

PLANO DE ENSINO

INF01121 - MODELOS DE LINGUAGENS DE PROGRAMAÇÃO

Semestre: 2009/2

Carga horária: 60 h

Créditos: 04

Professor(es): Leandro Krug Wives

Objetivos:

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

Súmula:

1. Tipos, escopo, tempo de amarração de variáveis. 2. Subrotina, co-rotinas, passagem de parâmetros, recursividade. 3. Paralelismo, sincronização, etc. 4. Paradigmas de Programação: Imperativo, Funcional, Lógico, OO.

Conteúdo Programático:

Introdução a linguagens de programação. Qualidades desejáveis em linguagens de programação. Modelos de linguagens de programação. Modelo funcional. Modelo Lógico. Modelo Imperativo. Amarrações (bindings). Sistemas de Tipos de Dados. Tipos Abstratos de Dados. Polimorfismo. Expressões. Operadores. Comandos de Controle de Fluxo. Subrotinas. Passagem de Parâmetros. Sistemas de Tratamento de Exceção. Mecanismos de suporte à Concorrência e ao Paralelismo em LPs.

Procedimentos Didáticos:

A disciplina será desenvolvida através de aulas expositivas intercaladas com aulas práticas de programação em linguagens imperativas, lógicas e Funcionais. Além disso, trabalhos práticos de programação de maior porte serão definidos para a fixação dos paradigmas estudados. Até 20% das aulas poderão ser realizados a distância (EAD).

Cópias das transparências apresentadas em aula serão disponibilizadas no ambiente Moodle Institucional ([HTTP://moodleinstitucional.ufrgs.br](http://moodleinstitucional.ufrgs.br)). O ambiente pode ser acessado por todos os alunos matriculados na disciplina, bastando para tal utilizar o seu número de matrícula (Cartão UFRGS) e sua senha pessoal do Portal do Aluno. O material disponibilizado no Moodle deve ser utilizado como auxílio ao estudo e como complementação aos apontamentos realizados em aula. Espera-se que os alunos tenham uma participação ativa nas aulas, realizando anotações, procurando compreender os conteúdos e, através de perguntas, resolvam suas dúvidas.

Atenção: sugere-se ao aluno que utilize e verifique constantemente o material disponível no ambiente Moodle citado, antes e após as aulas. Nele serão disponibilizados exercícios e atividades extraclasse, notas, especificações de trabalho e demais materiais que sejam necessários. Toda comunicação oficial do professor com os alunos será feita APENAS por e-mail através do Moodle. Assim, não se esqueça de verificar se seu e-mail está atualizado no sistema da UFRGS.

Método de Avaliação:

Ao longo do semestre, serão realizados:

- i. Três (3) provas, P1, P2 e P3, correspondendo, respectivamente, a 20%, 25% e 25% da nota final;
- ii. Um (1) trabalho prático (TP) envolvendo implementação de um programa em duas linguagens de programação, apresentação do resultado em aula e elaboração de relatório, correspondendo, no total, a 30% da nota final;

A nota do trabalho prático só será contabilizada na média final se o aluno realizar todas as atividades relacionadas com ele (implementação, elaboração e entrega de relatório e apresentação).

Provas, trabalhos e demais atividades serão avaliados com notas entre 0.0 e 10.0.

A média final (MF) será obtida por meio da seguinte fórmula: $MF = 0.20 * P1 + 0.25 * P2 + 0.25 * P3 + 0.3 * TP$

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

PLANO DE ENSINO

Média Final	Conceito
[9,0 – 10,0]	A (aprovado)
[7,5 – 9,0]	B (aprovado)
[6,0 – 7,5]	C (aprovado)
[4,0 – 6,0]	Sem conceito (recuperação), podendo passar para conceito C em caso de aprovação ou conceito D em caso de reprovação na recuperação
[0,0 – 4,0]	Conceito D (reprovado)
Faltas > 25%	Conceito FF (reprovado)

Atenção: as notas das provas e trabalhos serão disponibilizadas em até 15 dias depois de sua realização.

Observações sobre o Trabalho Prático

O trabalho prático tem o objetivo de permitir aos alunos verificar como as diversas questões de projeto de linguagem foram tratadas em diferentes linguagens de programação. O trabalho prático poderá ser feito em grupos de até dois (2) alunos. O grupo deverá escolher duas linguagens de programação para trabalhar no projeto, ao longo do semestre. O trabalho prático deverá ser implementado nas duas linguagens escolhidas pelo grupo.

No final do semestre, cada grupo apresentará a implementação do trabalho prático, comparando os resultados nas duas linguagens escolhidas, ressaltando suas diferenças, semelhanças, facilidades e dificuldades, utilizando também as métricas de qualidade de linguagens de programação estudadas em aula.

Observação sobre a Recuperação

- A recuperação das provas individuais será realizada somente para os casos previstos na legislação: saúde, parto, serviço militar, convocação judicial, luto, etc., devidamente comprovados.
- Somente serão calculadas as médias finais daqueles alunos que tiverem, ao longo do semestre, obtido um índice de frequência às aulas igual ou superior a 75% das aulas previstas. Aos que não satisfizerem esse requisito, será atribuído o conceito FF (Falta de Frequência).
- Recuperação do conceito “D”: está prevista no cronograma uma prova de recuperação no final do semestre, a ser aplicada somente aos alunos que obtiverem MF entre 4,0 e 6,0. Caso o aluno obtenha nota igual ou superior a 6,0 na prova de recuperação, ele terá o seu conceito modificado para “C”. Caso contrário, ficará com “D”.
- Pedidos de revisão de notas só serão avaliados se o aluno tiver pelo menos 75% de presenças.

Bibliografia Básica:

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

Varejão, F. Linguagens de programação: Java, C e C e outras : conceitos e técnicas. Rio de Janeiro: Campus, 2004. 334 p.

Bibliografia Complementar:

Scott, M. L. Programming Language Pragmatics. Morgan Kaufmann Pub., U.S, 2000.

Watt, D.A. Programming Language Design Concepts. John Wiley & Sons, 2004.

Watt, D.A. Programming Languages Concepts and Paradigms. Prentice Hall, UK, 1990.

Ghezzi, C.; Jazayeri, M. Programming Language Concepts. John Wiley & Sons, Third edition, 1998.

Melo, A. C. V. Princípios de Linguagens de Programação. Ed. Edgard Blücher, 2003.

Cronograma por aula:

Aula	Dia	Conteúdo
01	17/08	Apresentação da disciplina. Introdução a linguagens de programação: linguagens e o processo de desenvolvimento de programas. Qualidades desejáveis em linguagens de programação.
02	19/08	Sintaxe e semântica. Processadores de linguagens. Compilação, interpretação. Máquinas virtuais.

PLANO DE ENSINO

03	24/08	Modelo de programação funcional: características do modelo; semântica de programas. Linguagem funcional ML: tipos de dados primitivos e estruturados, funções primitivas, funções compostas.
04	26/08	Laboratório ML: introdução, tipos de dados primitivos e estruturados, funções. Apresentação do enunciado do trabalho prático.
05	31/08	Laboratório ML: funções compostas, recursividade, <i>pattern matching</i> , polimorfismo e sobrecarga.
06	02/09	Laboratório ML: tipos definidos pelo usuário e tipos estruturados de dados, tipos polimórficos.
	07/09	Feriado
07	09/09	Laboratório Prolog: Modelo de programação em lógica: características do modelo; fundamentos de lógica; semântica de programas.
08	14/09	Laboratório Prolog: aritmética, recursividade, estrutura de um programa, I/O.
09	16/09	Laboratório Prolog: listas, estruturas, bases de dados, gramáticas DCG.
10	21/09	Bindings (amarrações/vinculações) – parte 1.
11	23/09	Bindings (amarrações/vinculações) – parte 2.
12	28/09	Primeira verificação de aproveitamento.
13	30/09	Tipos de Dados: sistemas de tipos, tipos primitivos, estruturados e compostos, tipos definidos pelo usuário; Introdução à Orientação a Objetos.
14	05/10	Tipos de dados: representação, equivalência, compatibilidade, conversão e checagem de tipos.
15	07/10	Expressões e atribuição. Operadores. Comandos de controle de fluxo: condicionais e iterativos. Iteradores.
16	12/10	Tipos abstratos de dados (TAD).
17	14/10	Polimorfismo: sobrecarga de operadores e funções, coerção, polimorfismo paramétrico, estático e dinâmico.
	19/10	Semana Acadêmica
	21/10	
18	26/10	Laboratório Java: ambiente, programas, variáveis e seu alcance, visibilidade e duração; verificação e conversão de tipos.
19	28/10	Laboratório Java: encapsulamento, herança, composição e polimorfismo.
20	02/11	Laboratório Java: TAD, Reflexão e E/S.
21	04/11	Segunda verificação de aproveitamento.
22	09/11	Subrotinas (parte 1): Definição, Invocação. Registros de ativação. Acesso a variáveis não locais.
23	11/11	Subrotinas (parte 2): semântica de passagem de parâmetros. Modos de passagem de parâmetros e modelos de implementação Funções como parâmetros.
24	16/11	Laboratório Java: Fluxo de controle, iteradores, passagem de parâmetros.
25	18/11	Laboratório Java: Tratamento de Exceções e polimorfismo em Java.
26	23/11	Laboratório Java: concorrência em Java. Prog. concorrente/distribuída: aspectos sintáticos e semânticos. Execução sequencial e concorrente em um único programa. Mecanismos de controle programação concorrente/distribuída: semáforos, monitores e rendez-vous.
27	25/11	Terceira verificação de aproveitamento.
28	30/11	Apresentação trabalhos práticos
29	2/12	Apresentação trabalhos práticos
30	7/12	Recuperação
	9/12	
	14/12	
	16/12	
	21/12	
	23/12	

OBS: as aulas 16 e 17 serão realizadas na modalidade EAD (Moodle Institucional).