

PLANO DE ENSINO

1. Identificação da Disciplina

Código: 116785	Nome: Programação Orientada a Objetos	Turma: B	Período: 2014/2
Professor(es) Responsável(eis): Vander Ramos Alves			
Horário e Local: Terças e quintas, das 10:00 às 11:50, no PAT AT 029			

2. Ementa

Introdução; princípios de orientação a objetos; classes e objetos; herança; interfaces; polimorfismo; padrões de projeto.

3. Objetivos

- Estudar os principais conceitos presentes na orientação a objetos.
- Estudar as principais técnicas presentes na programação orientada a objetos.
- Aprender uma linguagem de programação orientada a objetos (**Java**).
- Fixar os conhecimentos através de exercícios práticos.

4. Programa

1. Introdução.
2. Classes (Conceito, Bibliotecas de classes, Atributos, Métodos, Visibilidades de atributos e métodos, Métodos abstratos, Classes abstratas, Interfaces)
3. Objetos (Instanciando classes, Mensagens, Interações entre objetos, Estados, eventos e exceções, Polimorfismo, Ligação estática e ligação dinâmica)
4. Herança (generalização e especialização, herança simples e múltipla, polimorfismo de inclusão, coerção, hierarquias de classes, polimorfismo paramétrico)
5. Arquitetura de software (importância da arquitetura, decomposição de um sistema de software, estilos arquiteturais, padrões de projeto, projeto e programação)
6. Persistência

5. Bibliografia Básica e Complementar

- Harvey M. Deitel , Paul J. Deitel. *Java How to Program: Early Objects Version* (9th edition), Prentice Hall, 2011. **(bastante didático)**
- James Gosling, Bill Joy, Guy Steele, Gilad Bracha. *The Java Language Specification*, 2005. <http://java.sun.com/docs/books/jls/> **(detalhes da linguagem Java)**
- MEYER, B. *Object-Oriented Software Construction*, 2nd Edition, Prentice Hall, 2000. **(conceitos centrais de orientação a objetos)**
- GAMMA, E. et al. *Design Patterns: Elements of Reusable Object Oriented Software*. Addison Wesley, 1994. **(soluções gerais para problemas recorrentes em programas OO)**
- BUSCHMANN, F. et al. *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*. Wiley, 1996. **(estilos arquiteturais para estruturar aplicações)**

5. Metodologia

Atividades Teóricas:	Aulas expositivas com slides; estudo dirigido no Moodle
Atividades Práticas:	Exercícios em sala de aula e em casa.

6. Critérios de Avaliação

Do Aluno:	<p>Forma de avaliação: duas provas, cada uma valendo 10 pontos, e dois trabalhos, cada um valendo 10 pontos.</p> <p>Cálculo da média: $MP \cdot 0,6 + MT \cdot 0,4$ onde</p> <p>$MP = (Nota Prova 1 + Nota Prova 2) / 2$ e</p> <p>$MT = (Nota Trabalho1 + Nota Trabalho2) / 2$</p> <p>Critério de aprovação: o aluno será aprovado se, e somente se, todas as seguintes três condições forem satisfeitas:</p> <ul style="list-style-type: none">• $MP \geq 5$;• $MT \geq 5$;• frequência às aulas ser igual ou superior a 75% do total de aulas.
Da Disciplina:	Avaliação anônima conduzida pelo DEG no MatriculaWeb

7. Observações

- **A doutoranda do Programa de Pós-graduação em Informática da UnB, Giselle Machado, sob minha supervisão, está cursando a disciplina Estágio Docência e, conforme previsto em regimento da UnB, ministrará algumas aulas nesta disciplina e conduzirá algumas atividades de avaliação.**
- Cópia ou tentativa de cópia de prova ou trabalho implica nota zero para quem copiou e para quem cedeu cópia.
- Moodle: <http://aprender.unb.br/course/view.php?id=3669> Senha: 116785
- Datas de provas e de entrega de trabalhos serão comunicadas com pelo menos duas semanas de antecedência.

Brasília _11___/_08___/2014

Prof. Vander Ramos Alves