



PLANO DE ENSINO

ANO LETIVO	QUADRIMESTRE	TURNO	CAMPUS
2022	Q2	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.

COMPETÊNCIA

A disciplina deve permitir que o aluno seja capaz de compreender os conceitos básicos da programação orientada a objetos e aplicar adequadamente os conceitos e técnicas de programação orientada a objetos no projeto e desenvolvimento de sistemas computacionais.

SOBRE A DISCIPLINA

T-P-I: 2-2-4

Docente: Paulo Roberto Miranda Meirelles | paulo.meirelles@ufabc.edu.br

Horário semanal:

- Terça-feira: 19h às 21h (teórica/sala: 307-2)
- Quinta-feira: 21h às 23h (prática/lab: 404-2)

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

- Curso: NA3MCTA018-13SA - Programação Orientada a Objetos - Paulo Meirelles - 2022.2

Atividades previstas

Semana	Horas	Tópicos	Atividades
01 07/06 e 09/06	4h	<ul style="list-style-type: none"> ● Apresentação da disciplina ● Software Livre ● Controle de versão ● Ambiente de desenvolvimento ● Introdução ao Java 	<ul style="list-style-type: none"> ● Aulas ● Ler material ● Exercício (Lab)
02 14/06 - 16/06: Feriado	2h	<ul style="list-style-type: none"> ● Introdução à Orientação à Objetos ● Classes, Objeto, Encapsulamento, UML 	<ul style="list-style-type: none"> ● Aula ● Ler material
03 21/06 e 23/06	4h	<ul style="list-style-type: none"> ● Princípios básicos de Orientação à Objetos e Modelagem ● Classes: Membros estáticos ● UML 	<ul style="list-style-type: none"> ● Aulas ● Ler material ● Exercício (Lab)
04 28/06 e 30/06	4h	<ul style="list-style-type: none"> ● Testes automatizados 	<ul style="list-style-type: none"> ● Aulas ● Ler material ● Exercício (Lab)
05 05/07 e 07/07	4h	<ul style="list-style-type: none"> ● Herança ● Modificadores de acesso ● Construtores ● Encapsulamento ● Métodos de acesso ● Classes abstratas ● UML 	<ul style="list-style-type: none"> ● Aulas ● Ler material ● Exercício (Lab)
06 12/07 e 14/07	4h	<ul style="list-style-type: none"> ● Revisão 1 ● Prova 1 (14/07) 	<ul style="list-style-type: none"> ● Aula ● Prova 1

07	4h	<ul style="list-style-type: none"> • Classes Abstratas e Polimorfismo 	<ul style="list-style-type: none"> • Aulas • Ler material • Exercício (Lab)
19/07 a 21/07			
08	4h	<ul style="list-style-type: none"> • Interface e Polimorfismo 	<ul style="list-style-type: none"> • Aulas • Ler material • Exercício (Lab)
26/07 e 28/07			
09	4h	<ul style="list-style-type: none"> • Tipos genéricos • Coleções • Tratamento de exceções 	<ul style="list-style-type: none"> • Aulas • Ler material • Exercício (Lab)
02/08 a 04/08			
10	4h	<ul style="list-style-type: none"> • Análise e projeto orientados a objetos • Padrões de Projeto 	<ul style="list-style-type: none"> • Aulas • Ler material • Exercício (Lab)
09/08 a 11/08			
11	4h	<ul style="list-style-type: none"> • Revisão 2 • Prova 2 (18/08) 	<ul style="list-style-type: none"> • Aula • Prova 2
16/08 a 18/08			
12	6h	<ul style="list-style-type: none"> • Revisão de notas • Avaliação substitutiva (25/08) • Recuperação (29/08, segunda-feira) 	<ul style="list-style-type: none"> • Prova substitutiva • Prova de recuperação
23/08 e 25/08 + 29/08			

Onde:

- **Exercícios (Lab):** serão disponibilizados exercícios ao longo do quadrimestre. Todos os exercícios são individuais (devem ser resolvidos pelo próprio discente). Os exercícios deverão ser submetidos via Github Classroom (teremos uma aula sobre controle de versão e o uso do Git, e o discente deve criar um usuário no Github).
 - Cada exercício pode ter pontuação máxima e prazo de entrega diferentes (essas informações estarão disponíveis no enunciado).
 - Os exercícios serão anunciados em Java, poderão ter como modelo um código em Java, acompanhado de testes automatizados.
 - Os repositórios com os códigos usados da disciplina estarão disponíveis em <https://github.com/UFABC-OO>.
 - Poderá ser utilizado um sistema de correção automática (do Github Classroom) para os exercícios (Lab).
 - Todos os exercícios (Lab) serão disponibilizados no Moodle com antecedência de, no mínimo, 72 horas do prazo de entrega.

Avaliação do período letivo regular

A Média Final (MF) será composta por:

- **Avaliação P1 (peso 30%):** 14/07/2022 às 21h (avaliação presencial, realizada durante o horário da aula). Será atribuída uma nota de 0 a 10 na Avaliação P1.
- **Avaliação P2 (peso 40%):** 18/08/2022 às 21h (avaliação presencial, realizada durante o horário da aula). Será atribuída uma nota de 0 a 10 na Avaliação P2.
- **Média de exercícios (peso 30%):** média das notas obtidas em todos os exercícios (MLabs) disponibilizados no Github Classroom (a média será uma nota de 0 a 10). Exercícios não entregues no prazo receberão nota zero.
- **Bônus:** poderá ser adicionado um bônus de até 0,5 ponto na média final para presença de quem comparecer em mais de 60% das aulas: 100% = 0,5 | 90% = 0,4 | 80% = 0,3 | 70% = 0,2 | 60% = 0,1

A Média Final (MF) será obtida da seguinte forma:

$$MF = (P1 * 0,30) + (P2 * 0,4) + (0,3 * MLabs) + Bônus$$

Conceito final

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

- **A** = $MF \geq 9,0$
- **B** = $9,0 > MF \geq 8,0$
- **C** = $8,0 > MF \geq 7,0$
- **D** = $7,0 > MF \geq 6,0$
- **F** = $MF < 6,0$
- **O** = reprovação por ausência de avaliação

Atividade Substitutiva (SUB)

Estarão habilitados para a avaliação substitutiva os discentes que se ausentaram de uma das avaliações do período regular (P1 ou P2) e contemplados pelo benefício de acordo com a Resolução CONSEPE no. 227, de 23 de abril de 2018. Nessa hipótese, o aluno deverá apresentar uma justificativa válida para o docente pelo menos 72h antes do dia da avaliação substitutiva. A nota da avaliação SUB substitui a nota da P1 ou da P2.

- *Data da Avaliação Substitutiva (SUB): 25/08/2022 às 21h (avaliação presencial, realizada durante o horário da aula)*

Caso o discente se ausente de mais de uma avaliação do período regular, o conceito da avaliação substitutiva será concedido para UMA ÚNICA avaliação não realizada. Aqueles que fizerem todas as avaliações não terão direito à avaliação substitutiva.

Mecanismo de Recuperação (REC)

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

- *Data da Avaliação de Recuperação (REC): 29/08/2022 às 21h (avaliação presencial, realizada durante o horário da aula)*

A média final com recuperação (MR) é obtida da seguinte forma, em que REC é a nota (0 a 10) obtida na Avaliação de Recuperação:

$$MR = (MF * 0,5) + (REC * 0,5)$$

O Conceito Final com Recuperação (CFR) é obtido conforme tabela a seguir (o CFR será no máximo C):

- **C** = $MR \geq 7,0$
- **D** = $7,0 > MR \geq 6,0$
- **F** = $MR < 6,0$

Análise contra Plágio

Caso seja identificado plágio em qualquer exercício ou prova entregue, será atribuído Conceito Final igual a **O** (reprovação por ausência de avaliação) para todos os envolvidos. Algumas consequências que o aluno poderá sofrer estão descritas no Código de Honra anexo a este plano no Moodle.

Ambiente de desenvolvimento e ferramentas de apoio

- Computador com acesso à internet
- Navegador web compatível o Moodle da UFABC e Github Classroom
- Java Development Kit
 - recomendado o Open Java Development Kit (OpenJDK) 11
- IDE Eclipse (recomendendo), GEdit ou qualquer editor de sua preferência
- Recomenda-se o uso do sistema operacional GNU/Linux (Debian ou Ubuntu, por exemplo)

Atividades de apoio e atendimentos

Esta disciplina prevê um horário de atendimento extraclasse para atividades de apoio aos estudantes regulares desta turma, conforme disposto na Resolução CONSUNI nº 183, de 31 de outubro de 2017.

O horário de atendimento semanal será realizado no seguinte dia, local e horário, com agendamento:

- Quinta-feira, das 18h às 20h, sala 504-2, bloco A, Santo André (Prof. Paulo Meirelles)

Bibliografia

Básica:

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

Bibliografia complementar:

1. BARNES, D. J.; KOLLING, M. Programação orientada a objetos com Java. 4. ed. São Paulo, SP: Pearson Prentice Hall, 2009.
2. BRUEGGE, B.; DUTOIT, A. H. Object-oriented software engineering: using UML, patterns, and Java. 2. ed. Upper Saddle River, NJ: Prentice Hall, 2003.
3. 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.
4. 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.

Material de Apoio

1. Introdução a Ciência da Computação com Java e Orientação a Objetos
 - <http://ccsl.ime.usp.br/files/publications/files/2008/intro-java-cc.pdf>
2. Java e Orientação a Objetos - Caelum Apostila FJ11
 - <https://www.caelum.com.br/download/caelum-java-objetos-fj11.pdf>