



UNIVERSIDADE FEDERAL DE UBERLÂNDIA



FICHA DE COMPONENTE CURRICULAR

| | | |
|--|--|------------------------------|
| CÓDIGO: FACOM33307 | COMPONENTE CURRICULAR: Programação Orientada a Objetos I | |
| UNIDADE ACADÊMICA OFERTANTE: Faculdade de Computação | | SIGLA: FACOM |
| CH TOTAL TEÓRICA: 30 horas | CH TOTAL PRÁTICA: 30 horas | CH TOTAL: 60 horas |

1. **OBJETIVOS**

Geral: Capacitar o aluno no entendimento e aplicação dos conceitos fundamentais da Programação Orientada a Objetos (POO). Exercitar os conceitos apresentados com atividades e projetos práticos, utilizando plataformas, linguagens e ferramentas modernas e com suporte ao paradigma orientado a objetos.

Específicos:

- Analisar problemas computacionais e propor soluções utilizando conceitos fundamentais da POO, tais como classes, objetos, encapsulamento, herança e polimorfismo;
- Projetar, implementar, testar e depurar programas em uma linguagem orientada a objetos;
- Entender e aplicar o conceito de reuso de software por meio de bibliotecas de classe contendo algoritmos, estruturas de dados e demais tipos de código pré-existente;
- Interpretar modelos e diagramas representando abstrações de um domínio do problema, objetos e suas interações, projetos de classes e módulos de um sistema orientado a objetos;
- Utilizar diagramas UML como suporte ao entendimento dos conceitos de orientação a objetos.

2. **EMENTA**

Introdução à Programação Orientada a Objetos (POO); Paradigma Procedimental versus Orientado a Objetos (OO); Classes e Objetos; Variáveis e Tipos; Herança e Polimorfismo; Ciclo de vida de objetos; Tratamento de Exceções; Reuso com pacotes e bibliotecas de classes; Serialização, Sistema de Arquivos, Entrada e Saída; Conceitos básicos de Interfaces Gráficas OO.

3. **PROGRAMA**

1. Introdução à Programação Orientada a Objetos (POO)

1. Histórico e evolução da POO;
2. Visão geral e atualidades sobre plataformas, tecnologias e linguagens.

2. Paradigma Procedimental versus Orientado a Objetos (OO)

1. Elementos comuns: estruturas de controle e repetição, variáveis, atribuições e procedimentos;
2. Fundamentos da OO: Abstração, Encapsulamento, Modularidade, Herança e Polimorfismo;
3. Introdução à utilização de diagramas da UML como ferramenta de representação visual: Diagrama de Classes, Diagrama de Objetos; Diagrama de Sequência, Diagrama de Atividades, Diagrama de Pacotes, dentre outros.

3. Classes e Objetos

1. Atributos e métodos;
2. Instanciação de objetos;
3. Interação entre objetos e troca de mensagens;
4. Técnicas para projeto e extração de classes e objetos;
5. Encapsulamento, escopo e visibilidade;
6. Coesão e acoplamento;
7. Tipos básicos de relacionamentos: composição, agregação, dependência e associações.

4. Variáveis e Tipos

1. Tipos básicos, conversão e coerção entre tipos;
2. Referência a objetos e vetores;
3. Escopo de variáveis (local, instância, classe);
4. Constantes e variáveis globais.

5. Herança e Polimorfismo

1. Conceito de herança e polimorfismo;
2. Hierarquia de classes, generalização e especialização;
3. Reescrita e sobreposição de métodos;
4. Herança múltipla, classes abstratas e interfaces.

6. Ciclo de vida de objetos

1. Construtores, inicialização, sobrecarga e mudanças de estados;
2. Gerenciamento de memória automático versus manual;
3. Áreas de alocação: pilha e heap;
4. Destrutores e condições para limpeza da memória.

7. Tratamento de Exceções

1. Conceito e tipos de exceção;
2. Delimitação de blocos de código sujeito a exceções;
3. Regras para tratamento de exceções.

8. Reuso com pacotes e bibliotecas de classes

1. Manipulação e formatação de texto, datas e calendários;
2. Operações matemáticas e manipulação de números;
3. Estruturas de dados básicas: listas, conjuntos e dicionários.

9. Serialização, Sistema de Arquivos, Entrada e Saída

1. Conceito de Serialização;
2. Processamento de arquivos;
3. Classes e bibliotecas para entrada, saída e representação de dados.

10. Conceitos básicos de Interfaces Gráficas OO

1. Ferramentas de prototipação de interfaces gráficas;
2. Manipulação de componentes gráficos (gerenciadores de layout, botões, listas, campos, áreas de texto, dentre outros);
3. Tratamento de eventos do usuário, classes internas e anônimas.

4. BIBLIOGRAFIA BÁSICA

BARNES, David J.; KOLLING, Michael. **Programação orientada a objetos com Java**: uma introdução prática usando BlueJ. 4. ed. São Paulo: Prentice Hall, 2009.

DEITEL, Paul J.; DEITEL, Harvey M. **Java**: como programar. 8. ed. São Paulo: Prentice Hall, 2010.

SIERRA, Kathy; BATES, Bert. **Use a cabeça!**: Java. 2. ed. Rio de Janeiro: Alta Books, 2005.

5. BIBLIOGRAFIA COMPLEMENTAR

BLOCK, Joshua. **Java efetivo**. 2. ed. Rio de Janeiro: Alta Books, 2010.

BOOCH, Grady *et al.* **Object-oriented analysis and design with applications**. 3rd. ed. Upper Saddle River: Addison-Wesley, 2007.

ECKEL, Bruce. **Thinking in Java**. 4th. ed. Upper Saddle River: Prentice Hall, 2006.

FOWLER, Martin. **UML essencial**: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005.

MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. **Use a cabeça!**: análise e projeto orientado ao objeto. Rio de Janeiro: Alta Books, 2007.

6. APROVAÇÃO

Prof. Dr. Rafael Dias Araújo
Coordenador do Curso de Sistemas de Informação
Campus Monte Carmelo

Prof. Dr. Mauricio Cunha Escarpinati
Diretor da Faculdade de Computação



Documento assinado eletronicamente por **Rafael Dias Araújo, Coordenador(a)**, em 25/01/2023, às 22:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Mauricio Cunha Escarpinati, Diretor(a)**, em 26/01/2023, às 17:18, 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?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4053291** e o código CRC **3EFB539B**.

