

Plano de Ensino da Disciplina: Programação Orientada a Objetos (INF1123)

IDENTIFICAÇÃO

Universidade Católica de Pernambuco

Centro: Centro de Ciências e Tecnologia
Computação

Curso: Curso de Ciência da

Professor (a): Antonio Luiz de Oliveira Cavalcanti Júnior

Turno: Noite

Período: 4º

Turma: TS40

Carga Horária: 60h

Ano/Semestre letivo: 2018/02

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 manutenção de software. Estudar conceitos de programação orientada a objetos, ilustrando como os mesmos são representados em uma linguagem de programação.

CONTEXTUALIZAÇÃO

Quanto à Grade Curricular: esta disciplina apresenta-se no quarto semestre do curso de Bacharelado em Ciência da Computação, tendo por objetivo o estudo e aplicação de estruturas de dados. O pré-requisito para cursar esta disciplina é Estruturas de Dados I.

Quanto ao Mercado de Trabalho: o desenvolvimento de praticamente todo sistema de informação é realizado utilizando linguagens orientadas a objetos, pois as características como de reuso de código, fácil manutenção e baseada em componentes são imprescindíveis em sistemas de médio e grande porte.

OBJETIVOS:

GERAIS:

Proporcionar ao aluno o conhecimento de conceitos de orientação a objetos e ao desenvolvimento de um sistema de pequeno porte onde ele será capaz de demonstrar o seu conhecimento.

ESPECÍFICOS:

- Entender os principais conceitos de orientação a objetos.
- Aplicar os conceitos de orientação a objetos no desenvolvimento de uma aplicação de pequeno porte.
- Aplicar padrões e boas práticas de desenvolvimento orientado a objetos.

COMPETÊNCIAS DA DISCIPLINA

Cada competência possui o código das competências gerais e/ou específicas do currículo de referência da Sociedade Brasileira de Computação - SBC de outubro de 2017. Essa disciplina cobre todos os eixos de formação do mesmo documento.

C1. Compreender os conceitos básicos do paradigma orientado a objetos como classes, atributos e métodos.

CE-I

C2. Criar diagramas de classe em UML contendo classes, atributos, métodos e as principais associações entre classes.

CG-IV, CE-V

C3. Compreender os conceitos de coesão e acoplamento e o relacionamento entre eles.

CE-I

C4. Compreender os conceitos de encapsulamento, herança, polimorfismo, classes abstratas, interfaces.

CE-I

C5. Compreender como os conceitos de encapsulamento, herança, polimorfismo, classes abstratas, interfaces se relacionam entre si e com a coesão e o acoplamento.

CE-I

C6. Aplicar os conceitos de encapsulamento, herança, polimorfismo e classes abstratas, interfaces, coesão e acoplamento na construção de softwares orientados a objetos.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

C7. Construir softwares orientados a objetos a partir de problemas contextualizados.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

C8. Aplicar boas práticas de desenvolvimento orientadas a objetos.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

C9. Entender e aplicar padrões de projetos e arquiteturas no desenvolvimento de software orientado a objetos.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

C10. Identificar e corrigir problemas clássicos de software orientado a objetos.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

C11. Utilizar ferramentas que auxiliem na aplicação dos conceitos e boas práticas de desenvolvimento de software orientado a objetos.

CG-I, CG-III, CG-IV, CG-VIII, CE-II, CE-V, CE-VI, CE-VIII, CE-X e CE-XI

CONTEÚDOS: (Conhecimentos, Habilidades, Atitudes)

Introdução: história da orientação a objetos e características da linguagem orientada a objetos a ser utilizada.

Aspectos de qualidade de software: modularidade e idéias básicas sobre a estruturação de sistemas, coesão, acoplamento e como controlá-los.

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.

Estruturação de sistemas com o padrão MVC (Model View Control).

Boas práticas de desenvolvimento SOLID

Padrões de Projeto Gof.

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.

Plano de Ensino		
Aula	Data	Conteúdo
1	06/08/2018	Apresentação da disciplina e acordos pedagógicos; Conceitos Básicos e UML
2	09/08/2018	Conceitos Básico e UML
3	13/08/2018	Coesão, Acoplamento e Encapsulamento
4	16/08/2018	Coesão, Acoplamento e Encapsulamento
5	20/08/2018	Herança
6	23/08/2018	Herança
7	27/08/2018	Polimorfismo
8	30/08/2018	Polimorfismo
9	03/09/2018	Classes Abstratas e Interfaces
10	06/09/2018	Classes Abstratas e Interfaces
11	10/09/2018	MVC e construção do projeto
12	13/09/2018	MVC e construção do projeto
13	17/09/2018	SOLID
14	20/09/2018	SOLID
15	24/09/2018	Avaliação do 1º GQ – Primeira Chamada
16	27/09/2018	Avaliação do 1º GQ – Segunda Chamada
17	01/10/2018	Devolução de Provas
18	04/10/2018	Padrões de Projeto GoF
19	08/10/2018	Padrões de Projeto GoF
20	11/10/2018	Padrões de Projeto GoF
21	15/10/2018	Implementação GoF no Projeto - Flipped
22	18/10/2018	Implementação GoF no Projeto - Flipped
23	22/10/2018	Implementação GoF no Projeto - Flipped
24	25/10/2018	Implementação GoF no Projeto - Flipped

25	29/10/2018	Implementação GoF no Projeto - Flipped
26	01/11/2018	Acompanhamento de Projetos – GoF
27	05/11/2018	Code Smells
28	08/11/2018	Acompanhamento de Projetos – Code Smell
29	12/11/2018	Métricas de Código com o Sonar Qube
30	15/11/2018	Acompanhamento de Projetos
31	19/11/2018	Avaliação de 2º GQ – 1ª Chamada
32	22/11/2018	Avaliação de 2º GQ – 2ª Chamada
33	26/11/2018	Entrega de Avaliações
34	29/11/2018	Prova final

RECURSOS DIDÁTICOS:

Computador, projetor multimídia, quadro-branco, laboratório de informática, ambiente de educação a distância..

AValiação DA APRENDIZAGEM

Projeto no qual será desenvolvido utilizando técnicas de orientação a objetos, padrões de projeto e estruturado na arquitetura MVC, onde o aluno irá aplicar o conhecimento adquirido nas aulas PBL. Avaliações escritas explorando os conceitos e práticas da disciplina e as entregas de cada ciclo de execução PBL.

1º GQ: Entrega das atividades do ciclo PBL valendo 3 pontos. Entrega da primeira parte do projeto valendo 3,5 pontos - Aplicação desenvolvida em uma linguagem de programação OO aplicando todos os conceitos vistos em sala de aula. O projeto precisa ser apresentado por cada aluno do grupo. Caso o aluno não apresente formalmente o projeto receberá nota zero. Prova Escrita valendo 3,5 pontos.

2º GQ: Entrega das atividades do ciclo PBL valendo 3 pontos. Entrega da segunda parte do projeto valendo 3,5 pontos - Aplicação desenvolvida em uma linguagem de programação OO aplicando todos os conceitos vistos em sala de aula. O projeto precisa ser apresentado por cada aluno do grupo. Caso o aluno não apresente formalmente o projeto receberá nota zero. Prova Escrita valendo 3,5 pontos.

FONTES DE PESQUISA: (Bibliografia)

Básica:

- DEITEL, H. M.; DEITEL, P. J. **Java: como programar**. 3. ed. Porto Alegre: Bookman, 2001.
- ECKEL, Bruce. **Thinking in Java**. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, c2003.
- HORSTMANN, Cay S; CORNELL, Gary. **Core java 2**. 1. ed. São Paulo: Makron books, 2001.

Complementar:

- CARVALHO, Thiago, **Orientação a Objetos: Aprenda seus conceitos e suas aplicabilidades de forma efetiva**, Casa do Código, 2016.
- ANICHE, Mauricio, **Orientação a Objetos e SOLID para Ninjas: Projetando classes flexíveis**, 2015, CASA DO CÓDIGO.
- LEMAY, Laura; CADENHEAD, Rogers. **Aprenda em 21 dias JAVA 2: professional reference**. Rio de Janeiro: Ed. 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.

Recife, 01 de Agosto de 2018

Prof. Antonio Luiz de Oliveira Cavalcanti Júnior