



Plano de Ensino

1. Dados de Identificação

Curso: Ciência da Computação (Turno: Noturno)

Componente Curricular: Programação I

Fase: Terceira

Ano/Semestre: 2015/1

Numero de Créditos: 4

Carga horária Hora Aula: 72

Carga horária Hora Relógio: 60

Professor: Fernando Bevilacqua

Atendimento ao aluno: quarta-feira das 14:30h às 17:30h

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	Total Parc.	CONTEÚDO
Aula 1	3	Introdução Apresentação da disciplina; paradigmas de programação.
Aula 2	5	Introdução à linguagem Java Instalação e configuração do ambiente de desenvolvimento; Características; Aspectos de funcionamento.



Aula 3	8	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	10	Noções de Programação orientada a objetos Representação gráfica de classes e objetos. Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação)
Aula 5	13	Noções de Programação orientada a objetos Objetos e instanciação, representação na memória, métodos e propriedades.
Aula 6	15	Implementação de classes Classes e objetos; Atributos; Métodos; Mensagens.
Aula 7	18	Implementação de classes Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final
Aula 8	20	Encapsulamento Modificadores de visibilidade; Métodos de acesso e modificadores
Aula 9	23	Encapsulamento Exercícios
Aula 10	25	Herança Conceito e utilização; Construtores e propriedades herdadas.
Aula 11	28	Herança Especialização; Sobreposição de métodos; Utilização de super.
Aula 12	30	Herança Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
Aula 13	33	Associações Criação de classes compostas por outras classes
Aula 14	35	Associações Criação de getter/setter para propriedades associadas.
Aula 15	38	Avaliação 1 (P1)
Aula 16	40	Classes abstratas Conceito; Classes abstratas; Métodos abstratos
Aula 17	43	Classes abstratas Conceito; Classes abstratas; Métodos abstratos
Aula 18	45	Interfaces Conceito; Uso de interfaces
Aula 19	48	Polimorfismo Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica
Aula 20	50	Polimorfismo Casting; Chamada de um mesmo método em classes diferentes; Operador instanceof
Aula 21	53	Pacotes Definição; Organização de classes em pacotes;
Aula 22	55	Pacotes Organização de subpacotes; Empacotamento em arquivos JAR.
Aula 23	58	Exceções Tratamento de exceções
Aula 24	60	Exceções Tratamento de exceções



Aula 25	63	Interface gráfica Componentes do Swing. Gerenciadores de Layout
Aula 26	65	Persistência Leitura e gravação de arquivos de texto; Serialização e deserialização de objetos
Aula 27	68	Avaliação 2 (P2)
Aula 28	70	Trabalho final
Aula 29	72	Exame (29)

* 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:

$$\mathbf{G1 = (T_1 * K_1 + T_2 * K_2 + \dots T_n * K_n) / N}$$

onde **T_i** representa a nota de um trabalho, variando de 0 (zero) a 10, **K_i** representa o peso da nota de um trabalho e **N** representa a soma de todos os valores **K_i**, ou seja, $n = K_1 + K_2 + \dots K_n$. 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:

$$\mathbf{G2 = (T_1 * K_1 + T_2 * K_2 + \dots T_n * K_n) / N}$$

onde **T_i** representa a nota de um trabalho, variando de 0 (zero) até 10, **K_i** representa o peso da nota de um trabalho e **N** representa a soma de todos os valores **K_i**, ou seja, $n = K_1 + K_2 + \dots K_n$. 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 fechamento de nota algum estudante obter média menor **6,0**, será oferecida uma oportunidade, através de uma prova **PR** com todo o conteúdo, que será utilizada para calcular a nova média como:

$$NP'=(NP + PR)/2$$

Após a realização da prova **PR**, a nota **NP** do aluno passará a ser **NP'**.

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.
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.

Professor

Coordenador do curso