Unidade Curricular

“Padrões e Desenho de Software”

António José Ribeiro Neves

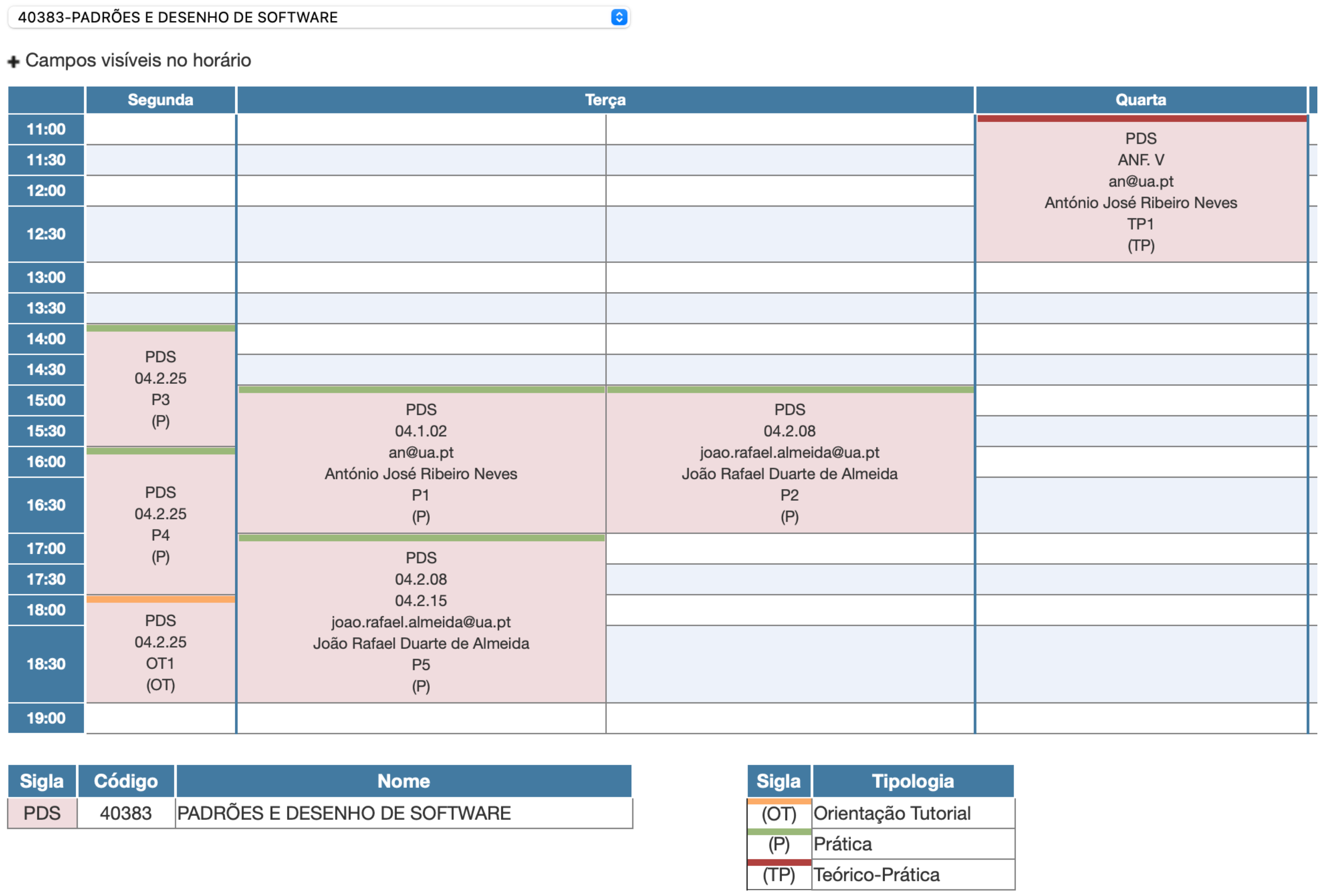
an@ua.pt

https://www.ua.pt/pt/uc/12275

Outline

Instructores and Timetable

- António Neves
- João Almeida
- Rafael Direito
- Lúcia Sousa



40383-PADRÕES E DESENHO DE SOFTWARE

+ Campos visíveis no horário

	Segunda	Terça	Terça	Quarta
11:00				PDS ANF. V an@ua.pt António José Ribeiro Neves TP1 (TP)
11:30				
12:00				
12:30				
13:00				
13:30				
14:00	PDS 04.2.25 P3 (P)			
14:30				
15:00		PDS 04.1.02 an@ua.pt António José Ribeiro Neves P1 (P)	PDS 04.2.08 joao.rafael.almeida@ua.pt João Rafael Duarte de Almeida P2 (P)	
15:30				
16:00	PDS 04.2.25 P4 (P)			
16:30				
17:00		PDS 04.2.08 04.2.15 joao.rafael.almeida@ua.pt João Rafael Duarte de Almeida P5 (P)		
17:30				
18:00	PDS 04.2.25 OT1 (OT)			
18:30				
19:00				

Sigla	Código	Nome
PDS	40383	PADRÕES E DESENHO DE SOFTWARE

Sigla	Tipologia
(OT)	Orientação Tutorial
(P)	Prática
(TP)	Teórico-Prática

Motivation

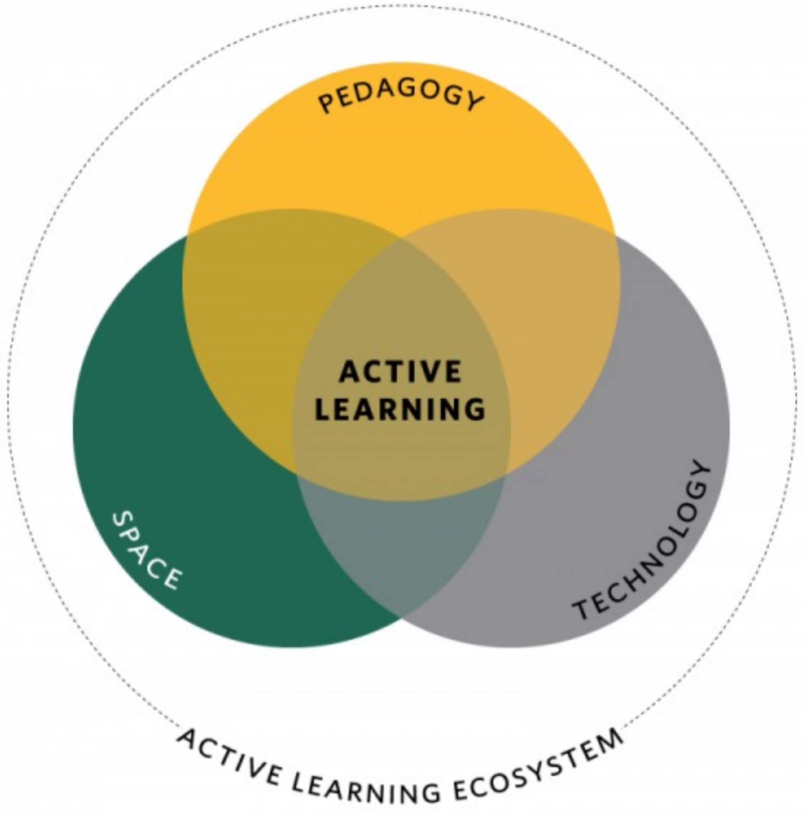
- Design patterns are general **reusable solutions** to commonly **occurring problems** within a specific context in **software design**
- They represent **best practices** for solving certain design problems and are aimed at improving the **maintainability**, **scalability**, and **efficiency** of software systems
- Design patterns provide a common language for developers to communicate solutions to design problems and **facilitate code reusability**
- This course focuses on the most modern approach to design patterns, emphasizing their role in object-oriented programming and their implementation in Java

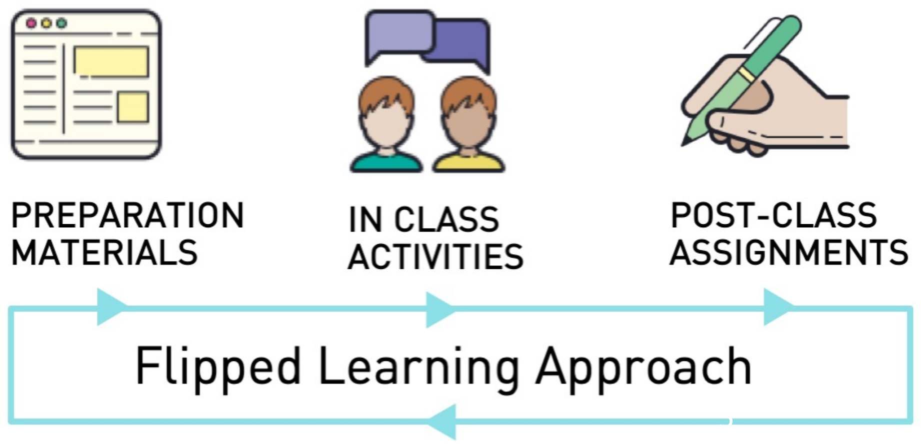
Learning Outcomes

- Identify and understand the different types of design patterns: creational, structural, and behavioral.
- Explain the principles and motivations behind each design pattern.
- Recognize the appropriate situations to apply each design pattern.
- Develop practical applications of design patterns using Java programming language.
- Analyze existing software systems to identify and apply design patterns effectively.
- Discuss the evolution and modern trends in design pattern usage.

Active learning

- This course will focus not only on technical knowledge but also on the development of critical thinking skills and self-regulated learning
- Students will:
    - learn how to analyze complex software design problems and select appropriate design patterns to solve them, considering factors such as scalability, maintainability, and performance
    - develop critical thinking skills by evaluating the trade-offs and implications of using different design patterns in various contexts, and making informed decisions based on their analysis
    - engage in collaborative learning activities, such as group projects and peer reviews
    - apply design patterns in practical software development projects
    - explore the ethical and professional implications of design pattern usage
    - recognize the importance of lifelong learning in the field of software engineering, and develop a mindset of continuous improvement and adaptation to new technologies and methodologies





Contents

Introduction to Design Patterns

- Definition and importance of design patterns in software engineering
- Overview of common design principles and their role in software design

Creational Design Patterns

- Singleton pattern
- Factory method pattern
- Abstract factory pattern
- Builder pattern
- Prototype pattern (optional)

Structural Design Patterns

- Adapter pattern
- Decorator pattern
- Proxy pattern
- Facade pattern
- Composite pattern
- Bridge pattern (optional)

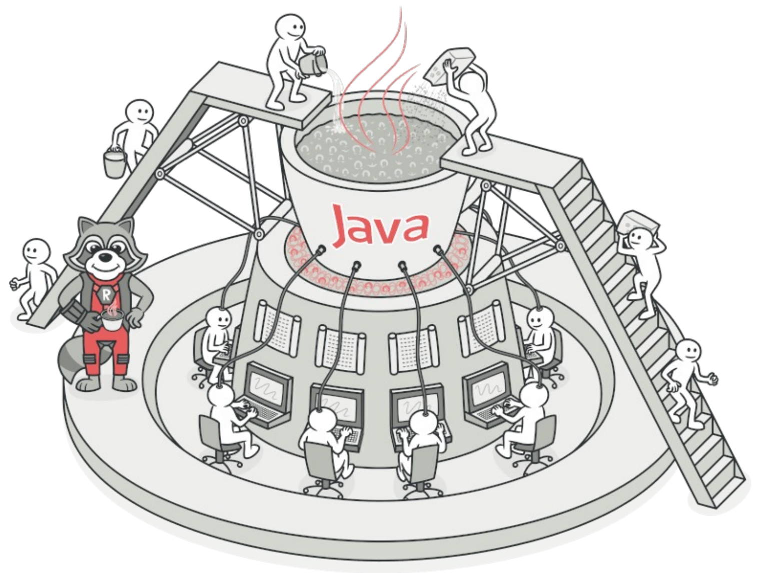
Behavioral Design Patterns

- Observer pattern
- Strategy pattern
- Command pattern
- Template method pattern
- Iterator pattern
- State pattern (optional)

Software architecture Design Patterns

Best Practices and Guidelines

Hands-On Exercises and Projects

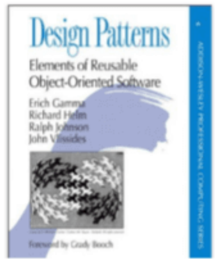


John Vlissides

Erich Gamma

Ralph Johnson

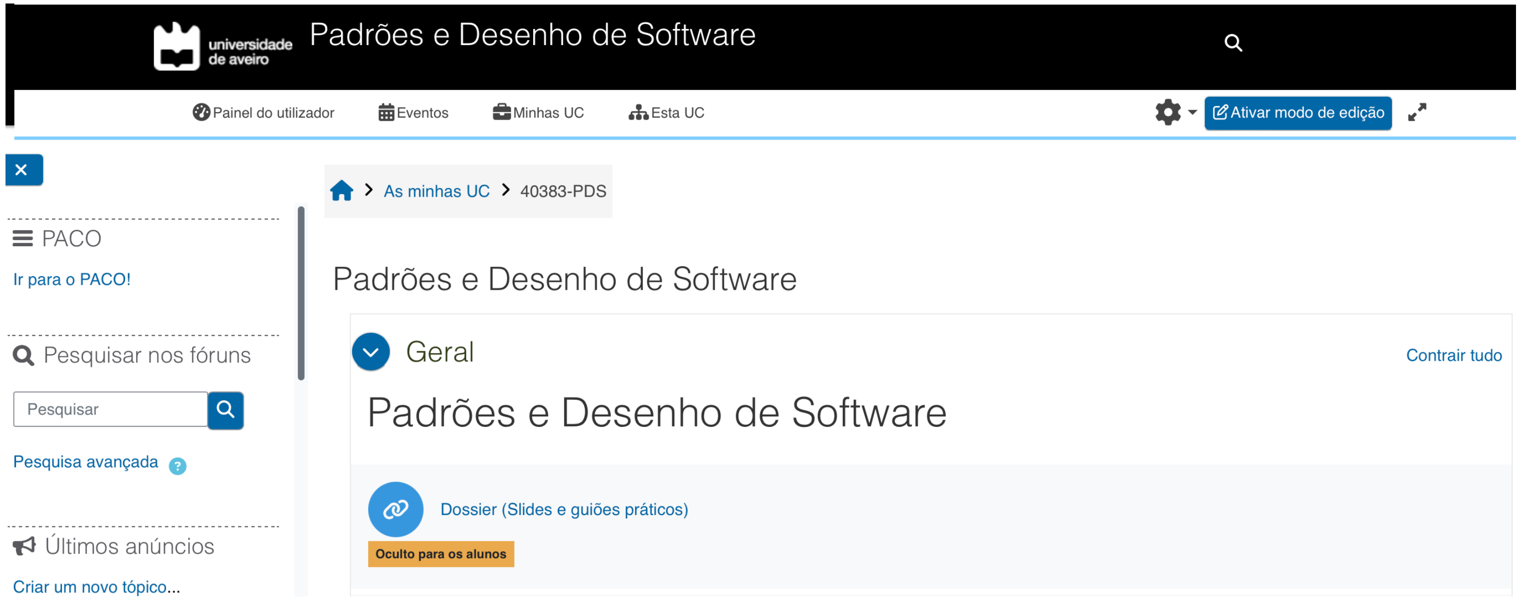
Richard Helm


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Program to an interface not an implementation

Favor object composition over inheritance

Learning tools


universidade de aveiro Padrões e Desenho de Software

Painel do utilizador · Eventos · Minhas UC · Esta UC · Ativar modo de edição

As minhas UC > 40383-PDS

PACO

Ir para o PACO!

Pesquisar nos fóruns

Pesquisar

Pesquisa avançada

Últimos anúncios

Criar um novo tópico...

Padrões e Desenho de Software

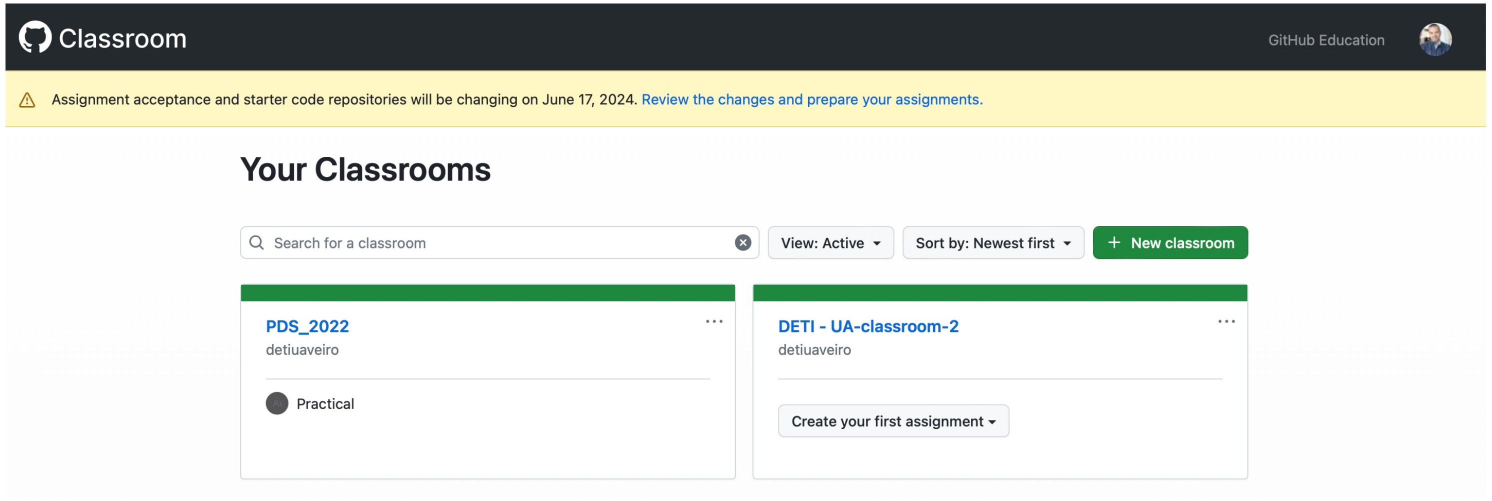
Geral

Contrair tudo

Padrões e Desenho de Software

Dossier (Slides e guiões práticos)

Oculto para os alunos


Classroom

GitHub Education

Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments.

Your Classrooms

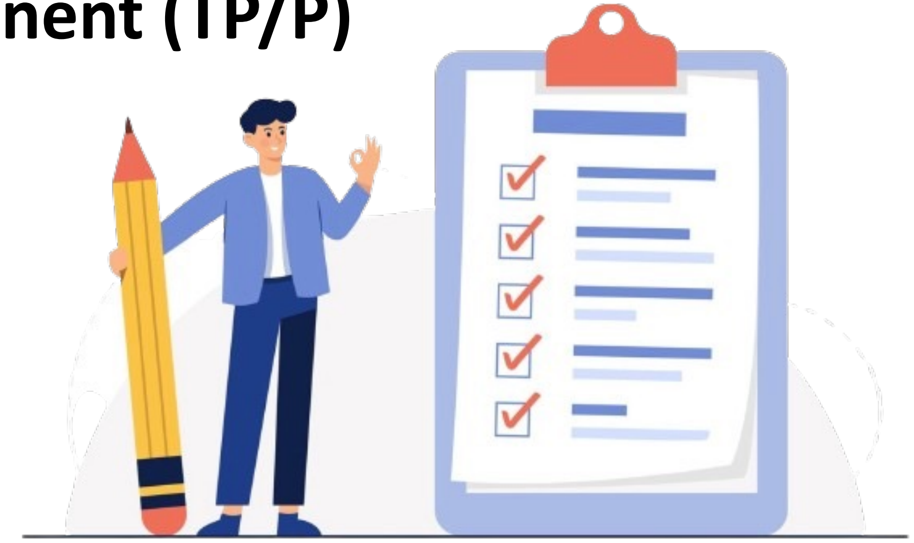
Search for a classroom · View: Active · Sort by: Newest first · New classroom

PDS_2022
detiuaveiro
Practical

DETI - UA-classroom-2
detiuaveiro
Create your first assignment

Assessment (still learning tools)

- 40% - Quizzes or other tasks during theoretical classes
- 20% - Code submissions, code analysis, presentations during practical classes
- 40% - Final practical exam (5th June 2024, afternoon)
- **7 values minimum in each component (TP/P)**



Assessment (still learning tools)

- **Overall Assessment Criteria**:
  - Understanding of design patterns: Demonstrated ability to identify, describe, and apply relevant design patterns to solve software design problems
  - Coding proficiency: Proficiency in implementing design patterns in code, considering factors such as code readability, maintainability, and performance
  - Critical thinking and problem-solving skills: Ability to analyze complex problems, evaluate alternative solutions, and make informed design decisions
  - Communication skills: Effective communication of design choices through written code, presentations, and oral explanations
  - Collaboration and teamwork (where applicable): Ability to work collaboratively in a team setting, contributing to group projects and providing constructive feedback to peers

Bibliography

- Weekly documentation is available on elearning
  - Websites
  - Tutorials
  - Book chapters
  - Slides
  - Problems

Final remarks

- This is a 6 ECTS course, implying, on average, a total of about $6 \times 30 = 180$ hours of work.
- Academic dishonesty will not be tolerated. Academic dishonesty involves acts, such as,
  - Cheating on an examination or quiz.
  - Substituting for another person during an examination or allowing such substitution for one's self.
  - Plagiarism. Act of appropriating passages from the work of another individual, either word for word or in substance, and representing them as one's own work. This includes any submission of written work other than one's own.
  - Collusion with another person in the preparation or editing of assignments submitted for credit, unless such collaboration has been approved in advance by the instructor.
- If you have doubts regarding a certain action, ask the instructors.