

# Programação Orientada a Objetos - ECo

Roland Teodorowitsch

Programação Orientada a Objetos - ECo - Curso de Engenharia de Computação - PUCRS

1 de agosto de 2024

# Apresentação da Disciplina

# Sobre o professor

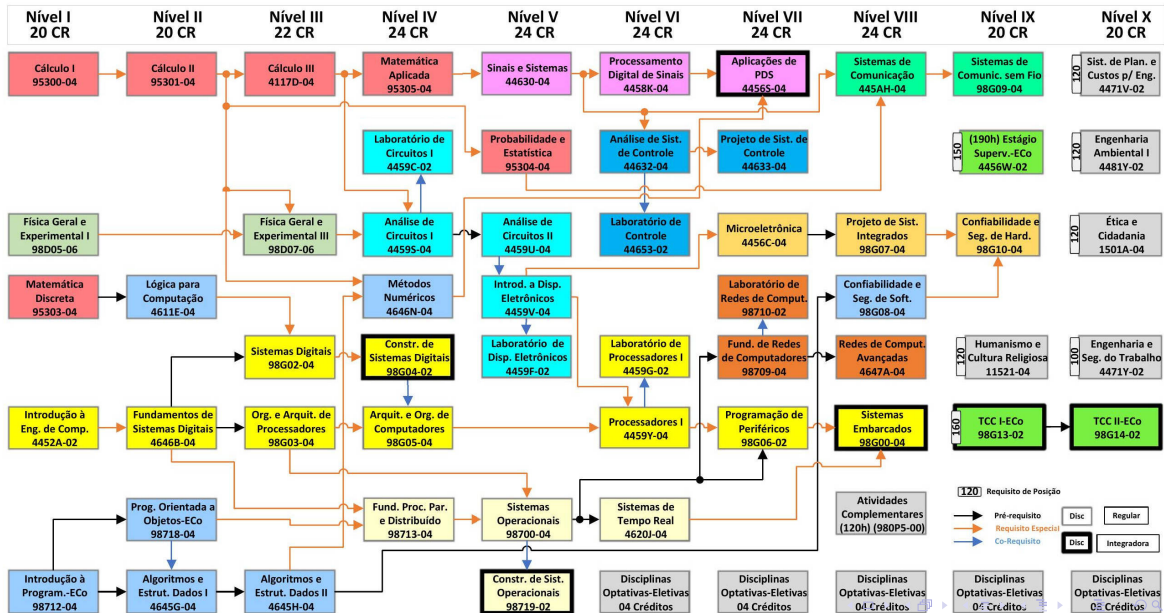
- Nome:
  - Roland Teodorowitsch
- Formação:
  - Bel. em Informática (PUCRS, 1990)
  - Msc. em Ciência da Computação (UFRGS, 1994)
- Áreas de Interesse:
  - SD, SO, STR, PP, Sist. Embarc., Arq. de Comp.
- E-mail:
  - roland.teodorowitsch@pucrs.br

# Sobre a disciplina

- Nome: Programação Orientada a Objetos - ECo
- Código: 98718-04
- Turma: 10
- Créditos: 4
- Carga-horária: 60 horas-aula
- Horário:
  - Turma 10: 2AB 4AB (AB=8h-9h30min)
- Modalidade: presencial

# Sobre a disciplina

- Semestre: 2
- Pré-requisito:
  - Introdução à Programação - ECo
- Co-requisito:
  - Algoritmos e Estruturas de Dados I
- É pré-requisito para:
  - Fundamentos do Processamento Paralelo e Distribuído



# Objetivos

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

- 1 Conhecer e utilizar de forma precisa conceitos e termos relacionados ao paradigma de orientação a objetos.
- 2 Desenvolver as competências e habilidades para a criação de sistemas de complexidade média, formado por múltiplos componentes, e expressar estas soluções na forma de um sistema de classes em uma linguagem de programação.
- 3 Empregar adequadamente ponteiros para manipulação de estruturas encadeadas e memória.
- 4 Construir abstrações para tipos de dados, usando os conceitos de classe, objeto-mensagem, herança e polimorfismo.
- 5 Compreender os conceitos avançados em orientação a objetos.

# Conteúdo



# Ementa

Paradigma de programação orientada a objetos; Abstração; Encapsulamento; Mensagens; Relacionamento entre classes (Composição, Referência); Parametrização de tipos (Templates); Uso de Ponteiros para Estruturas Encadeadas; Fluxos de Entrada e Saída; Modularização; Herança; Polimorfismo; STL; Conceitos avançados em Orientação a Objetos.

# Conteúdo (1/4)

3 grandes unidades

- Orientação a objetos básica
- Manipulação de dados
- Orientação a objetos avançada

# Conteúdo (2/4)

1. Orientação a objetos básica
  - 1.1. Conceitos de orientação a objetos
    - 1.1.1. Classes e objetos
    - 1.1.2. Atributos e métodos: classe e instância
  - 1.2. Visibilidade de atributos e métodos
  - 1.3. Princípios de projeto orientado a objetos
    - 1.3.1. Mensagem
    - 1.3.2. Abstração
    - 1.3.3. Encapsulamento
  - 1.4. Detalhamento de classes
    - 1.4.1. Relacionamento entre classes (composição, referência)
    - 1.4.2. Construtores
    - 1.4.3. Sobrecarga
    - 1.4.4. Autorreferência
    - 1.4.5. Modularização (agrupamento de classes relacionadas)

# Conteúdo (3/4)

## 2. Manipulação de dados

- 2.1. Ponteiros

- 2.2. Alocação dinâmica

- 2.3. Estruturas encadeadas

- 2.4. Fluxos de entrada e saída

## Conteúdo (4/4)

- 3. Orientação a objetos avançada
  - 3.1. Generalização/especialização
    - 3.1.1. Herança simples e múltipla
    - 3.1.2. Hierarquia de classes
  - 3.2. Polimorfismo
  - 3.3. Tratamento de exceções
  - 3.4. Parametrização de tipos (*Templates*)
    - 3.4.1. STL (*Standard Template Library*)
  - 3.5. Conceitos avançados em Orientação a Objetos

# Bibliografia Básica

RAMNATH, S.; DATHAN, B. **Object-Oriented Analysis, Design and Implementation**: an integrated approach. 2 ed. Heidelberg: Springer, 2010. 471 p. (ou anterior)

WEISFELD, M. **The Object-oriented Thought Process**. 4 ed. Upper Saddle River, NJ: Addison Wesley, 2013. 336 p.

MEYER, B. **Object Oriented Software Construction**. 2 ed. Upper Saddle River, NJ: Prentice Hall, 1997. 1296 p.

## Bibliografia Complementar

BOOCH, G. et al. **Object-oriented Analysis and Design with Applications**. 3rd ed. Upper Saddle River, NJ: Addison Wesley, 2007. 720 p. (ou anterior)

DEITEL, H. M; DEITEL, P. J. **C++: como programar**. 5 ed. São Paulo: Pearson Education do Brasil, 2006. 1208 p. (ou anterior)

FARRELL, J. **An Object-oriented Approach to Programming Logic and Design**. 4 ed. Boston, MA: Cengage Learning, 2012. 560 p.

GAMMA, E. et al. **Padrões de Projeto**: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364 p.

SILVA Fo., A. M. **Introdução a Programação Orientada a Objetos com C++**. Rio de Janeiro, RJ: Elsevier/Campus, 2010. 312 p.

# Avaliação



# Avaliação

$$G1 = \frac{2 \times MT + 4 \times P_1 + 4 \times P_2}{10}$$

- $MT$  = Média de Trabalhos
- $P_1$  = Prova referente à primeira parte da disciplina
- $P_2$  = Prova referente a todo o conteúdo da disciplina

## Índices de Aprovação

SEM.	NÚM. AL.	APROV.	REP.	REP. FALTAS	CANC.	CANC. NC
2019/1	14	10 (71,4%)	1 (7,1%)	1 (7,1%)	2 (14,3%)	0 (0,0%)
2019/2	25	16 (64,0%)	2 (8,0%)	0 (0,0%)	7 (28,0%)	0 (0,0%)
2020/1	13	2 (15,4%)	3 (23,1%)	0 (0,0%)	8 (61,5%)	0 (0,0%)
2020/2	31	17 (54,8%)	4 (12,9%)	0 (0,0%)	10 (32,3%)	0 (0,0%)
2021/1	17	6 (35,3%)	2 (11,8%)	0 (0,0%)	9 (52,9%)	0 (0,0%)
2021/2	27	9 (33,3%)	11 (40,7%)	0 (0,0%)	2 (7,4%)	5 (18,5%)
2022/1	31	10 (32,3%)	4 (12,9%)	2 (6,5%)	12 (38,7%)	3 (9,7%)
2022/2	12	3 (25,0%)	1 (8,3%)	1 (8,3%)	7 (58,3%)	0 (0,0%)
2023/1	20	11 (55,0%)	2 (10,0%)	3 (15,0%)	4 (20,0%)	0 (0,0%)
2023/2 (10)	9	4 (44,4%)	2 (22,2%)	0 (0,0%)	3 (33,3%)	0 (0,0%)
2023/2 (11)	23	13 (56,5%)	2 (8,7%)	1 (4,3%)	7 (30,4%)	0 (0,0%)
2024/1	21	8 (38,1%)	2 (9,5%)	3 (14,3%)	7 (33,3%)	1 (4,8%)

Tabela: Índices de Aprovação

## Informações Gerais

# Avisos: Presenças e Faltas

- 1 encontro = 2 presenças ou faltas
- em 2024-2: 34 encontros (68 horas/aula) com presença contabilizada
- na semana de G2, a presença NÃO é contabilizada
- frequência mínima para aprovação: 75%
- limite de faltas = 25% de 68 = 17
- portanto, com 17 faltas (8 encontros e meio) o aluno está REPROVADO POR FALTAS

# Avisos: Moodle

- Avisos pelo Mural
- Dúvidas pelo fórum
- Material de apoio
- Entrega de trabalhos e exercícios

# Avisos: Diversos

- Listas de exercícios com múltiplas entregas

# Avisos: Mapas Mentais

- São uma forma de representar conhecimento sobre determinado assunto
- Recomenda-se fortemente que o aluno construa Mapas Mentais dos conteúdos da disciplina
- Os Mapas Mentais poderão ser usados nas avaliações da disciplina (com exceção das provas PS e G2)
- Regras:
  - Um Mapa Mental por unidade do conteúdo programático
  - NÃO incluir textos grandes e/ou copiados e colados nos nodos do mapa
  - Cada Mapa Mental em uma folha A4 impressa
  - Todos os Mapas Mentais devem ser entregues uma semana antes da avaliação em que serão utilizados
- Mapas Conceituais também são uma alternativa interessante

## Avisos: Recomendações

- Dificuldades não esclarecidas em disciplinas anteriores NÃO serão resolvidas sem esforço adicional
- Consultem as obras disponíveis na biblioteca (livros físicos e on-line)
- Aproveitar a aula para realizar os exercícios
- Não deixe acumular trabalhos
- Não se aprende a programar estudando na véspera da prova...
- Reserve um horário de estudo na semana para cada uma das disciplinas
- "Disciplina é liberdade..." (letra da música Há Tempos, da banda Legião Urbana)
- Existe uma monitoria que presta apoio aos alunos com dúvidas (verifique horários na página de monitoria)



Dúvidas?

# Dúvidas?



## Mensagens Finais

# Reflexões

`https://youtu.be/eBGRY6aeaqa`

