



Ministério da Educação  
Universidade Federal da Fronteira Sul

## PLANO DE ENSINO

### 1. Dados de Identificação

Curso: Ciência da Computação (Turno: Matutino)

Componente Curricular: GEX098 - Programação I

Turma: 13220

Fase: 3ª

Ano/Semestre: 2016/1

Numero de Créditos: 4

Carga horária Hora Aula: 72

Carga horária Hora Relógio: 60

Professora: Márcia Boell

Aulas: Quintas-feiras 7:30 e sextas-feiras 10:10

Atendimento ao aluno: quinta feira das 10:00h às 11:50h e sextas-feiras 7:30 às 9:50

### 2. Objetivo Geral 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. Objetivo

#### 4.1 Geral

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

#### 4.2 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.

## 5. Cronograma e Conteúdo Programático

Encontro	Conteúdo
1	<b>Introdução</b> Apresentação da disciplina; paradigmas de programação orientada a objetos.
2	<b>Noções de Programação orientada a objetos</b> Objetos e instanciação, representação na memória, métodos e propriedades.
3	<b>Noções de Programação orientada a objetos</b> Representação gráfica de classes e objetos; Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação).
4	Exemplos da diferença entre Programação Procedimental x Programação Orientada a Objetos.
5	<b>Introdução à linguagem Java</b> Instalação e configuração do ambiente de desenvolvimento; Características. Aspectos de funcionamento. <b>Sintaxe básica:</b> Estrutura de um programa; Variáveis; Tipos de dados; Operadores; Comandos de entrada e saída; Estruturas de decisão; Estruturas de repetição; Controle de fluxo; Arrays.
6	<b>Implementação de classes</b> Classes e objetos; Atributos; Métodos; Mensagens.
7	<b>Implementação de classes</b> Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final.
8	<b>Encapsulamento</b> Modificadores de visibilidade; Métodos de acesso e modificadores.
9	<b>Herança</b> Conceito e utilização; Construtores e propriedades herdadas.
10	<b>Herança</b> Especialização; Sobreposição de métodos; Utilização de super.
11	Apresentação de trabalhos, entrega de lista de exercícios.
12	<b>Herança</b> Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
13	<b>Associações</b> Criação de classes compostas por outras classes; <b>Associações</b> Criação de getter/setter para propriedades associadas.
14	<b>Interface gráfica</b> Aplicações gráficas em java
15	Apresentação de trabalhos, entrega de lista de exercícios.
16	Revisão.
17	<b>Avaliação 1 (Todo o conteúdo trabalhado até o dia da prova)</b>
18	Discussões.

	<b>Classes abstratas</b> Conceito; Classes abstratas; Métodos abstratos.
19	<b>Interfaces</b> Conceito; Uso de interfaces.
20	<b>Interfaces</b> Conceito; Uso de interfaces.
21	<b>Polimorfismo</b> Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica.
22	<b>Polimorfismo</b> Chamada de um mesmo método em classes diferentes.
23	<b>Persistência</b> Java e Bancos de Dados – JDBC; Hibernate.
24	<b>Pacotes</b> Definição; Organização de classes em pacotes; Organização de subpacotes; Empacotamento em arquivos JAR.
25	<b>Exceções</b> Tratamento de exceções.
26	Atividade prática
27	<b>Avaliação de recuperação</b>
28	<b>RMI- Remote Method Invocation</b> (Invocação de Métodos Remotos)
29	Atividade prática
30	Entrega de atividade
<b>31</b>	<b>Avaliação 2</b>
32	Discussão sobre avaliação 2; Entrega de listas de exercícios

\* O plano e cronograma podem ser alterados pelo professor ao longo do semestre.

## 6. Procedimentos Metodológicos (estratégias de ensino, equipamentos, entre outros)

Conduzir a disciplina com aulas expositivas/dialogadas enquanto discutidos os itens de cunho teórico, evoluindo em tópicos específicos para exercícios práticos, demonstrações, contextualização baseada em publicações atualizadas. Uso de atividades em laboratórios com o objetivo de apresentar/exercitar os conceitos estudados.

O plágio e a cola serão tratados de forma rígida (nota 0 para os envolvidos na atividade em questão).

Não será permitido o uso de equipamentos eletrônicos (computadores, smartphones, tablets, etc) que não sejam explicitamente para uso na disciplina. O não atendimento a esta regra será passível de punição para toda a turma.

## 7. Avaliação do Processo Ensino Aprendizagem

Uso de abordagens tais como: avaliações teóricas e práticas, exercícios extraclasse, trabalhos de implementação.

As avaliações serão agrupadas em dois momentos: 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 bônus (C1), conforme o seguinte cálculo:  $NP1 = P1 \cdot 0,7 + G1 \cdot 0,3 + C1$  sendo G1 calculado da seguinte forma:  $G1 = (T1 + T2 + \dots + Tn) / n$  onde T1 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 bônus (C2), conforme o seguinte cálculo:  $NP2 = P2 \cdot 0,5 + G2 \cdot 0,5 + C2$  sendo G2 calculado da seguinte forma:

$G2 = (T1 \cdot K1 + T2 \cdot K2 + \dots + Tn \cdot Kn) / n$  onde T1 representa a nota de um trabalho, variando de 0 (zero) até 10, K1 representa o peso do trabalho em questão, variando de 1 (um) até 50 (cinquenta), e n representa a soma de todos os valores K1, ou seja,  $n = K1 + K2 + \dots + Kn$ .

O coeficiente bônus C2 pode ser positivo (acréscimo de nota) ou negativo (decréscimo de nota).

A média final (MF) será calculada como:  $MF = (NP1 + NP2) / 2$ .

O aluno que obtiver MF maior ou igual a 6,0 estará aprovado na disciplina.

Em relação à correção de trabalhos 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, entre outros meios, 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 nome Forma Camelo, identificação 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, assim como também constar a explicação das principais funcionalidades do código).
- Haverá um desconto de 50% da nota do trabalho na entrega, com prazo máximo de três dias de atraso;
- Programas que não compilarem receberá nota **zero** instantânea (nenhuma avaliação será realizada).

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

## 7.1 Recuperação: novas oportunidades de aprendizagem e avaliação

As avaliações e trabalhos serão discutidos em sala de aula. Esta discussão tem como objetivo oferecer uma nova oportunidade de aprendizagem do conteúdo avaliado. Todos os estudantes participarão da discussão.

Estudante que obter **MF < 6,0** será oferecida uma oportunidade, através de uma prova exame **PR** com todo o conteúdo, que será utilizada para calcular a nova média final como: **MF=(MF + PR)/2**

Após a realização do exame, a nota final do aluno passará a ser **MF**.

Se **MF** for maior ou igual a **6,0**, o aluno será aprovado na disciplina, caso contrário será reprovado.

## **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 Complementares**

1. ECKEL, Bruce. **Thinking in Java**. PrenticeHall, 2000.

2. LEWIS, J., LOFTUS, W.. **Java Software Solutions Foundations of Program Design**. AddisonWesley, 1999.

3. KEOGH, Jim; GRANNINI, Mario. **OOP Desmistificado – Programação Orientada a Objetos**. Alta Books, 2005.

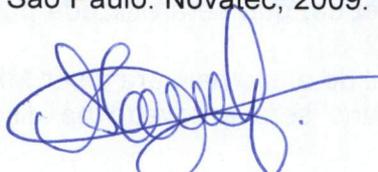
4. HEMRAJANI, Anil. **Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse**. 1.ed. Pearson, 2007.

5. LARMAN, C. **Utilizando UML e Padrões: uma Introdução à Análise e ao Projeto Orientados a Objetos**. São Paulo: Bokkman Companhia, 3a. ed, 2007.

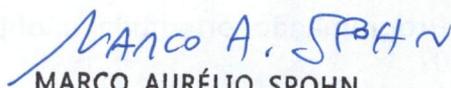
6. CARDOSO, C. **Orientação a Objetos na Prática**. Rio de Janeiro: Ciência Moderna, 2006.

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.



professora



**MARCO AURÉLIO SPOHN**  
Siape nº.1521671  
Coord. do Curso de Ciência da Computação  
Universidade Federal da Fronteira Sul-UFFS  
Campus Chapecó-SC