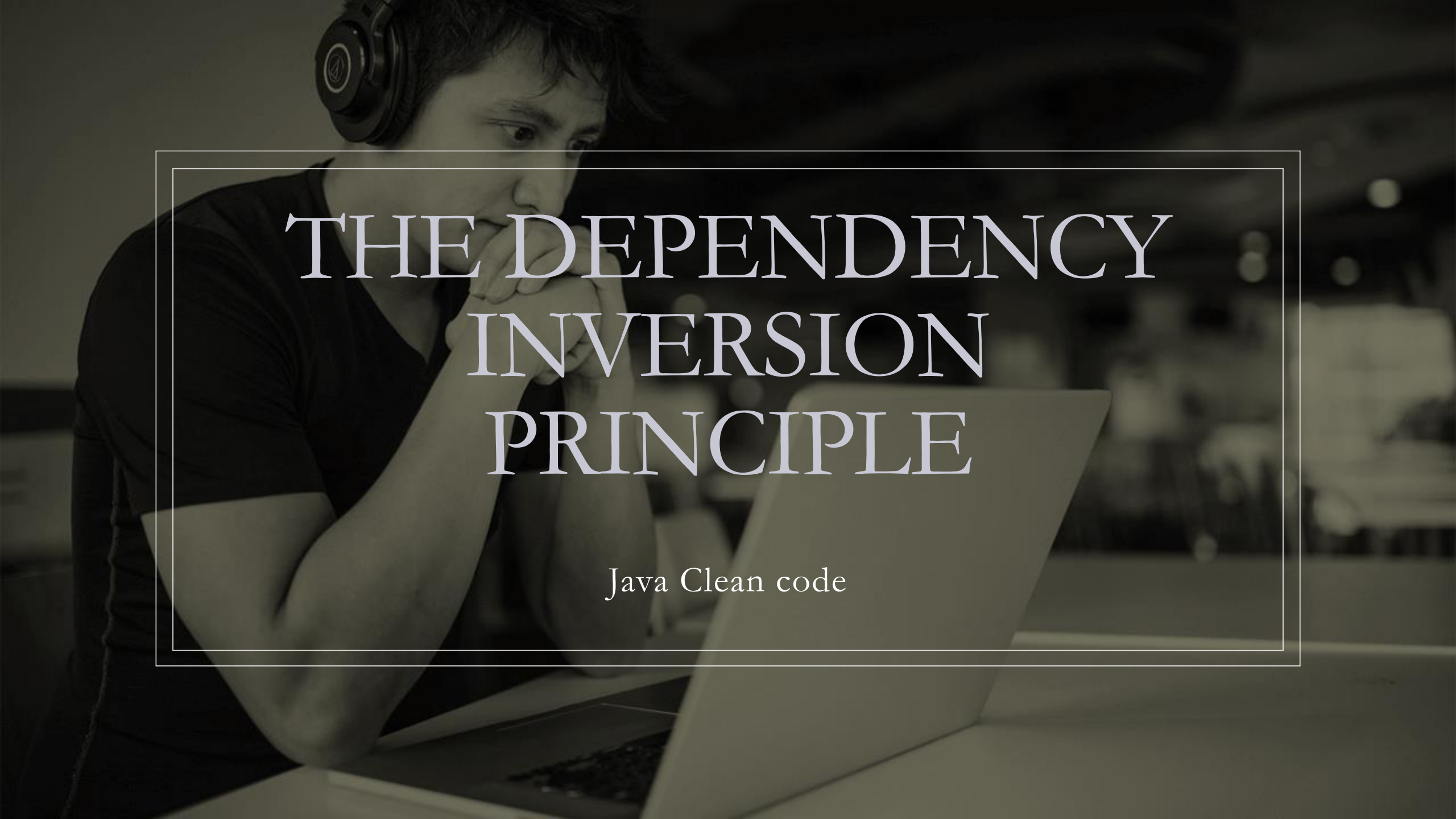


THE DEPENDENCY INVERSION PRINCIPLE

Java Clean code

Outline

Lesson 6.

The Single Responsibility Principle

Lesson 7.

The Open Closed Principle

Lesson 8.

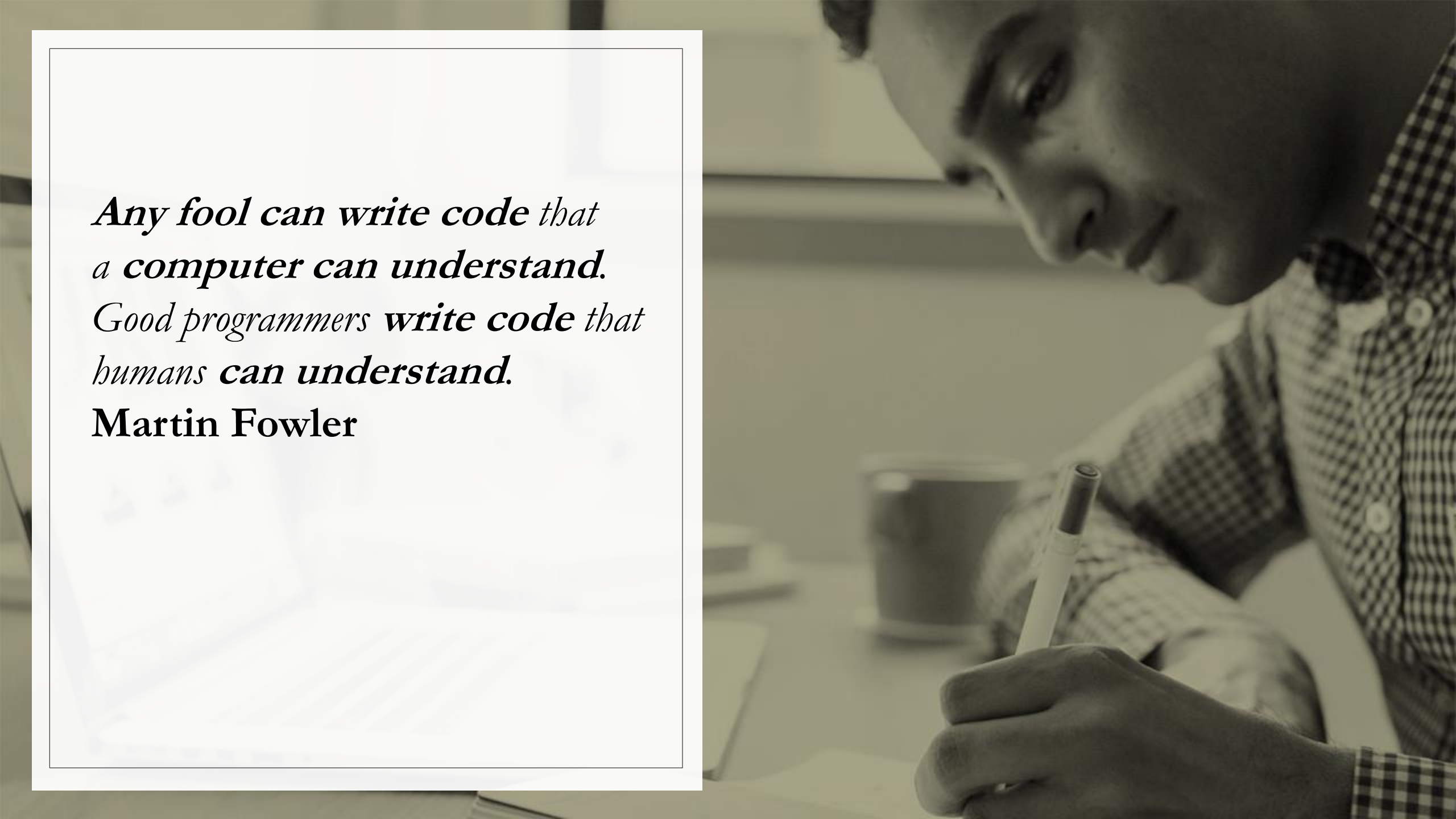
The Liskov Substitution Principle

Lesson 9.

The Interface Segregation Principle

Lesson 10.

The Dependency Inversion Principle



Any fool can write code *that a* ***computer can understand.*** *Good programmers* ***write code*** *that humans* ***can understand.***

Martin Fowler

The Dependency Inversion Principle

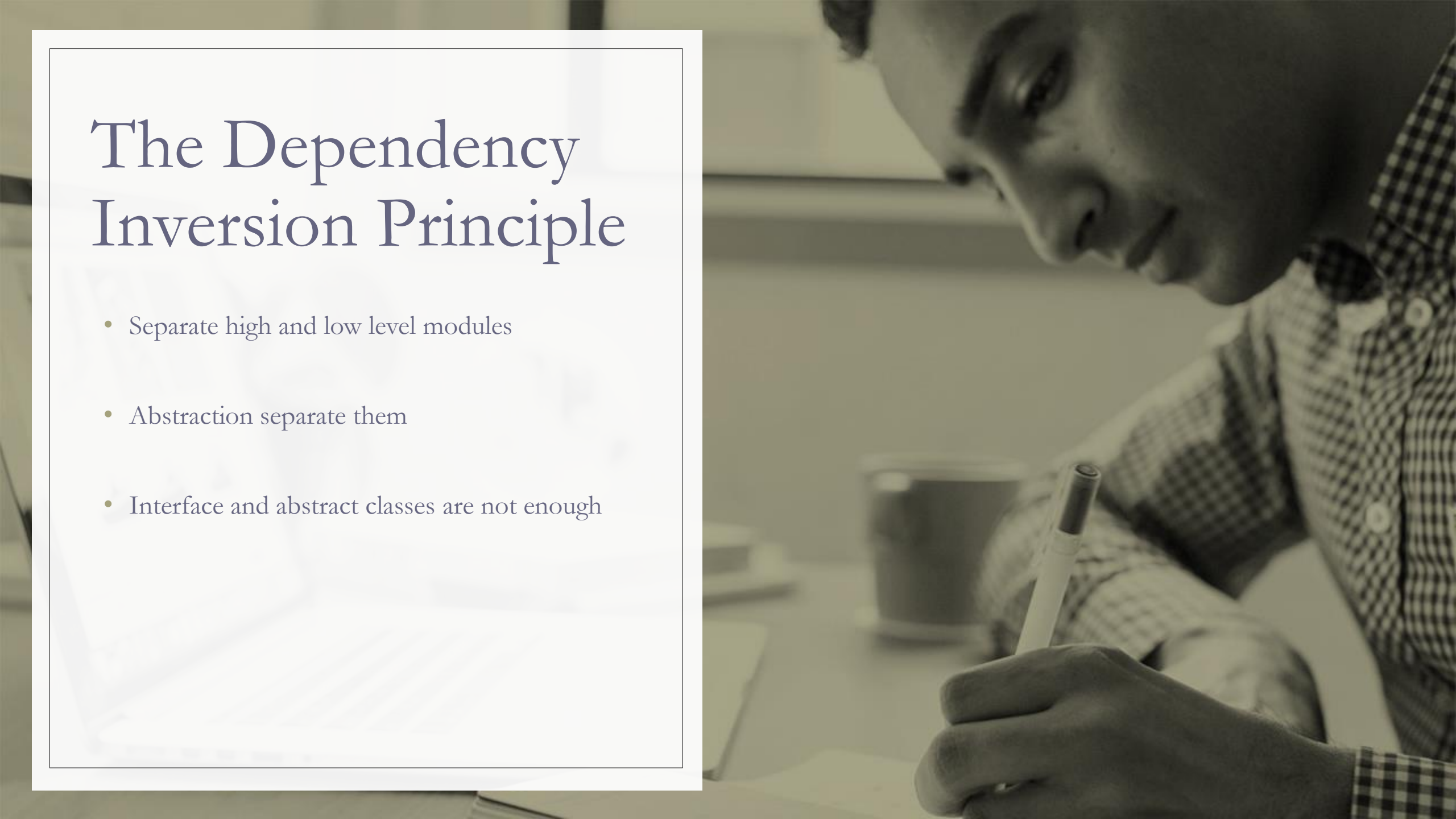
- Based on the OCP & LSP
- Maintainability
- Testability
- Extensible
- Reusable

The Dependency Inversion Principle

- *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.*

The Dependency Inversion Principle

- Separate high and low level modules
- Abstraction separate them
- Interface and abstract classes are not enough



Dependency injection

- How you acquire the dependency
- Inversion of control

Runtime & Compile time

- Compile time not a problem
- Deployment
- Big projects need to work independently

Dependency Inversion Principle Summary

- Maintainability
- Testability
- Extensible
- Reusable

Course Progress

Lesson 6	Lesson 7	Lesson 8	Lesson 9	Lesson 10
The Single Responsibility Principle	The Open Closed Principle	The Liskov Substitution Principle	The Interface Segregation Principle	The Dependency Inversion Principle

DIP

https://martinfowler.com/articles/dipInTheWild.html#SynopsisOfTheDip


THANK YOU!