



Apresentação do curso

Prof. Renato Pimentel

2024/2



Objetivos



- Ao final do curso, o aluno será capaz de:
 - ① Analisar problemas computacionais e propor soluções utilizando conceitos de programação orientada a objetos, como: classes, objetos, herança e polimorfismo;
 - ② Desenvolver programas em uma linguagem de programação orientada a objetos.
- Objetivos específicos:
 - ① Propiciar transição entre a programação estruturada e a programação orientada a objetos (POO);
 - ② Projetar, implementar e testar programas orientados a objetos;
 - ③ Introduzir conceitos de classes, objetos, herança e polimorfismo;
 - ④ Apresentar visão geral dos recursos da linguagem Java.



- Introdução aos conceitos básicos de POO: classes, objetos, encapsulamento, herança e polimorfismo;
- Conhecimento dos membros que tipicamente compõem classes: construtores, destrutores, variáveis e métodos;
- Entendimento e aplicação dos conceitos de OO em linguagens de programação que suportem tal paradigma;
- Desenvolvimento de sistemas usando POO.



Bibliografia básica



- BARNES, D. J. *Programação orientada a objetos com Java: uma introdução prática usando o BlueJ*. São Paulo: Prentice Hall, 2009.
- CARDOSO, C. *Orientação a objetos na prática: aprendendo orientação a objetos com Java*. Rio de Janeiro: Ciência Moderna, 2006.
- DEITEL, H. M.; DEITEL P. J. *Java: Como Programar*. 6. ed. Boston: Pearson, 2005



- BOOCH, G., RUMBAUGH, J., JACOBSON, I. *UML, Guia do Usuário*. Rio de Janeiro: Campus, 2000.
- GOSLING, J. *The java application programming interface*. Reading: Addison-Wesley, 1996.
- HORSTMANN, C. S. *Padrões e projeto orientados a objetos*. Porto Alegre: Bookman, 2007.
- SEDGEWICK, R.; WAYNE, K. *Introduction to programming in Java: an interdisciplinary approach*. Boston: Addison-Wesley, 2008.
- SIERRA, K.; BATES, B. *Use a cabeça! Java*. Rio de Janeiro: Alta Books, 2005.



Conteúdo previsto



- Introdução à POO
- Programação procedimental versus POO
- Classes e interfaces
- Objetos, atributos, métodos, construtores e destrutores
- Membros de classe e membros de objetos
- Métodos concretos e abstratos
- Visibilidade e encapsulamento
- Generalização, especialização e herança
- Polimorfismo
- Tratamento de exceções
- Manipulação de arquivos
- GUI
- JDBC



- 2 provas teóricas:
 - ▶ 12/03/2025 ($P1$) – 35 pontos
 - ▶ 30/04/2025 ($P2$) – 35 pontos

Nota de prova:

$$NP = P1 + P2$$

- Trabalho de implementação semestral ($TF = 30$ pontos): a ser apresentado ao final do curso, com arguição.
 - ▶ Data de entrega final: 02/05/2025.
 - ▶ Maiores detalhes ao longo do semestre.
- Pontos extras (no máximo 10 pontos), na forma de entrega de itens específicos ou a totalidade das atividades de laboratório, no tempo estipulado.
- Nota final (aproveitamento):

$$NF = NP + TF$$

Observação importante

Atividades práticas completamente ou parcialmente copiadas receberão nota **zero**, *tanto de quem copiou quanto quem forneceu o material*. Mesmo nos casos em que o aluno assuma que copiou ou forneceu, a nota será mantida. Somente trabalhe por si ou com os integrantes de seu grupo – no caso do projeto – e não forneça seu material aos demais.

O mesmo se aplica a materiais copiados, mesmo que parcialmente, de outras fontes, como Internet.



- Alunos que obtiverem ao final do curso $NF < 60$ (somente) terão direito a uma prova substitutiva (SUB), valendo 35 pontos, de modo a tentar aumentar o aproveitamento final
- Data: 09/05/2025
- O conteúdo da prova será o visto ao longo de todo o semestre;
- Elimina a menor nota, dentre $P1$ e $P2$, substituindo-a;
- A NF neste caso será dada por

$$NF = \begin{cases} 60, & \text{se } \max(P1, P2) + SUB + TF \geq 60 \\ \max(P1, P2) + SUB + TF, & \text{caso contrário,} \end{cases}$$

ou seja, o aluno que ficou de SUB **terá nota máxima 60**, caso atinja na mesma a pontuação necessária para ser aprovado.



- O aluno que tiver frequência inferior a 75% é reprovado por faltas.
- A assiduidade será computada através da chamada em sala durante as aulas, em um horário aleatório após 10 minutos do início de cada encontro diário. O professor poderá adotar, a seu critério, caso haja demasiada desistência de continuidade em sala, uma segunda chamada ao final do segundo horário de aula.

- **Falta em dia de prova:** o aluno somente terá direito a fazer prova em nova data caso apresente justificativa prevista pelas Normas de Graduação.
- É responsabilidade do aluno controlar sua frequência, de modo a evitar reprovação por falta.
- O professor não corrigirá atividades de alunos que já tenham sido reprovados por falta. Inclusive, os mesmos não terão direito à SUB (Normas da Graduação).



Aulas



- Quartas-feiras: 19:00 até 20:40 – **Sala 1B106**
- Sextas-feiras: 20:50 até 22:30 – **Sala 1B106** – ou **Lab. 1B220 (LAB04)** para práticas.

Atividades extraclasse

- Listas de exercícios (fixação) – também para complementação de carga horária.
- **Teams:** Equipe **FACOM32305 – 2024/2**, código de inscrição **98upp3v**.



- Professor: Renato Pimentel
 - ▶ Página: <http://www.facom.ufu.br/~rpimentel>
 - ▶ E-mail: [rpimentel @ ufu . br](mailto:rpimentel@ufu.br)
 - ▶ Sala 1B139
- Atendimento (agendar previamente através de e-mail):
 - ▶ Quartas-feiras, 14:00 até 15:40, sala 1B139
 - ▶ Quintas-feiras, 19:00 até 20:40, sala 1B139
- Material da disciplina:
 - ▶ <http://www.facom.ufu.br/~rpimentel> > Ensino > 2024/2 > FACOM32305 – Programação Orientada a Objetos