Ministério da Educação
Universidade Federal da Fronteira Sul

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: **2012/2**
Professor (a): **Fernando Bevilacqua**
Horário de Atendimento: **Terça-feira das 18h00 às 19h00**

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; Estruturas de decisão; Estruturas de repetição; Controle de fluxo; Arrays
Aula 4	**2**	**Noções de Programação orientada a objetos** Representação gráfica de classes e objetos. Operações de abstração (classificação/instanciação, generalização/ especialização, agregação, associação)
Aula 5	**2**	**Noções de Programação orientada a objetos** Objetos e instanciação, representação na memória, métodos e propriedades.
Aula 6	**2**	**Implementação de classes** Classes e objetos; Atributos; Métodos; Mensagens.
Aula 7	**2**	**Implementação de classes** Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final
Aula 8	**2**	**Encapsulamento** Modificadores de visibilidade; Métodos de acesso e modificadores
Aula 9	**2**	**Encapsulamento** Exercícios
Aula 10	**2**	**Herança** Conceito e utilização; Construtores e propriedades herdadas.
Aula 11	**2**	**Herança** Especialização; Sobreposição de métodos; Utilização de super.
Aula 12	**2**	**Herança** Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
Aula 13	**2**	**Associações** Criação de classes compostas por outras classes
Aula 14	**2**	**Associações** Criação de getter/setter para propriedades associadas.
Aula 15	**2**	Exercícios
Aula 16	**2**	Revisão
Aula 17	**2**	**Avaliação 1 (P1)**
Aula 18	**2**	Correção da prova e discussões.
Aula 19	**2**	**Classes abstratas** Conceito; Classes abstratas; Métodos abstratos

Aula 20	2	**Recuperação 1 (R1)**
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**	**Pacotes** Definição; Organização de classes em pacotes;
Aula 25	**2**	**Pacotes** Organização de subpacotes; Empacotamento em arquivos JAR.
Aula 26	**2**	**Exceções** Tratamento de exceções
Aula 27	**2**	**Exceções** Tratamento de exceções
Aula 28	**2**	**Interface gráfica** Componentes do Swing
Aula 29	**2**	**Interface gráfica** Gerenciadores de Layout
Aula 30	**2**	**Persistência** Leitura e gravação de arquivos de texto; Serialização e deserialização de objetos
Aula 31	2	Revisão
Aula 32	**2**	**Avaliação 2 (P2)**
Aula 33	2	Correção da prova e discussões.
Aula 34	2	**Trabalho final**
Aula 35	2	**Trabalho final**
Aula 36	**2**	**Recuperação 2 (R2)**

7. AVALIAÇÃO

Uso de abordagens tais como: avaliações teóricas e práticas, exercícios extra-classe, trabalhos de implementação.

As avaliações serão agrupadas em dois momentos (conforme instrução normativa No. 001/Prograd/ 2010) Notas Parciais 1 e 2 (NP1 e NP2, respectivamente). A **NP1** será composta por uma avaliação escrita (**P1**) , trabalhos (**G1**) realizados até a data da prova e um coeficiente bonus (**C1**), conforme o seguinte cálculo:

$$NP1 = P1*0,75 + G1*0,25 + C1$$

sendo **G1** calculado da seguinte forma:

$$G1 = (T1 + T2 + \ldots Tn) / n$$

onde **Ti** representa a nota de um trabalho, variando de 0 (zero) a 10, e **n** representa o número de trabalhos solicitados. O coeficiente bônus **C1** pode ser positivo (acréscimo de nota) ou negativo (decréscimo de nota).

A **NP2** será composta por uma avaliação escrita (**P2**) , trabalhos (**G2**) realizados até a data da prova e um coeficiente bonus (**C2**), conforme o seguinte cálculo:

NP2 = P2*0,6 + G2*0,4 + C2

sendo **G2** calculado da seguinte forma:

G2 = (T1 * K1 + T2 * K2 + … Tn * Kn) / n

onde **Ti** representa a nota de um trabalho, variando de 0 (zero) até 10, **Ki** representa o peso do trabalho em questão, variando de 1 (um) até 50 (cinquenta), e n representa a soma de todos os valores **Ki**, ou seja, **n = K1 + K2 + … Kn**. O coeficiente bônus **C2** pode ser positivo (acréscimo de nota) ou negativo (decréscimo de nota).

A média final (**MF**) será calculada como:

MF = (NP1 + NP2)/2

Para cada NP será ofertada prova de recuperação (**RP**) em horário extra-classe. A reposição de nota se aplica somente à prova, não substituindo os trabalhos. Além disso, RP **não substitui** P no mesmo teor, mas sim compõe uma média com **P**. Dessa forma, para os alunos que prestarem RP o cálculo da NP em questão é definido substituindo-se a nota da antiga prova (**Pi**) pela ***média*** entre **RPi** e **Pi**, da seguinte forma:

Pi = (Pi + RPi)/2

Os demais elementos que compõem a NP em questão permanecem com seus valores sem alteracões decorrentes da RP. 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 relação à correção detrabalhos e provas:

- Em caso de plágio e/ou cola, todos os alunos envolvidos recebem nota zero.

- Para os trabalhos, o uso de conteúdo da Internet, livros, colegas, etc. é permitido desde que a fonte seja citada. Contudo, a nota do trabalho será proporcional ao conteúdo original.

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 terças-feiras, das 18h00 às 19h00, na sala 03 de professores..

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 Introduçao 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.