

PLANO DE ENSINO

INF01120 – TÉCNICAS DE CONSTRUÇÃO DE PROGRAMAS

Semestre: 2009/2

Carga horária: 04 h

Créditos: 04

Professor(es):

Eduardo Borges () – Turma A

Érika Cota (erika@inf.ufrgs.br) – Turma B

Marcelo Pimenta (mpimenta@inf.ufrgs.br) – Turma C

Súmula:

Discutir, desenvolver e empregar técnicas de programação que produzam programas de "boa qualidade"; apresentação das técnicas utilizadas para este fim; construção, durante o semestre de um programa de médio porte, onde estas técnicas devem ser exercitadas.

Conteúdo Programático:

O objetivo é capacitar o aluno a implementar, testar e refinar programas complexos a partir de um conjunto de técnicas de programação que visem a qualidade do software sendo desenvolvido. Como parte deste objetivo, esta disciplina oferece os conceitos básicos de qualidade de software, modularidade, tipos abstratos de dados e uma introdução ao paradigma da programação orientada a objetos.

Procedimento Didático:

As aulas serão de cunho teórico-expositivo intercaladas com discussões sobre os exercícios e os trabalhos extra-classe, além de aulas em laboratório, quando couber. Além disso, estão previstas aulas de discussão onde os alunos irão identificar e discutir técnicas específicas de programação ou aspectos de linguagens de programação que afetem a qualidade do software. Por fim, um trabalho prático de programação será definido para a fixação das técnicas discutidas durante as aulas.

Método de Avaliação:

Serão realizadas até 2 (duas) provas em datas a serem fixadas. A cada prova será atribuído um grau entre 0 (zero) a 10 (dez). Serão realizados trabalhos teóricos e/ou práticos para consolidação e aplicação dos conceitos vistos em aula.

O conceito final do aluno será atribuído levando-se em consideração a sua participação nas atividades em classe e extra-classe (10%), a média dos trabalhos realizados pelo aluno (50%) e a média das provas (40%). Em particular, a média das provas não poderá ser inferior a 6,0 (seis)

RECUPERAÇÃO

Os alunos com conceito final insuficiente (menor que C) podem realizar uma Prova de Recuperação, com data e conteúdo a serem fixados pelo professor. **O aluno com nota inferior a 4,0 (quatro) em alguma das provas não poderá fazer a Prova de Recuperação.**

A recuperação de cada prova individualmente 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 pelo professor.

Bibliografia Básica:

1. MEYER, B. Object-Oriented Software Construction, 2nd Edition, 1997.
2. MEYER, B. Touch of Class - Learning to Program Well with Objects and Contracts, Springer-Verlag, 2009 .
3. McCONNELL, Steve. Code Complete: A Practical Handbook of Software Construction. Microsoft Press, 2a. edição, Julho, 2004.
4. BRAUDE, Eric. Projeto de Software, Bookman, 2005.

PLANO DE ENSINO

Bibliografia Complementar:

5. HORSTMANN, C. Object-Oriented Design & Patterns, Joyhn Wiley & Sons, 2004.
6. BECK, K. Implementation Patterns, Pearson Education, Addison-Wesley, 2008.
7. GUZDIAL, M. ; ERICSON, B. Introduction to Computing and Programming with Java: A Multimedia Approach (GOAL Series), Prentice Hall, 2006.
8. Artigos de publicações como ACM Computing Surveys, Communications of ACM, IEEE Transactions on Software Engineering, IEEE Software, Revista Brasileira de Computação, entre outras.
9. Artigos recentes publicados em Conferências como International Conference on Software Engineering (ICSE), Simpósio Brasileiro de Engenharia de Software (SBES), entre outros.

Cronograma por aula: (para o caso de mais de uma turma)

Aula	Conteúdo
1	Apresentação da disciplina. Apresentação e discussão do problema do software.
2	Qualidade de software: fundamentos, critérios e fatores de qualidade
3	Modularidade: Critérios de modularidade. Acoplamento e coesão.
4	Modularidade: Regras e princípios de modularidade.
5	TP – parte 1: análise de código
6	TP – parte 1: análise de código Enunciado do Trabalho prático
7	Projeto de software visando o reuso.
8	Decomposição de software. Tipos de dados abstratos.
9	Introdução à programação orientada a objetos: classes e objetos
10	Laboratório Java: Introdução à POO com Java #1
11	Laboratório Java: Estrutura de classes para o TP
12	Primeira verificação de aproveitamento (P1)
13	Introdução à programação orientada a objetos: herança e polimorfismo
14	Convenções e boas práticas de programação: identificadores.
14	Convenções e boas práticas de programação: tipos de dados, controle.
15	Laboratório Java: Introdução à POO com Java #2 (sobre o TP)
16	Laboratório Java: Introdução à POO com Java #3 (sobre o TP) Semana Acadêmica Semana Acadêmica
17	Laboratório Java: boas práticas de programação (sobre o TP)
18	Conceitos básicos de Teste de Software. Teste de desenvolvedor Milestone 1: entrega primeira versão do trabalho
19	Depuração de código
20	Laboratório Java: JUnit
21	Refatoração
22	Laboratório Java: Avaliação de métricas e refatoração no Eclipse
23	Laboratório Java: Teste e refatoração no TP
24	Seminário: C++/ Python / Ruby
25	Seminário: C++/ Python / Ruby
26	Segunda verificação de aproveitamento (P2).
27	Apresentação dos trabalhos práticos
28	Apresentação dos trabalhos práticos
29	Apresentação dos trabalhos práticos
30	Recuperação
-	AULA DE RESERVA

PLANO DE ENSINO

Calendário por Turma

Aula	Turma A	Turma B	Turma C
1	18/08	18/08	17/08
2	25/08	25/08	19/08
3	27/08	27/08	24/08
4	01/09	01/09	26/08
5	03/09	03/09	31/08
6	08/09	08/09	02/09
7	10/09	10/09	14/09
8	15/09	15/09	16/09
9	17/09	17/09	21/09
10	22/09	22/09	23/09
11	24/09	24/09	28/09
12	29/09	29/09	30/09
13	01/10	01/10	05/10
14	06/10	06/10	07/10
15	08/10	08/10	14/10
16	13/10	13/10	19/10
17	15/10	15/10	28/10
18	27/10	27/10	04/11
19	29/10	29/10	09/11
20	03/11	03/11	11/11
21	05/11	05/11	16/11
22	10/11	10/11	18/11
23	12/11	12/11	23/11
24	17/11	17/11	25/11
25	19/11	19/11	30/11
26	24/11	24/11	02/12
27	26/11	26/11	07/12
28	01/12	01/12	09/12
29	03/12	03/12	14/12
30	08/12	08/12	16/12
-	10/12	10/12	21/12