



## Plano de Ensino

### 1. DADOS DE IDENTIFICAÇÃO

Curso: Ciência da Computação

Componente curricular: GEX098 - Programação I

Fase: 3ª

Ano/semestre: 2016.2

Número da turma: 16052

Número de créditos: 4

Carga horária – Hora aula: 72

Carga horária – Hora relógio: 60

Professores: Andressa Sebben

Atendimento ao Aluno: Segunda-feira das 18h40 às 19h00 e Quarta-feira das 20h30 às 21h00

### 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

<b>Encontro</b>	<b>Conteúdo</b>
1 01/08	<b>Introdução</b> Apresentação da disciplina; paradigmas de programação.
2 03/08	<b>Introdução à linguagem Java</b> Instalação e configuração do ambiente de desenvolvimento; Características. Aspectos de funcionamento.
3 08/08	<b>Sintaxe básica</b> Estrutura de um programa; Variáveis; Tipos de dados; Operadores; Comandos de entrada e saída
4 10/08	<b>Sintaxe básica</b> Estruturas de decisão; Estruturas de repetição; Controle de fluxo; Arrays; Lista de exercícios.
5 15/08	<b>Sintaxe básica</b> Correção de exercícios
6 17/08	<b>Noções de Programação orientada a objetos</b> Representação gráfica de classes e objetos. Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação)
7 22/08	<b>Noções de Programação orientada a objetos</b> Objetos e instanciação, representação na memória, métodos e propriedades.
8 24/08	<b>Implementação de classes</b> Classes e objetos; Atributos; Métodos; Mensagens.
9 29/08	<b>Implementação de classes</b> Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final
10 31/08	<b>Encapsulamento</b> Modificadores de visibilidade; Métodos de acesso e modificadores
11 05/09	<b>Encapsulamento</b> Exercícios
12 12/09	<b>Herança</b> Conceito e utilização; Construtores e propriedades herdadas.
13 14/09	<b>Herança</b> Especialização; Sobreposição de métodos; Utilização de super.

14 19/09	<b>Herança</b> Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
15 21/09	<b>Associações</b> Criação de classes compostas por outras classes.
16 26/09	<b>Associações</b> Criação de getter/setter para propriedades associadas.
17 28/09	Revisão
18 03/10	<b>Avaliação 1 (P1)</b>
19 05/10	<b>Feedback</b>
20 10/10	<b>Classes abstratas</b> Conceito; Classes abstratas; Métodos abstratos
21 17/10	<b>Interfaces</b> Conceito; Uso de interfaces
22 19/10	<b>Polimorfismo</b> Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica
23 24/10	<b>Polimorfismo Casting;</b> Chamada de um mesmo método em classes diferentes; Operador instanceof
24 26/10	<b>Polimorfismo</b> Aprofundamento e utilização de arrays com classes pai e filho misturadas.
25 31/10	<b>Pacotes</b> Definição; Organização de classes em pacotes;
26 07/11	<b>Pacotes</b> Organização de subpacotes; Empacotamento em arquivos JAR.
27 09/11	<b>Exceções</b> Tratamento de exceções
28 14/11	<b>Exceções</b> Tratamento de exceções
29 16/11	<b>Exceções</b> Tratamento de exceções
30	<b>Interface gráfica</b>

21/11	Componentes do Swing
31	<b>Interface gráfica</b>
23/11	Gerenciadores de Layout
32	<b>Persistência</b>
28/11	Serialização e deserialização de objetos
33	<b>Persistência</b>
30/11	Leitura e gravação de arquivos de texto
34	<b>Trabalho final</b>
05/12	
35	<b>Trabalho final</b>
07/12	
36	<b>Avaliação 2 (P2)</b>
12/12	

## 6. PROCEDIMENTOS METODOLÓGICOS

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.

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

A avaliação tem caráter integrador, uma vez que existe um crescimento gradativo na mobilização dos conhecimentos, habilidades e atitudes desenvolvidos. O resultado é expresso sob a forma de nota que varia de 0 (zero) a 10 (dez), com intervalos de 0,1 (um décimo). É exigido, no mínimo, a média 6,0 (seis) para fins de aprovação no componente curricular.

Os instrumentos de avaliação serão agrupados em 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 extraclasse (G1) realizados até a data da prova, com o seguinte cálculo:

$$NP1 = P1*0,6 + G1*0,4$$

sendo G1 calculado da seguinte forma:

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

onde Ti representa a nota de um trabalho, variando de 0 (zero) até 10.



## Universidade Federal da Fronteira Sul

A NP2 será composta por uma avaliação escrita (P2) e trabalhos extra-classe (G2) realizados até a data da prova, com o seguinte cálculo:

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

sendo G2 calculado da seguinte forma:

$$G2 = (T1 + T2 + \dots Tn) / n$$

onde  $T_i$  representa a nota de um trabalho, variando de 0 (zero) até 10.

A média final (MF) será calculada como:

$$MF = (NP1+NP2)/2$$

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.

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.

O formato dos trabalhos extraclasse será definido pelo professor no decorrer do processo de ensino-aprendizagem, tendo em vista o caráter processual da avaliação.

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, 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).
- Haverá um desconto de 50% da nota do trabalho no caso de entrega em atraso, com prazo máximo de três dias;
- 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**

A recuperação será realizada da seguinte forma: para cada NP será ofertada prova de recuperação (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



# Universidade Federal da Fronteira Sul

cálculo de NP não é alterado. Para os que a realizarem, haverá reposição apenas da nota da prova, não havendo portanto substituição das notas dos trabalhos. Dessa forma, para os alunos que prestarem RP, o cálculo de NP passa a ser definido por:

$$NP = RP * 0,6 + G * 0,4$$

## 8. Referências

### 8.1 Básicas

SANTOS, Rafael. **Introdução à programação orientada a objetos usando Java**. 8. reimp. Rio de Janeiro: Elsevier, 2003.

DEITEL, Harvey M.; DEITEL, Paul J. **Java Como Programar**. 8. ed. São Paulo: Pearson, 2010.

BORATTI, Isaias Camilo. **Programação orientada a objetos em JAVA**. Florianópolis: Visual Books, 2007.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, Servlets, Java Server Faces, Hibernate, EJB 3 Persistence e Ajax**. 1. ed. Rio de Janeiro: Ciência Moderna, 2007.

CORNELL, G.; HORSTMANN, C. S. **Core Java – Fundamentos**. São Paulo: Prentice Hall Brasil, 2009. v. 1.

### 8.2 Complementares

ECKEL, Bruce. **Thinking in Java**. Prentice-Hall, 2000.

LEWIS, J.; LOFTUS, W. **Java Software Solutions - Foundations of Program Design**. Addison-Wesley, 1999. KEOGH, Jim;

GRANNINI, Mario. **OOP Desmistificado – Programação Orientada a Objetos**. Rio de Janeiro: Alta Books, 2005.

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

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

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

Andressa Sellen

Professor 1770127

MARCO A. SPOHN

Coordenador do curso

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