



## Ficha 2

|   |  |                                     |                     |  |                 |                      |                            |
|---|--|-------------------------------------|---------------------|--|-----------------|----------------------|----------------------------|
| Disciplina: <b>Programação Orientada a Objetos</b>  |  |                                     |                     |  |                 | Código: <b>TE353</b> |                            |
| Natureza:<br>(X) Obrigatória<br>( ) Optativa  |  | (X) Semestral ( ) Anual ( ) Modular |                     |  |                 |                      |                            |
| Pré-requisito:  |  | Co-requisito:                       |                     | Modalidade: (X) Presencial ( ) Totalmente EaD ( ) ..... % EaD* |                 |                      |                            |
| CH Total: 60<br>CH semanal: 04  |  | Padrão (PD): 60                     | Laboratório (LB): 0 | Campo (CP): 0  | Estágio (ES): 0 | Orientada (OR): 0    | Prática Específica (PE): 0 |
| <b>EMENTA</b>   |  |                                     |                     |  |                 |                      |                            |
| Encapsulamento; Herança e Hierarquia; Composição e Derivação; Construtores; Polimorfismos; Modelos.   |  |                                     |                     |  |                 |                      |                            |
| <b>PROGRAMA</b>   |  |                                     |                     |  |                 |                      |                            |
| Revisão de Algoritmos e de Programação Estruturada usando Java.<br>Introdução Família Java/Java SE e IDE Eclipse. JRE, JVM, SDKs, portabilidade.<br>Programação Orientada a Objetos (POO): Classes, objetos e métodos. Herança. Polimorfismo. Interfaces.<br>Tipos de dados e operadores. Abstrações para projeto. Encapsulamento de informação. Tipos genéricos.<br>Correções do programa. Invariantes. Pacotes. Tratamento de Exceções. Manipulação de arquivos (texto, binário, randômico). Serialização. Componentes de biblioteca. |  |                                     |                     |  |                 |                      |                            |
| <b>OBJETIVO GERAL</b>   |  |                                     |                     |  |                 |                      |                            |
| Desenvolver programas aplicando o paradigma de Orientação a Objetos a projetos implementados na linguagem Java.   |  |                                     |                     |  |                 |                      |                            |
| <b>OBJETIVOS ESPECÍFICOS</b>  |  |                                     |                     |  |                 |                      |                            |
| <ul style="list-style-type: none"><li>▶ Utilizar o paradigma de Orientação a Objetos para modelar problemas e implementar programas</li><li>▶ Conhecer e empregar boas práticas de programação e projeto Orientado a Objetos</li><li>▶ Escrever e depurar programas Java usando uma IDE</li><li>▶ Empregar classes da biblioteca Java no desenvolvimento de programas</li></ul>   |  |                                     |                     |  |                 |                      |                            |

## PROCEDIMENTOS DIDÁTICOS

A disciplina será desenvolvida usando metodologias de aprendizagem ativa, sobretudo PjBL (*Project-Based Learning*) e TBL (*Team-Based Learning*). As aulas expositivas serão empregadas para discussão dos conteúdos curriculares teóricos, da articulação teórico-prática e para o esclarecimento de dúvidas de conteúdo, de metodologia, de avaliação e para o feedback do processo avaliativo.

Os trabalhos que serão propostos se constituem de atividades práticas para fixação de conhecimento e apropriação das técnicas através do aprender fazendo. Eles estarão relacionados aos conceitos em estudo, incluindo realização de exercícios e projetos individuais e em equipe que são componentes da avaliação.

Ambiente Virtual de Aprendizagem:

A plataforma Microsoft TEAMS, disponível gratuitamente para todos os estudantes com registro ativo na UFPR, será utilizada para todas as comunicações oficiais, agenda, datas importantes, disponibilização de material adicional: artigos, slides, pdf etc e para interação/comunicação durante o semestre, incluindo a entrega dos trabalhos.

## FORMAS DE AVALIAÇÃO

A avaliação será baseada no desenvolvimento de trabalhos de implementação realizados em equipe e/ou individualmente e em duas provas bimestrais individuais, da seguinte forma:

\* Provas individuais (duas bimestrais):

- 2ª. Quinzena de julho e
- 1ª. Quinzena de setembro.

\* Trabalhos em equipe, com apresentação do protótipo e avaliação baseada em rubricas pré-determinadas e constantes da especificação, com datas de entrega compatíveis com as datas das avaliações.

Peso na composição da média:

- Provas 1 e 2: 30% cada, 60% total
- Trabalhos: 40% total

A aprovação segue o padrão da universidade, ou seja:

- Média  $\geq 7.0$ , aprovado sem exame;
- $4.0 \leq$  Média  $< 7.0$ , em exame final;
- Média  $< 4.0$ , reprovado sem direito a exame final.

O exame final será realizado no período compreendido entre 19 e 24 de setembro de 2022.

A frequência mínima para aprovação deve ser maior ou igual a 75%.

### BIBLIOGRAFIA BÁSICA

- HORSTMANN, Cay S. **Conceitos de computação com Java**: compatível com Java 5 & 6. 5. ed Porto Alegre: Bookman, 2009. 720 p., il., 25 cm. Inclui índice. ISBN 978-85-7780-352-1.
- PREISS, Bruno R. **Estruturas de dados e algoritmos**: padroes de projetos orientados a objetos com Java. Rio de Janeiro: Campus, c2001. xvi, 566. Inclui bibliografia e índice. ISBN 8535206930 (broch.).
- BARKER, Jacquie. **Beginning Java Objects**: From Concepts to Code. Second Edition Berkeley, CA: Jacquie Barker, 2005. Ebook. v.: digital. (Professional Computing and Web Design (Springer; 12059-ZDB-2-CWD). Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0036-9>. Acesso em: 1 set. 2021.

### BIBLIOGRAFIA COMPLEMENTAR

- HORSTMANN, Cay S. **Big Java**. 2. ed. Hoboken: J. Wiley, c2006. xxxii, 1216 p., il. Inclui apêndices, glossário e índice. ISBN 0471697036 (broch.).
- SPELL, Brett. **Pro Java Programming**. Second Edition Berkeley, CA: Brett Spell, 2005. Ebook. v.: digital. (Professional Computing and Web Design (Springer; 12059-ZDB-2-CWD). Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0032-1>. Acesso em: 1 set. 2021.
- JANDL JUNIOR, Peter. **Java**: guia do programador. São Paulo: Novatec, 2007. 681p., il. Inclui bibliografia e índice. ISBN 9788575221099 (broch.).
- CHAN, Mark C; GRIFFITH, Steven W; IASI, Anthony F. **Java 1001 dicas de programação**. São Paulo: Makron Books, 1999. 714 p. ISBN 978-8534608954.
- WINDER, R. **Desenvolvendo software em Java**. 3. ed Rio de Janeiro: LTC, c2009. xxii, 696 p., il., 28 cm. Inclui índice e bibliografia. ISBN 978-85-216-1658-0.

**Professor da Disciplina:** Henri Frederico Eberspacher  
**Documento assinado digitalmente**

**Chefe de Departamento ou Unidade equivalente:** Luiz Antonio Belinaso  
**Documento assinado digitalmente**

**Plano Válido para 2022/1º. Semestre**