

MC322 C – Programação Orientada a Objetos

1º Semestre 2019

Prof. Leonardo Montecchi

Instituto de Computação

leonardo@ic.unicamp.br

	Dia	Horário	Sala
Aula	Terça-feira	21h-23h	CC00 (Laboratório)
	Quinta-feira	19h-21h	CB02 (Teoria)
Atendimento	O horário de atendimento extra-classe será realizado sempre depois das aulas, ou com agendamento prévio em outros horários.		

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

Na primeira aula do curso, será aplicada uma *avaliação diagnóstica*, com o objetivo de verificar a necessidade de uma revisão dos conhecimentos considerados fundamentais para um bom desempenho na disciplina.

Site da Disciplina

Todas as comunicações, a disponibilização do material didático, assim como a entrega dos trabalhos práticos, serão realizadas por meio do sistema Moodle (<https://www.ggte.unicamp.br/ea/>), na área correspondente à disciplina. É responsabilidade dos alunos verificar que o e-mail configurado na própria conta Moodle seja correto ou, em alternativa, acessar periodicamente a página para ler as comunicações.

Conteúdo

Os tópicos a serem apresentados 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 n testes teóricos *individuais* de duração 40 minutos, no final do horário das aulas teóricas, em datas específicas. Os testes teóricos contribuirão ao 30% da nota final.
- Serão aplicados m testes práticos *individuais* no laboratório (programação), em datas específicas. Os testes práticos contribuirão ao 30% da nota final.
- Os alunos desenvolverão um projeto software *em grupos* de 3-4 integrantes. O projeto contribuirá ao 40% da nota final.

A média M da disciplina será calculada então como:

$$M = \left(0.3 \sum_{i=1}^n T_i \right) + \left(0.3 \sum_{j=1}^m L_j \right) + 0.4P$$

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

O cronograma definido a seguir podem sofrer alterações devido a imprevistos ou necessidades relacionadas ao desenvolvimento da disciplina. Qualquer alteração será comunicada com antecedência.

21/03/2019	Teste teórico T_1
26/03/2019	Avaliação prática L_1
11/04/2019	Teste teórico T_2
16/04/2019	Avaliação prática L_2
07/05/2019	Avaliação prática L_3
09/05/2019	Teste teórico T_3
30/05/2019	Teste teórico T_4
04/06/2019	Avaliação prática L_4
30/06/2019	Entrega Projeto

11/07/2019	Exame
------------	-------

Monitoria

- Elder de Oliveira Rodrigues Júnior (PED)

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

- A presença é obrigatória em todas as aulas, incluindo as atividades no laboratório. Frequência inferior a 75% causa reprovação.
- Não haverá testes ou laboratórios substitutivos.
- Qualquer tipo de fraude acarretará em nota final F=0 (zero) para todos os envolvidos.