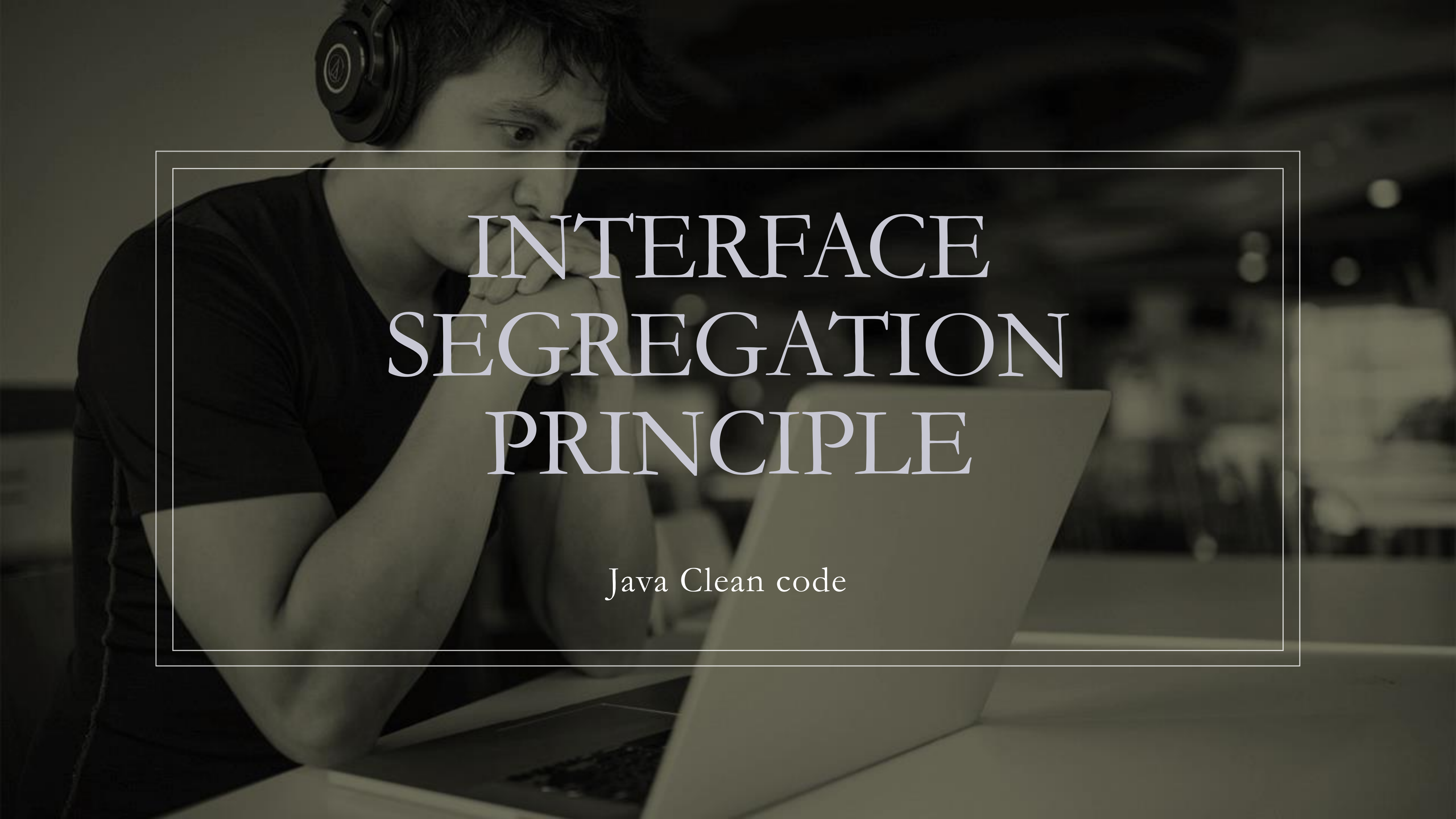
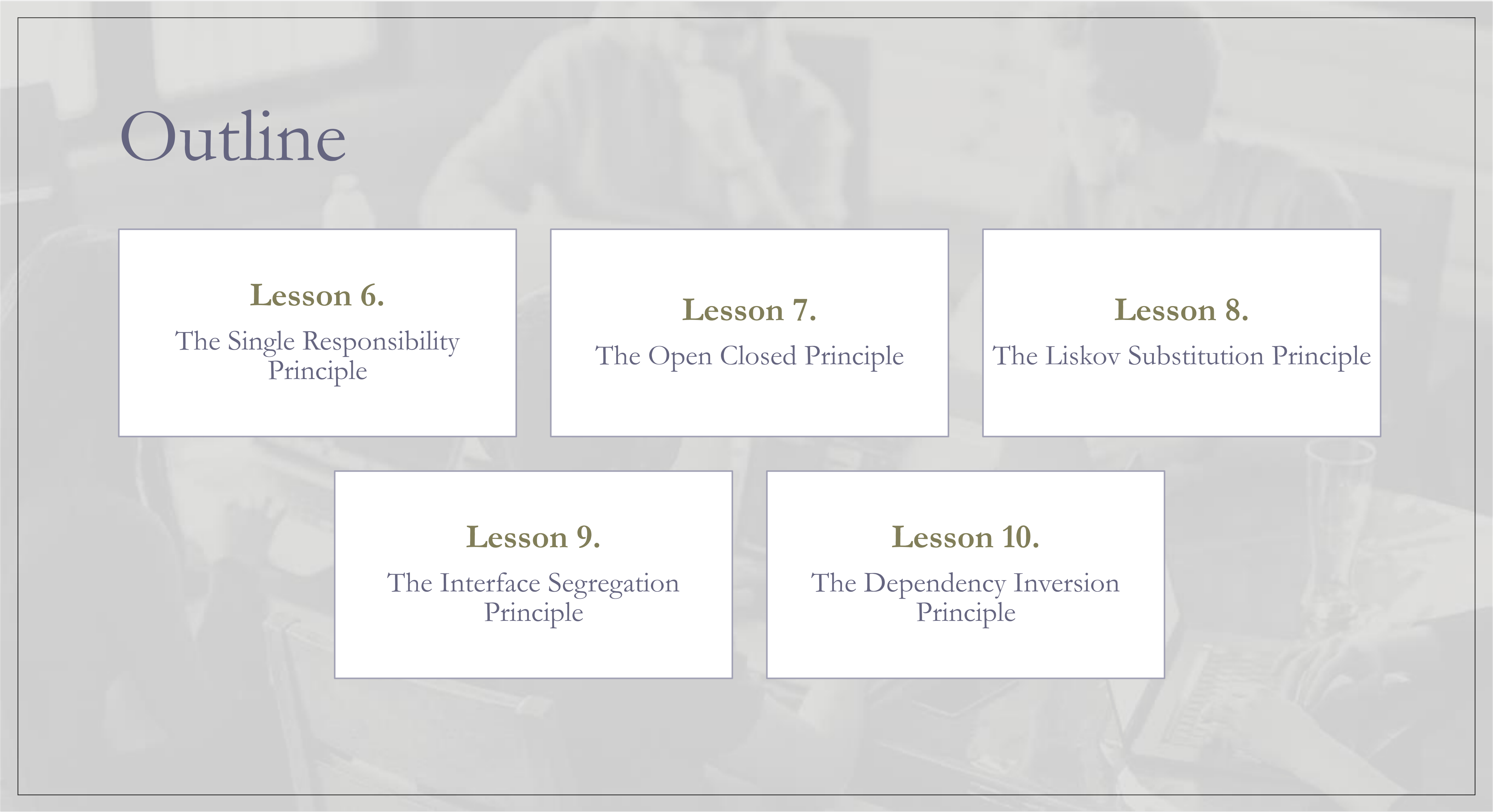


INTERFACE SEGREGATION PRINCIPLE

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

Interface segregation principle

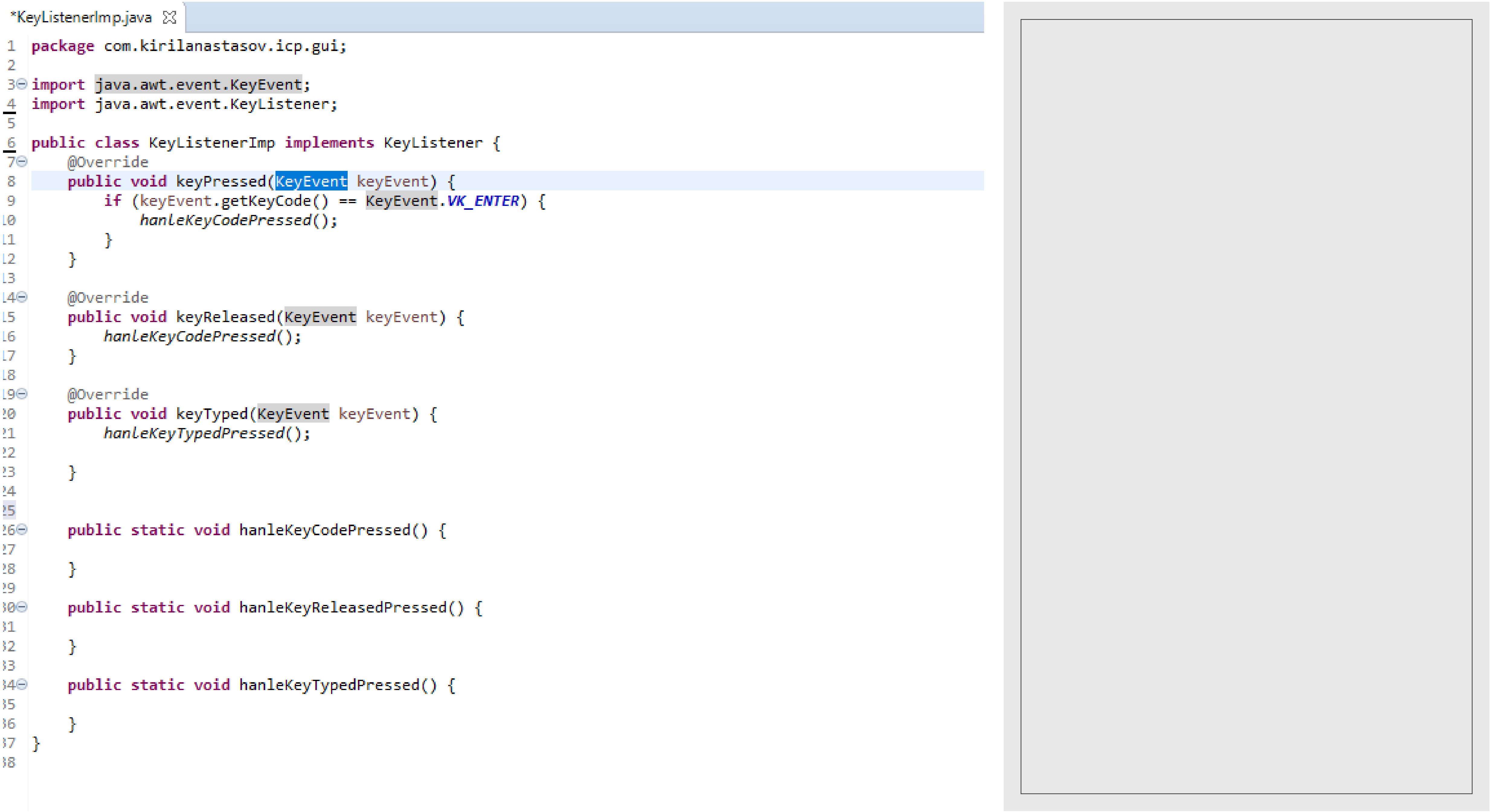
- *“Clients should not be forced to depend upon interfaces that they do not use.”*
- Code is easier to maintain and debug
- Interfaces that are responsible for a client specific needs

Fat classes

- A lot of methods
- Tight Coupling
- Dependencies

ISP Java Examples

- FocusListener
- KeyListener
- MouseMotionListener
- TestListener
- WindowFocustListener

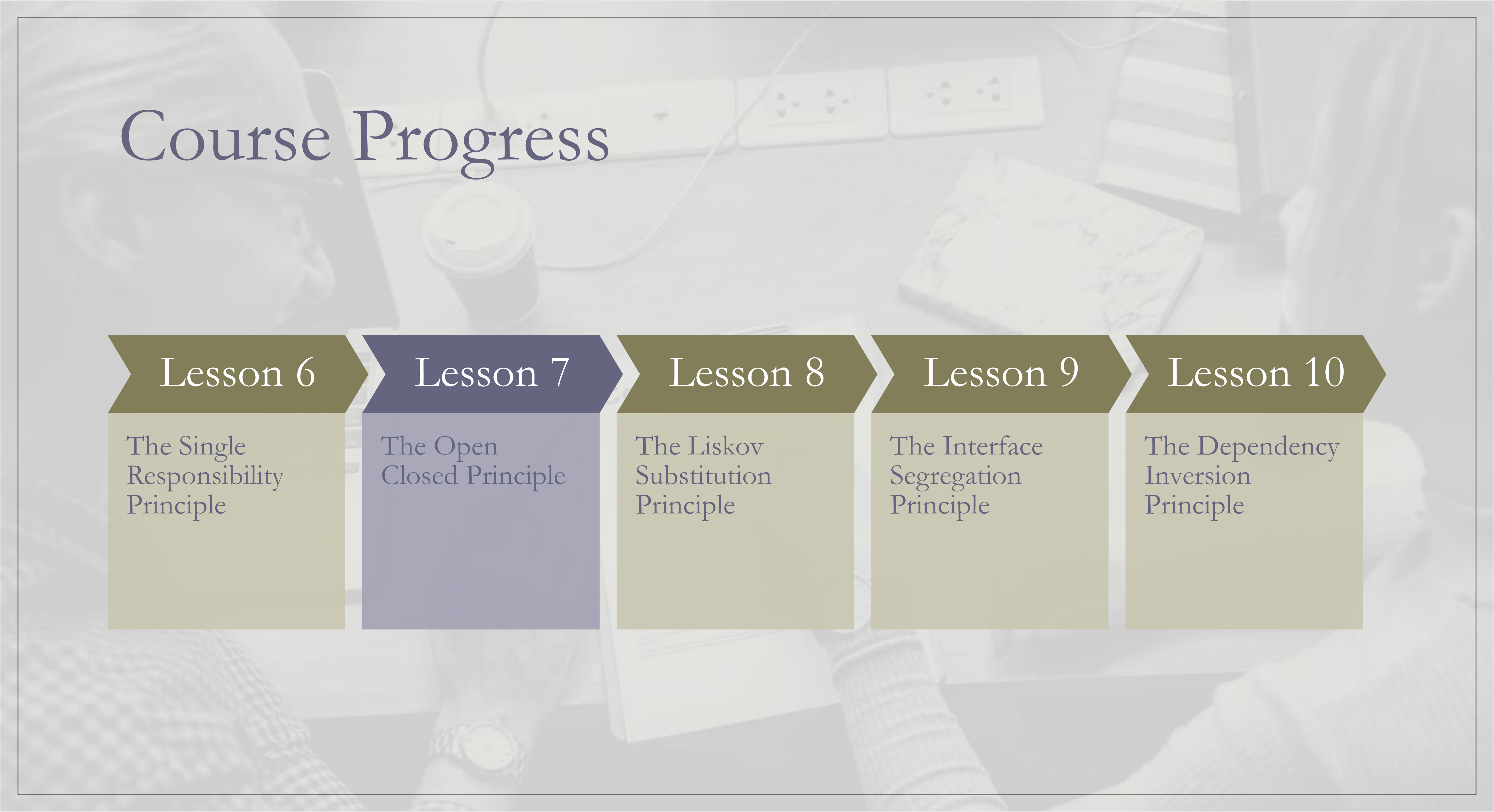


```java
package com.kirilanastasov.icp.gui;

import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class KeyListenerImp implements KeyListener {
    @Override
    public void keyPressed(KeyEvent keyEvent) {
        if (keyEvent.getKeyCode() == KeyEvent.VK_ENTER) {
            hanleKeyCodePressed();
        }
    }

    @Override
    public void keyReleased(KeyEvent keyEvent) {
        hanleKeyCodePressed();
    }

    @Override
    public void keyTyped(KeyEvent keyEvent) {
        hanleKeyTypedPressed();

    }


    public static void hanleKeyCodePressed() {

    }

    public static void hanleKeyReleasedPressed() {

    }

    public static void hanleKeyTypedPressed() {

    }
}
```

Interface Segregation Principle Summary

- Clients should not be forces to implement unnecessary code
- Code is easier to maintain and debug
- Interfaces that are responsible for a client specific needs

Course Progress



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

THANK YOU!