



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
 Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



PLANO DE ENSINO

1. IDENTIFICAÇÃO

| | | | | | | | |
|------------------------|-----------------------------------|----------------|-----------|---------------|--------|------------------|---------------|
| Componente Curricular: | Programação Orientada a Objetos 2 | | | | | | |
| Unidade Ofertante: | FACOM | | | | | | |
| Código: | GSI020 | Período/Série: | 4 | Turma: | S | | |
| Carga Horária: | | | Natureza: | | | | |
| Teórica: | 30 | Prática: | 30 | Total: | 60 | Obrigatória: (X) | Optativa: () |
| Professor(A): | Fabíola S. F. Pereira | | | Ano/Semestre: | 2023/1 | | |
| Observações: | | | | | | | |

2. EMENTA

Padrões de análise. Projeto de software orientado a objetos. Arquitetura de software. Projeto detalhado de software. Princípios de projeto orientado a objetos. Tecnologia de componentes. Padrões arquiteturais. Padrões de projeto. Desenvolvimento de software orientado a aspectos. Tópicos avançados em projeto de software.

3. JUSTIFICATIVA

Hoje o mercado exige desenvolvedores que não apenas dominam determinada linguagem de programação, mas que também sejam versados em técnicas avançadas, capazes de avaliar problemas, saber manipular ferramentas que auxiliem projetos de software e que sejam capazes de tomar boas decisões em projeto de software. Linguagens do paradigma de Orientação a Objetos são o padrão no mercado de desenvolvimento de software atual, sendo o Java uma das mais proeminentes. O conhecimento também é de grande valia no meio acadêmico. É também mais uma oportunidade fundamental para que a prática da programação seja exercida pelos alunos de forma prática no curso de Bacharelado em Sistemas de Informação.

4. OBJETIVO

Objetivo Geral:

Munir o aluno de mais profundos conhecimentos relativos à Programação Orientada a Objetos, relacionados a ferramentas de desenvolvimento, análise e projeto de software e técnicas de reuso de software e projeto eficiente.

Objetivos Específicos:

Aplicar técnicas avançadas de análise e projeto empregadas no desenvolvimento de software, enfatizando formas de melhorar o reuso de software através do paradigma de Orientação a Objetos. Especificamente:

- compreender o papel dos padrões na reutilização de colaborações entre classes e objetos em modelos conceituais e modelos de software;
- projetar uma arquitetura de software usando padrões arquiteturais;
- aplicar padrões de projeto, principalmente aqueles mais importantes no desenvolvimento de frameworks;
- analisar os padrões de projeto mais apropriados ao software a ser desenvolvido;
- compreender alguns padrões picos de análise, i.e., modelos conceituais de objetos reutilizáveis;
- compreender anti-padrões;
- compreender os conceitos de frameworks e como eles permitem reutilizar a análise de problemas e o projeto de soluções, permitindo assim escrever aplicações relacionadas com eficácia;
- analisar frameworks concretos;

- compreender uma metodologia de desenvolvimento de frameworks;
- desenvolver software usando as técnicas avançadas de análise e projeto de software

5. PROGRAMA

1 — Programação genérica com classes e métodos genéricos

- 1.1 - Introdução
- 1.2 Motivação para métodos genéricos
- 1.3 - Métodos genéricos: implementação e tradução em tempo de compilação
- 1.4 - Métodos que utilizam uma variável de tipo como tipo de retorno
- 1.5 - Sobrecarregando métodos genéricos
- 1.6 - Classes genéricas
- 1.7 - Tipos brutos (raw types)
- 1.8 - Curingas (wildcards) em métodos que aceitam parâmetros de po
- 1.9 - Genéricos e Herança
- 1.10 Conclusão

2—Multithreading

- 2.1 — Introdução
- 2.2 - Estados de uma thread
- 2.3 - Prioridades
- 2.4 - Criando e executando threads
- 2.5 - Sincronização de threads
- 2.6 - Relacionamento produtor / consumidor
- 2.7 - Multithreading com GUI
- 2.8 - Interfaces Callable e Future
- 2.9 - Aplicações de threads

3 - RMI - Remote Method Invocation

- 3.1 - Introdução à invocação remota de métodos
- 3.2 - Conceitos básicos
- 3.3 - Camadas do RMI
- 3.4 - Serviços de registro e naming
- 3.5 - Aplicações de RMI

4 - Frameworks para elaboração de MVC, acesso a dados, testes e geração de relatórios

- 4.1 - Struts
- 4.2 - Hibernate
- 4.3 - Junit
- 4.4 - Jasper Report / iReport

5 - Princípios e Padrões de análise, arquitetura e projeto de Software

- 5.1 - Princípios de projeto de classes: (SRP) Single responsibility principle; (OCP) The open-closed principle; (LSP) The liskov substitution principle; (DIP) The dependency inversion principle; (ISP) The interface segregation principle.
- 5.2 - Princípios de coesão e acoplamento de pacotes: (REP) The reuse/release equivalency principle; (CCP) The common closure principle; (CRP) The common reuse principle; (ADP) The acyclic dependencies principle; (SDP) The stable dependencies principle; (SAP); The stable abstractions principle.
- 5.3 - Principais padrões de análise: party, organizational hierarchy, accountability, knowledge level, quantity, range, temporal patterns, accounting patterns.
- 5.4 - Principais padrões de projeto: observer, template method, strategy, abstract factory, builder, iterator, composite, decorator, façade, adapter, proxy, singleton, factory method, visitor, bridge, mediator, command, flyweight.
- 5.5 - Principais padrões arquiteturais: layer, microkernel, MVC, black board, broker;
- 5 - Introdução a orientação a aspectos
- 5.1- Limitações da orientação a objetos: entrelaçamento e espalhamento de código;
- 5.2 - Definição de pointcuts e advices;
- 5.3 - Implementação de aspectos: aspectJ;
- 5.4- Exemplos de uso de aspectos para melhoria de modularidade em sistemas;
- 5.5 - Conceituação de aspectos: concerns, scattering, tangling, weaving;

- 5.6 - Manutenção separada de aspectos com módulos de casos de uso; Estabelecimento de arquitetura de software baseada em casos de uso e aspectos;
- 6 - Orientação a aspectos. Apresentar o desenvolvimento de software orientado a aspectos.
 - 6.1 - Limitações da orientação a objetos: entrelaçamento e espalhamento de código;
 - 6.2 - Definição de pointcuts e advices;
 - 6.3 - Implementação de aspectos: AspectJ;
 - 6.4 - Exemplos de uso de aspectos para melhoria de modularidade em sistemas;
 - 6.5 - Conceituação de aspectos: concerns, scattering, tangling, weaving; Modelagem e captura de aspectos com casos de uso;
 - 6.6 - Manutenção separada de aspectos com módulos de casos de uso; Estabelecimento de arquitetura de software baseada em casos de uso e aspectos;
 - 6.7 - UML: modelagem de aspectos e casos de uso slices;
 - 6.8 - Padrões de uso de aspectos.
- 7 - Estudo de caso.
 - 7.1 - Desenvolver uma aplicação empregando adequadamente princípios e padrões, multithreading, RMI e frameworks apresentados;
 - 7.2 - Utilizar persistência em bancos de dados;
 - 7.3 - Documentar adequadamente todas as fases do desenvolvimento em UML

6. METODOLOGIA

6.1 - Sobre as atividades

As atividades acontecerão presencialmente às quartas-feiras, de 20:50 às 22:30, e às sextas-feiras, de 19:00 às 20:40. Compreendem aulas expositivas, aulas práticas em laboratório, aulas de resolução de exercícios e atividades avaliativas (por exemplo, provas).

| Aula | Data | Conteúdo |
|------|------------|---|
| | 02/08/2023 | Apresentação do curso. Discussão da ementa e formas de avaliação. |
| | 04/08/2023 | Propriedades de Projeto |
| | 09/08/2023 | Princípios de Projeto |
| | 11/08/2023 | Aula prática: Propriedades & Princípios de Projeto |
| | 16/08/2023 | Padrões de projeto criacionais I |
| | 18/08/2023 | Aula prática: Padrões de projeto criacionais I |
| | 23/08/2023 | Padrões de projeto criacionais 2 |
| | 25/08/2023 | Aula prática: Padrões de projeto criacionais 2 |
| | 30/08/2023 | Padrões de projeto estruturais I |

01/09/2023 Aula prática: Padrões de projeto estruturais I

06/09/2023 Padrões de projeto estruturais 2

08/09/2023 Aula prática: Padrões de projeto estruturais 2

13/09/2023 Padrões de análise

15/09/2023 Revisão

20/09/2023 Prova P1

Aula Extra - 1 Vista de Prova

22/09/2023 Padrões de projeto comportamentais I

27/09/2023 Aula prática: Padrões de projeto comportamentais I

29/09/2023 Padrões de projeto comportamentais 2

04/10/2023 FACOM Tech Week

06/10/2023 FACOM Tech Week

11/10/2023 Aula prática: Padrões de projeto comportamentais 2

13/10/2023 Padrões Arquiteturais

18/10/2023 Frameworks de persistência

20/10/2023 Aula prática: Frameworks de persistência

25/10/2023 Revisão

27/10/2023 Prova P2

Aula Extra - 2 Vista 2a prova

Aula Extra - 3 Aula de dúvidas sobre trabalho final

01/11/2023 Frameworks de teste e TDD

03/11/2023 Aula prática: Trabalho Final

08/11/2023 Aula prática: Trabalho Final

10/11/2023 Trabalho Final

17/11/2023 Encerramento da disciplina

22/11/2023 Prova Sub

Aula Extra - 4 Vista da Prova Sub

6.2 - Atendimento ao discente

Horário de atendimento presencial, previamente agendado com a professora: terça-feira, das 8:40h às 09:40h. Dúvidas também poderão ser encaminhadas via mensagem direta à professora. O tempo de resposta é de até 48h.

6.3 - Plataformas e softwares que serão utilizados

Site da disciplina:

Plataforma MS Teams UFU

Curso: GSI020 - POO2 - 2023/1

Inscrição: link a ser informado durante primeira aula presencial

Canal de comunicação com discentes:

Todos os avisos e comunicações necessárias serão postados pela professora no site da disciplina (MS Teams). Eventualmente, a professora utilizará como complemento à comunicação postada no site da disciplina, o envio de email para o aluno. Neste caso, será utilizado o email que o aluno cadastrou ao se matricular na disciplina.

Para comunicar com a professora, o aluno pode contactá-la via qualquer uma das seguintes opções:

Email (fabiola.pereira@ufu.br)

Mensagem direta via MS Teams (fabiola.pereira)

6.4 - É necessário que o discente:
possua acesso à Plataforma MS Teams.

6.5 - Informações gerais

Ao se matricular nesta disciplina, o aluno manifesta total concordância com o estabelecido neste plano de ensino.

7. AVALIAÇÃO

O aluno estará aprovado se a sua frequência for superior ou igual a 75% e a nota final for maior ou igual a 60 pontos. Se a frequência for inferior a 75% será reprovação automática.

7.1 Avaliação de desempenho

A avaliação do desempenho dos alunos será somativa, de forma individual, sendo realizada por meio de:

Prova P1 (20 pontos), individual, a ser realizada durante aula presencial, no dia 20/09/2023.

Prova P2 (25 pontos), individual, a ser realizada durante aula presencial, no dia 27/10/2023.

Exercícios práticos (20 pontos), a serem entregues individualmente, via site da disciplina (vide Seção 6.3), com prazo de entrega de aproximadamente 7 dias a contar da data de disponibilização da atividade. O valor de cada atividade será distribuído uniformemente em relação ao número de atividades aplicadas, totalizando 20 pontos.

Trabalho final (35 pontos), a ser entregue no dia 10/11/2023, que consistirá no desenvolvimento de um software aplicando os conceitos da disciplina. A professora irá apresentar a especificação detalhada do trabalho final durante a disciplina.

Os critérios de avaliação serão a corretude em programas conforme entradas e saídas estabelecidas, a exatidão nas respostas aos questionamentos, no percentual de itens entregues e na formatação de apresentações.

7.2 Recuperação de aprendizagem

Prova escrita substitutiva (ficará em substituição da pior nota das duas provas teóricas) a ser resolvida em sala no horário normal da aula. Será garantida a realização da atividade avaliativa de recuperação de aprendizagem ao estudante que não obtiver o rendimento mínimo para aprovação e com frequência mínima de 75% (setenta e cinco por cento) no componente curricular. Com a prova substitutiva a nota final na disciplina não poderá ser maior que 60. Data de realização: 22/11/2023.

7.3 Validação da assiduidade

A assiduidade será computada com base nas presenças nas aulas (chamada oral). O aluno deverá ter 75% de frequência para aprovação.

8. BIBLIOGRAFIA

Básica

DEITEL, H. M. Java: como programar, 8. ed. São Paulo: Prentice Hall, 2010.

FREEMAN, Eric. FREEMAN, Elisabeth. Use a cabeça! padrões de projeto. Rio de Janeiro: Atlas Books, 2005.

JACOBSON, Ivar. Aspect-oriented software development with use cases. Upper Saddle River: Addison-Wesley, 2004.

Complementar

GAMMA, Erich et al. Design Patterns. Upper Saddle River: Addison Wesley, 1995.

LADDAD, Ramnivas. Aspect in action: practical aspect oriented programming. Greenwich Mannign, 2003.

KRUCHTEN, Philippe. The rational unified process: an introduction. 3rd ed. Boston: Addison Wesley, 2003.

KRUCHTEN, Philippe. The rational unified process made easy: practitioner's guide to the RUP. Boston: Addison Wesley, 2003.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Porto Alegre: Bookman, 2000.

SOMMERVILE, I. Engenharia de software. 8th ed. Harlow: Addison Wesley, 2007.

FOWLER, Martin. Refactoring: Improving the design of existing code. Addison Wesley, 2000.

MENDES, Antonio. Arquitetura de Software, Ed. Campus, 2002.

SZYPERSKI, C. Component Software: Beyond Object Oriented Programming. Addison Wesley, 1998.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Porto Alegre: Bookman, 2001.

Marco Tulio Valente. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 2020.

9. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: ____/____/____

Coordenação do Curso de Graduação: _____



Documento assinado eletronicamente por **Fabíola Souza Fernandes Pereira, Professor(a) do Magistério Superior**, em 04/10/2023, às 15:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site

[https://www.sei.ufu.br/sei/controlador_externo.php?](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **4734273** e o código CRC **C2C2D1F6**.