



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

FICHA DE DISCIPLINA

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS 1

CÓDIGO: GSI015		UNIDADE ACADÊMICA: FACULDADE DE COMPUTAÇÃO		
PERÍODO/SÉRIE: 3º		CH TOTAL TEÓRICA:	CH TOTAL PRÁTICA:	CH TOTAL:
OBRIGATÓRIA: (X)	OPTATIVA: ()	60	00	60
OBS:				
PRÉ-REQUISITOS: NÃO HÁ		CÓ-REQUISITOS: NÃO HÁ		

OBJETIVOS

Ao final do curso o aluno será capaz de:

1. Analisar problemas computacionais e propor soluções utilizando conceitos de programação orientada a objetos, como classes, objetos, herança e polimorfismo
2. Desenvolver programas em uma linguagem de programação orientada a objetos.

EMENTA

Introdução dos conceitos fundamentais de programação orientada a objetos.
 Processo de Desenvolvimento de Software; Análise e Projeto Orientado a Objetos. Notação UML.
 Técnicas para extração e reconhecimento de objetos e classes do mundo real em elementos de *software*.
 Conceitos fundamentais de orientação a objetos – Classes, objetos, operações, mensagens, métodos, estados; Tipos e classes; Polimorfismo; Identificação de objetos; Abstrações, Generalização, Sub-classes, Super Classes, Instanciação; Herança; Encapsulamento; Abstração de Agregação; Abstração de Composição, Objeto complexo, Propagação. Construtores e Destrutores; Ligação estática e dinâmica; Herança múltipla e interfaces. Uso de bibliotecas em linguagens orientadas a objetos. Aplicações e estudos de caso.

DESCRIÇÃO DO PROGRAMA

1. **Introdução à programação orientada a objetos:**
 - 1.1. Histórico da programação orientada a objetos;
 - 1.2. Programação procedimental versus programação orientada a objetos.
2. **Processo de Desenvolvimento de Software:**
 - 2.1. Análise e Projeto Orientada a Objetos;
 - 2.2. Notação UML;
 - 2.3. Técnicas para extração e reconhecimento de objetos e classes do mundo real em elementos de software.
3. **Conceitos fundamentais de programação orientada a objetos:**
 - 3.1. Classes, objetos, atributos, métodos, construtores e *destrutores*;
 - 3.2. Interação entre objetos
 - 3.3. Classes concretas, classes abstratas e interfaces;
 - 3.4. Classes auto-referenciais
 - 3.5. Membros de classe e membros de objetos;
 - 3.6. Métodos concretos e métodos abstratos;
 - 3.7. Pacotes, visibilidade e encapsulamento;
 - 3.8. Generalização, especialização, herança e métodos finais;
 - 3.9. Herança múltipla e interfaces;
 - 3.10. Sobrecarregamento e sobrescrita de métodos
 - 3.11. Polimorfismo;
 - 3.12. Ligação estática e ligação dinâmica.
 - 3.13. Composição, agregação, associação, herança x composição
4. **Uso de bibliotecas em linguagens orientadas a objetos:**
 - 4.1. Uso de API básica (e.g., *java.lang* e *java.util*);
 - 4.2. Tratamento de exceções ;
 - 4.3. Strings, caracteres e expressões regulares (e.g. *classes pattern e matcher*)
 - 4.4. Arquivos, fluxos e serialização de objetos (e.g., *java.io*);
 - 4.5. Interfaces gráficas (e.g., *java.awt* e *javax.swing*);
5. **Desenvolvimento de um sistema orientado a objetos:**
 - 5.1. Estudo de caso;
 - 5.2. Implementação usando programação orientada a objetos;
 - 5.3. Documentação do sistema desenvolvido.

BIBLIOGRAFIA

Básica

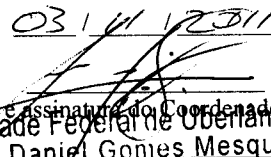
BARNES, D.J., KOLLING, M. Programação Orientada a Objetos com Java.4a ed São Paulo:Pearson Prentice Hall, 2009.
ARNOLD, Ken; GOSLING, James; HOLMES, David. **A Linguagem de Programação Java**, 4ed. Bookman, 2007.
FLANAGAN, David. **Java: o guia essencial**, 5ed. Porto Alegre: Bookman, 2006.

Complementar

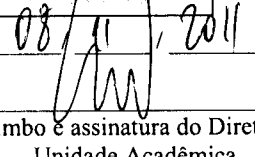
HORSTMANN, Cay S. **Big Java**. Porto Alegre: Bookman, 2004.
HUBBARD, John R. **Teoria e Problemas da Programação com Java**, 2ed. Porto Alegre: Bookman, 2006.
DEITEL, Harvey M. **Java: Como Programar**, 8ed. São Paulo: Prentice Hall, 2010.
DEITEL, Harvey M. **Advanced java 2 platform: how to program**. Upper Sadler River: Prentice Hall, 2002.

SANTOS, Rafael. **Introdução à Programação Orientada a Objetos Usando Java**. Editora Campus, 2003.
HORSTMANN, Cay S. **Core Java 2**. São Paulo: Pearson Education, 2001.
BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.
LARMAN, C. **Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos**. Porto Alegre: Bookmann, 2001.
FOWLER, M. Scott, K. **UML Essencial**. POA: Bookmann, 2000.
ECKEL B.: **Thinking in Java**, Prentice Hall, Segunda Edição, 2000.
GOODRICH, Michael T.; TAMASSIA, Roberto. **Estrutura de Dados e Algoritmos em Java**, 4ed. Bookman, 2007.

APROVAÇÃO

03/11/2011


Carimbo e assinatura do Coordenador do curso
Universidade Federal de Uberlândia
Prof. Dr. Daniel Gomes Mesquita
Coordenador do Curso de Sistemas de Informação
Portaria R Nº. 1257/10

08/11/2011


Universidade Federal de Uberlândia
Prof. Ilmério Reis da Silva
Diretor da Faculdade de Computação
Portaria R Nº. 757/11