
CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: Orientação a Objetos

CÓDIGO 195341

TURMA: A

CRÉDITOS: 04

SEMESTRE: 2020 / 1

PROFESSOR: Vandor Roberto Vilardi Rissoli

PLANO DE ENSINO

1. OBJETIVOS DA DISCIPLINA

Abordar as principais características do paradigma de orientação a objetos, tais como: classes, objetos, atributos, métodos, herança, polimorfismo e encapsulamento. Introduzir os conceitos fundamentais necessários para o desenvolvimento de programas de computador baseados neste paradigma. Introduzir os conceitos de análise e concepção orientadas a objetos (modelagem orientada a objetos), com uso inicial da notação UML (*Unified Modeling Language*).

2. EMENTA

1. Problemas no desenvolvimento de software;
 2. Introdução às boas práticas de programação;
 3. Introdução à programação orientada a objetos;
 4. Projetos e implementação de sistema orientado a objetos.
-

3. HORÁRIO DAS AULAS E DE ATENDIMENTO

Aulas teóricas / práticas: Terças e Quintas-feiras das **16h às 17h50** no laboratório **I-07** do prédio UAC

Horário de atendimento: Terças-feiras das **10h às 12h** na sala **4** do prédio UED

4. CONTEÚDO PROGRAMÁTICO

01. Introdução a Orientação a Objetos
 02. Classes e objetos
 03. Introdução a Linguagem JAVA
 04. Encapsulamento
 05. Herança
 06. Polimorfismo
 07. Desenvolvimento de aplicações orientadas a objetos
-

5. METODOLOGIA

Serão ministradas aulas expositivas e práticas em laboratório pelo professor e auxiliadas por monitores (se disponíveis). O objetivo das práticas é permitir que os estudantes desenvolvam habilidades essenciais para a programação orientada a objetos, além de fixar os conceitos abordados nas aulas teóricas e das compreensões lógicas e organizacionais ao emprego adequado da tecnologia de orientação a objetos.

Com o intuito de estabelecer a aprendizagem nessa disciplina, as aulas serão complementadas com atividades de exercícios, trabalhos e demandas extraclasse, disponibilizadas por meio do Ambiente Virtual de Aprendizagem

(MOODLE) e do Sistema Tutor Inteligente (SAE), além do uso de um Ambiente de Desenvolvimento Integrado (Eclipse – IDE - *Integrated Development Environment*).

6. CRITÉRIOS DE AVALIAÇÃO

A avaliação dos estudantes na disciplina será composta por 7 atividades avaliativas (**V1, V2, P1, V3, V4, TF e V5**), consistindo em um processo de aprendizagem continuada. Cada avaliação será realizada em laboratório e englobará toda a matéria apresentada anteriormente (conteúdo acumulativo).

Cada uma das avaliações identificadas como virtuais (**V1, V2, V3, V4 e V5**) exige que o estudante tenha atingido a situação de acompanhamento SATISFATÓRIO no Sistema de Apoio Educacional (SAE), em todos os conteúdos que estarão sendo avaliados por cada uma destas avaliações. Somente com esta situação atendida o resultado da avaliação em **V1, V2, V3, V4 ou V5** será contabilizada na Média Final (**MF**). Caso isso **não** venha a ocorrer o resultado de qualquer uma dessas avaliações (**V1, V2, V3, V4, V5**) será **ZERO** para cada estudante que não atender a essa exigência da disciplina.

Um outro item a ser incluído na avaliação de cada estudante é a participação dentro e fora da sala de aula, com a elaboração, apresentação e entrega de trabalhos, listas e exercícios (**E**) no decorrer de todo o período letivo dessa disciplina.

A atividade avaliativa Trabalho Final (**TF**) será definida pelo docente no momento indicado pelo item 7 (Cronograma de Atividades) desse Plano de Ensino e poderá envolver todo conteúdo explorado por essa disciplina.

Assim, a Média Final (**MF**) na disciplina será calculada respeitando a seguinte equação:

$$MF = ((V1 \times 0,03) + (V2 \times 0,045) + (P1 \times 0,225)) + ((V3 \times 0,09) + (V4 \times 0,12) + (TF \times 0,27) + (V5 \times 0,12)) + (E \times 0,1)$$

Para ser **aprovado** na disciplina o estudante deverá:

- Obter pelo menos **75%** de presença nas aulas;
- Possuir nota igual ou superior a **4,0** (quatro pontos) na atividade **TF** (Trabalho Final);
- Atingir Média Final (**MF**) maior ou igual a **5,0** pontos.

Os estudantes poderão participar de qualquer atividade avaliativa desde que cheguem no local da aplicação até dez minutos depois do início da mesma, sendo seu prazo limite para encerramento devidamente informado pelo professor, após o início de cada prova (atividade avaliativa). As atividades extraclasse envolvem a realização de exercícios, sua implementação e a entrega em ambiente virtual no padrão lecionado e exigido pelo professor. Só assim serão considerados realizados a contento aos objetivos da disciplina. As atividades só são consideradas realizadas quando forem entregues, exatamente, no ambiente em que foram solicitadas, respeitando seus prazos limites de elaboração e entrega definidos e divulgados pelo professor, além dos padrões adotadas pela disciplina.

A impossibilidade de comparecimento de qualquer estudante a uma das atividades avaliativas deverá ser comprovada por documento oficial e original (Atestado Médico, Declaração de Serviço Militar, etc.) indicando o motivo justificável perante a lei de sua ausência, em que será possível que este estudante realize uma Prova de Reposição (**PR**) na data previamente agendada para o final deste semestre letivo (ver Cronograma de Atividades). Todo o conteúdo lecionado por esta disciplina fará parte desta avaliação (**PR**).

7. CRONOGRAMA DE ATIVIDADES

SEMANA	DATA / SALA	ATIVIDADES / CONTEÚDOS
1ª	10/03 / I-07	Apresentação do Plano de Ensino Introdução ao Paradigma de Programação Orientado a Objeto com Linguagem Java Acesso e configuração do MOODLE Enquete Pessoal e possíveis Negócios das famílias dos estudantes da turma Acesso ao CAE Disponibilização no CAE dos tópicos iniciais de estudo na disciplina
	12/03 / I-07	Ambiente de Desenvolvimento Orientação sobre a instalação do ambiente de desenvolvimento (Eclipse e Java) Atividade de Programação e preparação do ambiente para a disciplina (Eclipse e Java) Tipos Primitivos e Operadores Convenções Controle de Fluxo (condicional e repetição)

2ª	17/03 / I-07	Introdução a <i>Arrays</i> Exercícios de fixação envolvendo Controle de Fluxo
	19/03 / I-07	Introdução a <i>String</i> Exercícios de fixação envolvendo Controle de Fluxo
3ª	24/03 / I-07	Classes e Objetos Declarações de classes e métodos Instanciação e referência de objetos
	26/03 / I-07	<i>Arrays</i> com mais dimensões e <i>Strings</i> mutáveis Introdução a Exceções (tratamento)
4ª	31/03 / I-07	Classes e Objetos (continuação) Modelagem e Notação UML (Casos de Uso) PRIMEIRA AVALIAÇÃO VIRTUAL (V1)
	02/04 / I-07	Encapsulamento
5ª	07/04 / I-07	Pacotes (na organização da implementação) Exercícios de Fixação
	09/04 / I-07	Coleções - Vector <i>Generics</i> (parametrização)
6ª	14/04 / I-07	Coleções <i>ArrayList</i>
	16/04 / I-07	Herança (início)
7ª	21/04 / I-07	Feriado – Tiradentes e Aniversário de Brasília
	23/04 / I-07	Exercícios de Revisão SEGUNDA AVALIAÇÃO VIRTUAL (V2)
8ª	28/04 / I-07	PRIMEIRA AVALIAÇÃO (P1)
	30/04 / I-07	Correção da Avaliação P1 Janela de Diálogo DIVULGAÇÃO DO TRABALHO FINAL
9ª	05/05 / I-07	Herança (continuação)
	07/05 / I-07	Polimorfismo Notação UML (Diagrama de Classe)
10ª	12/05 / I-07	Classe Abstrata TERCEIRA AVALIAÇÃO VIRTUAL (V3)
	14/05 / I-07	Interface
11ª	19/05 / I-07	Coleções (<i>Collection</i>) Exceções (propagação e criação)
	21/05 / I-07	Comparadores
12ª	26/05 / I-07	Programação com interface gráfica (GUI)
	28/05 / I-07	Programação com Eventos Padrões de Projeto (MVC)
13ª	02/06 / I-07	Armazenamento Secundário (BD) Padrão DAO QUARTA AVALIAÇÃO VIRTUAL (V4)
	04/06 / I-07	Exercícios de Fixação
14ª	09/06 / I-07	Introdução ao Desenvolvimento na Web (HTML e HTTP)
	11/06 / I-07	Feriado – Corpus Christi
15ª	16/06 / I-07	Introdução ao Desenvolvimento Java na Web (<i>Servlets</i>)
	18/06 / I-07	Introdução ao Desenvolvimento Java na Web (JSP)
16ª	23/06 / I-07	Acompanhamento com apresentação dos Trabalhos Finais (TF)
	25/06 / I-07	Apresentação e Entrega dos Trabalhos Finais (TF)
17ª	30/06 / I-07	Apresentação e Entrega dos Trabalhos Finais (TF) QUINTA AVALIAÇÃO VIRTUAL (V5) Pesquisa Educacional na disciplina
	02/07 / I-07	PROVA DE REPOSIÇÃO (PR)
18ª	07/07 / I-07	Divulgação dos Resultados Finais
	09/07 / I-07	Possível aula de reposição prevista no calendário da UnB (se for necessário)

8. BIBLIOGRAFIA

BÁSICA

STROUSTRUP, B. The C++ Programming Language, 4th Edition, Addison-Wesley Professional, 2013.

HORSTMANN, C. S.; CORNELL, G. Core Java, Volume I - Fundamentals, 8th Edition, Prentice Hall, 2016.

(eBrary) LANO, K. UML 2 Semantics and Applications, Wiley, 2009.

COMPLEMENTAR

IAN D. CHIVERS, An Introduction to C++ and Object Oriented Programming, Springer, 2001.

(eBrary) BARCLAY, K.; SAVAGE, J. Object-Oriented Design with UML and Java, Elsevier, 2003.

(eBrary) SHAUGHNESSY, P. Ruby Under a Microscope: An Illustrated Guide to Ruby Internals, No Starch Press, 2013.

(eBrary) PHILLIPS, D. Python 3 Object Oriented Programming, Packt Publishing Ltd, 2010.

DEITEL, H. M., DEITEL, P. J. Java: Como Programar, 8 ed. Pearson do Brasil, 2010.

HORSTMANN, C. S. e CORNELL, G. Core Java2, volume 2, Makron Books, 2001.

9. OBSERVAÇÕES

1. Conforme a conveniência, a data das avaliações e a distribuição dos conteúdos podem ser alteradas, desde que com prévia comunicação e anuência dos estudantes.
2. O estudante que faltar a uma das atividades avaliativas terá direito a fazer uma **Prova de Reposição (PR)** prevista no Cronograma de Atividades, desde que apresente atestado/declaração que justifique a ausência e seja coerente aos aspectos legais que aceitam tal ausência como justificada no Brasil.
3. As aulas marcadas, ou sombreadas, no Cronograma de Atividades (item 7) identificam datas de recesso, feriados ou atividades complementares, podendo prever atividades de estudo dirigido por meio da leitura de material indicado pelo professor, em que a realização de tarefas, fora do horário de aula, será necessária e esclarecida pelo professor da disciplina.
4. Caso haja necessidade de reposição de aula no decorrer desse semestre letivo, as mesmas serão ministradas em datas e horários não previstos neste cronograma (item 7), sendo estas, antecipadamente, combinadas entre o professor e os estudantes desta disciplina/turma.