



Unidade Curricular

“Padrões e Desenho de Software”

#14 – Design Patterns Interview Preparation Workshop

António José Ribeiro Neves

an@ua.pt

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



universidade
de aveiro



IEETA





Outline

Challenge presentations

Questions preparation

Interviews

Discussion

Design Patterns Interview Prep. Workshop

Review all design patterns covered in the course by simulating a job interview setting. This activity will help you understand how design patterns are applied in real-world scenarios and prepare you for common interview questions related to design patterns.

Steps:

- 5 min. to explain the activity
- 5 min. to decide the groups that will interview / will be interviewed
- Each group decides who will record a video of the interview (max 2 min)
- 30 minutes - Revise the material of the classes in pure silence OR prepare the questions for the interview (in silence also)
- 30 minutes interview outdoor
- 15 minutes final discussion



References

- (google: “job interview questions about software design patterns in java”
 - <https://www.geeksforgeeks.org/top-30-java-design-patterns-interview-question/>
 - <https://www.javatpoint.com/java-design-pattern-interview-questions>
 - <https://www.interviewbit.com/design-patterns-interview-questions/>
 - ...
- AI 😊



AI proposal for an interview

- **What are design patterns and why are they important in software development?**
 - Follow-up: Can you name a few commonly used design patterns?
- **Can you explain the Singleton pattern and provide an example of where it might be used in a Java application?**
 - Follow-up: How would you implement a Singleton in Java?
- **Describe the Factory Method pattern. How does it differ from the Abstract Factory pattern?**
 - Follow-up: Can you provide a simple Java code example of the Factory Method pattern?
- **What is the Observer pattern and how does it work?**
 - Follow-up: Can you describe a scenario where you might use the Observer pattern in a Java application?
- **How does the Decorator pattern enhance or modify the behaviour of an object?**
 - Follow-up: Can you give an example of how you might implement the Decorator pattern in Java?

Time for
discussion 😊

