

Instructions



How is it going?

0 responses

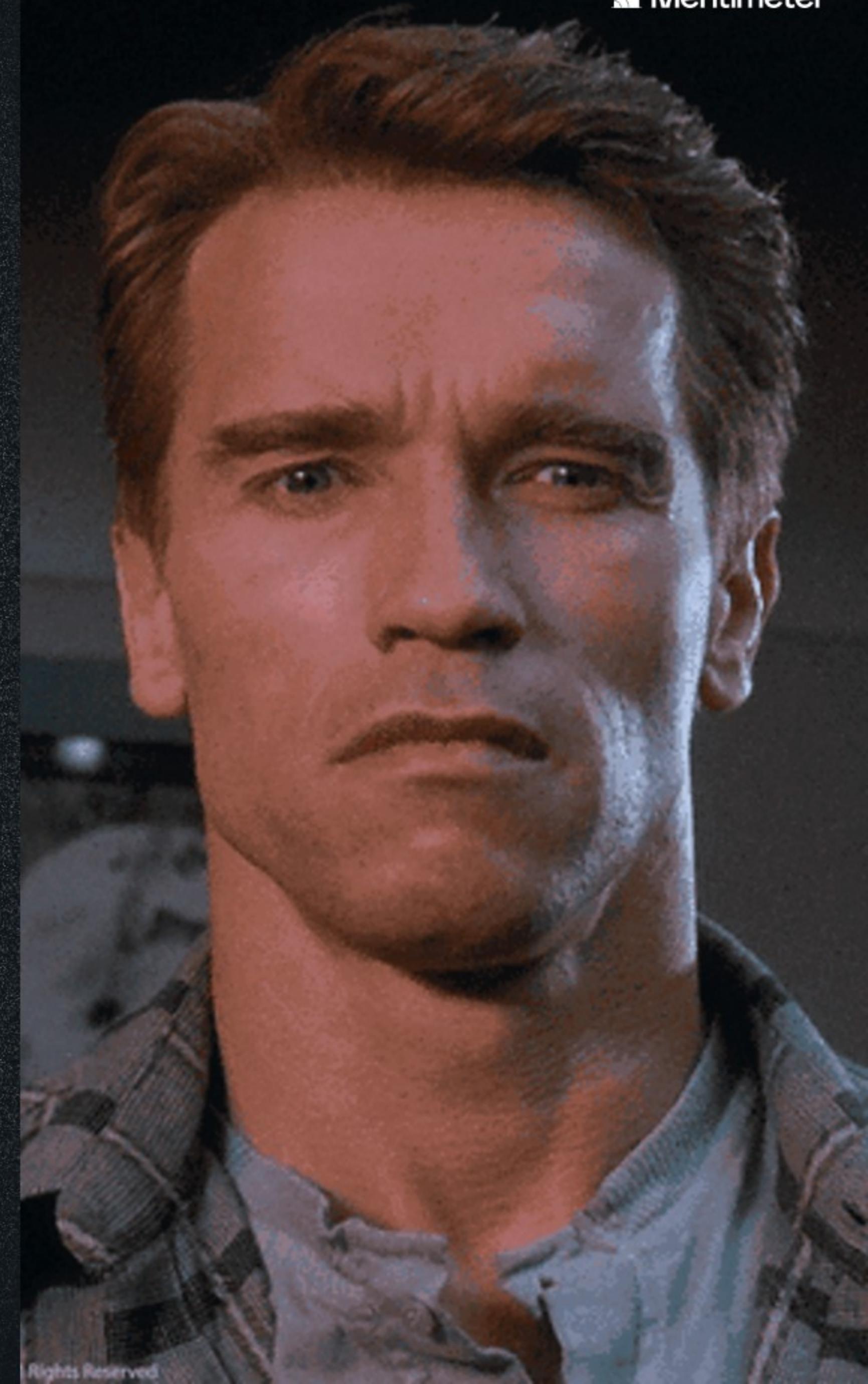


Do you have problems with practice?

Agenda

- Recall last session
- Discuss current topic
- Practice

Total Recall



What do you remember most from the previous session?



DIP is about

0 

High-Level and low-level models depend on Abstraction, which ALL depends on Details

0 

High-Level and low-level models AND Details ALL depend on Abstraction

0 

everyone is independent



DIP by Martin

High-level modules should not depend on low-level modules.

Both should depend on abstractions.

Abstractions should not depend on details. Details should depend on abstractions.

DIP basic rule

0 

Combine layers

0 

Separate layers

0 

Remove layers

IoC types



Dependency Injection



Dependency Lookup

What is IoC Container?

0 responses



Conclusion

DI is when you create objects separately and put one into another (i.e. inject): a pattern that let's you test, maintain code easier.

IoC helps you to do DI using instructions (e.g. xml), store objects, reuse, control lifecycle (main player is ioc-container).

DIP provides rules to setup modules\classes\objects and their relations (a principle with rules)



Discuss current
topic



Random fact

The two highest IQ rates ever recorded on Earth belong to women.

Have you looked into MVC?

0



yes

0



no

What modules divide an application according to the MVC scheme?

0 

Model, View, Controller

0 

domain model, user
interaction

MVC benefits

0 responses

MVC benefits

- Organizes large-size web applications
- Supports Asynchronous Method Invocation
- MVC Model Returns the Data Without Formatting
- Develop Multiple View Components for your Model
- Modifications do not Affect the Entire Model
- Accelerated Development Process (including Testing)

Which MVC implementation is sometimes called MVP?

0 

Model View Passive

0 

Model View Presenter

0 

When M and V are designed to act passively

What is the observer pattern used for in MVC?



What is the facade pattern used for in MVC?



Advantages of the MVC Design Pattern

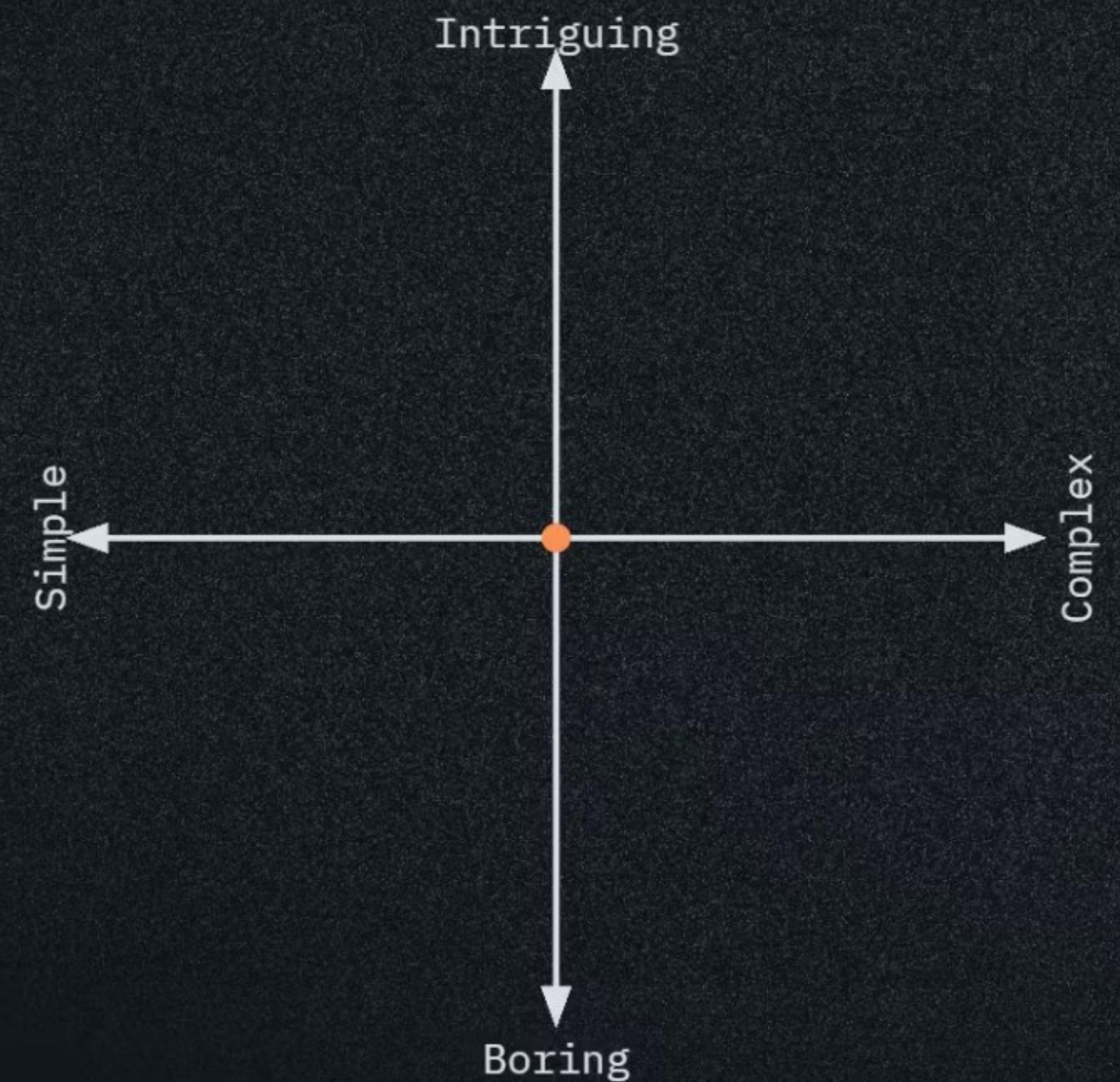
- Separation of Concerns: MVC separates data, UI, and logic, making the code easier to understand, maintain, and modify
- Modularity: Each component (Model, View, Controller) can be developed and tested separately: reusability and scalability
- Flexibility: changes to one component do not affect the others: easier updates and modifications
- Parallel Development: Multiple developers can work on different components simultaneously, speeding up dev process
- Code Reusability: components can be reused in other parts of app or in other projects, reducing dev time and effort
- Item 6

Disadvantages of the MVC Design Pattern

- Complexity: Implementing MVC can add complexity to code, especially for simple apps, leading to overhead in dev
- Learning Curve: Devs need to understand the concept of MVC and how to implement it effectively
- Overhead: communication between components can lead to overhead, affecting the performance of the app
- Potential for Over-Engineering: In some cases, devs may over-engineer the app by adding unnecessary abstractions and lay
- Increased File Count: MVC can result in a larger number of files and classes compared to simpler architectures

10 minute break

Express your opinion



1 What do you think
about current course



Q&A part

0 questions
0 upvotes

Quiz leaderboard

No results yet

Top Quiz participants will be displayed here once there are results!