



Universidade Federal de Goiás
Instituto de Informática
Bacharelado em Engenharia de Software
Programação Orientada a Objetos

Componente Curricular

Código Disc.	Nome	Carga Horária	
		Teórica	Prática
10000082	Programação Orientada a Objetos	32	32

Turma A: 2020/1

Professor: Fábio Nogueira de Lucena & Plínio de Sá Leitão Júnior

EMENTA DA DISCIPLINA

1. Abstração e tipos abstratos de dados.
2. Classes, métodos, encapsulamento, interface. Mensagens, instâncias e inicialização. Herança e composição. Polimorfismo.
3. Uso de uma linguagem orientada a objetos.
4. Noções de UML.
5. Noções de padrões de projeto orientado a objetos.

OBJETIVOS DA DISCIPLINA

Objetivo geral

Aplicar a orientação a objetos no desenvolvimento de programas.

Objetivos específicos

- Conhecer e compreender o modelo orientado a objetos.
- Desenvolver a habilidade necessária para modelar problemas conforme o modelo orientado a objetos.
- Especificar modelos orientados a objetos usando a UML.
- Mapear um modelo orientado a objetos para código correspondente em uma linguagem de programação.

PROGRAMA		
<ul style="list-style-type: none"> • Contexto de desenvolvimento de software. • Origem (e motivação) de orientação a objetos. • Definir orientação a objetos (OO). • Terminologia de orientação a objetos. • UML. • Classes. Objetos. Instâncias. • Identificação de classes. • Métodos. Mensagens. • Associações. • Agregação. • Composição. • Herança. • Polimorfismo. • Implementação em Java 		
LEGENDA DOS PROCEDIMENTOS DIDÁTICOS		
Legenda	Descrição	Objetivo
AEX	Aula teórica	Transmitir conhecimento utilizando quadro ou slides.
AP	Aula prática	Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica.
ED	Estudo dirigido	Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise.
OTR	Outros	Transmitir conhecimento utilizando quadro ou slides.
RE	Aula teórica com resolução de exercícios	Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções de problemas.
SE	Seminários	Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo.
TG	Trabalho em grupo	Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo.
FERRAMENTAS E RECURSOS PARA O ENSINO REMOTO		
Fórum	Pode ser utilizado isolado ou associado a outras ferramentas em atividades dirigidas. Nele, os participantes podem expressar sua opinião. É uma ferramenta assíncrona. Sua utilização considera aspectos qualitativos e quantitativos.	
Diário	Permite aos participantes postarem suas reflexões acerca de um tema e o relato, de modo assíncrono, do seu processo de aprendizagem. Possibilita a interação apenas entre estudante e professor-tutor.	
Wiki	Ferramenta assíncrona de escrita colaborativa. Permite edição coletiva dos documentos e atualização dinâmica. É necessário estar articulada a outra ferramenta, como o fórum e o <i>chat</i> , para que os participantes possam organizar suas ideias e traçar suas metas.	

Chat	Ferramenta de comunicação síncrona, exigindo que os participantes da discussão estejam conectados simultaneamente, de modo síncrono, para que o processo de comunicação seja efetuado; bate-papo por meio de aplicativos de <i>instant message</i> (exemplo: Whats app, Telegram, Discord, etc.).
Lista de discussão	Ferramentas de comunicação assíncronas. Caracteriza-se pelo recebimento e envio de mensagens por e-mail.
Blog/ Videoblog	Páginas pessoais da Internet cujo mecanismo possibilita registrar e atualizar em ordem cronológica, opiniões, fatos, emoções, imagens, além de outros conteúdos que se queira disponibilizar;
Tarefa	Consiste na descrição ou no enunciado de uma atividade a ser desenvolvida pelo participante, de modo assíncrono, que pode ser enviada em formato digital ao servidor da plataforma e será verificada posteriormente pelo professor-tutor;
Aula remota	Aula (preleção, demonstração, orientação) mediada por ambiente virtual em que professor e estudantes participam de modo síncrono, contendo recursos de interatividade (áudio, vídeo e/ou <i>chat</i>). Exemplos: Google Meet, MS Teams, Zoom, BBB, Whereby, etc.)
Glossário	Ferramenta que permite criar e atualizar uma lista de definições como em um dicionário, de modo assíncrono.
Questionário	Ferramenta assíncrona de recepção de materiais, possibilitando ao estudante submeter arquivos digitais, para posterior avaliação e <i>feedback</i> do professor.
Podcast/ Videocast	Ato de difundir mensagens de áudio e/ou vídeo pela Internet (ou de modo listado) em servidores de vídeos (Youtube, SnapTube, etc.). Permite ao participante criar arquivos de áudio ou vídeo para publicação e/ou difusão de conhecimentos, episódios, demonstrações, etc.
Enquete online	Recurso de interação para obtenção de respostas acerca de um tema, pesquisa ou questão posta no modo síncrono.

PROCEDIMENTOS METODOLÓGICOS

A disciplina possui um momento síncrono (25%) e o restante assíncrono (75%). De forma simplificada, serão disponibilizados tópicos e materiais, além de exemplos e atividades sobre as quais serão computadas as presenças e a nota final na disciplina. A modalidade específica empregada por cada atividade será definida à medida que insumos forem coletados ao longo da execução da disciplina. **O ensino remoto reflete um momento excepcional que demanda uma abordagem também excepcional, detalhada doravante.**

Toda terça-feira, das 19h às 20h ocorrerá o momento síncrono da disciplina. Este momento síncrono contará como 25% da carga horária da disciplina e, em particular, 25% das presenças. Ele ocorrerá por meio da sala indicada abaixo:

<https://meet.google.com/lookup/awkznsp2o3>

OS DEMAIS 75% SERÃO REALIZADOS DE FORMA ASSÍNCRONA.

Em conformidade com *Toward Understanding Students' Learning Performance in an Object-Oriented Programming Course: The Perspective of Program Quality*, SUN, Q. et al, IEEE Access, vol 8, 2020 ([aqui](#)) ([pdf](#)), várias atividades serão realizadas pelos estudantes, cobrindo a ementa da disciplina, à medida que cada tópico é apresentado, juntamente com exemplos suficientes para a realização de cada atividade.

Nesta edição da disciplina, portanto, o ciclo “Apresentação de tópico”, seguido de “Exemplos” e “Atividades realizadas pelos estudantes” ocorrerão ao longo de toda a disciplina. Nenhuma destas ações será síncrona. Ou seja, no portal da disciplina (<https://github.com/kyriosdata/oo>) e ao longo da mesma, serão disponibilizados a “Apresentação de tópico” e “Exemplos”, juntamente com as “Atividades a serem realizadas pelos estudantes”. O resultado da execução destas últimas será empregado tanto para a avaliação (nota) quanto a presença.

A EXECUÇÃO DE CADA ATIVIDADE RESULTA EM ARTEFATOS. Por exemplo, texto, diagramas, ou código, dentre outros. TODO ARTEFATO PRODUZIDO PELO ESTUDANTE DEVE SER REGISTRADO NO GITHUB. O portal do Github é <https://github.com>.

Novamente, você, estudante, deverá registrar TUDO o que produzir na realização das atividades previstas na sua conta e em projeto específico no Github. Você será orientado, aula por aula, sobre o que é esperado que você faça (atividade), quais artefatos deve produzir e como usar o Github para efetuar o registro correspondente.

A interação entre docentes e estudante, nesta disciplina, será mediada pelo Github. O que você registrar (ou não), assim como a resposta (ou não) dos docentes, ficará registrado.

Observações:

1. O Github apóia o desenvolvimento de software por equipe possivelmente distribuída.
2. O Github é de uso gratuito, e seguramente está entre os principais serviços empregados no mundo para desenvolvimento de software.
3. O Github será nosso Ambiente Virtual de Aprendizagem (AVA) nesta disciplina.
4. Programação Orientada a Objetos, nossa disciplina, é 100% desenvolvimento de software e, mais especificamente, construção de software.
5. O uso do Github exige acesso a internet similar ao do ambiente Turing (Instituto de Informática).
6. O ensino remoto cria uma “equipe virtual de desenvolvimento de software”, formada pelos docentes e por você, estudante. Nossa equipe está distribuída, e o Github contempla este cenário.

CRITÉRIO DE AVALIAÇÃO

A avaliação de desempenho e também da frequência (presenças) do estudante na disciplina é determinada pelo registro dos artefatos definidos para cada uma das atividades previstas. Novamente, ao estudante será apresentado um tópico, acompanhado de exemplos e das atividades correspondentes. O registro no Github dos artefatos correspondentes e, naturalmente, o conteúdo destes artefatos será empregado na definição do desempenho e da frequência do estudante na disciplina.

A avaliação é contínua. Todo tópico (aula) tem uma ou mais atividades associadas. Todas elas serão avaliadas. Ou seja, teremos tantas avaliações quanto tópicos e, possivelmente, um por semana, por exemplo. A avaliação produz um resultado COM APROVEITAMENTO ou SEM APROVEITAMENTO. A nota final na disciplina será definida com base no total de atividades para as quais o resultado foi COM APROVEITAMENTO. Por exemplo, se forem definidas 10 atividades e você teve aproveitamento em 6 ou mais delas, então será considerado aprovado. Caso contrário, será considerado reprovado por média. Formalmente, a nota final é a razão entre o total de atividades com aproveitamento e o total de atividades, multiplicada por 10.

Toda atividade tem um prazo bem-definido para a sua realização. Atividade realizada fora do prazo não será considerada, nem na perspectiva do desempenho nem tampouco da frequência. Aquelas realizadas no prazo estipulado, mesmo que o estudante não obtenha aproveitamento, conforme avaliação dos docentes, receberá a frequência correspondente.

Observações:

(1) Estará aprovado o(a) aluno(a) que atingir nota final (NF) igual ou superior a 6.0 (seis pontos) e frequência às atividades acadêmicas igual ou superior a 75% da carga horária da disciplina. Se tivermos 10 tópicos, então terá que ter aproveitamento de suas atividades em pelo menos 8 destes tópicos.

(2) Será atribuída a nota 0 (zero) a qualquer atividade avaliativa não realizada na data estipulada. Adicionalmente, a presença correspondente (frequência), não será computada, neste caso.

(3) Todas as atividades assíncronas são supervisionadas. Os docentes irão acompanhar o que foi feito ou não, por meio do Github.

(4) Os alunos que se envolverem em plágio (desvios de conduta, seja como facilitador ou como beneficiário) receberão nota 0 (zero) para a atividade correspondente. O caso será levado ao conhecimento da Coordenação do Curso, do Núcleo Docente Estruturante e do Conselho Diretor do Instituto de Informática para as providências cabíveis e legais. O pedido de segunda chamada deverá ser protocolado conforme condições estipuladas na Resolução CONSUNI específica (RGCG) em vigor.

(5) Este Plano está amparado pelas normativas e portarias emanadas dos órgãos governantes superiores, pelas resoluções, instruções normativas e diretrizes didático-pedagógicas da UFG e do INF, em vigor, que definem e regulam o funcionamento do ensino remoto excepcional.

BIBLIOGRAFIA

Bibliografia Básica

1. Morelli, R. e WALDE, R., Java, Java, Java: Object Oriented Problem Solving, 3rd edition, disponível gratuitamente em <http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf>.

BORATTI, I. C. Programação orientada a objetos em Java. Visual Books, 2007.

DEITEL, P. J; DEITEL, H. M. Java como programar 6.ed. São Paulo:Prentice Hall, 2005.

Bibliografia Complementar

1. *On the Criteria To Be Used in Decomposing Systems into Modules*, D. L. Parnas, CACM, 15 (12), 1972. Disponível [aqui](#).
2. *Software Structures: A careful look* ([aqui](#)) ([pdf](#)).

3. *Toward Understanding Students' Learning Performance in an Object-Oriented Programming Course: The Perspective of Program Quality*, SUN, Q. et al, IEEE Access, vol 8, 2020 ([aqui](#)) ([pdf](#))