

Java Best Practices

Why Care About Best Practices?



Andrejs Doronins

Software Developer in Test

Java Best Practices

Version Check



Version Check



This course was created by using:

- Java 17



Version Check

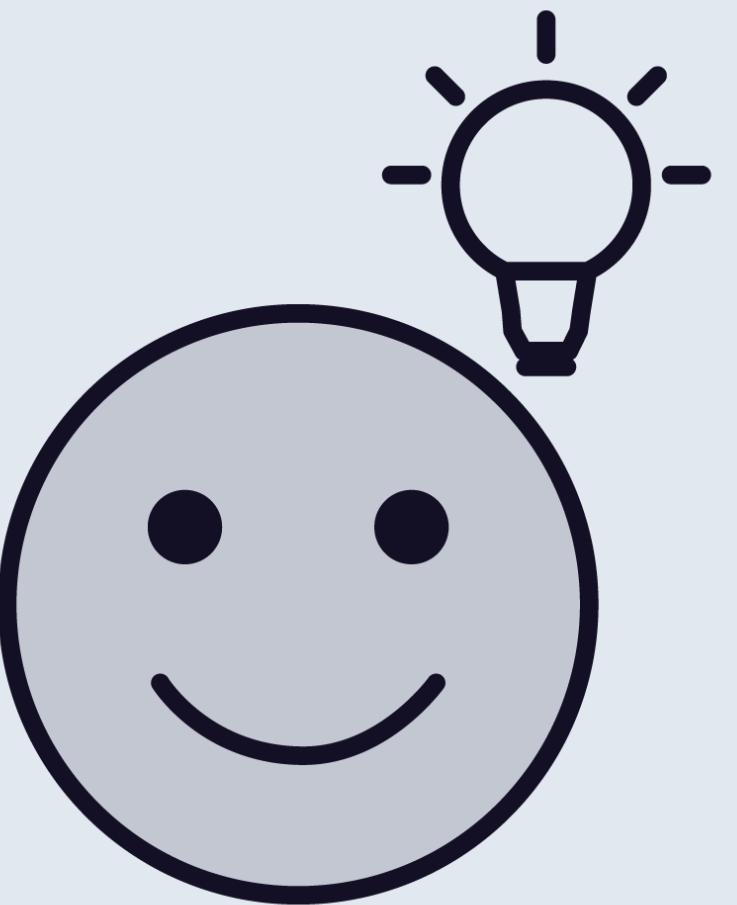


This course is 100% applicable to:

- Java 17-21



```
if(isAnyInputInvalid) {  
    // throw  
}  
  
return someList.stream()  
    .filter(...)  
    .filter(...)  
    .toList();
```



```
if(isInputOneValid) {  
    if(isInputTwoValid) {  
        for (String e : someList) {  
            if (e.equals(thing)) {  
                // do something  
            }  
        }  
    }  
}  
} else {  
    // throw  
}
```



Why can't I understand this code?

Do I need more experience?



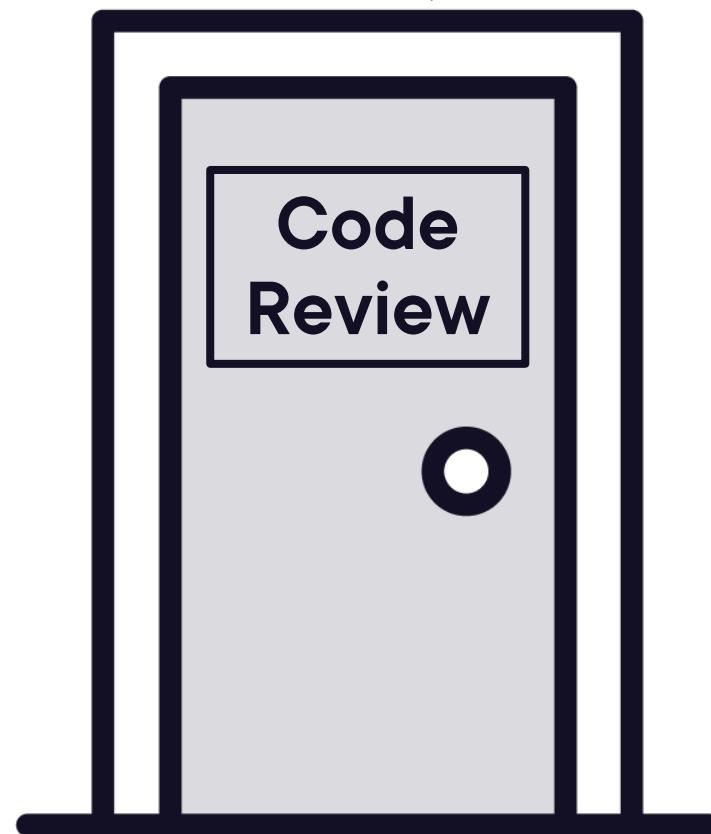
Poor unreadable code is widespread



Do I write clean code?



What the ... ?!

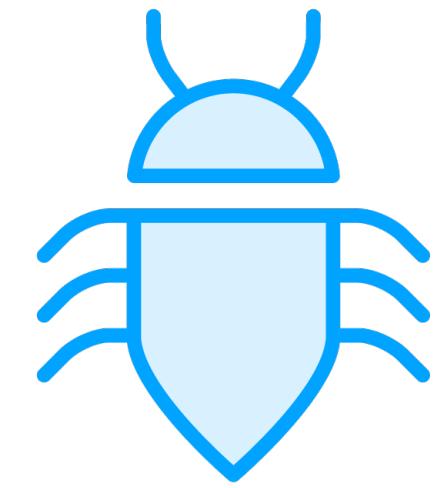


**Clean code is not
a nice-to-have.**

It's a necessity.



Poor Code Consequences



More bugs



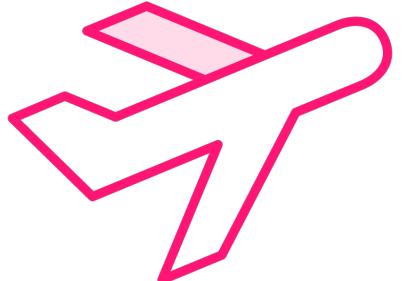
**Decreased
productivity**



Lower job satisfaction



Bug Consequences



Decades-old bug wiped out UK air traffic control

Source: <https://www.itpro.co.uk/business-intelligence/24682/decades-old-bug-wiped-out-uk-air-traffic-control>



Knight Shows How to Lose \$440 Million in 30 Minutes

Source: <https://www.bloomberg.com/news/articles/2012-08-02/knight-shows-how-to-lose-440-million-in-30-minutes>



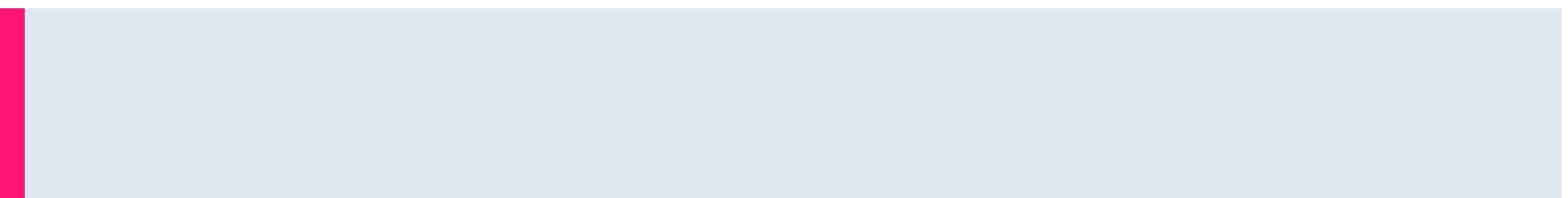
Time Spent on...

Reading code



50%

Writing code



50%



Time Spent on...

Reading code



90% 10%

Treat technical debt like credit card debt

Robert C. Martin

“The ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code.”

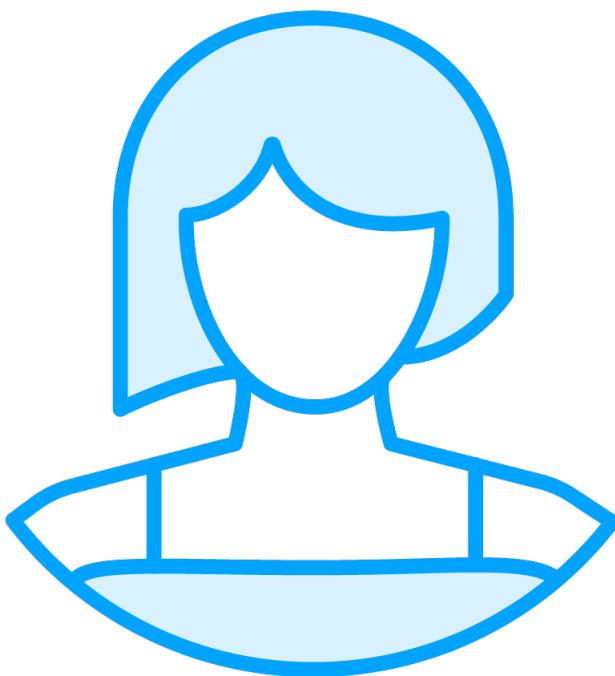


Robert C. Martin

“...[Therefore,] making it easy to
read makes it easier to write.”



Who This Course Is For



Developers



Test Automation Engineers

Both must check their own work!



Prerequisites



Java:

- OOP
- 1+ years of professional experience

Expectations:

- JDK installed
- Using an IDE
- Basics of a build tool



Demo



Project setup



[Table of contents](#)

[Description](#)

[Transcript](#)

[Exercise files](#)



Overview



- What's in a name?
- Creating objects the right way
- Most code (and mistakes) are in methods
- A closer look at strings and numbers
- Branching and iterating
- Handling exceptions gracefully
- Comments aren't always good
- Respect test code

