

TDD in Java SE 17

What Is TDD and Why Use It?



Andrejs Doronins

Software Developer in Test

TDD in Java SE 17

Version Check



Version Check



This course was created by using:

- Java 17



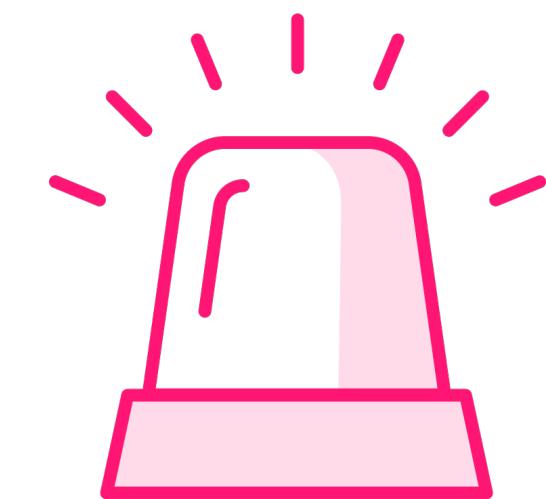
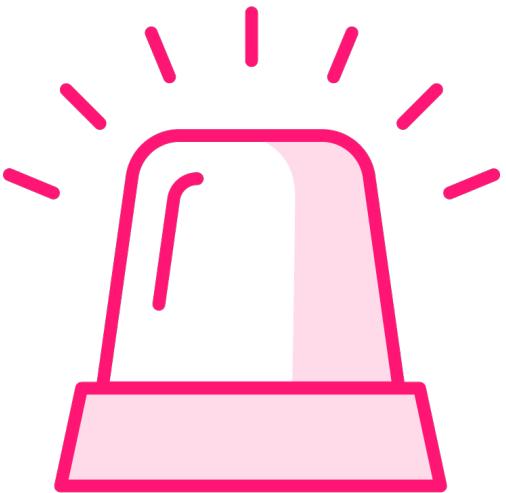
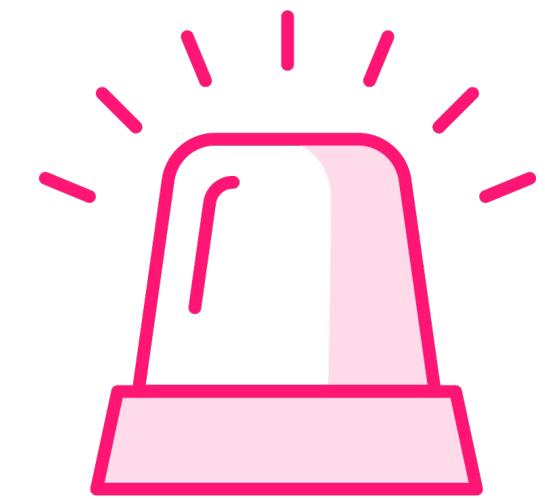
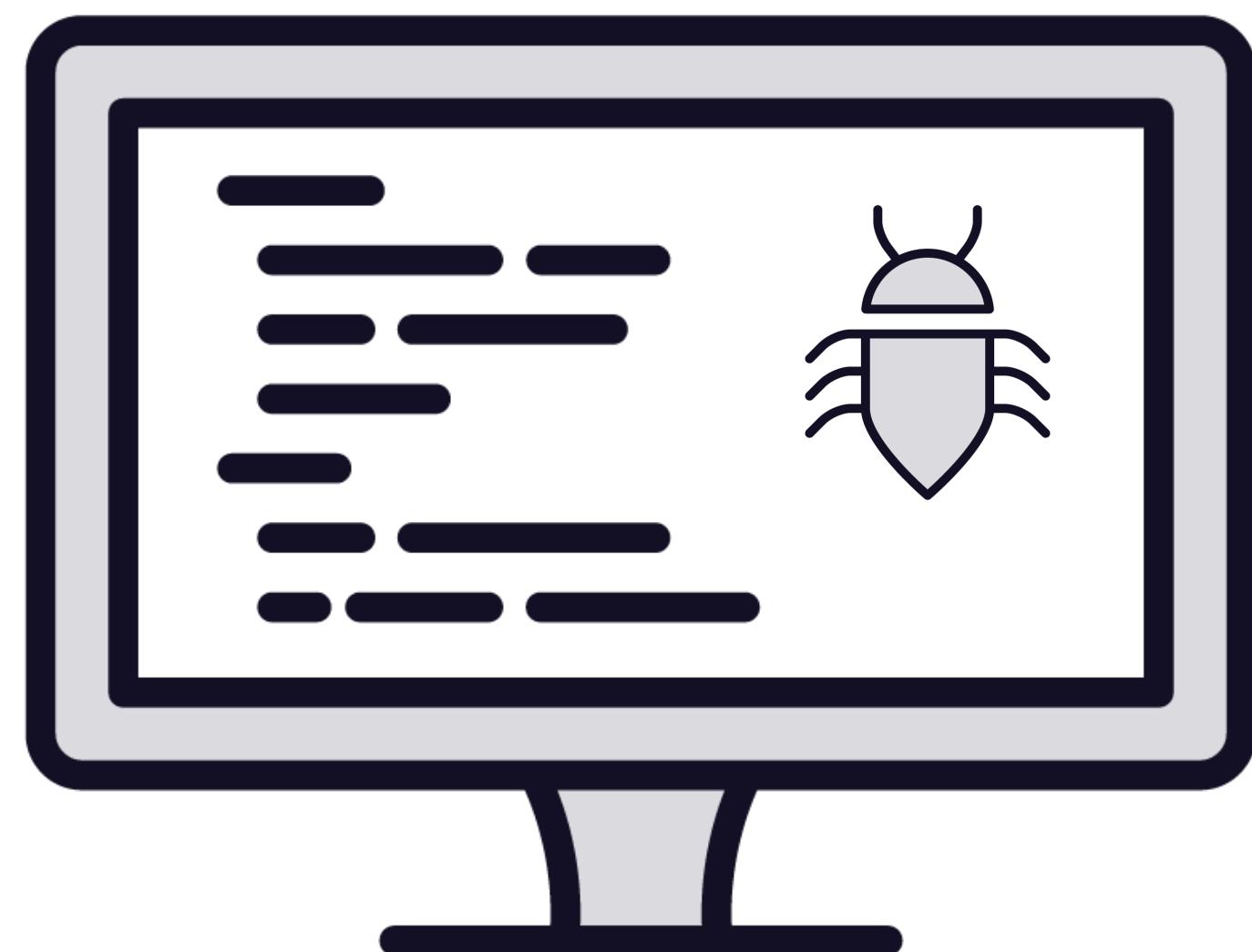
Version Check



This course is 100% applicable to:

- Java 17





Compiler: incompatible types

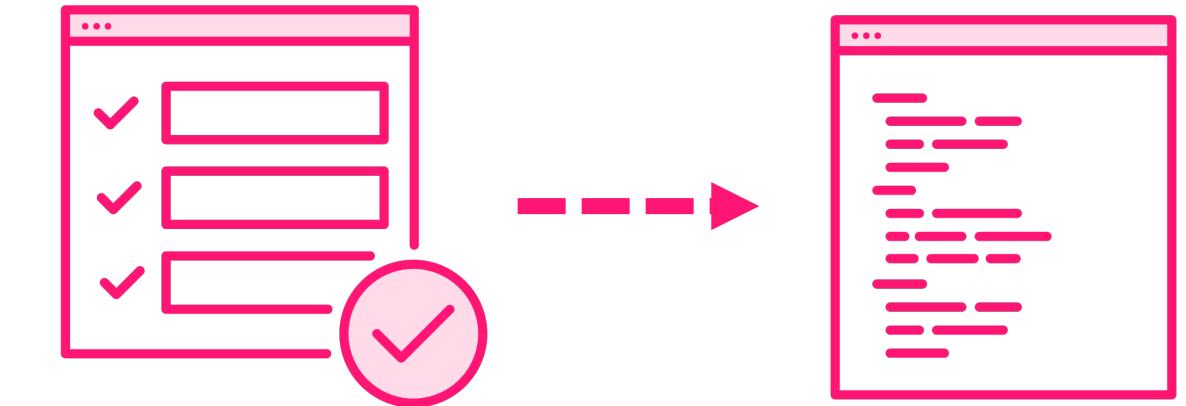
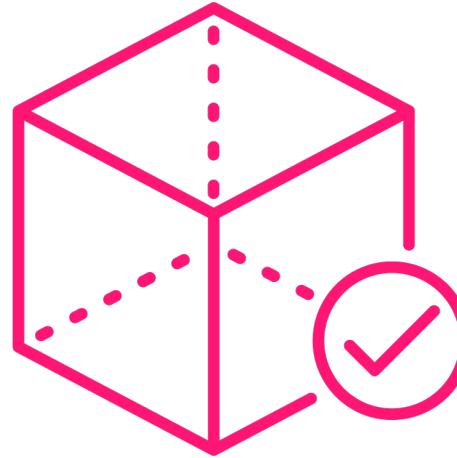
```
String s = 1;
```

```
for(int i=1; i < 10; i--) { }
```

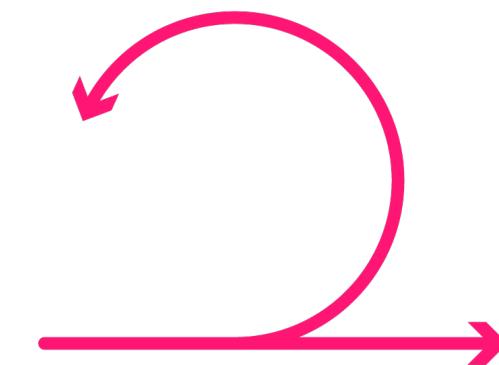
- Always true (oops!)



TDD



TDD helps produce well-designed, well-tested,
and well-factored code in small, verifiable steps



Book: The Art of Agile Development





So TDD is unit testing, right?

+ drive the design

+ tiny steps & run tests (very) frequently

+ robust test suite





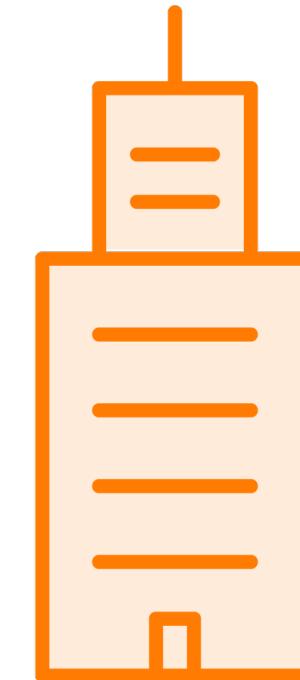
Fear is minimized

Less debugging



Living docs for others

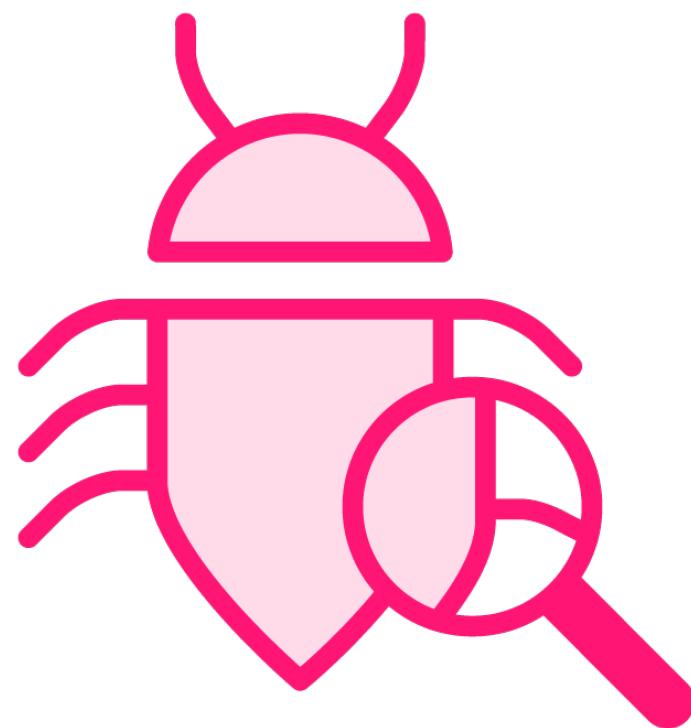
Faster Code Reviews



Faster delivery of value



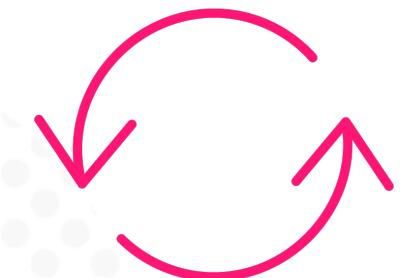
TDD vs. DDD

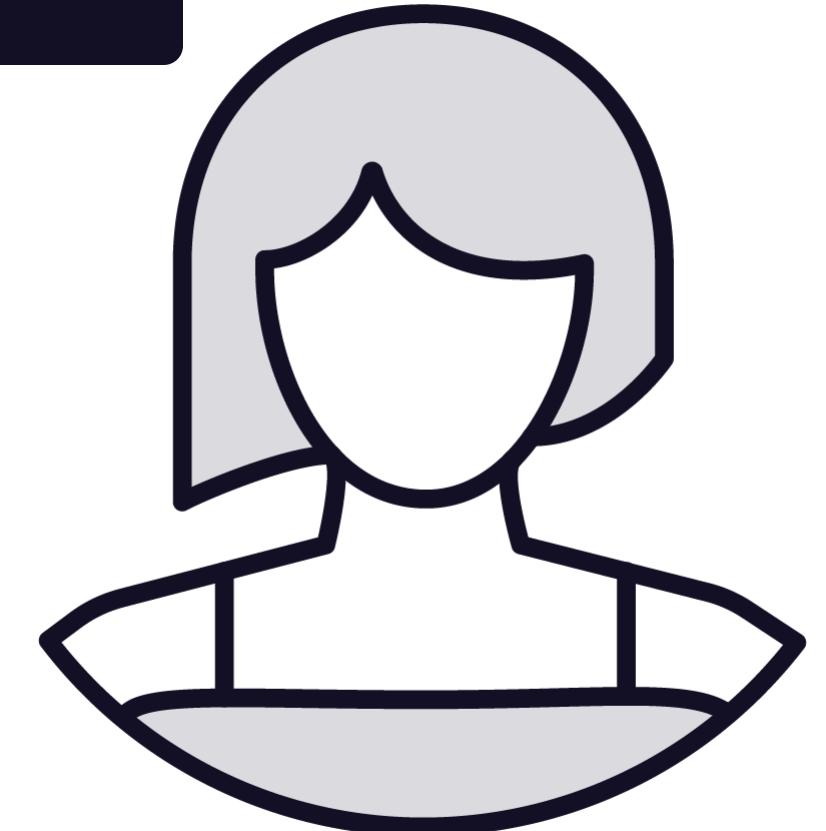


Debugging is a useful skill

Not breaking things is even better

DDD: Debugger-Driven-Development





**TDD has a learning curve.
It slows you down!**

**True.
So does learning Git or IDE.**





TDD is time-consuming - you must write much more code



**And the time spent debugging
and hot-fixing regressions?**



TDD



Tracking things that happened

vs.

**Tracking things that never happened
(prevented)**





**Tests are a big cost!
(And they break)**



Criticism of tests, not TDD

**Minimize overhead with
best practices**



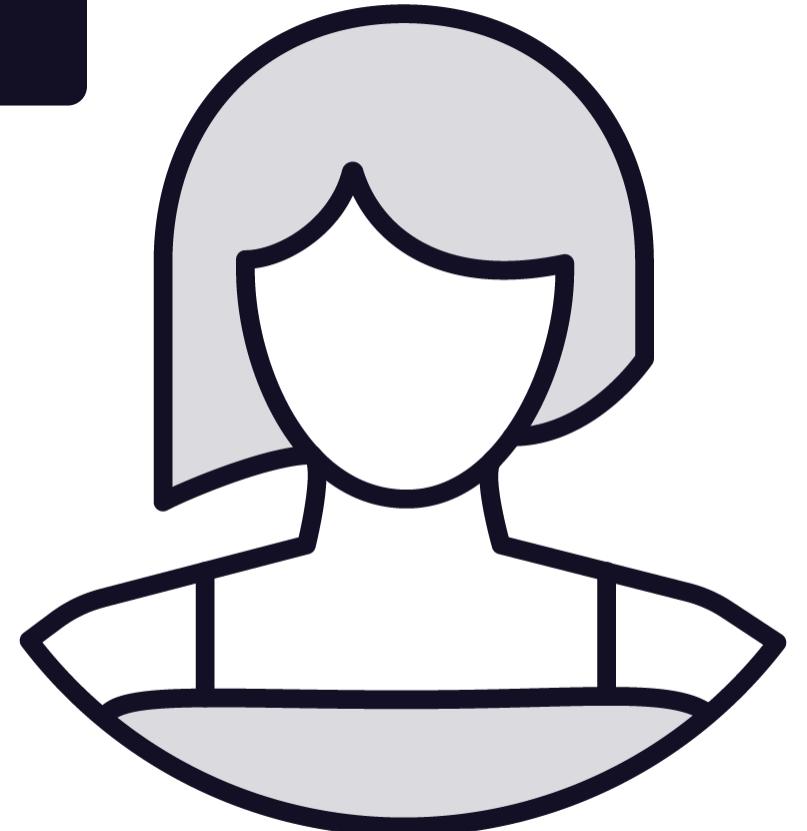
**Production code
best practices**

**Test code
best practices**





**False sense of security,
difficult to apply in some places**



Those are valid concerns!



Prerequisites



You can create a blank Java project

Add dependencies

Stack: Windows, IntelliJ, JUnit 5



Overview

RGR: Red-Green-Refactor

Start simple: implement a single method

Next level: use TDD for a small app

Nuances of TDD

