THE SINGLE RESPONSIBILITY 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



SOLID

- Single Responsibility Principle
- Open Closed Principle
- Liskov Substitute Principle
- Interface Segregation Principle
- The Dependency Inversion Principle

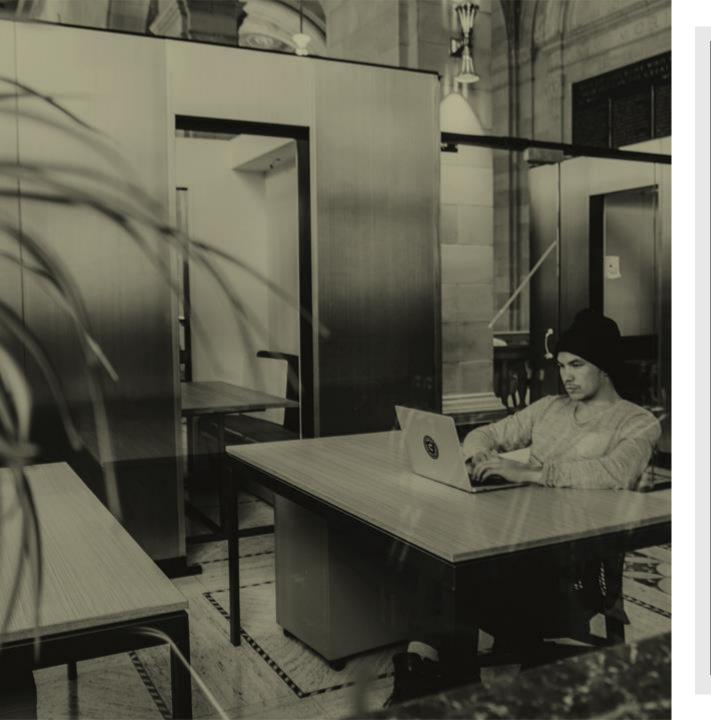


The Single Responsibility Principle

Class should have one and only one reason to change

- Loose coupling
- Code is easier to maintain
- Easier to add new functionality
- Easier to debug





Spring Data Repository

```
public interface DriverRepository
extends CrudRepository<Driver,

Long> {
    List findByLastName(String)
lastName);
```

```
package com.kirilanastasov.srp;
3⊕ import java.io.File;
 13
     public class FileManager {
 15
         public void downloadFile(String location) throws IOException {
 16⊖
             URL website = new URL("http://www.google.com/");
 17
             ReadableByteChannel rbc = Channels.newChannel(website.openStream());
 18
Q<sub>1</sub>19
             FileOutputStream fos = new FileOutputStream("google.html");
             fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
 20
 21
 22
 23⊝
         public void parseTheFile(File file) throws FileNotFoundException {
             Scanner input = new Scanner(file);
 24
 25
             while (input.hasNext()) {
Q<sub>1</sub>26
                 String nextToken = input.next();
 27
                 // or to process line by line
Q<sub>6</sub>28
                 String nextLine = input.nextLine();
 29
             input.close();
 30
 31
 32
 33⊝
         public void persistTheData(List<String> data) {
 34
             Employer entity = new Employer(100, "name");
 35
             entity.random = 1234;
36
             Object gson;
 37
 38 }
 39
```

```
🕡 FileDownloader.java 🛭 🕡 FileParser.java
                                                                            FileManager.java
                  Employer.java
 package com.kirilanastasov.srp;
 3⊖ import java.io.FileOutputStream;
 4 import java.io.IOException;
 5 import java.net.URL;
 6 import java.nio.channels.Channels;
   import java.nio.channels.ReadableByteChannel;
   public class FileDownloader {
10
11⊝
        public void downloadFile(String location) throws IOException {
12
            URL website = new URL("http://www.google.com/");
13
            ReadableByteChannel rbc = Channels.newChannel(website.openStream());
            FileOutputStream fos = new FileOutputStream("google.html");
15
           fos.getChannel().transferFrom(rbc, 0, Long.MAX VALUE);
16
17 }
18
```

```
🕡 FileParser.java 🛭 🔎 DataPersist.jav
Employer.java
                                  package com.kirilanastasov.srp;
  2
  3⊖ import java.io.File;
  4 import java.io.FileNotFoundException;
  5 import java.util.Scanner;
    public class FileParser {
        public void parseTheFile(File file) throws FileNotFoundException {
            Scanner input = new Scanner(file);
           while (input.hasNext()) {
               String nextToken = input.next();
%12
               String nextLine = input.nextLine();
13
14
            input.close();
15
16 }
 17
```

```
🔝 DataPersist.java 🛭
                   Employer.java
FileManager.java

√ FileDownloader.java

√ FileParser.java

  package com.kirilanastasov.srp;
  3 import java.util.List;
    public class DataPersist {
        public void persistTheData(List<String> data) {
             Employer entity = new Employer(100, "name");
            entity.random = 1234;
  8
            Object gson;
 10
 11
12
13
```



Single Responsibility Lesson Summary

- A Class should have one and only one reason to change.
- SRP is one of the most commonly used principles in software development.
- It is easy to violate the Single Responsibility
 Principle.

Course Progress

Lesson 6

The Single Responsibility Principle

Lesson 7

The Open Closed Principle

Lesson 8

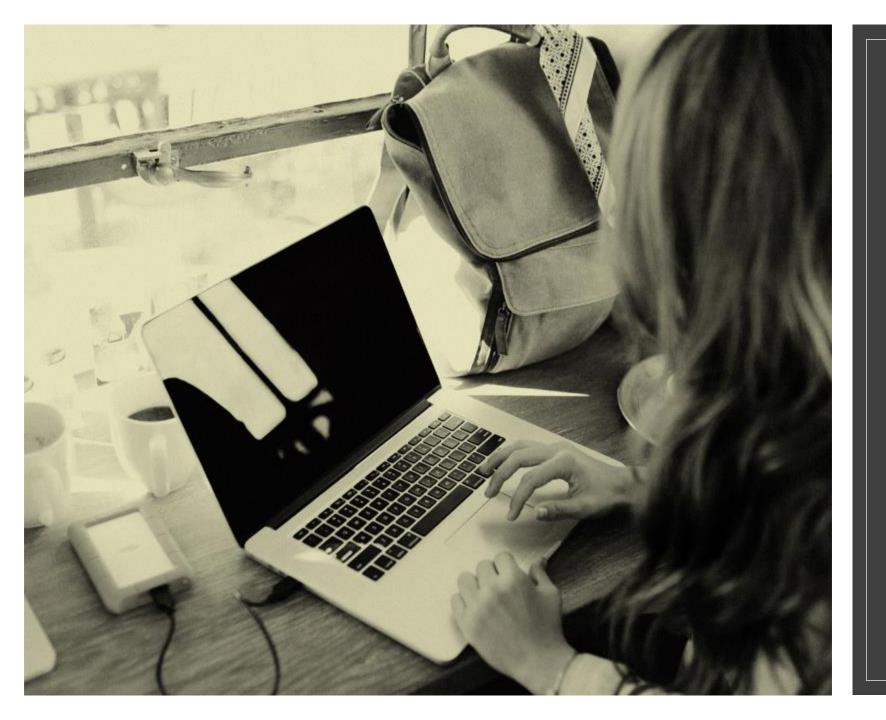
The Liskov Substitution Principle

Lesson 9

The Interface Segregation Principle

Lesson 10

The Dependency Inversion Principle



The Single Responsibili ty Principle

https://blog.cleancoder.com/uncle bob/2014/05/08/SingleReponsibilityPrinciple html

