

## MCTA018-13 Programação Orientada a Objetos

Turma A2 noturno (NA2MCTA018-13SA)

2º quadrimestre de 2022

Terça-feira, 19h às 21h; Quinta-feira, 21h às 23h

### Professor:

Bruno Augusto Dorta Marques

bruno.marques@ufabc.edu.br

### Informações básicas:

Os slides, vídeos e exercícios serão disponibilizados no Moodle.

O link para videochamadas será disponibilizado no Moodle.

Serão utilizadas as plataformas Google Meet, RNP ou outra similar.

Atenção: Houve uma mudança na oferta desta disciplina que será ofertada excepcionalmente na **modalidade remota**.

### Cronograma de atividades:

Semana 1 (06/06 a 11/06)	Apresentação do curso; Apresentação do ambiente de desenvolvimento; Introdução à linguagem Java
Semana 2 (13/06 a 18/06)	Revisão da linguagem Java; Tipagem; Bytecode; Entrada e Saída
Semana 3 (20/06 a 25/06)	Introdução à orientação a objetos; Definição de classes e manipulação de objetos; Encapsulamento; Modificadores de acesso
Semana 4 (27/06 a 02/07)	Linguagem UML; Projeto de software
Semana 5 (04/07 a 09/07)	<i>Estruturas de dados em Java: Arrays, Listas, Pilhas, Coleções; Conceitos de alocação em memória (Heap, Pilha);</i>
Semana 6 (11/07 a 16/07)	Herança; Polimorfismo; Sobrecarga; Classes abstratas; interfaces; Classes estáticas
Semana 7 (18/07 a 23/07)	Tratamento de exceções e asserções
Semana 8 (25/07 a 30/07)	Serialização de objetos e persistência
Semana 9 (01/08 a 06/08)	Applets; Interfaces Gráficas
Semana 10 (08/08 a 13/08)	Introdução a programação paralela em Java; Threads; sistema produtor-consumidor
Semana 11 (15/08 a 20/08)	Comunicação; Socket; sistema Cliente-Servidor
Semana 12 (22/08 a 27/08)	Entrega do Trabalho Final

## Metodologia:

- As aulas assíncronas serão gravadas e disponibilizadas no Moodle.
- Conteúdo suplementar será disponibilizado na forma de textos, slides, imagens e códigos fontes.
- Os encontros síncronos serão reservados para resolução de exercícios práticos e plantão de dúvidas.
- Comunicação extraclasse será feita através da plataforma Moodle através de mensagens e da funcionalidade Avisos e Fórum.

## Plantão de dúvidas e Encontros Síncronos:

Atendimento via Google Meet (link de convite no Moodle).

Necessário entrar com e-mail da UFABC.

No horário da aula: todas as terças-feiras das 19h às 21h

Plantão de dúvidas agendado: mediante marcação prévia com horário e dia a combinar

Para agendar um plantão de dúvidas, seguir instruções da página no Moodle ou enviar um e-mail para [bruno.marques@ufabc.edu.br](mailto:bruno.marques@ufabc.edu.br) com o assunto: [POO-Q2-2022] dúvida.

## Critérios de avaliação:

Nesta disciplina, estão previstas três atividades que assumem conceitos e são definidas como a seguir:

- **$T_1$** : Projeto de uma aplicação (software) seguindo os conceitos de orientação a objetos. O projeto deverá ser entregue baseado em diagramas UML. A documentação deve ser entregue na forma de um relatório descritivo do problema a ser resolvido pelo software e detalhes da solução proposta.
- **$T_2$** : Implementação de uma aplicação (software), na linguagem JAVA, seguindo os conceitos de orientação a objetos. Idealmente a implementação será referente ao projeto apresentado em  $T_1$ . Acompanhado a implementação (código fonte e executável) deverá ser entregue um relatório com o projeto atualizado, diagramas UML e descrição de uso do software. A avaliação será realizada através do material entregue e da apresentação via videoconferência ou vídeo de apresentação pré-gravado.
- **$E$** : Conceito referentes aos exercícios entregues via Moodle. Os exercícios serão disponibilizados nas aulas gravadas e em atividades no formato de questionários ou listas de exercícios. O conceito da Atividade EX será baseado na avaliação de todos os exercícios entregues.

Os exercícios referentes a atividade  **$E$**  deverão ser entregues de forma individual.

As atividades  **$T_1$**  e  **$T_2$**  poderão ser entregues de forma individual ou em dupla. A dupla deverá ser a mesma na atividade  **$T_1$**  e  **$T_2$** .

A entrega das atividades  $T_1$  e  $T_2$  será pelo Moodle, contendo minimamente:

- Um relatório completo da aplicação contendo motivação, descrição da solução, descrição de como os conceitos de orientação a objetos foram aplicados. Preferencialmente no formato PDF.
- Diagramas UML do software.
- Para a atividade  $T_2$ : código fonte em JAVA, entregue em um arquivo compactado.

### Nota Final:

A nota final  $N$  da disciplina será atribuído como o seguinte:

$$N = 0.25T_1 + 0.6T_2 + 0.15E$$

### Métrica de avaliação:

Nesta disciplina as atividades serão avaliadas com notas numérica. A nota final  $N$  será convertida baseada nas seguintes regras:

- [9.0, 10.0] → Desempenho excepcional, o aluno demonstra excelente compreensão e execução da atividade proposta, o código-fonte, quando aplicável, é eficiente e de fácil compreensão, documentação excelente incluindo comentários, imagens, diagramas, e material suplementar à submissão, nenhum erro aparente de execução e implementação.
- [7.5 a 9.0[ → Bom desempenho, o aluno demonstra boa capacidade de uso dos conceitos da disciplina. Executou a atividade conforme instruções. O código-fonte é funcional, podendo apresentar pequenas falhas que não atrapalham a execução. A documentação é organizada e apresenta todo o conteúdo essencial da atividade proposta.
- [6.0 a 7.5[ → Desempenho regular, o aluno demonstra capacidade adequado de compreensão dos conceitos da disciplina. Execução a tarefa conforme instruções, apresentou habilidade de resolução de problemas satisfatória, empregando soluções algorítmicas simples. O código-fonte é funcional, podendo apresentar algumas falhas que não afetam de maneira significativa a usabilidade do usuário. O aluno não explorou de maneira satisfatório um ou mais conceitos vistos em aula.
- [5.0 a 6.0[ → Desempenho mínimo na atividade. O aluno executou de maneira parcial a atividade. A atividade entregue apresenta deficiências na implementação que afetam a usabilidade do usuário ou documentação inadequada.
- < 5.0 → Desempenho insuficiente. O aluno não executou a atividade proposta de maneira minimamente satisfatória.

A recuperação (N com conceito D ou F) será uma nova entrega da Atividade  $T_2$ , podendo esta seguir o mesmo tema entregue anteriormente ou um novo tema a escolha do aluno.

O algoritmo a seguir apresenta o critério para composição do conceito final  $N_F$  após a recuperação. O algoritmo leva em conta o conceito obtido pré-recuperação  $N_P$ , e o conceito da atividade entregue na recuperação  $N_R$ . Importante notar que o conceito máximo obtido após a recuperação é C:

INÍCIO ALGORITMO NOTA RECUPERAÇÃO

Se  $N_P = D$  então:

Se  $N_R = (A \text{ ou } B \text{ ou } C)$  então:

$N_F \leftarrow C$

Senão:

$N_F \leftarrow D$

Se  $N_P = F$  então:

Se  $N_R = (A \text{ ou } B)$  então:

$N_F \leftarrow C$

Se  $N_R = (C)$  então:

$N_F \leftarrow D$

Senão

$N_F \leftarrow F$

FIM ALGORITMO NOTA RECUPERAÇÃO