LESSON 10



Java

Exceptions & Files

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

JAVA PROGRAMMING

LESSON 10

Exceptions and File I/O

try/catch/finally, checked vs unchecked, custom exceptions, try-with-resources and java.ni

try, catch, finally

Parse.java

```
try {
    int n = Integer.parseInt(input);
    System.out.println(100 / n);
} catch (NumberFormatException e) {
    System.out.println("Not a number: " + input);
} catch (ArithmeticException e) {
    System.out.println("Cannot divide by zero");
} finally {
    System.out.println("done");
}
```

- When something goes wrong, Java throws an exception object. try wraps risky code, catch …
- Catch the most specific exception you can, and never silently swallow one with an empty catch.

01

Checked vs Unchecked



- Checked exceptions, like IOException, must be caught or declared with throws. The compiler ...
- Unchecked exceptions extend RuntimeException, like NullPointerException, and usually signal ...

02

Throwing and Custom Exceptions

Bank.java

```
class InsufficientFundsException extends Exception {
    InsufficientFundsException(String msg) { super(msg); }
}

void withdraw(double amt) throws InsufficientFundsException {
    if (amt > balance) {
        throw new InsufficientFundsException("balance is " + balance);
    }
    balance -= amt;
}
```

- throw new creates and throws an exception. Make your own by extending Exception or …
- Pass the original exception as the cause, so the full story appears in the stack trace.

03

try-with-resources

ReadLines.java

```java
try (BufferedReader r = Files.newBufferedReader(Path.of("scores.txt"))) {
    String line;
    while ((line = r.readLine()) != null) {
        System.out.println(line);
    }
} catch (IOException e) {
    System.err.println("Cannot read: " + e.getMessage());
}
```

- Files, network connections and database connections must be closed. Put them in parentheses …
- Any class that implements AutoCloseable works this way.

04

The Files Class

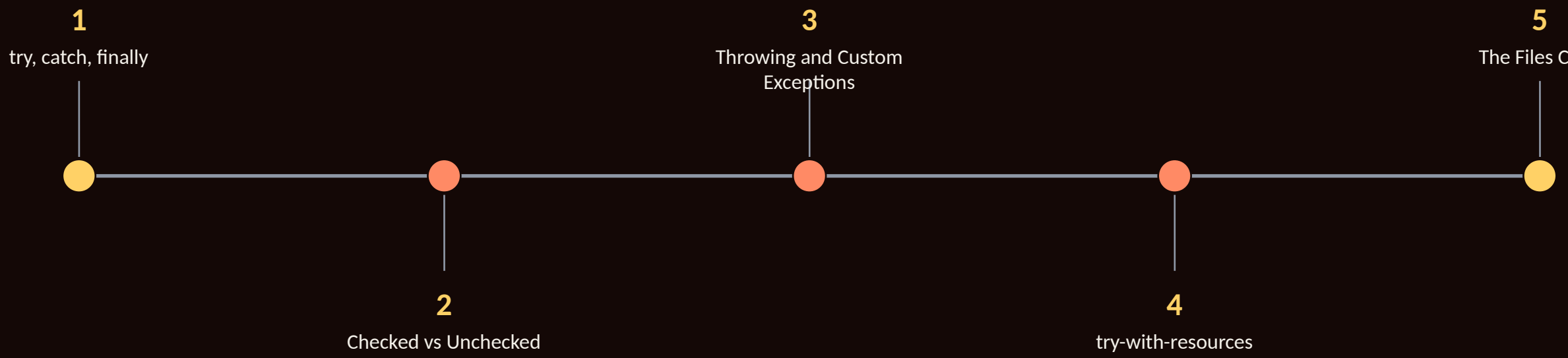
FilesDemo.java

```
Path p = Path.of("data", "report.txt");
Files.createDirectories(p.getParent());
Files.writeString(p, "Kabul,95\nHerat,88\n");
List<String> lines = Files.readAllLines(p);
try (Stream<String> s = Files.lines(p)) {
    long n = s.filter(l -> l.contains("Kabul")).count();
}
```

- java dot nio dot file makes simple file work one line: readAllLines, readString, …
- For big files, Files dot lines gives a stream so you never load everything into memory.

05

Lesson Map


1
try, catch, finally
2
Checked vs Unchecked
3
Throwing and Custom Exceptions
4
try-with-resources
5
The Files Class

Sources and Further Reading

Parse.java

```java
try {
    int n = Integer.parseInt(input);
    System.out.println(100 / n);
} catch (NumberFormatException e) {
    System.out.println("Not a number: " + input);
} catch (ArithmeticException e) {
    System.out.println("Cannot divide by zero");
} finally {
    System.out.println("done");
}
```

JAVA PROGRAMMING · 10

Bank.java

```java
class InsufficientFundsException extends Exception {
    InsufficientFundsException(String msg) { super(msg); }
}

void withdraw(double amt) throws InsufficientFundsException {
    if (amt > balance) {
        throw new InsufficientFundsException("balance is " + balance);
    }
    balance -= amt;
}
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari