

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Instituição: **Universidade Federal da Fronteira Sul**
Curso: **Bacharelado em Ciência da Computação**
Disciplina: **Programação I**
Fase: **3ª (noturno)**
Créditos: **04**
Carga Horária: **72 H/A**
Período Letivo: **2011/2**
Professor (a): **Andressa Sebben e Fernando Bevilacqua**
Horário de Atendimento: **Segundas-feiras das 18h00 às 21h00**

2. OBJETIVO DO CURSO

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3. EMENTA

Conceitos de programação orientada a objetos. Classes. Herança. Encapsulamento. Polimorfismo. Associações. Reusabilidade de software. Componentes. Criação e uso de bibliotecas de classes. Interface gráfica com o usuário. Persistência de dados e de objetos. Tratamento de exceções e erros. Aspectos de projeto orientado a objetos. Prática de programação usando uma linguagem de programação orientada a objetos.

4. JUSTIFICATIVA

O paradigma de programação orientado a objetos desempenha papel fundamental no desenvolvimento de softwares simples e complexos. A capacitação do aluno nesse novo conceito de programação busca incrementar e aprimorar suas técnicas de resolução de problemas e modelagem de programas.

5. OBJETIVOS:

5.1. GERAL:

Compreender os conceitos fundamentais do paradigma de programação orientada a objetos e aplicá-los no desenvolvimento de soluções de software.

5.2. ESPECÍFICOS:

- Conhecer os conceitos do paradigma de programação orientada a objetos;
- Instalar e configurar o ambiente de desenvolvimento para a linguagem Java;
- Projetar, codificar, testar e depurar programas utilizando orientação a objetos em Java.

6. CONTEÚDO PROGRAMÁTICO

ENCONTRO	CH	CONTEÚDO
Aula 1	2	Introdução Apresentação da disciplina; paradigmas de programação.

Aula 2	2	Introdução à linguagem Java Instalação e configuração do ambiente de desenvolvimento; Características; Aspectos de funcionamento.
Aula 3	2	Sintaxe básica Estrutura de um programa; Variáveis; Tipos de dados; Operadores; Comandos de entrada e saída
Aula 4	2	Sintaxe básica Estruturas de decisão; Estruturas de repetição; Controle de fluxo; Arrays; Lista de exercícios.
Aula 5	2	Sintaxe básica Correção de exercícios
Aula 6	2	Noções de Programação orientada a objetos Representação gráfica de classes e objetos. Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação)
Aula 7	2	Noções de Programação orientada a objetos Objetos e instanciação, representação na memória, métodos e propriedades.
Aula 8	2	Implementação de classes Classes e objetos; Atributos; Métodos; Mensagens.
Aula 9	2	Implementação de classes Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final
Aula 10	2	Encapsulamento Modificadores de visibilidade; Métodos de acesso e modificadores
Aula 11	2	Encapsulamento Exercícios
Aula 12	2	Herança Conceito e utilização; Construtores e propriedades herdadas.
Aula 13	2	Herança Especialização; Sobreposição de métodos; Utilização de super.
Aula 14	2	Herança Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
Aula 15	2	Associações Criação de classes compostas por outras classes
Aula 16	2	Associações Criação de getter/setter para propriedades associadas.
Aula 17	2	Revisão
Aula 18	2	Avaliação 1 (P1)
Aula 19	2	Feedback

Aula 20	2	Classes abstratas Conceito; Classes abstratas; Métodos abstratos
Aula 21	2	Interfaces Conceito; Uso de interfaces
Aula 22	2	Polimorfismo Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica
Aula 23	2	Polimorfismo Casting; Chamada de um mesmo método em classes diferentes; Operador instanceof
Aula 24	2	Polimorfismo Aprofundamento e utilização de arrays com classes pai e filho misturadas.
Aula 25	2	Pacotes Definição; Organização de classes em pacotes;
Aula 26	2	Pacotes Organização de subpacotes; Empacotamento em arquivos JAR.
Aula 27	2	Exceções Tratamento de exceções
Aula 28	2	Exceções Tratamento de exceções
Aula 29	2	Exceções Tratamento de exceções
Aula 30	2	Interface gráfica Componentes do Swing
Aula 31	2	Interface gráfica Gerenciadores de Layout
Aula 32	2	Persistência Serialização e deserialização de objetos
Aula 33	2	Persistência Leitura e gravação de arquivos de texto
Aula 34	2	Trabalho final
Aula 35	2	Trabalho final
Aula 36	2	Avaliação 2 (P2)

7. AVALIAÇÃO

As avaliações serão agrupadas em dois momentos (conforme instrução normativa No. 001/Prograd/2010): Notas Parciais 1 e 2 (NP₁ e NP₂, respectivamente). A média final do semestre será a média aritmética simples entre as notas NP₁ e NP₂. A média final mínima para aprovação é 6,0 (seis). A frequência mínima para aprovação é de 75% (setenta e cinco por cento).

A NP₁ será composta por uma avaliação escrita (P₁) e trabalhos extra-classe (G₁) realizados até a data da prova, com o seguinte cálculo:

$$NP_1 = P_1 * 0,6 + G_1 * 0,4$$

sendo G_1 calculado da seguinte forma:

$$G_1 = (T_1 + T_2 + \dots T_n) / n$$

onde T_i representa a nota de um trabalho, variando de 0 (zero) até 1.

A NP_2 será composta por uma avaliação escrita (P_2) e trabalhos extra-classe (G_2) realizados até a data da prova, com o seguinte cálculo:

$$NP_2 = P_2 * 0,6 + G_2 * 0,4$$

sendo G_2 calculado da seguinte forma:

$$G_2 = (T_1 + T_2 + \dots T_n) / n$$

onde T_i representa a nota de um trabalho, variando de 0 (zero) até 1.

A média final (MF) será calculada como

$$MF = (NP_1 + NP_2) / 2$$

Para cada NP será ofertada prova de recuperação (RP). A reposição de nota se aplica somente à prova, não substituindo os trabalhos. Além disso, RP não substitui P, mas é calculada uma média entre RP e P. Dessa forma, para os alunos que prestarem RP, o cálculo de NP é definido por:

$$NP = ((P + RP) / 2) * 0,6 + G * 0,4$$

Durante os 5 minutos iniciais de RP, o aluno terá a oportunidade de avaliar a prova e decidir entre prestar ou não a mesma. Para os que decidirem por não prestar RP, o cálculo de NP não é alterado.

Em caso de plágio as seguintes regras serão aplicadas.

Prova:

–O aluno recebe nota zero na prova onde o plágio foi detectado

Trabalhos:

–É permitido usar conteúdo da internet, livros, colegas, etc., contanto que uma citação seja feita. A nota do trabalho será proporcional ao conteúdo original.

–Caso seja detectado plágio o aluno recebe zero no trabalho em questão, além disso, por demonstrar prática não aceitável o caso será levado ao conhecimento do colegiado.

O formato dos instrumentos de avaliação será definido pelo professor no decorrer do processo de ensino-aprendizagem, tendo em vista o caráter processual da avaliação. Os mesmos poderão ser realizados na forma de avaliações escritas, práticas em laboratório, trabalho individual ou em grupo.

As notas serão divulgadas em até no máximo 10 dias após a realização da avaliação. As avaliações corrigidas serão entregues aos alunos e os resultados serão analisados e discutidos de forma coletiva.

Em relação à avaliação dos trabalhos, os seguintes elementos serão levados em consideração:

- Funcionamento correto (o programa precisa cumprir seu objetivo conforme a descrição do trabalho);
- Legibilidade do código (nomes de classes com a primeira letra maiúscula, métodos e propriedades no formado nomeFormaCamelo, **indentação correta**, etc);
- Comentários (o código fonte deve conter um bloco de comentário no começo informando o propósito do programa e o nome/email do seu autor).
- Haverá um desconto de 50% da nota do trabalho por dia de atraso na entrega, com prazo máximo de 3 dias de atraso;
- Programas que não compilarem receberão nota **zero** instantânea (nenhuma avaliação será realizada).

Os demais aspectos referentes à avaliação seguirão as normas vigentes na UFFS.

OBSERVAÇÕES GERAIS

O atendimento extraclasse aos alunos será realizado nas segundas-feiras, das 18h00 às 21h00, na sala 03 de professores ou no laboratório de informática.

O cronograma de aulas poderá sofrer alterações conforme a disponibilidade dos recursos necessários.

8. REFERÊNCIAS

8.1.BÁSICAS:

1. SANTOS, Rafael. **Introdução à programação orientada a objetos usando Java**. 8.reimp. Rio de Janeiro: Elsevier, 2003.
2. DEITEL, Harvey M.; DEITEL, Paul J. **Java Como Programar**. 8.ed. Pearson, 2010.
3. BORATTI, Isaias Camilo. **Programação orientada a objetos em Java**. Florianópolis: Visual Books, 2007.
4. GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, Servlets, Java Server Faces, Hibernate, EJB 3 Persistence e Ajax**. 1.ed. Ciência Moderna, 2007.
5. CORNELL, G., HORSTMANN, C. S. **Core Java, V.1 – Fundamentos**. São Paulo: Prentice Hall Brasil, 2009.

8.2.COMPLEMENTAR:

1. ECKEL, Bruce. **Thinking in Java**. Prentice-Hall, 2000.
2. LEWIS, J., LOFTUS, W.. **Java Software Solutions - Foundations of Program Design**. Addison-Wesley, 1999.
3. KEOGH, Jim; GRANNINI, Mario. **OOP Desmistificado – Programação Orientada a Objetos**. Alta Books, 2005.
4. HEMRAJANI, Anil. **Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse**. 1.ed. Pearson, 2007.
5. LARMAN, C. **Utilizando UML e Padrões: uma Introdução à Análise e ao Projeto Orientados a Objetos**. São Paulo: Bokkman Companhia, 3a. ed, 2007
6. CARDOSO, C. **Orientação a Objetos na Prática**. Rio de Janeiro: Ciência Moderna, 2006.
7. SIERRA, Kathy; BATES, Bert. **Use a Cabeça! Java**. 2. ed. Rio de Janeiro: Alta Books, 2007.
8. MENDES, Douglas Rocha. **Programação Java com Ênfase em Orientação a Objetos**. São Paulo: Novatec, 2009.