



## PLANO DE ENSINO

### 1. Dados de Identificação

Curso: Ciência da Computação (Turno: Noturno)  
Componente Curricular: Programação I  
Fase: Terceira  
Ano/Semestre: 2014/2  
Numero de Créditos: 4  
Carga horária Hora Aula: 72  
Carga horária Hora Relógio: 60  
Professor: Fernando Bevilacqua  
Atendimento ao aluno: sexta-feira das 18:30h às 21:00h

### 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 | Aulas | Total Parc. | CONTEÚDO  |
|----------|-------|-------------|---|
| Aula 1   | 2     | 2           | <b>Introdução</b><br>Apresentação da disciplina; paradigmas de programação.   |
| Aula 2   | 2     | 4           | <b>Introdução à linguagem Java</b><br>Instalação e configuração do ambiente de desenvolvimento; Características; Aspectos de funcionamento.   |
| Aula 3   | 2     | 6           | <b>Sintaxe básica</b><br>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     | 8           | <b>Noções de Programação orientada a objetos</b><br>Representação gráfica de classes e objetos.<br>Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação) |
| Aula 5   | 2     | 10          | <b>Noções de Programação orientada a objetos</b><br>Objetos e instanciação, representação na memória, métodos e propriedades.   |
| Aula 6   | 2     | 12          | <b>Implementação de classes</b><br>Classes e objetos; Atributos; Métodos; Mensagens.  |
| Aula 7   | 2     | 14          | <b>Implementação de classes</b><br>Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final   |
| Aula 8   | 2     | 16          | <b>Encapsulamento</b><br>Modificadores de visibilidade; Métodos de acesso e modificadores   |
| Aula 9   | 2     | 18          | <b>Encapsulamento</b><br>Exercícios   |
| Aula 10  | 2     | 20          | <b>Herança</b><br>Conceito e utilização; Construtores e propriedades herdadas.  |
| Aula 11  | 2     | 22          | <b>Herança</b><br>Especialização; Sobreposição de métodos; Utilização de super.   |
| Aula 12  | 2     | 24          | <b>Herança</b><br>Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.  |
| Aula 13  | 2     | 26          | <b>Associações</b><br>Criação de classes compostas por outras classes   |
| Aula 14  | 2     | 28          | <b>Associações</b><br>Criação de getter/setter para propriedades associadas.  |
| Aula 15  | 2     | 30          | Exercícios  |
| Aula 16  | 2     | 32          | Revisão   |
| Aula 17  | 2     | 36          | <b>Avaliação 1 (P1)</b>   |
| Aula 18  | 2     | 38          | Correção da prova e discussões.   |
| Aula 19  | 2     | 40          | <b>Classes abstratas</b><br>Conceito; Classes abstratas; Métodos abstratos  |
| Aula 20  | 2     | 42          | <b>Classes abstratas</b>  |



|                |          |           |  |
|----------------|----------|-----------|--|
|                |          |           | Conceito; Classes abstratas; Métodos abstratos   |
| <b>Aula 21</b> | <b>2</b> | <b>44</b> | <b>Interfaces</b><br>Conceito; Uso de interfaces   |
| <b>Aula 22</b> | <b>2</b> | <b>46</b> | <b>Polimorfismo</b><br>Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica               |
| <b>Aula 23</b> | <b>2</b> | <b>48</b> | <b>Polimorfismo</b><br>Casting; Chamada de um mesmo método em classes diferentes; Operador instanceof    |
| <b>Aula 24</b> | <b>2</b> | <b>50</b> | <b>Pacotes</b><br>Definição; Organização de classes em pacotes;  |
| <b>Aula 25</b> | <b>2</b> | <b>52</b> | <b>Pacotes</b><br>Organização de subpacotes; Empacotamento em arquivos JAR.                              |
| <b>Aula 26</b> | <b>2</b> | <b>54</b> | <b>Exceções</b><br>Tratamento de exceções  |
| <b>Aula 27</b> | <b>2</b> | <b>56</b> | <b>Exceções</b><br>Tratamento de exceções  |
| <b>Aula 28</b> | <b>2</b> | <b>58</b> | <b>Interface gráfica</b><br>Componentes do Swing   |
| <b>Aula 29</b> | <b>2</b> | <b>60</b> | <b>Interface gráfica</b><br>Gerenciadores de Layout  |
| <b>Aula 30</b> | <b>2</b> | <b>62</b> | <b>Persistência</b><br>Leitura e gravação de arquivos de texto; Serialização e deserialização de objetos |
| <b>Aula 31</b> | <b>2</b> | <b>64</b> | Revisão  |
| <b>Aula 32</b> | <b>2</b> | <b>66</b> | <b>Avaliação 2 (P2)</b>  |
| <b>Aula 33</b> | <b>2</b> | <b>68</b> | Correção da prova e discussões.  |
| <b>Aula 34</b> | <b>2</b> | <b>70</b> | <b>Trabalho final</b>  |
| <b>Aula 35</b> | <b>2</b> | <b>72</b> | <b>Exame</b>   |

\* 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 extra-classe, 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 bonus (**C1**), conforme o seguinte cálculo:

$$\mathbf{NP1 = P1*0,6 + G1*0,4 + C1}$$

sendo **G1** calculado da seguinte forma:

$$\mathbf{G1 = (T1 + T2 + \dots Tn) / n}$$

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:

$$\mathbf{NP2 = P2*0,5 + G2*0,5 + C2}$$

sendo **G2** calculado da seguinte forma:

$$\mathbf{G2 = (T1 * K1 + T2 * K2 + \dots Tn * Kn) / n}$$

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,  $n = K1 + K2 + \dots Kn$ . O coeficiente bônus **C2** pode ser positivo (acréscimo de nota) ou negativo (decréscimo de nota).

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

$$\mathbf{MF = (NP1 + NP2)/2}$$

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, 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.

## 7.1 Recuperação: novas oportunidades de aprendizagem e avaliação

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

Se no fim do semestre algum estudante 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**. 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: Borkman Companhia, 3a. ed, 2007
6. CARDOSO, C. **Orientação a Objetos na Prática**. Rio de Janeiro: Ciência Moderna, 2006.



Ministério da Educação  
Universidade Federal da Fronteira Sul

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.