

# MC322 CD – Programação Orientada a Objetos

1º Semestre 2021

Prof. Leonardo Montecchi

Instituto de Computação

[leonardo@ic.unicamp.br](mailto:leonardo@ic.unicamp.br)

Site da Disciplina	<a href="https://classroom.google.com/u/0/c/MjYzMTMyMzMwMjUy">https://classroom.google.com/u/0/c/MjYzMTMyMzMwMjUy</a>
Grupo Discord	<a href="https://discord.gg/KmZNzXNVhM">https://discord.gg/KmZNzXNVhM</a>

## Objetivo

Ao final desta disciplina o aluno deverá ser capaz de implementar sistemas de software aplicando o paradigma de programação orientada a objetos (POO).

## Conhecimentos Requeridos

- Conceitos básicos de programação procedural
- Tipos abstratos de dados (TADs)

## Site da Disciplina

Todas as comunicações, a disponibilização do material didático, assim como a entrega dos trabalhos práticos, serão realizadas por meio do sistema Google Classroom (<http://classroom.google.com>), na área correspondente à disciplina. **É responsabilidade do aluno verificar de ter acesso ao e-mail configurado na própria conta Classroom** ou, em alternativa, acessar periodicamente a página da disciplina para ler as comunicações.

## Aulas à Distância – Metodologia

A disciplina é organizada em 15 semanas de ensino. Cada semana abordará um tópico da disciplina, e será organizada da seguinte forma:

- Segunda: Disponibilização de aula gravada (assíncrona), slides e outro material de apoio e exercícios da semana.
- Terça (19h-23h): Horário reservado para trabalhar nos exercícios da semana (laboratórios), com supervisão dos monitores e do professor.
- Quinta (19h-21h): Aula síncrona – serão revisados os assuntos da semana e esclarecidas eventuais dúvidas (Google Meet). **Espera-se que o aluno tenha assistido aos vídeos disponibilizados na segunda.**

## Conteúdo

Os tópicos que serão tratados no curso incluem:

• Conceitos básicos de programação a objetos: objeto, classe, interface pública, estado, mensagens, encapsulamento • Introdução ao Java: tipos primitivos, estruturas de controle, especificidades da linguagem • Classes e Objetos: declaração de classes, métodos e variáveis, instanciação de objetos, construtores, sobrecarga de métodos, visibilidade de métodos e atributos • Métodos e atributos: métodos e atributos estáticos, variáveis e métodos finais • Vetores estáticos e dinâmicos: Arranjos unidimensionais e multidimensionais, classe Array e ArrayList • Enumeração • Relacionamentos entre classes: Hierarquias de generalização/especialização, Hierarquias de agregação/decomposição, Associação entre classes • Herança, classes abstratas e interfaces • Polimorfismo, ligação estática e

dinâmica, sobrescrita de métodos • Tratamento de exceções • Metaclasses e reflexão • Introdução aos Padrões de Projeto • Introdução à modelagem UML.

A linguagem de programação utilizada na disciplina será **Java**.

## Avaliação

A avaliação da disciplina será realizada com base nas seguintes atividades:

- Serão aplicados 4 testes teóricos *individuais*. O conjunto de testes teóricos contribuirá ao 30% da nota final (7.5% cada teste).
- Serão aplicados 4 testes práticos *individuais* (programação). O conjunto de testes práticos contribuirá ao 30% da nota final (7.5% cada teste).
- Os alunos desenvolverão um projeto software *em grupos* de até 3 integrantes. O projeto contribuirá ao 40% da nota final.

Seja  $M_T = \sum^n T_i$  a média dos testes teóricos, e  $M_L = \sum^m L_j$  a média dos testes no laboratório. Caso  $M_T < 5.0$  ou  $M_L < 5.0$  será aplicado um desconto de 25% na nota do projeto. A média M da disciplina será calculada então como:

$$M = \begin{cases} 0.3 \sum^n T_i + 0.3 \sum^m L_j + 0.4 P & \text{se } M_T \geq 5.0 \text{ e } M_L \geq 5.0 \\ 0.3 \sum^n T_i + 0.3 \sum^m L_j + 0.4 (0.75 P) & \text{caso contrário} \end{cases}$$

Sendo  $T_i$  a nota do i-ésimo teste teórico,  $L_j$  a nota do j-ésimo teste prático e  $P$  a nota do projeto.

Os alunos com média  $2.5 \leq M < 5.0$ , poderão fazer um exame final. Seja  $E$  a nota do exame, a nota final da disciplina  $F$  será calculada como:

$$F = \begin{cases} \min\left(5.0, \frac{M+E}{2}\right) & \text{caso } 2.5 \leq M < 5.0 \text{ e o aluno tenha realizado o exame} \\ M & \text{caso contrário} \end{cases}$$

O aluno estará aprovado caso sua nota final  $F$  seja maior ou igual a 5.0, e estará reprovado caso contrário.

## Cronograma

As datas de entrega das avaliações (aluno entrega no Classroom) são definidas no cronograma a seguir. Os textos serão disponibilizados com alguns dias de antecedência. O cronograma poderá sofrer alterações, que serão sempre comunicadas com antecedência.

19/04/2021	Entrega teste teórico T <sub>1</sub> Entrega avaliação prática L <sub>1</sub>
10/05/2021	Entrega teste teórico T <sub>2</sub> Entrega avaliação prática L <sub>2</sub>
31/05/2021	Entrega teste teórico T <sub>3</sub> Entrega avaliação prática L <sub>3</sub>
21/06/2021	Entrega teste teórico T <sub>4</sub> Entrega avaliação prática L <sub>4</sub>
05/07/2021	Entrega Projeto

## Atendimento

Será realizado atendimento online por meio da ferramenta Discord ([www.discord.com](http://www.discord.com)). No horário das aulas de laboratório estarão disponíveis online os monitores e/ou o professor para tirar dúvidas.

Contatos:

- **Leonardo Montecchi (Professor)** – [leonardo@ic.unicamp.br](mailto:leonardo@ic.unicamp.br)
- Junior Cupe Casquina (PED) – [jcupe.cas@gmail.com](mailto:jcupe.cas@gmail.com)
- Fillipe dos Santos Silva (PED) – [fillipesantos00@gmail.com](mailto:fillipesantos00@gmail.com)
- Natan Rodrigues de Oliveira (PAD) – [natanrodriguesdeoliveira@gmail.com](mailto:natanrodriguesdeoliveira@gmail.com)
- Leonardo Yoshida (PAD) – [leonardo\\_yoshida@yahoo.com.br](mailto:leonardo_yoshida@yahoo.com.br)
- Caio Henrique Pardal dos Santos (PAD) – [pardal.henrique@gmail.com](mailto:pardal.henrique@gmail.com)

## Referências

As principais referências consideradas importantes para o cumprimento do conteúdo proposto encontram-se listadas a seguir. O material complementar a ser utilizado será indicado na página da disciplina.

- Bruce Eckel, “Thinking in Java”, Prentice Hall, 4th edition (2006)
- Robert C. Martin, “Clean Code: A Handbook of Agile Software Craftsmanship”, Prentice Hall (2008)
- Grady Booch et. al., “The Unified Modeling Language User Guide”, 2nd Edition (1999)

## Observações

- Qualquer tipo de fraude acarretará em nota final F=0 (zero) para todos os envolvidos.