

 CATÓLICA	UNIVERSIDADE CATÓLICA DE PERNAMBUCO RECONHECIDA EM 18 DE JANEIRO DE 1952 PELO DECRETO Nº 30.417 PRÓ-REITORIA ACADÊMICA DIVISÃO DE PROGRAMAÇÃO ACADÊMICA DIRETORIA DE GESTÃO ESCOLAR	CÓDIGO	INF1143			
		DISCIPLINA	PROGRAMACAO III			
		VIGÊNCIA	a partir de 2020.1			
		CRÉDITOS	CARGA HORÁRIA			SEMESTRAL
	SEMANAL					
	TEORIA		EXERCÍCIO	LAB / PRÁTICA		
PROGRAMAÇÃO ACADÊMICA DE DISCIPLINA	04	01	01	02	60	

EMENTA

Levar o aluno a discutir, exercitar e consolidar o uso de técnicas de programação que tenham um impacto considerável sobre a qualidade de software. Discutir aspectos de qualidade, modularidade, reusabilidade e produtividade no desenvolvimento e manutenção de software. Estudar conceitos de programação orientada a objetos e paradigma híbrido (orientação a objetos com extensões funcionais), ilustrando como os mesmos são representados em uma linguagem de programação. Dar ao aluno a capacidade de discernir como a orientação a objetos impacta na qualidade de software.

CONTEÚDO PROGRAMÁTICO E METODOLOGIA

Aspectos de qualidade de software: modularidade e ideias básicas sobre a estruturação de sistemas, coesão, acoplamento e como controlá-los. Características da linguagem orientada a objetos a ser utilizada e suas extensões funcionais. **Conceitos de orientação a objetos:** objeto, classe, método e atributo, encapsulamento, ocultamento de informação (information hiding). Criação e remoção de objetos, construtores, referências, aliasing, sobrecarga (overloading). Herança, subtipos e polimorfismo. Ligação dinâmica (dynamic binding). Classes abstratas. Interfaces. **Conceitos de hibridismo:** Funções lambda. Estruturação de sistemas com padrões arquiteturais. Boas práticas de desenvolvimento SOLID. Principais padrões de projeto GRASP e Gof.

Competências/Habilidades:

- Compreender os conceitos básicos do paradigma orientado a objetos como classes, atributos e métodos;
- Compreender os conceitos de coesão e acoplamento e o relacionamento entre eles;
- Compreender os conceitos de encapsulamento, herança, polimorfismo, classes abstratas, interfaces;
- Compreender como os conceitos de encapsulamento, herança, polimorfismo, classes abstratas, interfaces se relacionam entre si e com a coesão e o acoplamento;
- Aplicar os conceitos de encapsulamento, herança, polimorfismo e classes abstratas, interfaces, coesão e acoplamento na construção de softwares orientados a objetos;
- Construir softwares orientados a objetos a partir de problemas contextualizados;
- Aplicar boas práticas de desenvolvimento orientadas a objetos;
- Entender e aplicar padrões de projetos e arquiteturais no desenvolvimento de software orientado a objetos;
- Identificar e corrigir problemas clássicos de software orientado a objetos;
- Utilizar ferramentas que auxiliem na aplicação dos conceitos e boas práticas de desenvolvimento de software orientado a objetos.

Metodologia: A disciplina será ministrada com sessões de resolução de problemas, pesquisas individuais e em grupo, consolidação das pesquisas e report dos resultados seguindo a técnica educacional PBL (Problem Based Learning). Serão ciclos de 3 aulas para expor o problema e construir coletivamente premissas, investigação da validade das premissas e discussão de boas soluções, construção e apresentação das boas soluções escolhidas. Projeto de implementação de sistema orientado a objetos como objetivo final de disciplina.

BIBLIOGRAFIA

Básica

DEITEL, H. M.; DEITEL, P. J. **Java: como programar**. 8.ed. São Paulo: Pearson, 2011.
 ECKEL, Bruce. **Thinking in Java**. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, c2003.
 FURGERI, Sérgio. **Java 8, ensino didático: desenvolvimento e implementação de aplicações**. São Paulo: Erica, 2015.

Complementar

ANICHE, Mauricio. **Orientação a objetos e SOLID para ninjas: projetando classes flexíveis**. Casa do Código, 2015.
 CARVALHO, Thiago. **Orientação a objetos: aprenda seus conceitos e suas aplicabilidades de forma efetiva**. Casa do Código, 2016.
 LEMAY, Laura; CADENHEAD, Rogers. **Aprenda em 21 dias JAVA 2: professional reference**. Rio de Janeiro: Campus, 2001.
 MEYER, B. **Object oriented software construction**. New Jersey: Prentice Hall, 1997.
 RODRIGUES FILHO, Renato. **Desenvolva aplicativos com Java 2**. São Paulo: Érica, 2005.

----- FIM DA IMPRESSÃO -----

Curso Responsável CIENCIA DA COMPUTACAO	Autenticado pela Diretoria de Gestão Escolar Recife, 27 de junho de 2023 MARIA TERESA BARRETO DE M. PERETTI Matr. 3156
--	--