

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Instituto de Informática - Departamento de Informática Aplicada

Disciplina: INF01121 - MODELOS DE LINGUAGENS DE PROGRAMAÇÃO
Carga Horária: 4 horas-aula/semana
Créditos: 04 (quatro)
Professor: Érika Cota (erika@inf.ufrgs.br)
Período letivo: 2008/02
Horário: Segundas e quartas, 15h30min - 17h10min, sala 102, prédio 43425

OBJETIVOS

Estudar os princípios de projeto e as características dos principais modelos de linguagens de programação e sua adequação à solução de problemas.

SÚMULA

Tipos, escopo, tempo de amarração de variáveis. Sub-rotina, co-rotinas, passagem de parâmetros, recursividade. Paralelismo, sincronização, etc. Paradigmas de Programação: Imperativo, Funcional, Lógico, Orientado a Objetos.

PROCEDIMENTOS DIDÁTICOS

A disciplina será desenvolvida através de aulas expositivas intercaladas com aulas práticas de programação em linguagens ML, Prolog e Java. Além disso, estão previstas aulas de discussão onde os alunos irão identificar e discutir decisões de projeto de outras linguagens de programação. Por fim, trabalhos práticos de programação serão definidos para a fixação dos paradigmas de programação.

FORMATO DAS AULAS DE DISCUSSÃO E IMPLEMENTAÇÃO DO TRABALHO PRÁTICO

As aulas de discussão e o trabalho prático têm o objetivo de permitir aos alunos verificar como as diversas questões de projeto foram tratadas em diferentes linguagens de programação.

A primeira aula de discussão tratará das linguagens de Script. Os artigos distribuídos apresentam argumentos a favor e contra essas linguagens, bem como suas características. Nesta aula, os alunos deverão ver o conceito de Linguagens Script, as aplicações mais comuns, suas vantagens e desvantagens.

Na segunda aula de discussão, cada grupo deverá apresentar as características gerais da linguagem escolhida (filosofia, histórico, objetivos, etc.) bem como as decisões de projeto mais importantes (que diferenciem a linguagem de outras, que tenham sido importantes para implementação do trabalho prático, etc.). Pretende-se com isso propor uma discussão comparativa entre as linguagens.

Na terceira aula de discussão, cada grupo apresentará a implementação do trabalho prático em uma linguagem de programação. Nesta aula, com base no programa implementado e nas métricas de qualidade de uma linguagem de programação, uma comparação quantitativa entre as linguagens estudadas poderá ser feita.

A presença nas aulas de discussão é **obrigatória** e a avaliação será feita individualmente, de acordo com a contribuição de cada aluno para a discussão.

SISTEMA DE AVALIAÇÃO

Provas, trabalhos e aulas de discussão serão avaliados com notas entre 0.0 e 10.0. Ao longo do semestre, serão realizados:

- i. Duas (2) provas, P1 e P2, correspondendo, cada uma, a 30% da nota final;
- ii. Um (1) trabalho prático (TP) envolvendo implementação e duas aulas de discussão, correspondendo, no total, a 25% da nota final;
- iii. Uma (1) aula de discussão (D1) correspondendo a 15% da nota final.

A presença nas aulas de discussão é obrigatória e a ausência na aula de discussão implica nota zero nesta avaliação específica. A nota será dada individualmente de acordo com a participação nas discussões.

A nota do trabalho prático só será contabilizada na média final se o aluno atingir 45% de aproveitamento pelas avaliações restantes.

A média geral (MG) será obtida por meio da seguinte fórmula, a partir da média parcial (MP):

$$MP = 0.3 * P1 + 0.3 * P2 + 0.15 * D1$$

SE $MP \geq 4.5$ ENTAO

$$MG = MP + 0.25 * TP$$

SENAO

Reprovado.

A conversão da MG para conceitos é feita por meio da seguinte tabela:

9,0 = MG \leq 10,0 : conceito A (aprovado).

7,5 = MG < 9,0 : conceito B (aprovado).

6,0 = MG < 7,5 : conceito C (aprovado).

0,0 = MG < 6,0 : conceito D (reprovado).

RECUPERAÇÃO

A recuperação será realizada somente para os casos previstos na legislação: saúde, parto, serviço militar, convocação judicial, luto, etc., devidamente comprovados, em data e horário a serem definidos pela professora.

Recuperação do conceito "D": está prevista no cronograma uma prova de recuperação no final do semestre, a ser aplicada somente aos alunos que obtiverem o conceito D (reprovado). Caso o aluno obtenha nota igual ou superior a 6,0 nesta prova de recuperação, ele terá o seu conceito modificado para "C".

ATENDIMENTO EXTRA-CLASSE

- O atendimento extra-classe para esclarecimento de dúvidas ou discussão de assuntos relacionados à disciplina pode ser agendado por e-mail com a professora (erika@inf.ufrgs.br).

IMPORTANTE

Uma página da disciplina foi criada no Moodle (<http://moodle.inf.ufrgs.br>). Nesta página serão colocados os slides, especificações de trabalhos, notas e outros materiais que sejam necessários. Toda comunicação oficial do professor com os alunos será feita APENAS por e-mail através do Moodle. Assim, acesse IMEDIATAMENTE o sistema e registre, no link "Dados Pessoais", um e-mail que seja lido com frequência. Código de acesso: MLP20082

BIBLIOGRAFIA

Livro Texto: Sebesta, R. W. Conceitos de Linguagens de Programação. Editora Bookman, 2000.

Referências adicionais:

- 1) Scott, M. L. Programming Language Pragmatics. Morgan Kaufmann Pub., U.S, 2000.
- 2) Melo, A.C.V. Princípios de Linguagens de Programação. Ed. Edgard Blücher, 2003.
- 3) Watt, D.A. Programming Languages Concepts and Paradigms. Prentice Hall, UK, 1990.
- 4) Ghezzi, C.; Jazayeri, M. Programming Language Concepts. John Wiley & Sons, Third edition, 1998
- 5) Watt, D.A. Programming Language Design Concepts. John Wiley & Sons, 2004.

Material de apoio: <http://moodle.inf.ufrgs.br>

UFRGS - INSTITUTO DE INFORMÁTICA - DEPARTAMENTO DE INFORMÁTICA APLICADA

INF01121 - MODELOS DE LINGUAGENS DE PROGRAMAÇÃO - Semestre 2008/2

Conteúdo Programático e cronograma

Aula	Dia	Conteúdo
01	04/08	Apresentação da disciplina. Introdução a linguagens de programação: linguagens e o processo de desenvolvimento de programas.
02	06/08	Qualidades desejáveis em linguagens de programação. Sintaxe e semântica. Processadores de linguagens. Compilação, interpretação. Máquinas virtuais.
03	11/08	Modelo de programação funcional: características do modelo; semântica de programas. Linguagem funcional ML: tipos de dados primitivos e estruturados, funções primitivas, funções compostas.
04	13/08	(laboratório) ML: introdução, tipos de dados primitivos e estruturados, funções.
05	18/08	(laboratório) ML: funções compostas, recursividade, <i>pattern matching</i> , polimorfismo e sobrecarga.
06	20/08	(laboratório) ML: tipos definidos pelo usuário e tipos estruturados de dados, tipos polimórficos.
07	25/08	(laboratório) Prolog: Modelo de programação em lógica: características do modelo; fundamentos de lógica; semântica de programas.
08	27/08	(laboratório) Prolog: aritmética/recursividade, estrutura de um programa, I/O.
09	01/09	(laboratório) Prolog: listas, estruturas, bases de dados, gramáticas DCG.
	03/09	Não haverá aula – SBCCI
10	08/09	Discussão D1: Linguagens de Script.
11	10/09	Bindings (amarrações/vinculações) - parte 1
12	15/09	Bindings (amarrações/vinculações) - parte 2
13	17/09	Tipos de Dados: sistemas de tipos, tipos primitivos, estruturados e compostos, tipos definidos pelo usuário; Introdução à Orientação a Objetos.
14	22/09	Tipos de dados: representação, equivalência, compatibilidade, conversão e checagem de tipos.
15	24/09	(laboratório) Java: ambiente, programas, variáveis e seu alcance, visibilidade e duração; verificação e conversão de tipos.
16	29/09	Primeira verificação de aproveitamento (P1).
17	01/10	Tipos abstratos de dados (TAD). Classes como tipos de dados: composição e extensão de tipos (herança). Herança múltipla.
18	06/10	(laboratório) Java: Reflexão; E/S; TAD
19	08/10	Polimorfismo: sobrecarga de operadores e funções, coerção, polimorfismo paramétrico, estático e dinâmico.
20	13/10	(laboratório) Java: encapsulamento, herança, composição e polimorfismo.
21	15/10	Expressões e atribuição. Operadores. Comandos de controle de fluxo: condicionais e iterativos. Iteradores
	20/10	Semana Acadêmica
	22/10	
22	27/10	Sub-rotinas e semântica de passagem de parâmetros. Invocações. Registros de ativação.
23	29/10	Modos de passagem de parâmetros e modelos de implementação. Funções como parâmetros.
24	03/11	(laboratório 104) Java: Sistemas de tratamento de exceções; Tratamento de Exceções e polimorfismo em Java.
25	05/11	Prog. concorrente/distribuída: aspectos sintáticos e semânticos. Execução seqüencial e concorrente em um único programa. Mecanismos de controle prog. concorrente/distribuída: semáforos, monitores e rendez-vous.
26	10/11	(laboratório 104) Java: concorrência em Java.
27	12/11	Discussão D2: LPs adicionais - características gerais, filosofia, pontos importantes para trabalho prático.
28	17/11	Discussão D3: Apresentação trabalhos práticos
29	19/11	Segunda verificação de aproveitamento.
30	24/11	Recuperação do conceito D
	26/11	
	01/12	
	03/12	
	08/12	