



UNIVERSIDADE FEDERAL DO ABC – UFABC
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PLANO DE ENSINO

para o Quadrimestre Suplementar (QS) – Resolução CONSEPE 240/2020

ANO LETIVO	QUADRIMESTRE	TURNO	CAMPUS
2020	QS	Noturno	Santo André

CÓDIGO	NOME
MCTA018-13SA	Programação Orientada a Objetos
Turma	RECOMENDAÇÃO
NA3MCTA018-13SA	Processamento da Informação Programação Estruturada

EMENTA

Conceitos básicos: classes, objetos, mensagens, encapsulamento, herança, polimorfismo. Programação orientada a objetos utilizando uma linguagem de programação orientada a objetos (Java). Análise e projeto orientados a objetos. UML. Padrões de projeto de software.

SOBRE DA DISCIPLINA

A disciplina será conduzida no ambiente virtual de aprendizagem (AVA) Moodle UFABC:

<https://moodle.ufabc.edu.br/>

Curso: **NA3MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2020.QS**

Fique atento aos comunicados realizados no Moodle UFABC, que será utilizado para a condução da disciplina.

Mapa de Atividades

Disciplina: Programação Orientada a Objetos – T-P-I 2-2-4

Docente: Paulo Henrique Pisani

Quadrimestre: 2020.QS

Descrição das atividades no mapa de atividades:

- **Horas:** a coluna horas na tabela é soma de T+P. Contudo, pode ser necessário tempo adicional para estudo, conforme indicado no T-P-I da disciplina.
- **Assistir videoaulas (gravadas):** serão disponibilizados links para videoaulas sobre o conteúdo da disciplina. É necessário um navegador web com capacidade de executar vídeos armazenados no Google drive (alguns vídeos podem estar no Youtube). Durante as videoaulas, além de apresentar o conteúdo, podem ser realizados exemplos práticos de implementação dos conceitos. Ao longo dos vídeos, poderão ser propostos exercícios (não avaliativos).
- **Ler material:** será disponibilizado material para leitura (por exemplo, slides da aula em PDF ou outro material sobre o conteúdo da disciplina). É recomendável ler o material, além de assistir às videoaulas.
- **Exercícios para submissão pelo Moodle:** serão disponibilizados exercícios ao longo do quadrimestre. Todos os exercícios são individuais (devem ser resolvidos pelo próprio aluno). Os exercícios deverão ser submetidos pelo Moodle. Cada exercício pode ter pontuação máxima e prazo de entrega diferentes (essas informações estarão em cada exercício no Moodle). Os exercícios de programação para nota possuem a indicação **[EP]** no Moodle.

Avaliação dos exercícios de programação:

Poderá ser utilizado um sistema de correção automática. A princípio, a correção automática (caso esteja disponível) aplicará o seguinte critério:

- *Exercícios com pontuação máxima até 1 ponto:* é necessário passar em todos os casos de teste para obter 1 ponto. Caso contrário, ficará com zero.
- *Exercícios com pontuação máxima superior a 1 ponto:* se não passar nos casos de teste abertos (que tem entrada e saída disponíveis/abertas), ficará com zero. Se passar nos casos de teste abertos, a nota poderá ser maior que zero. Dessa forma, estes exercícios podem receber pontuação parcial.
- **Importante:** o professor poderá atribuir nota manualmente nos exercícios e alterar a nota atribuída pelo sistema de correção automática também (por exemplo, devido a pontos que o corretor automático não pode identificar ou problemas/erros na avaliação).

Semana	Horas (T+P)	Tema / Subtema	Objetivos específicos	Atividades teóricas	Atividades práticas
01 21/09/2020 a 27/09/2020	4h	Introdução ao Java <ul style="list-style-type: none"> • Tipos de dados • Entrada/Saída • Operadores aritméticos, relacionais, lógicos • Estrutura condicional • Estruturas de repetição • Vetores e matrizes • Métodos • Estilo de codificação 	Aplicar a linguagem Java para implementação de programas, incluindo os programas propostos na disciplina.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
02 28/09/2020 a 04/10/2020	4h	Classes, objetos, encapsulamento, UML <ul style="list-style-type: none"> • Classes, atributos e métodos • Classe vs Instância • Modificadores de acesso • Pacotes • UML (representação de uma classe) • UML (relacionamentos: associação, agregação e composição) 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
03 05/10/2020 a 11/10/2020	4h	Construtores, sobrecargas <ul style="list-style-type: none"> • Construtores • Sobrecargas 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

04 12/10/2020 a 18/10/2020	4h	Herança, polimorfismo <ul style="list-style-type: none"> • Herança • Polimorfismo • UML (herança no diagrama de classes) 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
05 19/10/2020 a 25/10/2020	4h	Interfaces, polimorfismo <ul style="list-style-type: none"> • Classe abstrata • Interfaces • UML (interface no diagrama de classes) 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
06 26/10/2020 a 01/11/2020	4h	Construtores, cópia de objetos, final <ul style="list-style-type: none"> • Chamada a outros construtores • Acesso a construtores • Cópia de objetos • Modificador <i>final</i> 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

07 02/11/2020 a 08/11/2020	4h	Estruturas lineares no contexto de orientação a objetos <ul style="list-style-type: none"> • Modificadores de acesso em classes • Aplicação dos conceitos vistos em estruturas lineares 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
---	----	---	--	--	--

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

08 09/11/2020 a 15/11/2020	4h	Tratamento de exceções <ul style="list-style-type: none"> • Hierarquia de exceções • Estruturas para lançar e capturar exceções 	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
---	----	---	--	--	--

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

09 16/11/2020 a 22/11/2020	4h	Generics	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
---	----	----------	--	--	--

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

10	4h	Padrões de projeto de software	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
23/11/2020 a 29/11/2020					

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

11	4h	Android	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
30/11/2020 a 06/12/2020					

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

12	4h	Revisão	Entender os conceitos de programação orientada a objetos apresentados até esta semana. Aplicar a linguagem Java nos problemas propostos na disciplina, implementando com uso dos conceitos de programação orientada a objetos apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	
07/12/2020 a 12/12/2020					

Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.

AVALIAÇÃO

Avaliação do Período Letivo Regular: a média final será composta por exercícios de programação passados ao longo do quadrimestre. Cada exercício pode ter pontuação máxima diferente assim como prazos de entrega diferentes. Os exercícios de programação para nota possuem a indicação **[EP]** no Moodle. Exercícios não entregues no prazo receberão nota zero. Todos os exercícios são individuais (devem ser resolvidos pelo próprio aluno). Será utilizada ferramenta para identificação de submissões com alta similaridade no Moodle.

A média final (MF) será a média ponderada da pontuação obtida em todos os exercícios:

$$MF = 10 \times \frac{\text{soma da pontuação obtida nos exercícios (EPs)}}{\text{soma da pontuação máxima dos exercícios (EPs)}}$$

A partir da média final (MF), o conceito final (CF) será obtido por meio da seguinte tabela:

Conceito Final (CF)	Média Final (MF)
A	$MF \geq 9$
B	$9 > MF \geq 7,5$
C	$7,5 > MF \geq 6$
D	$6 > MF \geq 5$
F	$MF < 5$
O	(reprovação por ausência de avaliação)

Mecanismo de Recuperação (REC):

Estarão habilitados para o mecanismo de recuperação os alunos que obtiverem conceito final **D** ou **F**, obedecendo as regras indicadas na Resolução CONSEPE no. 182 de 23 de outubro de 2014.

Como mecanismo de recuperação, o aluno terá a possibilidade de submeter exercícios passados ao longo do quadrimestre. A nota obtida em cada exercício entregue substituirá a nota previamente obtida naquele exercício. A submissão será por meio de atividade para recuperação no Moodle.

Prazo para entrega dos exercícios (recuperação): **18/12/2020**.

Após a substituição das notas dos exercícios entregues, a média final com recuperação (MR) será obtida com a mesma fórmula usada para a média final (considerando todos os exercícios passados do quadrimestre). Será aplicada a tabela de conversão de conceito a seguir (o conceito final com recuperação será no máximo C):

Conceito Final com Recuperação (CFR)	Média Final com Recuperação (MR)
C	$MR \geq 6$
D	$6 > MR \geq 5$
F	$MR < 5$

Caso seja identificado plágio em qualquer exercício entregue, será atribuído conceito final F.

FERRAMENTAS

Para acompanhar a disciplina é necessário computador com acesso à internet:

- Recomenda-se o uso do sistema operacional Ubuntu Linux 18.04 64-bit;
- Caixa de som ou fone de ouvido;
- Navegador web compatível com o Moodle UFABC e Google Meet; O navegador web deve permitir a execução de vídeos a partir do Google drive e Youtube.
- Editor de texto (por exemplo, gedit, Notepad++, etc).
- Java Development Kit: necessário para desenvolvimento de programas em Java - é recomendado o Open Java Development Kit (OpenJDK) 11 – versão disponível no Ubuntu Linux 18.04 64-bit.
- Opcional: Android Studio 4.0.1 (será usado na semana 11, mas não haverá atividade avaliativa que precisará do Android Studio para ser realizada).

ATIVIDADES DE APOIO

O professor estará disponível nos seguintes dias/horários para atendimento aos alunos (dúvidas sobre o conteúdo da disciplina):

- **terça-feira: das 19h às 21h** – atendimento por videoconferência;
- **quinta-feira: das 18h às 20h** – atendimento por chat ou por e-mail.

Para o atendimento por videoconferência, será utilizado o Google Meet (a ferramenta poderá ser outra a critério do professor). Haverá link para acesso disponível na página da disciplina no Moodle. Para o atendimento por chat, será usado o chat do Moodle.

Em dias que forem feriados, o atendimento não será síncrono. Contudo, será possível enviar dúvidas para resposta em um momento posterior (de forma assíncrona). O chat do Moodle, por exemplo, permite o envio de mensagens de forma assíncrona também.

O atendimento síncrono depende de disponibilidade de internet. Em caso de indisponibilidade de internet no momento do atendimento, a resposta para as dúvidas ocorrerá de forma assíncrona. Os alunos podem enviar suas perguntas por e-mail ou pelo chat do Moodle e o professor poderá responder assim que possível.

BIBLIOGRAFIA

Bibliografia básica:

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro, RJ: Campus, 2005.
- DEITEL, H. M.; DEITEL, P. J. Java: como programar. 6. ed. Porto Alegre, RS: Bookman, 2005.
- GUEDES, G. T. A. UML 2: uma abordagem prática. São Paulo, SP: Novatec, 2009.

Bibliografia complementar:

- BARNES, D. J.; KOLLING, M. Programação orientada a objetos com Java. 4. ed. São Paulo, SP: Pearson Prentice Hall, 2009.
- BRUEGGE, B.; DUTOIT, A. H. Object-oriented software engineering: using UML, patterns, and Java. 2. ed. Upper Saddle River, NJ: Prentice Hall, 2003.

- FLANAGAN, D. Java: o guia essencial. 5. ed. Porto Alegre, RS: Bookman, 2006.
- FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre, RS: Bookman, 2005.
- LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento interativo. 3. ed. Porto Alegre, RS: Bookman, 2007.

PROFESSOR(ES) RESPONSÁVEL(IS)

Prof. Dr. Paulo Henrique Pisani