



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Bacharelado em Ciência da Computação

Componente Curricular: Programação I

Fase: 3ª (diurno)

Ano/Semestre: 2004/1

Número de Créditos: 04

Carga Horária - Hora aula: 72h

Carga Horária - Hora relógio: 60h

Professor: Fernando Bevilacqua

Atendimento ao Aluno: Segunda-feira das 14h00 às 18h00

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. OBJETIVOS:

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
Aula 1	Introdução Apresentação da disciplina; paradigmas de programação.
Aula 2	Introdução à linguagem Java Instalação e configuração do ambiente de desenvolvimento; Características; Aspectos de funcionamento.
Aula 3	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	Noções de Programação orientada a objetos Representação gráfica de classes e objetos. Operações de abstração (classificação/instanciação, generalização/especialização, agregação, associação)
Aula 5	Noções de Programação orientada a objetos Objetos e instanciação, representação na memória, métodos e propriedades.
Aula 6	Implementação de classes Classes e objetos; Atributos; Métodos; Mensagens.
Aula 7	Implementação de classes Métodos construtores; Passagem de parâmetros; Membros estáticos; Modificador Final
Aula 8	Encapsulamento Modificadores de visibilidade; Métodos de acesso e modificadores
Aula 9	Encapsulamento Exercícios
Aula 10	Herança Conceito e utilização; Construtores e propriedades herdadas.
Aula 11	Herança Especialização; Sobreposição de métodos; Utilização de super.
Aula 12	Herança Utilização de construtores herdados, aprofundamento sobre métodos sobrepostos.
Aula 13	Associações Criação de classes compostas por outras classes
Aula 14	Associações Criação de getter/setter para propriedades associadas.
Aula 15	Avaliação 1 (P1)
Aula 16	Correção da prova e discussões.
Aula 17	Classes abstratas Conceito; Classes abstratas; Métodos abstratos
Aula 18	Recuperação 1 (R1)
Aula 19	Interfaces Conceito; Uso de interfaces
Aula 20	Polimorfismo Assinatura de um método; Sobreposição; Sobrecarga; Ligação dinâmica
Aula 21	Polimorfismo



	Casting; Chamada de um mesmo método em classes diferentes; Operador instanceof
Aula 22	Pacotes Definição; Organização de classes em pacotes;
Aula 23	Pacotes Organização de subpacotes; Empacotamento em arquivos JAR.
Aula 24	Exceções Tratamento de exceções
Aula 25	Exceções Tratamento de exceções
Aula 26	Interface gráfica Teoria e exercícios.
Aula 27	Persistência Leitura e gravação de arquivos de texto; Serialização e deserialização de objetos
Aula 28	Avaliação 2 (P2)
Aula 29	Recuperação 2 (R2)

6. PROCEDIMENTOS METODOLÓGICOS

Uso de aulas teóricas e práticas, exercícios extra-class e trabalhos de implementação.

7. AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM

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:

$$NP1 = P1*0,7 + G1*0,3 + C1$$

sendo **G1** calculado da seguinte forma:

$$G1 = (T1 * K1 + T2 * K2 + ... Tn * Kn) / n$$

onde **Ti** representa a nota de um trabalho, variando de 0 (zero) a 10, **Ki** representa o peso do trabalho em questão, variando de 1 (um) até 50 (cinquenta), e **n** representa o número de trabalhos solicitados. O coeficiente bônus **C1** pode ser positivo (acréscimo de nota) ou negativo (decréscimo de nota), sendo utilizado para atividades de avaliação em classe (como trabalhos rápidos).

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:

$$NP2 = P2*0,7 + G2*0,3 + C2$$

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), sendo utilizado para atividades de avaliação em classe (como trabalhos rápidos).

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

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

7.1 RECUPERAÇÃO: NOVAS OPORTUNIDADES DE APRENDIZAGEM E AVALIAÇÃO

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:

$$P_i = (P_i + R P_i)/2$$

Os demais elementos que compõem a NP em questão permanecem com seus valores sem alterações 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 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.



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 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.

8.3 COMPLEMENTAR:

Professor

Coordenador do curso