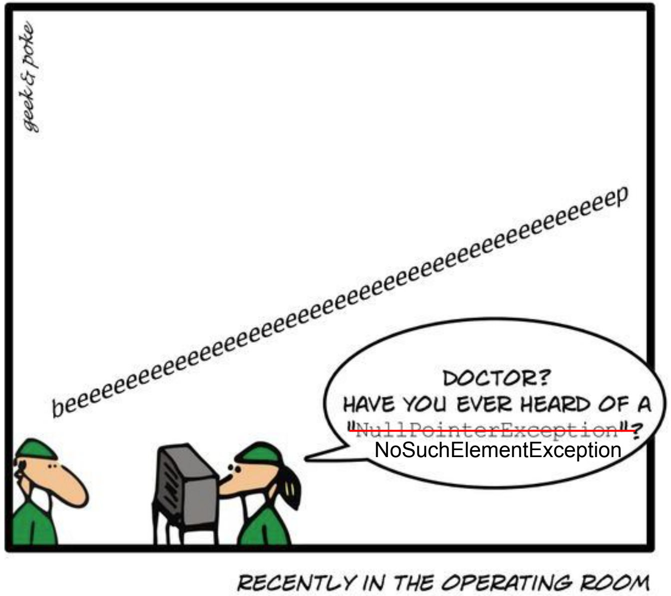
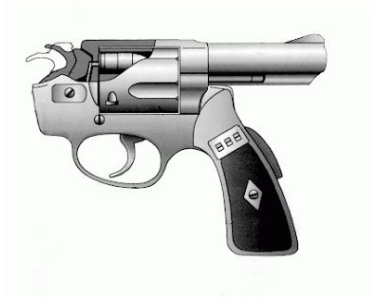
Java 8 Optional Checker Framework



Justin Kotalik (jkotalik)
Jasper Hugunin (jasperh)
Han Zhang (gzezhang)

Motivation

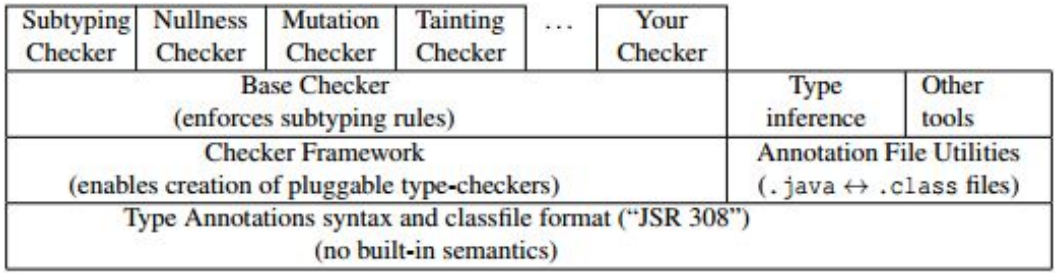
- People are inclined to use the Optional Class to solve NullPointerExceptions
- In general a bad idea:
  - If you modify to .get() on optional objects...
  - Transfers the error to a NoSuchElementException :(
  - Also adds an extra layer of uncertainty with introducing a new class
- Both are masking lacking null checks



Problem to solve: finding improper uses of .get() with Optional Objects.

Approach

- Use the Checker Framework:
- Adapt Nullness Checker
- Null <-> Not present; dereference <-> Optional.get()



dereference of possibly-null reference y y.charAt(2);	Main.java	/TestProject/src	line 6	Checker Framework Problem

Main Challenge

- We need to understand the Nullness Checker
- But, the Nullness Checker is complicated
- Luckily, one of the main developers, Michael Ernst, is our teacher
- We can ask him for help if we get stuck understanding something
- The Checker Framework has lots of simpler examples as well