



Pontifícia Universidade Católica do Paraná

Plano de Ensino

Escola/Campus:	Politécnica			
Cursos:	Tecnologia em Jogos Digitais Bacharelado em Jogos Digitais Cibersegurança Ciência da Computação Sistemas de Informação Engenharia de Software			Ano/Semestre: 2024/2
Código/Nome da disciplina:	Programação Orientada a Objetos (POO)			
Carga Horária:	120 horas-aula (90 horas-relógio)			
Requisitos:	Não há			
Créditos:	6	Período: 2º	Turma: B	Turno: Noite
Professor Responsável:	Marina de Lara			

1. Ementa

A disciplina, de natureza teórico-prática, é ofertada pelos cursos da área de Computação da Escola Politécnica. O estudante aplica conceitos da orientação a objetos na resolução de problemas estruturados. Ao término da disciplina, é capaz de codificar produtos de software utilizando linguagem de programação orientada a objetos de forma sistematizada e aderente às especificações. Os estudantes trabalham em equipes para verificar os benefícios de orientação a objetos no desenvolvimento de sistemas de grande porte. Para frequentar a disciplina é desejável conhecimento em algoritmos.

2. Relação com disciplinas precedentes e posteriores

Esta disciplina é parte integrante do eixo de programação dos cursos. Não há pré-requisitos, mas é desejável conhecimento prévio de algoritmos e programação estruturada. A disciplina fornece a base conceitual e tecnológica necessária para a construção de programas orientados a objetos. Contribui com resultados de aprendizagem para as seguintes disciplinas:

Curso	Disciplinas Precedentes	Disciplinas Posteriores
Bacharelado em Engenharia de Software (BES)	<ul style="list-style-type: none">• Raciocínio Algorítmico (1º Período);• Programação Web (2º Período);• Banco de Dados (2º Período);	<ul style="list-style-type: none">• Resolução de Problemas Estruturados em Computação (4º Período);• Design de Software (4º Período);• Verificação e Validação (4º Período);• Desenvolvimento Orientado a Reuso (6º Período);• Projeto Final I (7º Período) e II (8º Período).
Bacharelado em Jogos Digitais	<ul style="list-style-type: none">• Game Programming Bootcamp (1º Período)• Raciocínio Algorítmico (2º Período)	<ul style="list-style-type: none">• Programação de Estruturas de Dados para Jogos (4º Período)• Experiência Criativa: Desenvolvimento de Jogos Sérios (5º Período)• Movimentos, Explosões e Partículas em Jogos (6º Período)• Game Interface (7º Período)
Tecnologia em Jogos Digitais	<ul style="list-style-type: none">• Game Programming Bootcamp (1º Período)• Raciocínio Algorítmico (2º Período)	<ul style="list-style-type: none">• Movimentos, Explosões e Partículas em Jogos (4º Período)• Rendering Pipeline (5º Período)

Cibersegurança	<ul style="list-style-type: none"> • Raciocínio Algorítmico (1º Período); • Fundamentos de Sistemas Ciberfísicos (1º Período) • Programação Imperativa (2º Período) 	<ul style="list-style-type: none"> • Resolução de Problemas Estruturados em Computação (4º Período) • Data Hiding (8º Período)
Ciência da Computação	<ul style="list-style-type: none"> • Raciocínio Algorítmico (1º Período); • Programação Imperativa (2º Período) • Arquitetura de Banco de Dados (2º Período) 	<ul style="list-style-type: none"> • Resolução de Problemas Estruturados em Computação (4º Período) • Modelagem de Sistemas Computacionais (4º Período) • Programação Distribuída (6º Período) • Experiência Criativa: Projeto Transformador I (7º Período) • Experiência Criativa: Projeto Transformador II (8º Período)
Sistemas de Informação	<ul style="list-style-type: none"> • Raciocínio Algorítmico (1º Período) • Banco de Dados (2º Período) • Programação Web (2º Período) 	<ul style="list-style-type: none"> • Resolução de Problemas Estruturados em Computação (4º Período) • Tecnologias para Desenvolvimento Web (4º Período) • Experiência Criativa: Projetando Sistemas de Informação (5º Período) • Padrões de Projeto e Multicamadas (5º Período) • Desenvolvimento de Serviços Web Seguros (6º Período) • Desenvolvimento Ágil de Produto I (7º Período) e II (8º Período)

3. Temas de estudo

- TE1: Encapsulamento (classe, método e atributo)
- TE2: Objeto (instanciação, referência, interação, teste e coleção de objetos)
- TE3: Herança entre classes (generalização, especialização e polimorfismo)
- TE4: Classe abstrata e interface
- TE5: Tratamento de exceção
- TE6: Persistência de objeto

4. Resultados de Aprendizagem

Resultados de Aprendizagem	Temas de Estudo	Elemento de Competência
Competência: Implementar softwares para problemas estruturados de solução algorítmica, utilizando técnicas e ferramentas de desenvolvimento, de forma autônoma e cooperativa.		
RA1: Codificar programas baseados em interação entre objetos instanciados a partir de classes associadas, em conformidade com o princípio de encapsulamento, com a validação do seu comportamento, de forma sistematizada.	TE1: Encapsulamento (classe, método e atributo) TE2: Objeto (instanciação, referência, interação, teste e coleção de objetos)	Definir a sequência lógica a partir da análise das especificações de software, (Internaliza) Testar produtos de software em relação aos requisitos e às especificações de forma sistematizada, integrada e autorregulada (Internaliza)
RA2: Codificar programas baseados em objetos definidos por hierarquia de classes.	TE3: Herança entre classes (generalização, especialização e polimorfismo) TE4: Classe abstrata e interface	Codificar produtos de software utilizando boas práticas de programação, conforme os requisitos e as especificações, de forma sistematizada e colaborativa. (Internaliza)
RA3: Codificar programas baseados em objetos que interajam com componentes de entrada e saída de dados, em conformidade com a especificação.	TE5: Tratamento de exceção TE6: Persistência de objeto	Criar modelos de soluções computacionais inovadores, utilizando a linguagem mais apropriada ao contexto para dados estruturados e não estruturados. (Internaliza)

5. Mapa Mental



6. Metodologia e Avaliação

Tabela de Resultados de Aprendizagem (RA), correspondentes Indicadores de Desempenho (ID), métodos de ensino e aprendizagem e processos de avaliação.

Alinhamento Construtivo			
Resultado de aprendizagem	Indicadores de desempenho	Processos de Avaliação	Métodos ou técnicas empregados**
RA1: Codificar programas baseados em interação entre objetos instanciados a partir de classes associadas, em conformidade com o princípio de encapsulamento, com a validação do seu comportamento, de forma sistematizada.	<p>ID1.1: Codifica classes, definindo seus membros (atributos e métodos), e aplicando proteção a seus membros (encapsulamento e sobreposição).</p> <p>ID1.2: Codifica métodos de classe, com parâmetros, referência a objeto, variáveis locais e invocação de outros métodos (estáticos ou não), validando seu comportamento por meio de testes unitários (invariante, pré-condição e pós-condição), de forma sistematizada.</p> <p>ID1.3: Codifica coleção de objetos, utilizado agregação ou composição.</p>	<p>[Formativo] Exercícios</p> <p>[Somativo] PBL, PjBL e Prova</p>	<p>Sala de aula invertida</p> <p>Feedback coletivo (em aula)</p> <p>Feedback individual</p> <p>Interação: ambiente virtual de aprendizagem</p>
RA2: Codificar programas baseados em objetos definidos por hierarquia de classes	<p>ID2.1: Codifica modelos de hierarquia de classes para compartilhamento de atributos e métodos, de forma colaborativa.</p> <p>ID2.2: Codifica polimorfismo no contexto de hierarquia entre classes.</p> <p>ID2.3: Codifica classes abstratas e interfaces, no contexto de hierarquia entre classes.</p> <p>ID2.4: Aplica mecanismo de tratamento de exceção para tratamento de erro.</p>	<p>[Formativo] Exercícios</p> <p>[Somativo] PBL, PjBL e Prova</p>	<p>Sala de aula invertida</p> <p>Feedback coletivo (em aula)</p> <p>Feedback individual</p> <p>Interação: ambiente virtual de aprendizagem</p>

RA3: Codificar programas baseados em objetos que interajam com componentes de entrada e saída de dados, em conformidade com a especificação.	ID3.1: Manipula arquivo para recuperar, alterar e armazenar dados. ID3.2: Codifica persistência de objeto usando serialização.	[Formativo] Exercícios [Somativo] PjBL	Sala de aula invertida Feedback coletivo (em aula) Feedback individual Interação: ambiente virtual de aprendizagem
---	---	---	---

Avaliações Somativa:

RAs	Somativas	Nota das atividades no RA	Nota do RA na nota semestral
RA1	Prova I (individual)	3,0	30%
RA2	Prova II (individual)	3,0	30%
RA3	PjBL (em equipe)	4,0	40%

Critério de aprovação: nota do semestre igual ou superior a **7** (sete) e frequência mínima de **75%**.

A composição da nota semestral é:

$$Nota = RA1 + RA2 + RA3$$

Recuperação Parcial:

- A recuperação parcial dos RAs 1, 2 e 3 ocorrerão por meio de provas específicas, conforme o cronograma.

- É elegível para realizar a recuperação parcial apenas o estudante que obteve nota menor que 2,1 no RA1 e RA2, e 2,8 no RA3.
- A nota máxima do estudante após a recuperação parcial será igual a 2,1 para o RA1 e RA2, e 2,8 para o RA3. Assim, os estudantes que conseguirem uma nota entre 2,1 e 3, para o RA1 e RA2, terão a nota máxima igual a 2,1 e nota entre 2,8 e 4 para o RA3 terão a nota máxima igual a 2,8.

Recuperação Final:

- A Semana de Recuperação Estendida permitirá recuperar os RAs 1, 2 e 3.
- Somente o estudante que obteve **nota semestral** maior ou igual a 4 e menor que 7 poderá fazer a recuperação da Semana de Recuperação Estendida.
- Somente poderão ser recuperados na Semana de Recuperação Estendida as notas inferiores a 2,1 no RA1 e RA2 e inferior a 2,8 no RA3.
- A média semestral para o estudante que realizar avaliações na Semana de Recuperação Estendida é limitada a 7 (sete).

7. Cronograma de atividades*

Período	RAs	Atividades Pedagógicas	Em aula / TDE	Carga horária da atividade
31/07 – 01/08 Semana 1	1	Apresentação da disciplina [Teoria] Introdução ao Java [Prática em Grupo] PBL 1	aula	6 horas-aula
07/08 – 08/08 Semana 2	1	[Teoria] Classes [Prática em Grupo] PBL 2	aula	6 horas-aula
		Git	TDE	8 horas-aula
14/08 – 15/08 Semana 3	1	[Teoria] Encapsulamento [Prática em Grupo] PBL 3	aula	6 horas-aula
21/08 – 22/08 Semana 4	1	[Teoria] Interação entre Objetos e Coleção de Objetos [Prática em Grupo] PBL 4	aula	6 horas-aula

28/08 – 29/08 Semana 5	1	[Revisão] Conteúdo RA1 [Avaliação Formativa] Simulado I [Feedback Coletivo] Correção do Simulado I	aula	6 horas-aula
04/09 – 05/09 Semana 6	1	[Avaliação Somativa] Prova I – RA1 [Feedback Coletivo] Correção da Avaliação [Entrega de Notas] RA1	aula	6 horas-aula
11/09 – 12/09 Semana 7	2	[Teoria] Hierarquia de Classes [Prática em Grupo] PBL 5	aula	6 horas-aula
18/09 – 19/09 Semana 8	2	[Teoria] Polimorfismo [Prática em Grupo] PBL 6	aula	6 horas-aula
25/09 – 26/09 Semana 9	-	[SEMANA ACADÊMICA]	-	6 horas-aula
02/10 – 03/10 Semana 10	2	[Teoria] Classes Abstratas, Métodos Abstratos e Interface [Prática em Grupo] PBL 7	aula	6 horas-aula
09/10 – 10/10 Semana 11	2	[Revisão] Conteúdo RA2 [Avaliação Formativa] Simulado II [Feedback Coletivo] Correção do Simulado II	aula	6 horas-aula
16/10 – 17/10 Semana 12	2	[Avaliação Somativa] Prova II – RA2	aula	6 horas-aula
	3	[Teoria] Tratamento de Exceções Interface Gráfica Armazenamento de dados em arquivos	TDE	12 horas-aula
23/10 – 24/10 Semana 13	3	[Teoria] Persistência de Objetos [Prática em grupo] PjBL – Desenvolvimento	aula	6 horas-aula
30/10 – 31/10 Semana 14	3	[Prática em grupo] PjBL – Desenvolvimento	aula	6 horas-aula
07/11 – 08/11 Semana 15	3	[Prática individual] PjBL – Desenvolvimento	aula	6 horas-aula
13/11 – 14/11 Semana 16	3	[Avaliação Somativa] PjBL – Entrega (Prova de Autoria)	aula	6 horas-aula

20/11 – 21/11 Semana 18	1, 2, 3	[Entrega de Notas] RA3 e Médias Finais [Avaliação Somativa] Recuperação parcial dos RAs 1, 2 e 3	aula	6 horas-aula
27/11 – 28/11 Semana Final	1, 2, 3	Semana de recuperação estendida	-	-

** este cronograma pode sofrer alterações conforme necessidades observadas durante o semestre*

8. Referências

Básica:

Todas as atividades contarão com o apoio de material preparado pelo professor sob a forma de slides, vídeos e exercícios implementados em linguagem de programação e disponibilizados no ambiente virtual de aprendizagem. Os seguintes livros poderão ser consultados na biblioteca da PUCPR pelos estudantes como fonte complementar:

1. DEITEL and DEITEL; Java Como Programar, Editora Bookman, 2007
2. SANTOS, Rafael; Introdução à Programação Orientada a Objetos Usando Java. Editora Campus. 2003.
3. HORSTMANN, Cay S; Cornell, Gary. Core Java, Vol.1 e 2. Makron, 2001.

Complementar

4. MEYER, Bertrand. OBJECT-ORIENTED SOFTWARE CONSTRUCTION. 2nd ed. Upper Saddle River: Prentice Hall PTR, c1997. 1254 p.1997 ISBN 0-13-629155-4.
5. TREMBLAY, Jean-Paul; CHESTON, Grant A. DATA STRUCTURES AND SOFTWARE DEVELOPMENT IN AN OBJECT-ORIENTED DOMAIN. Upper Saddle River: Prentice Hall, 2001. 1037 p. ISBN: 0-13-787946-6.
6. ARNOLD, Ken; GOSLING, James; HOLMES, David. A LINGUAGEM DE PROGRAMAÇÃO JAVA. Porto Alegre: Bookman, 2007. 799 p. ISBN 978-85-60031-64-1.
7. FLANAGAN, David. JAVA: O GUIA ESSENCIAL. tradução [da 5.ed. original de Edson Furmankiewicz.]. Porto Alegre: Bookman, 2006. 1099 p. ISBN 85-60031-07-3.
8. PREISS, Bruno R. ESTRUTURAS DE DADOS E ALGORITMOS: PADRÕES DE PROJETOS ORIENTADOS A OBJETO COM JAVA. [tradução Elizabeth Ferreira Gouvêa]. Rio de Janeiro: Campus, 2001. 566 p. ISBN: 85-7110-0693-0.

9. Acessibilidade**

Não houve necessidade de adaptação.

10. Adaptações para práticas profissionais**

Não se aplica.

** conforme nota técnica conjunta número 17/2020 CGLNRS/DPR/SERES/SERES