

CURSO DE ENGENHARIA DE SOFTWARE

1. CARACTERIZAÇÃO DA DISCIPLINA

Nome da Disciplina: Programação Orientada à Objetos	Ano/semestre: 2021/2
Código da Disciplina: 10174	Período: 3º/4º
Carga Horária Total: 80h/a	Carga Horária Prática: 80h/a
Pré-Requisito: Não se Aplica	Co-Requisito: Não se Aplica

2. PROFESSOR(ES)

William Pereira dos Santos Júnior, M.e.

3. EMENTA

Conhecer: Princípios da programação orientada a objetos: classes e objetos.

Compreender: Princípios da programação orientada a objetos. Programação concorrente, controle e tratamento de eventos, persistência dos dados, distribuição dos componentes, tratamento de erros e exceções, tolerância a falhas, interação e apresentação.

Aplicar: Abstração; encapsulamento; classe; objeto; herança; polimorfismo; coesão; acoplamento; contratos/interfaces. Interface, entrada e saída (streams). Aplicação em uma Linguagem de Programação. Ferramentas para programação orientada a objetos.

4. OBJETIVO GERAL

Proporcionar o conhecimento sobre como trabalhar com uma linguagem de programação orientada a objetos e compreender detalhes de um projeto de sistemas orientado a objetos.

5. OBJETIVOS ESPECÍFICOS

Unidades	Objetivos Específicos
Abstração, objetos;	Adquirir, entender e aplicar os conhecimentos de conversão de projetos orientados à objetos em programas orientados à objetos.
Aplicação em Linguagem de programação;	Desenvolver a prática na linguagem de programação JAVA: bibliotecas e sintaxe.
Classes, herança, polimorfismo, interface, entrada e saída.	Avaliar e criar programas orientados à objetos utilizando os recursos do paradigma em sua totalidade.

6. HABILIDADES E COMPETÊNCIAS

Gerais

- G.1. Identificar problemas que tenham solução algorítmica;
- G.2. Conhecer e compreender os limites da computação;
- G.3. Tomar decisões, avaliando criticamente, soluções computacionais, consciente dos aspectos humanos, éticos, legais e ambientais decorrentes;
- G.4. Gerir a própria aprendizagem e desenvolvimento pessoal-profissional, realizando trabalho em equipe, com visão trans e interdisciplinar;
- G.5. Desenvolver trabalhos e soluções, adotando metodologias diversificadas;
- G.7. Compreender as línguas materno e estrangeira, em seus aspectos estruturais, linguístico-discursivo e aplicá-las em diferentes contextos

comunicativos formais e informais na Engenharia de Software;
G.8. Resolver problemas usando ambientes de programação;

Específicas

E.1. Investigar, compreender e estruturar as características de domínios de aplicação em diversos contextos para a construção de sistemas de software, considerando questões éticas, sociais, legais e econômicas, individualmente e/ou em equipe;

E.2. Compreender e aplicar processos, técnicas e procedimentos de construção inerentes à produção e utilização de software, conhecendo os direitos e propriedades intelectuais;

E.3. Avaliar a qualidade e evolução de sistemas de software, aplicando adequadamente normas técnicas, através de padrões e boas práticas no desenvolvimento de software;

E.7. Analisar e criar novos modelos no desenvolvimento de software, identificando oportunidades e desenvolvendo soluções inovadoras.

7. CONTEÚDO PROGRAMÁTICO

Semana	Data	Conteúdo	Estratégia de ensino-aprendizagem	Aula Teórica/Prática	Local
1	05/08	Apresentação do Plano de Disciplina. Compreensão da mudança de paradigma.	Aula expositiva dialogada: Metodologia Ativa	Teórica	Sala Convencional
2	12/08	Trabalho em sala sobre o novo paradigma. Introdução à Linguagem Java	Metodologia Ativa – Mapa Mental	Prática	Laboratório de Desenvolvimento
3	19/08	Conceitos básicos de orientação a Objetos: Definição de abstração, classes e objetos. Exercícios em sala de aula. Trabalho sobre objetos para casa.	Oficina em laboratório: TIC (Softwares para desenvolvimento da prática de Programação Orientada à Objetos)	Prática	Laboratório de Desenvolvimento
4	26/08	Atividade prática sobre composição. Implementação Laboratório.	Estudo de caso: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
5	02/09	Atividade prática sobre encapsulamento. Implementação Laboratório	Debate: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
6	09/09	Correção das atividades Práticas	Oficina em laboratório: TIC (Softwares para desenvolvimento da prática de Programação Orientada à Objetos)	Prática	Laboratório de Desenvolvimento
7	16/09	Revisão para a 1ª VA através de atividade prática e fundamentação a partir de dúvidas	Discussão de temas: Metodologia Ativa	Prática	Auditório
8	23/09	1ª Verificação de Aprendizagem	Avaliação somativa	Teórica	Sala Convencional
9	30/09	Devolutiva da 1 VA. Recuperação de Conteúdo.	Oficina em laboratório: TIC (Softwares para desenvolvimento da prática de Programação Orientada à Objetos)	Prática	Laboratório de Desenvolvimento
10	07/10	Herança, Polimorfismo, sobrescrição e sobreposição de métodos: exemplos práticos. Trabalho em laboratório	Estudo dirigido: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
11	14/10	Utilização de composição e herança através da construção de interfaces gráficas. Atividade prática.	Oficina em laboratório: TIC (Softwares para desenvolvimento da prática de Programação Orientada à Objetos)	Prática	Laboratório de Desenvolvimento
12	21/10	Exercícios práticos sobre herança e polimorfismo	TBL: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
13	28/10	SINACEN	Estudo dirigido: Metodologia Ativa	Teórica	Sala de aula convencional
14	04/11	interfaces e classes abstratas. Revisão sobre o uso de herança e sobrescrição de métodos.	Oficina em laboratório	Prática	Laboratório de Desenvolvimento
15	11/11	2ª Verificação de Aprendizagem	Avaliação somativa	Prática	Laboratório de Desenvolvimento
16	18/11	Exercícios práticos sobre herança e	Seminário: Metodologia Ativa	Prática	Laboratório de

		polimorfismo: interfaces e classes abstratas.			Desenvolvimento
17	25/11	Trabalho de estudo sobre Java 8:	Estudo de texto: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
18	02/12	Apresentação e entregar do Trabalho sobre Java 8.	Discussão de temas: Metodologia Ativa	Prática	Laboratório de Desenvolvimento
19	09/12	3ª Verificação de Aprendizagem	Avaliação Formativa	Teórica	Sala Convencional
20	16/12	Devolutiva da 3 VA. Aplicação da prova substitutiva.	Sala de Aula	Teórica	Auditório/Sala de aula

8. PROCEDIMENTOS DIDÁTICOS

Metodologias de ensino/aprendizagem

A disciplina será ministrada através de material teórico em sala de aula e construção de aplicações práticas em laboratório, além de um constante acompanhamento quanto a qualidade e aplicação dos conceitos nos exercícios e trabalhos práticos.

Serão utilizados conceitos de construção de telas e estímulo na construção de jogos simples de tabuleiro e “plataforma” para ajudar no processo de aprendizagem sobre abstração de problemas em modelos orientados a objetos.

Os métodos de ensino utilizados são:

- Metodologias Ativas: Seminário, aula expositiva dialogada, Phillips 66, oficina em laboratório, estudo de caso, debate, avaliação somativa, discussão de temas, estudo dirigido, gamificação, estudo de texto
- Tecnologia da Informação e comunicação (TICs): softwares de programação orientada a objeto, ROBOCODE para desenvolvimento de jogos

Recursos educativos

Quadro-branco/pincel, projetor multimídia, livros, ebook, laboratório de computação, internet.

Recursos de Acessibilidade disponíveis aos acadêmicos

O curso assegura acessibilidade metodológica, digital, comunicacional, atitudinal, instrumental e arquitetônica, garantindo autonomia plena do discente.

9. ATIVIDADE INTEGRATIVA

A interdisciplinaridade no curso de Engenharia de Software é construída com o amparo das disciplinas de Projeto Interdisciplinar. Estas promovem a associação entre os diferentes conteúdos, habilidades e cenários em projetos que favoreçam a construção do conhecimento científico, tecnológico e de prática profissional aliado à autoaprendizagem, proatividade, resolução conjunta de problemas, trabalho em equipe, reflexividade, entre outros.

A proposta de cada disciplina de Projeto Interdisciplinar é variável, mas, obrigatoriamente, deve evoluir em uma constante de maturidade pessoal, interpessoal, científica e prática. Para o desenvolvimento dos projetos interdisciplinares o aluno percorre três momentos: ensino – por meio do diálogo entre as áreas de conhecimento; pesquisa – seguindo os rigores metodológicos necessários à construção do conhecimento científico e de extensão – oportunizando o compartilhamento dos projetos desenvolvidos para o público interno e externo.

Em função disto, as atividades de cada Projeto estão detalhadas em Plano de Ensino próprio.

10. PROCESSO AVALIATIVO DA APRENDIZAGEM

1ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação teórica, com valor de 0 a 60 pontos.

Avaliações processuais totalizam 40 pontos distribuídos da seguinte forma:

- Atividades realizadas em casa – 0 a 20.
- Atividades que utilizam TICs e metodologia ativa em sala de aula – 0 a 20. (**seção de conteúdo programático e procedimentos didáticos**)

A média da 1ª V. A. será a somatória da nota obtida na avaliação teórica (0-60 pontos) e as notas obtidas nas avaliações processuais (0-40 pontos).

(a devolutiva será realizada conforme Cronograma).

2ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação teórica, com valor de 0 a 60 pontos.

Avaliações processuais totalizam 40 pontos distribuídos da seguinte forma:

- Atividades realizadas em casa – 0 a 20.
- Atividades que utilizam TICs e metodologia ativa em sala de aula – 0 a 20. (**seção de conteúdo programático e procedimentos didáticos**)

A média da 2ª V. A. será a somatória da nota obtida na avaliação teórica (0-60 pontos) e as notas obtidas nas avaliações processuais (0-40 pontos).

(a devolutiva será realizada conforme Cronograma).

3ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação teórica, com valor de 0 a 50 pontos.

Avaliações processuais totalizam 50 pontos distribuídos da seguinte forma:

- Atividades realizadas em casa – 0 a 10.
- Atividades que utilizam TICs e metodologia ativa em sala de aula – 0 a 20. (**seção de conteúdo programático e procedimentos didáticos**)
- Projeto interdisciplinar – 0 a 30.

Obs.: Caso o aluno não esteja realizando o Projeto interdisciplinar o valor de 50 pontos será aplicado as outras avaliações processuais.

A média da 3ª V. A. será a somatória da nota obtida na avaliação teórica (0-50 pontos) e as notas obtidas nas avaliações processuais (0-50 pontos).

(a devolutiva será realizada conforme Cronograma).

ORIENTAÇÕES ACADÊMICAS

- Nas três VAs – O pedido para avaliação substitutiva tem o prazo de 3 (três) dias úteis a contar da data de cada avaliação com apresentação de documentação comprobatória (§ 1º e § 2º do art. 39 do Regimento Geral do Centro Universitário UniEVANGÉLICA). **A solicitação deverá ser feita através do Sistema Acadêmico Lyceum obrigatoriamente.**
- Nas três VAs – O pedido para Revisão de nota tem o prazo de 3 (três) dias úteis a contar da data da publicação, no sistema acadêmico Lyceum, do resultado de cada avaliação. (Art. 40 do Regimento Geral do Centro Universitário UniEVANGÉLICA).
- Atribui-se nota zero ao aluno que deixar de submeter-se às verificações de aprendizagem nas datas designadas, bem como ao que nela se utilizar de meio fraudulento. (Capítulo V Art. 39 do Regimento Geral do Centro Universitário UniEVANGÉLICA)
- A frequência será aferida predominantemente no final da aula. Exceto por processo de tratamento de casos excepcionais, não serão abonadas faltas, mesmo que o aluno tenha nota para ser aprovado na disciplina. Os Exercícios Práticos poderão ser aplicados em qualquer aula inclusive com entrega no mesmo dia. Para entrega de trabalhos por e-mail utilizar: **poo.marcelo@gmail.com**

Participação em eventos científicos:

Portaria – Frequência e nota dos alunos que apresentarem trabalhos em eventos científicos

Seguir as orientações presentes na Portaria Nº 01, de 7 de fevereiro de 2019, dos Bacharelados em Computação, que dispõe sobre os procedimentos de justificativa de ausência para alunos que apresentarem trabalhos em eventos científicos.

Condição de aprovação

Considera-se para aprovação do (a) acadêmico (a) na disciplina, frequência mínima igual ou superior a 75% da carga horária e nota igual ou superior a sessenta (60) obtida com a média aritmética simples das três verificações de aprendizagem.

11. BIBLIOGRAFIA

Básica:

ECKEL, Bruce. Thinking in Java. 3. ed. Estados Unidos: Prentice Hall, 2003. 1114 p. Winder, Russel. Desenvolvendo software em Java. 6 ed.

Rio de Janeiro: LTC, 2009. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-1994-9>

LARMAN, Craig. Utilizando UML e Padrões. Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo. 3ª ed.

Bookman, 2016. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577800476>.

Complementar:

ECKEL, Bruce. Thinking in Java. 3. ed. Prentice Hall, 2003

FEDELI, Ricardo Daniel; POLLONI, Enrico Giulio; PERES, Fernando Eduardo. Orientação a objeto com prototipação. São Paulo, SP, Brasil: Thomson, 2002. 119 p.

FEIJÓ, Bruno; CLUA, Esteban; SILVA, Flávio Soares Corrêa. Introdução à ciência da computação com jogos: aprendendo a programar com entretenimento. Rio de Janeiro, RJ, Brasil: Campus, 2010. 263 p.

HORSTMANN, Cays S; CORNELL, Gary. Core java 2: fundamentos. Tradutor de Marcelo SOARES, Andreza CARDOSO. 7. ed. Rio de Janeiro, RJ, Brasil: Alta Books, 2004. 424 p.

SANTOS, Rafael. Introdução à Programação Orientada a Objetos Usando JAVA. Campus, 2003. 319p.

Anápolis, 02 de Agosto de 2021.

Profa. M.e Natasha Sophie Pereira
COORDENADORA DO CURSO DE ENGENHARIA DE SOFTWARE DA UniEVANGÉLICA

Prof. M.e William Pereira dos Santos Júnior
COORDENADOR PEDAGÓGICO DO CURSO DE ENGENHARIA DE SOFTWARE DA UniEVANGÉLICA

Prof. M.e William Pereira dos Santos Júnior
PROFESSOR-RESPONSÁVEL PELA DISCIPLINA

