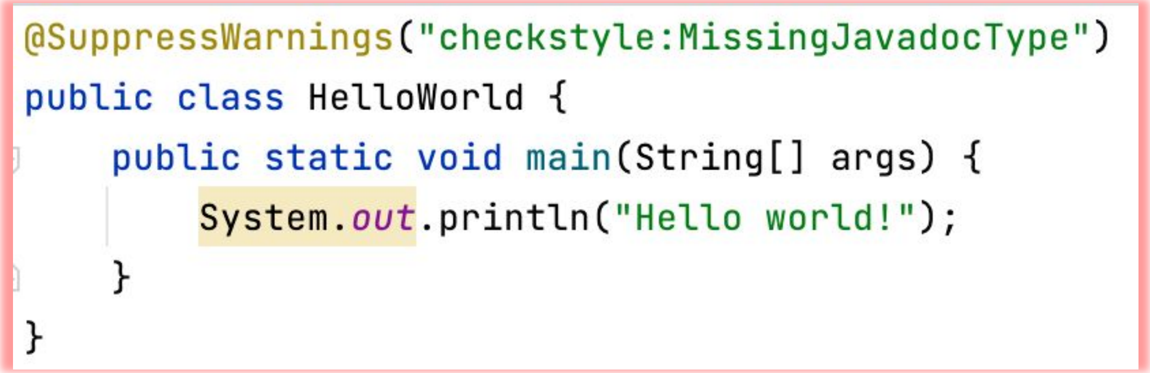
Lesson 4 – Homework review

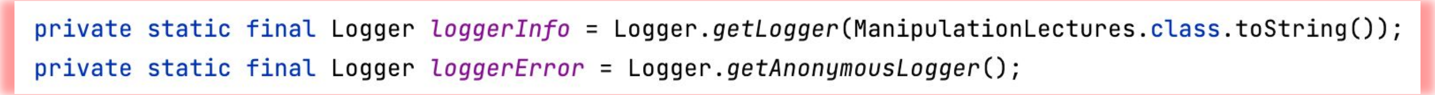
Throw away the past



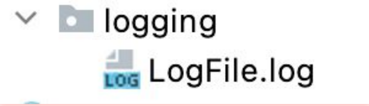
```java
@SuppressWarnings("checkstyle:MissingJavadocType")
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

HelloWorld

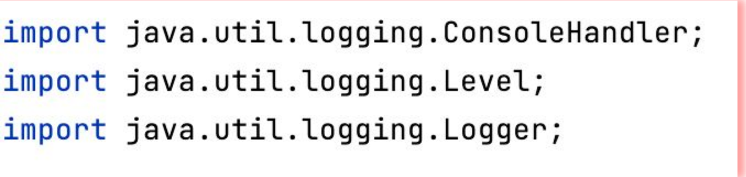
Task 1 Not implemented



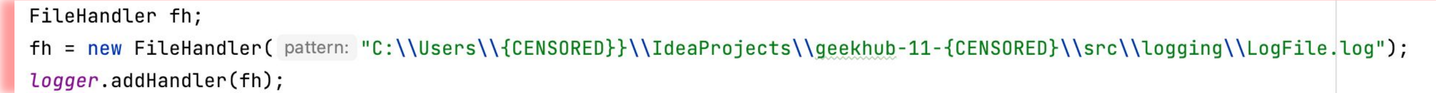
```java
private static final Logger loggerInfo = Logger.getLogger(ManipulationLectures.class.toString());
private static final Logger loggerError = Logger.getAnonymousLogger();
```



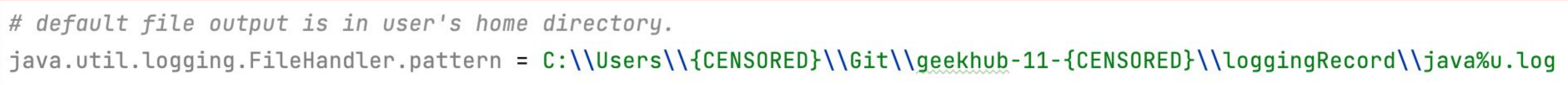
```
logging
    LogFile.log
```



```java
import java.util.logging.ConsoleHandler;
import java.util.logging.Level;
import java.util.logging.Logger;
```

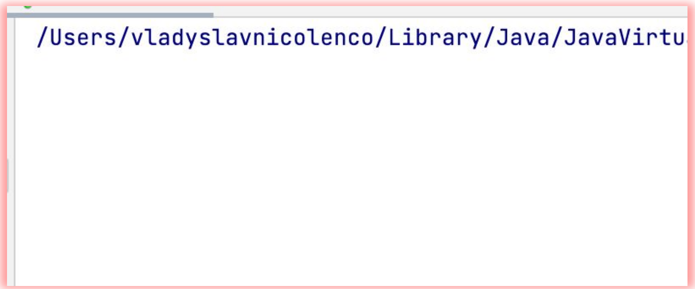


```java
FileHandler fh;
fh = new FileHandler( pattern: "C:\\Users\\{CENSORED}\\IdeaProjects\\geekhub-11-{CENSORED}\\src\\logging\\LogFile.log");
logger.addHandler(fh);
```

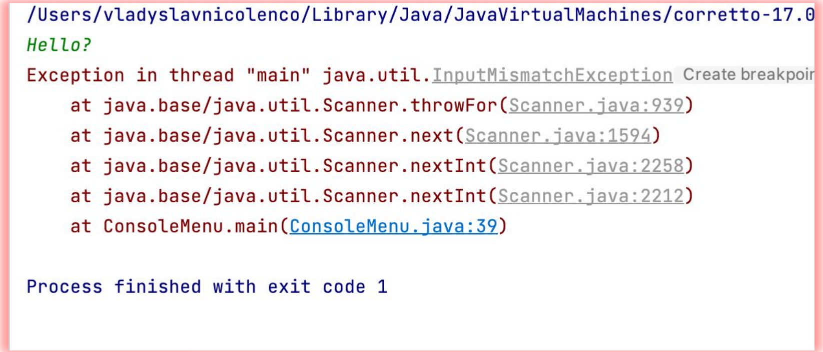


```
# default file output is in user's home directory.
java.util.logging.FileHandler.pattern = C:\\Users\\{CENSORED}\\Git\\geekhub-11-{CENSORED}\\loggingRecord\\java%u.log
```

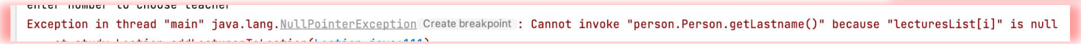
Task 2. For some reason….



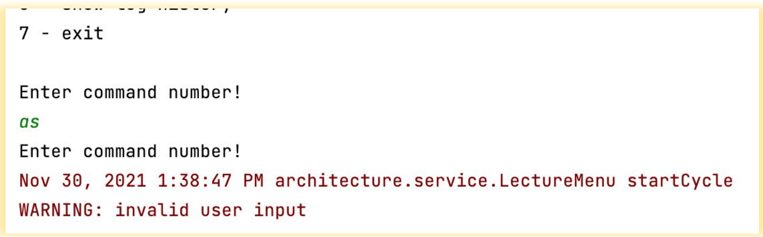
```
/Users/vladyslavnicolenco/Library/Java/JavaVirtu
```



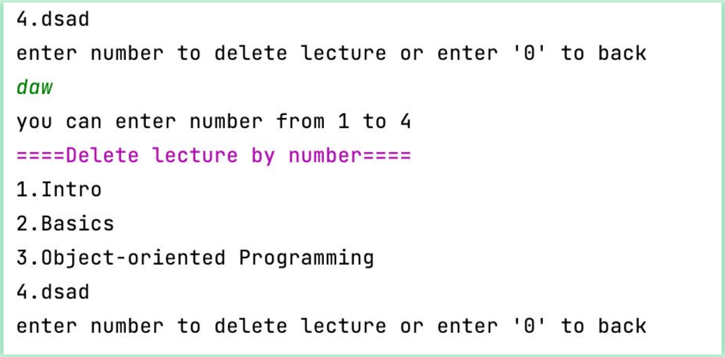
```
/Users/vladyslavnicolenco/Library/Java/JavaVirtualMachines/corretto-17.0
Hello?
Exception in thread "main" java.util.InputMismatchException Create breakpoir
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at ConsoleMenu.main(ConsoleMenu.java:39)

Process finished with exit code 1
```



```
enter number to choose teacher
Exception in thread "main" java.lang.NullPointerException Create breakpoint : Cannot invoke "person.Person.getLastname()" because "lecturesList[i]" is null
    at study.Lection.addLectureToLection(Lection.java:111)
```

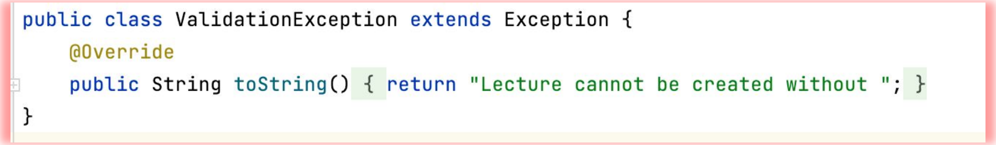


```
6 - show log history
7 - exit

Enter command number!
as
Enter command number!
Nov 30, 2021 1:38:47 PM architecture.service.LectureMenu startCycle
WARNING: invalid user input
```

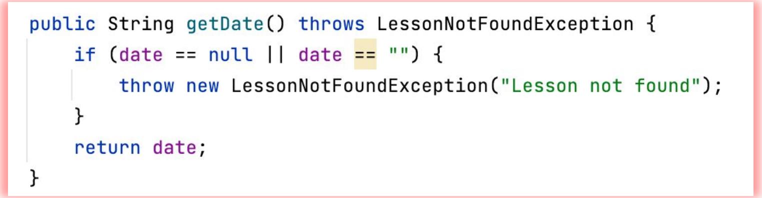


```
4.dsad
enter number to delete lecture or enter '0' to back
daw
you can enter number from 1 to 4
====Delete lecture by number====
1.Intro
2.Basics
3.Object-oriented Programming
4.dsad
enter number to delete lecture or enter '0' to back
```

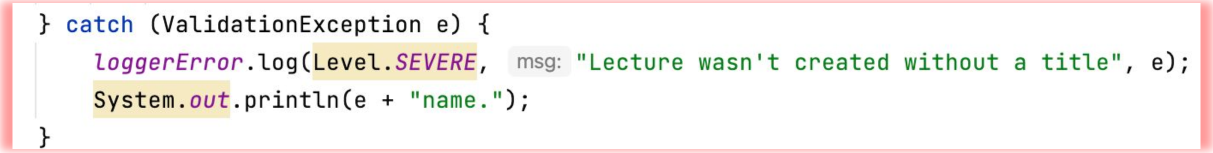
Use exceptions correctly



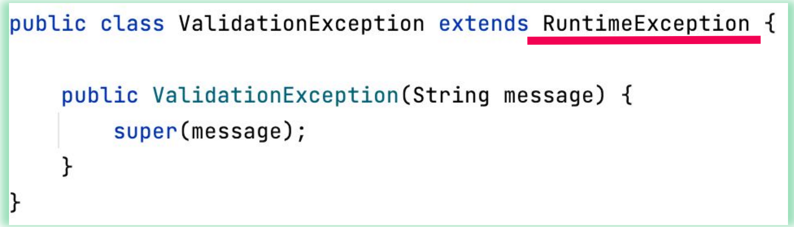
```
public class ValidationException extends Exception {
    @Override
    public String toString() { return "Lecture cannot be created without "; }
}
```



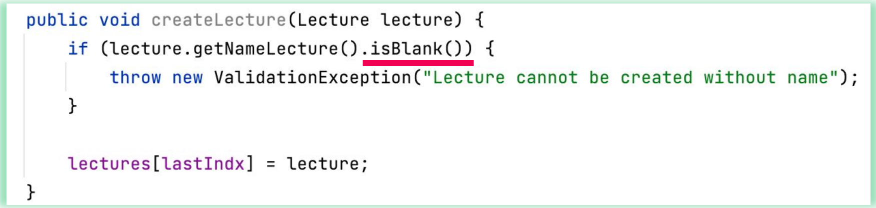
```
public String getDate() throws LessonNotFoundException {
    if (date == null || date == "") {
        throw new LessonNotFoundException("Lesson not found");
    }
    return date;
}
```



```
} catch (ValidationException e) {
    loggerError.log(Level.SEVERE, msg: "Lecture wasn't created without a title", e);
    System.out.println(e + "name.");
}
```

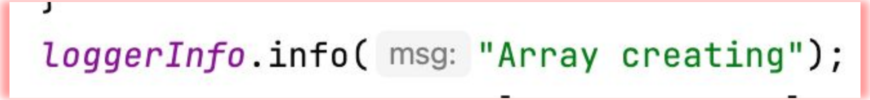


```
public class ValidationException extends RuntimeException {

    public ValidationException(String message) {
        super(message);
    }
}
```

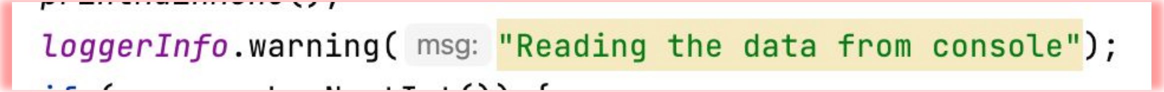


```
public void createLecture(Lecture lecture) {
    if (lecture.getNameLecture().isBlank()) {
        throw new ValidationException("Lecture cannot be created without name");
    }

    lectures[lastIndx] = lecture;
}
```

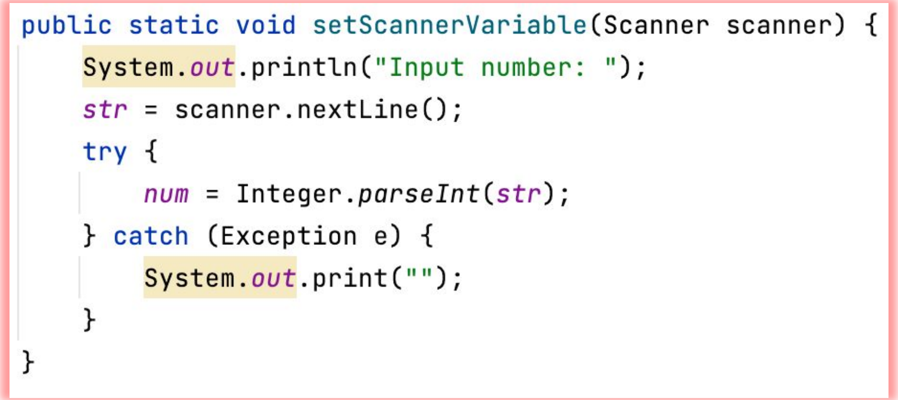
Make logs useful



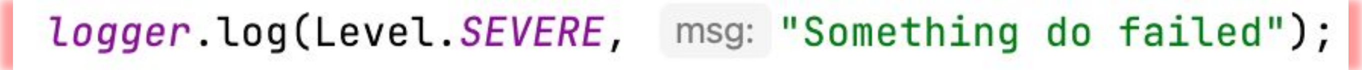
```
loggerInfo.info( msg: "Array creating");
```



```
loggerInfo.warning( msg: "Reading the data from console");
```

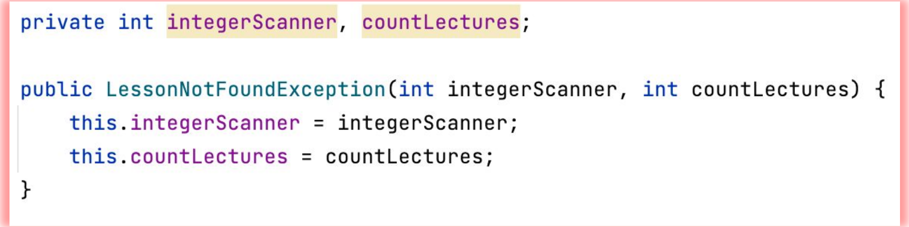


```
public static void setScannerVariable(Scanner scanner) {
    System.out.println("Input number: ");
    str = scanner.nextLine();
    try {
        num = Integer.parseInt(str);
    } catch (Exception e) {
        System.out.print("");
    }
}
```

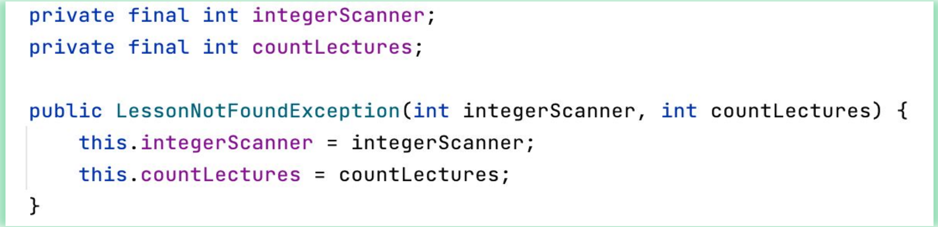


```
logger.log(Level.SEVERE, msg: "Something do failed");
```

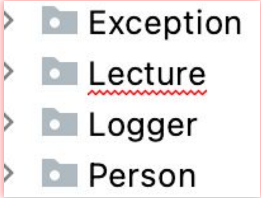
Follow conventions



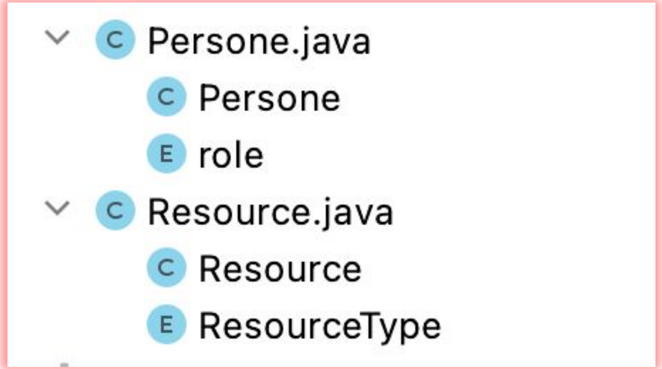
```java
private int integerScanner, countLectures;

public LessonNotFoundException(int integerScanner, int countLectures) {
    this.integerScanner = integerScanner;
    this.countLectures = countLectures;
}
```



```java
private final int integerScanner;
private final int countLectures;

public LessonNotFoundException(int integerScanner, int countLectures) {
    this.integerScanner = integerScanner;
    this.countLectures = countLectures;
}
```

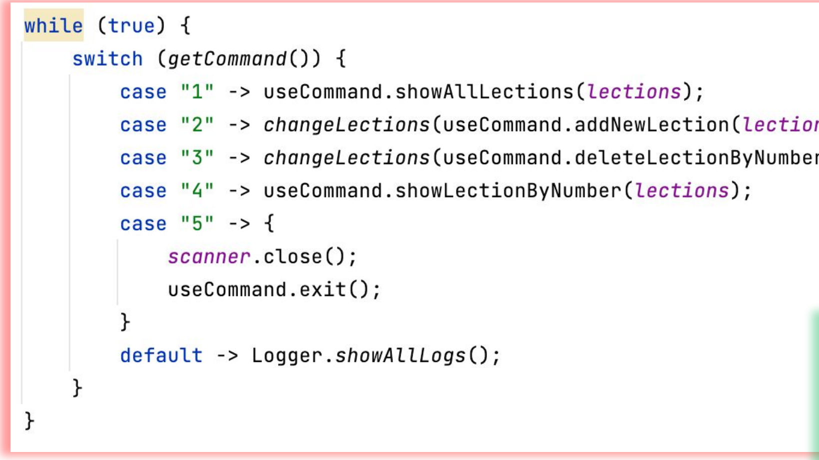


- Exception
- Lecture
- Logger
- Person

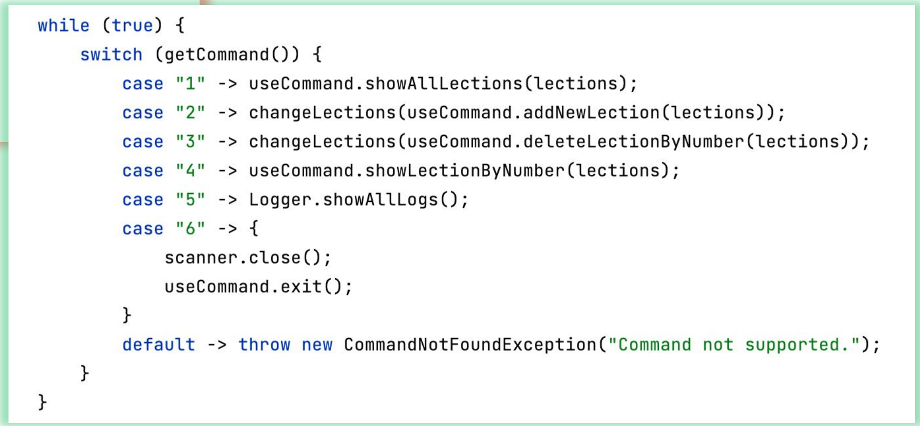


- Persone.java
  - Persone
  - role
- Resource.java
  - Resource
  - ResourceType

Handle edge cases

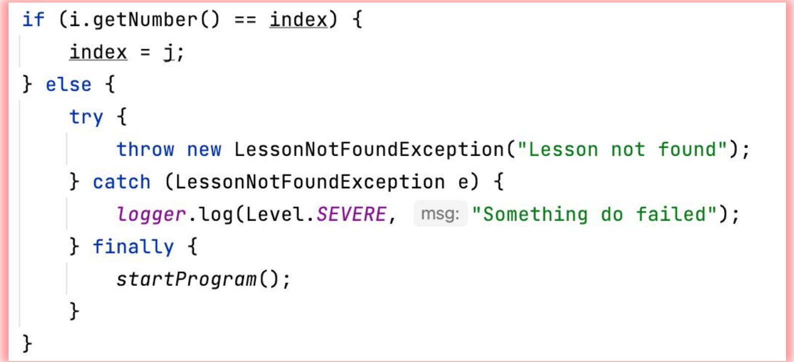


```java
while (true) {
    switch (getCommand()) {
        case "1" -> useCommand.showAllLections(lections);
        case "2" -> changeLections(useCommand.addNewLection(lections));
        case "3" -> changeLections(useCommand.deleteLectionByNumber(lections));
        case "4" -> useCommand.showLectionByNumber(lections);
        case "5" -> {
            scanner.close();
            useCommand.exit();
        }
        default -> Logger.showAllLogs();
    }
}
```

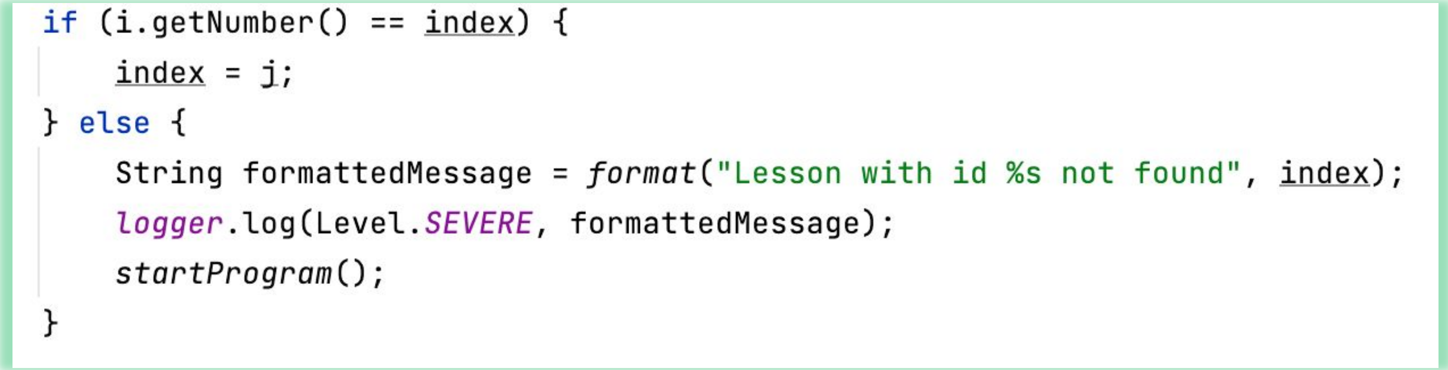


```java
while (true) {
    switch (getCommand()) {
        case "1" -> useCommand.showAllLections(lections);
        case "2" -> changeLections(useCommand.addNewLection(lections));
        case "3" -> changeLections(useCommand.deleteLectionByNumber(lections));
        case "4" -> useCommand.showLectionByNumber(lections);
        case "5" -> Logger.showAllLogs();
        case "6" -> {
            scanner.close();
            useCommand.exit();
        }
        default -> throw new CommandNotFoundException("Command not supported.");
    }
}
```

Useless exception usage

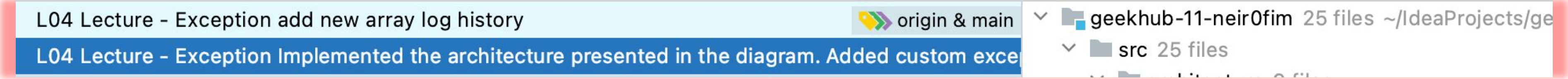


```java
if (i.getNumber() == index) {
    index = j;
} else {
    try {
        throw new LessonNotFoundException("Lesson not found");
    } catch (LessonNotFoundException e) {
        logger.log(Level.SEVERE,  msg: "Something do failed");
    } finally {
        startProgram();
    }
}
```

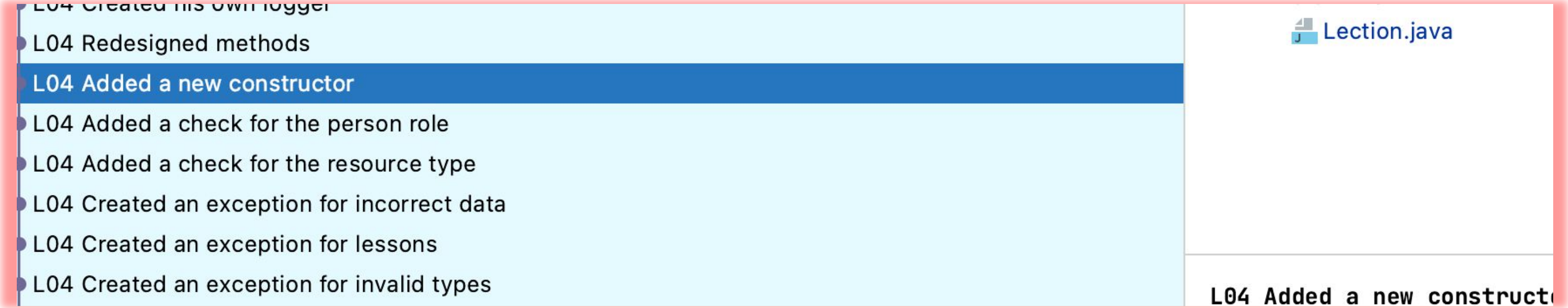


```java
if (i.getNumber() == index) {
    index = j;
} else {
    String formattedMessage = format("Lesson with id %s not found", index);
    logger.log(Level.SEVERE, formattedMessage);
    startProgram();
}
```

Make commits more atomic



```
L04 Lecture - Exception add new array log history                          origin & main
L04 Lecture - Exception Implemented the architecture presented in the diagram. Added custom exce
```



```
L04 Created his own logger
L04 Redesigned methods
L04 Added a new constructor
L04 Added a check for the person role
L04 Added a check for the resource type
L04 Created an exception for incorrect data
L04 Created an exception for lessons
L04 Created an exception for invalid types
```

```
L04 Added a new constructo
```

FIX THIS!