- 7.Estruturas de controle no nível de instrução
 - 7.1.Instruções compostas
 - 7.2.Estruturas de controle de condicionais
 - 7.3.Estruturas de controle iterativas (explícitas e implícitas)
 - 7.4.Controle global (desvio incondicional, interrupções e continuações)
 - 7.5.Comandos guardados

- 8.Subprogramas e parâmetros
 - 8.1.Fundamentos dos subprogramas
 - 8.2.Questões de projeto referentes aos subprogramas
 - 8.3.Ambientes de referência locais
 - 8.4.Métodos de passagem de parâmetros
 - 8.5.Parâmetros que são nomes de subprogramas
 - 8.6.Subprogramas sobrecarregados
 - 8.7.Subprogramas genéricos
 - 8.8.Questões de projeto referentes a funções
 - 8.9.Acessando ambientes não-Loais
 - 8.10.Operadores sobrecarregados definidos pelo usuário

- 9. Implementando subprogramas
 - 9.1. A semântica geral das chamadas e dos retornos
 - 9.2. Implementando subprogramas *FORTRAN 77*
 - 9.3. Implementando subprogramas em linguagens assemelhadas a *Algol e C*
 - 9.4. Implementando blocos
 - 9.5. Implementando o escopo dinâmico
 - 9.6. Implementando parâmetros que são nomes de subprograma

- 10. A representação de tipos
 - 10.1. Tipos em linguagens de programação
 - 10.2. Tipos primitivos de *hardware*
 - 10.3. Tipos definidos pelo programador e compostos
 - 10.4. Operações em objetos compostos e em tipos
 - 10.5. A semântica de tipos
 - 10.5.1. Descrição semântica e checagem de tipo
 - 10.5.2. Identidade de domínio
 - 10.5.3. Domínios definidos pelo programador

- 11. Tipos de dados abstratos
 - 11.1. O conceito de abstração e encapsulamento
 - 11.2. Introdução à abstração de dados
 - 11.3. Questões de projeto
 - 11.4. Tipos de dados abstratos parametrizados

- 12. Manipulação de exceções
 - 12.1. Introdução à manipulação de exceções
 - 12.2. Manipulação de exceções em PL/I, Ada, C++ e Java

4. BIBLIOGRAFIA

- 1. SEBESTA, R.W., *Conceitos de Linguagem de Programação*, 5ª Edição. Editora [Bookman Companhia](#), 2003.
- 2. VAREJÃO, F., *Linguagens de Programação: Conceitos e Técnicas*, Editora Campus, 2004.
- 3. GHEZZI, C., JAZAYERI, M. *Programming Language Concepts*, 3rd Edition, New York: John Willey & Sons, 1997.
- 4. FISHER, A.E. AND GRODZINSKY, F.S., *The Anatomy of Programming Languages*, Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- 5. GELERNTER, D. AND JAGANNATHAN, S., *Programming Linguistics*, Cambridge, Massachusetts: The MIT Press, 1990.
- 6. PRATT, T.W., *Programming Languages: Design and Implementation*, 3rd Edition. Prentice-Hall, 1996.
- 7. WIRTH, N., *Programming in Modula-2*, 3rd edition, Springer-Verlag, 1983.
- 8. BARNES, J.G.P., *Programming in ADA*, Addison-Wesley, 1982.
- 9. MEYER, B., *Object Oriented Software Construction*, Prentice-Hall, 1988.
- 10. Manuais das Linguagens *Java, Scheme, Prolog, Lua, Python, Ruby e SmallTalk*.

APROVAÇÃO DO DEPARTAMENTO
Assinatura do Chefe

APROVAÇÃO DO COLEGIADO
Assinatura do Coordenador