

# Object-oriented Programming in Java

Approaching Object-oriented Programming



**Paolo Perrotta**

Developer, Author

@nusco | [www.paoloperrotta.com](http://www.paoloperrotta.com)



## Version Check



**This course was created with:**

- Java SE 21 (LTS)



## Version Check



**This course is mostly applicable to:**

- Java 8 (2014) and later



**Object-oriented programming is  
a programming paradigm based  
on identifying abstract data  
types that connect data and  
operations on those data**

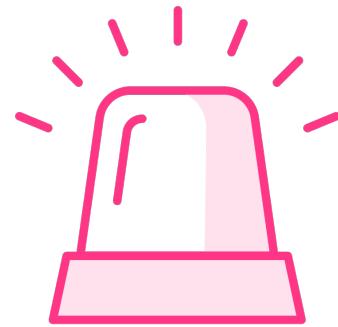




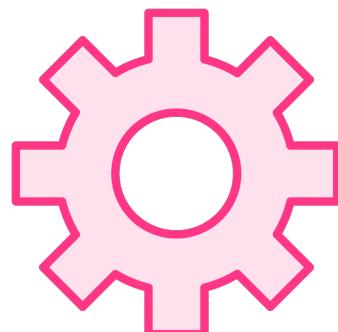
# OOP in 6 Minutes



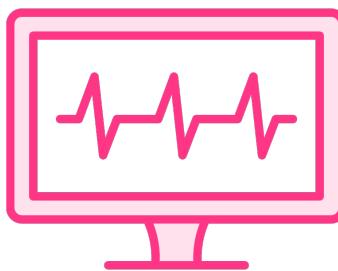
# An Application to Manage Alarms



**You can create alarms in an industrial environment**



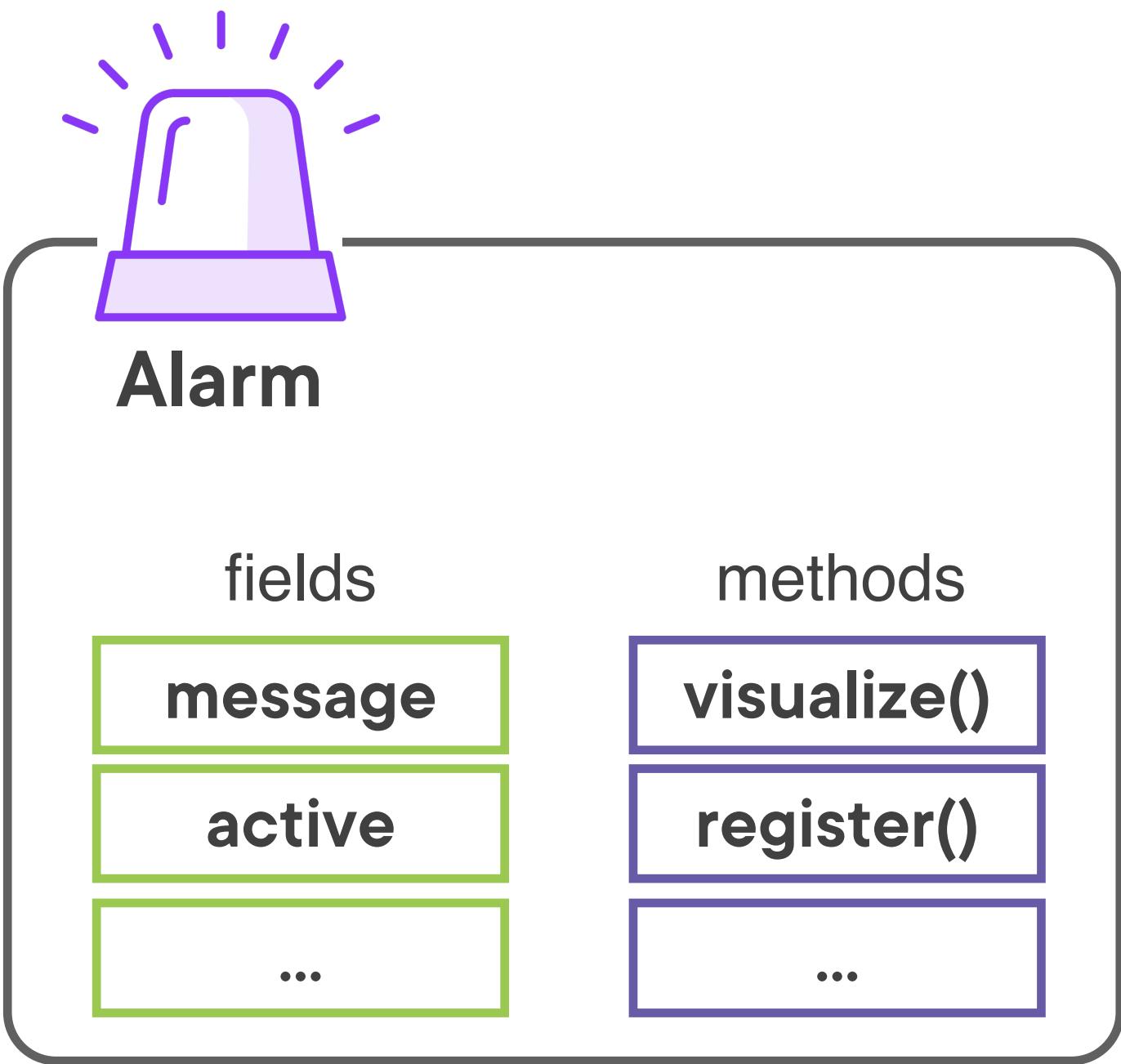
**You can configure alarms for different events**



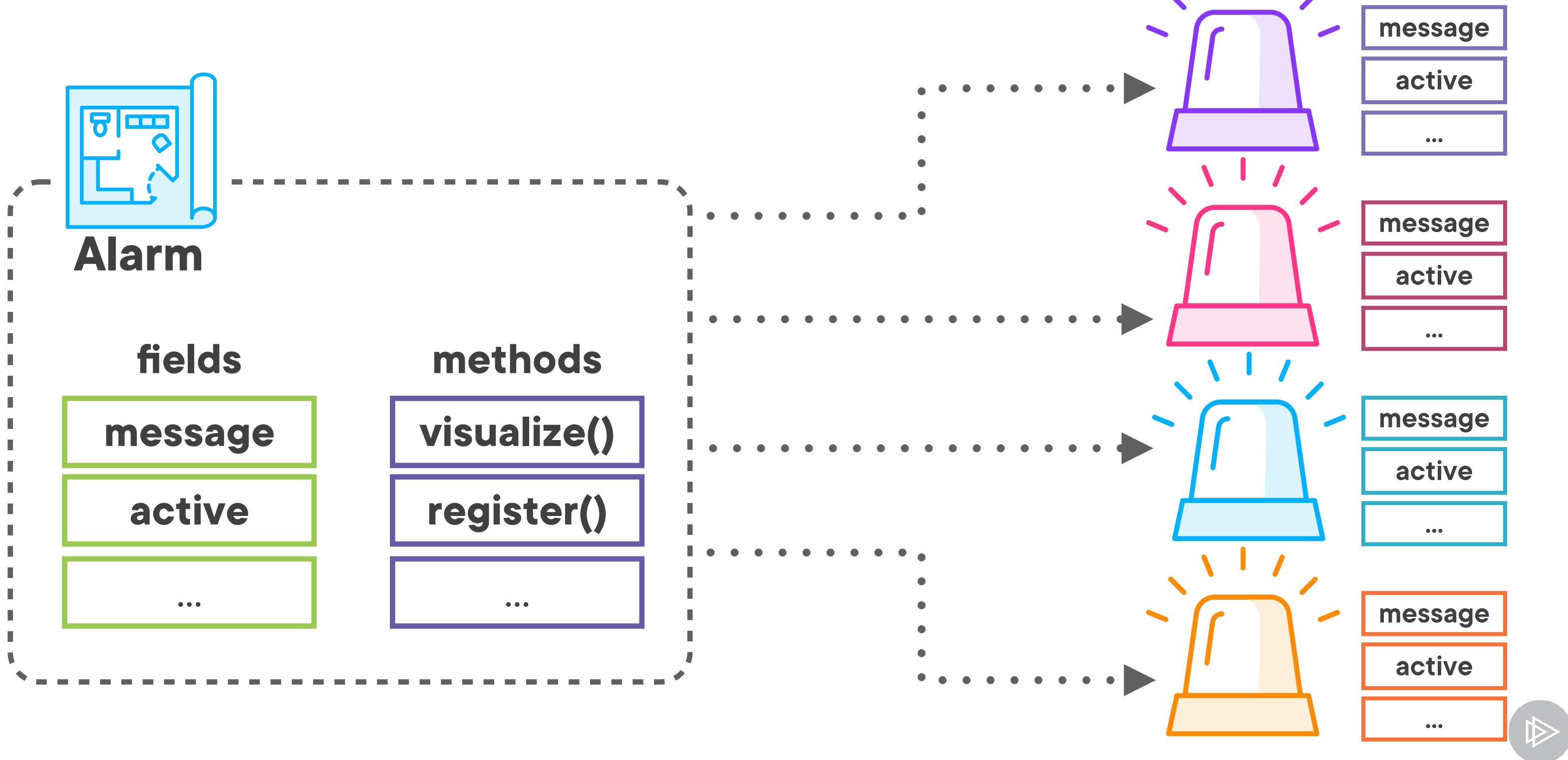
**You can put alarms on a dashboard**



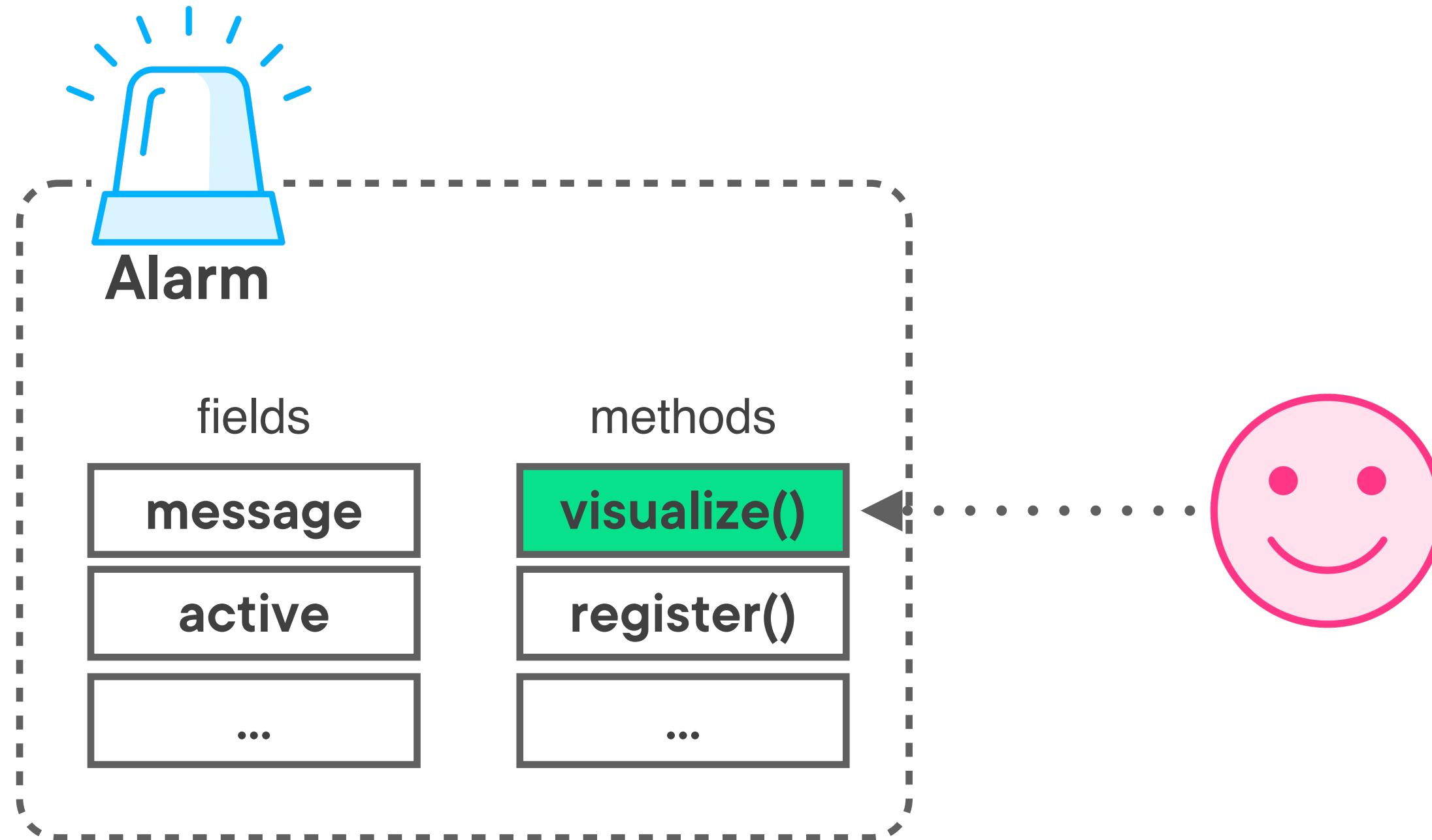
# Objects



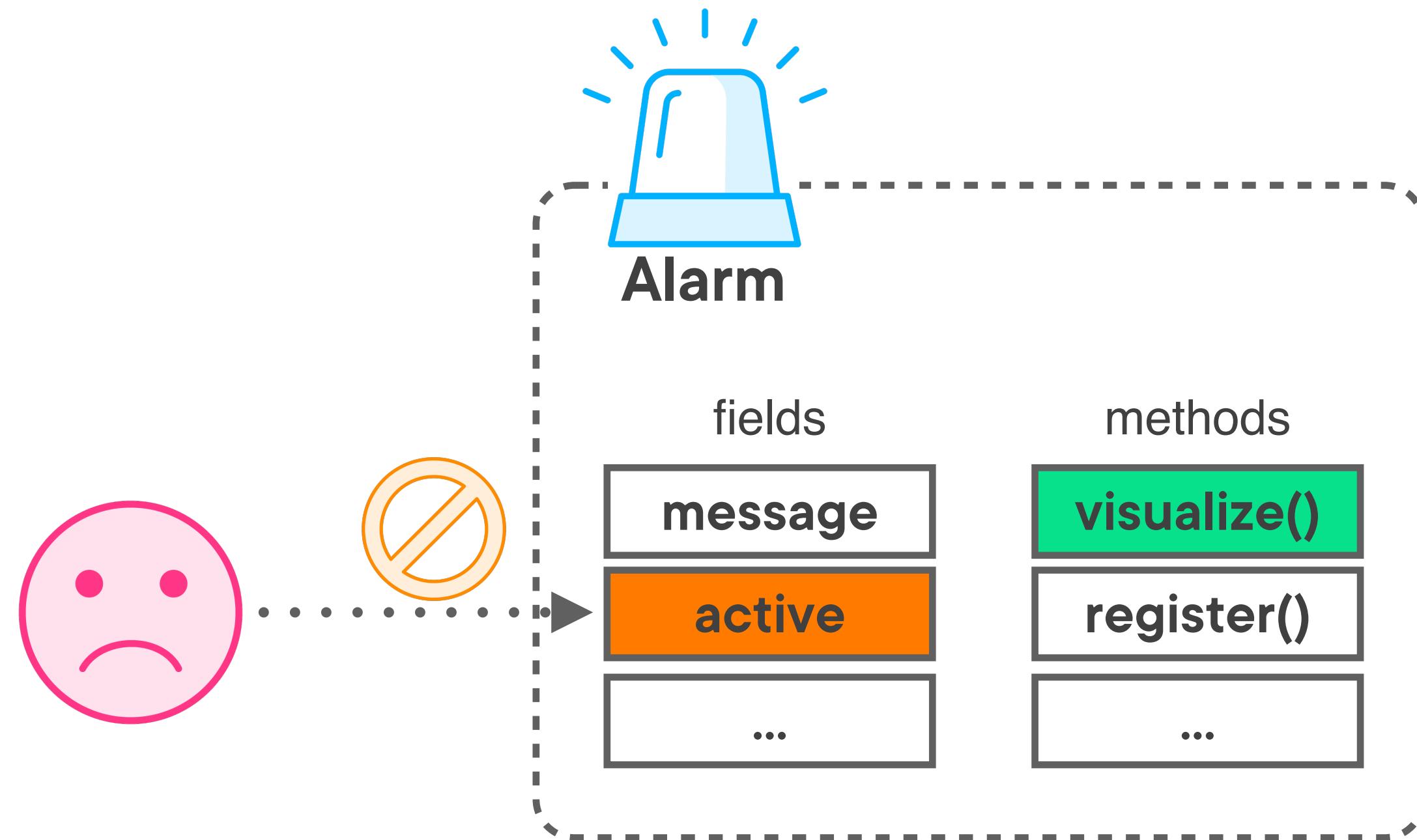
# Classes



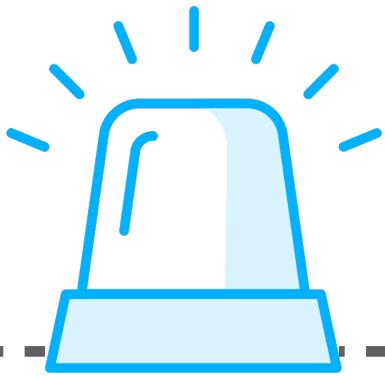
# Interface vs. Implementation



# Interface vs. Implementation



# Similar Classes



Alarm

fields

message

active

methods

visualize()

register()



SnoozableAlarm

fields

message

active

methods

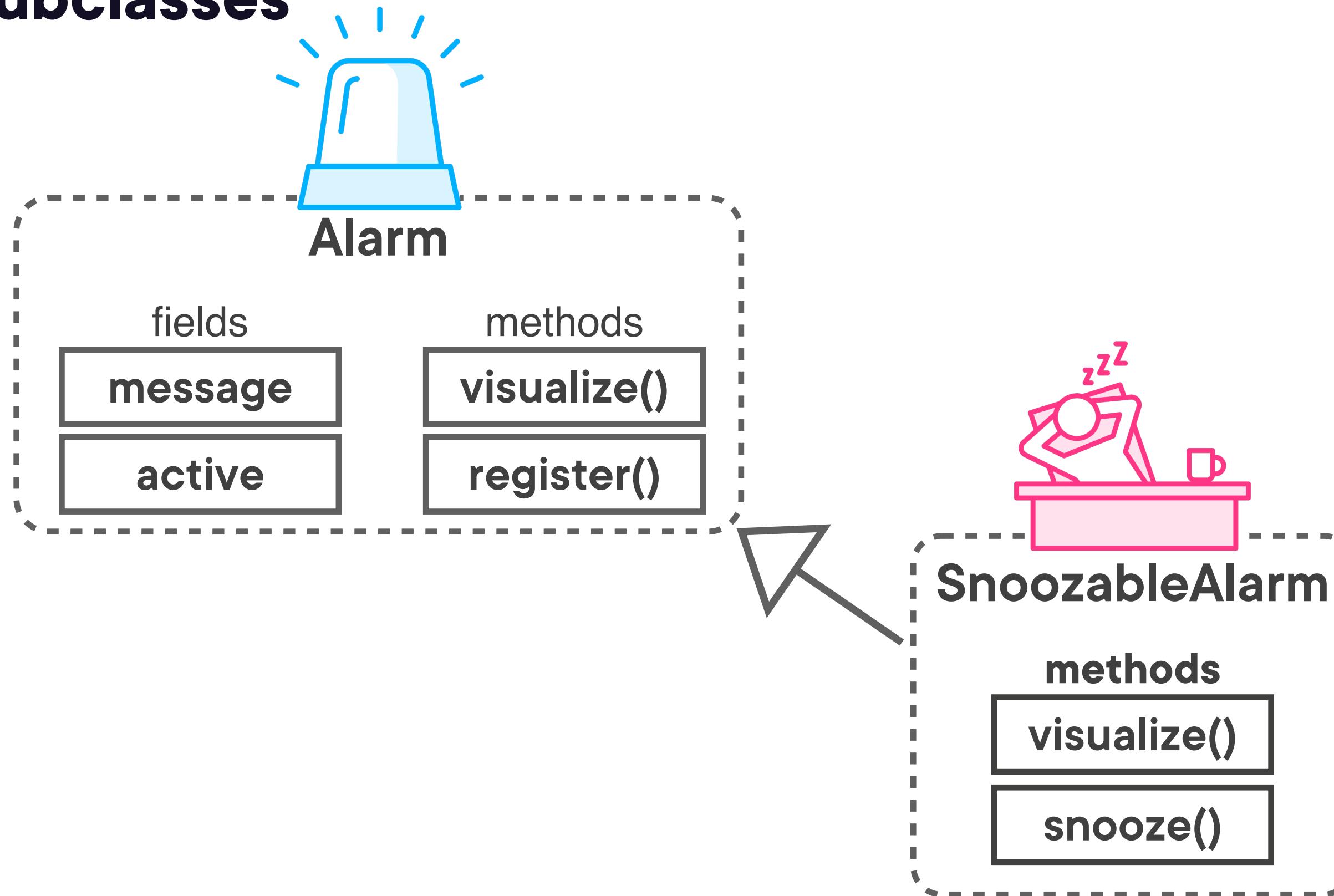
visualize()

register()

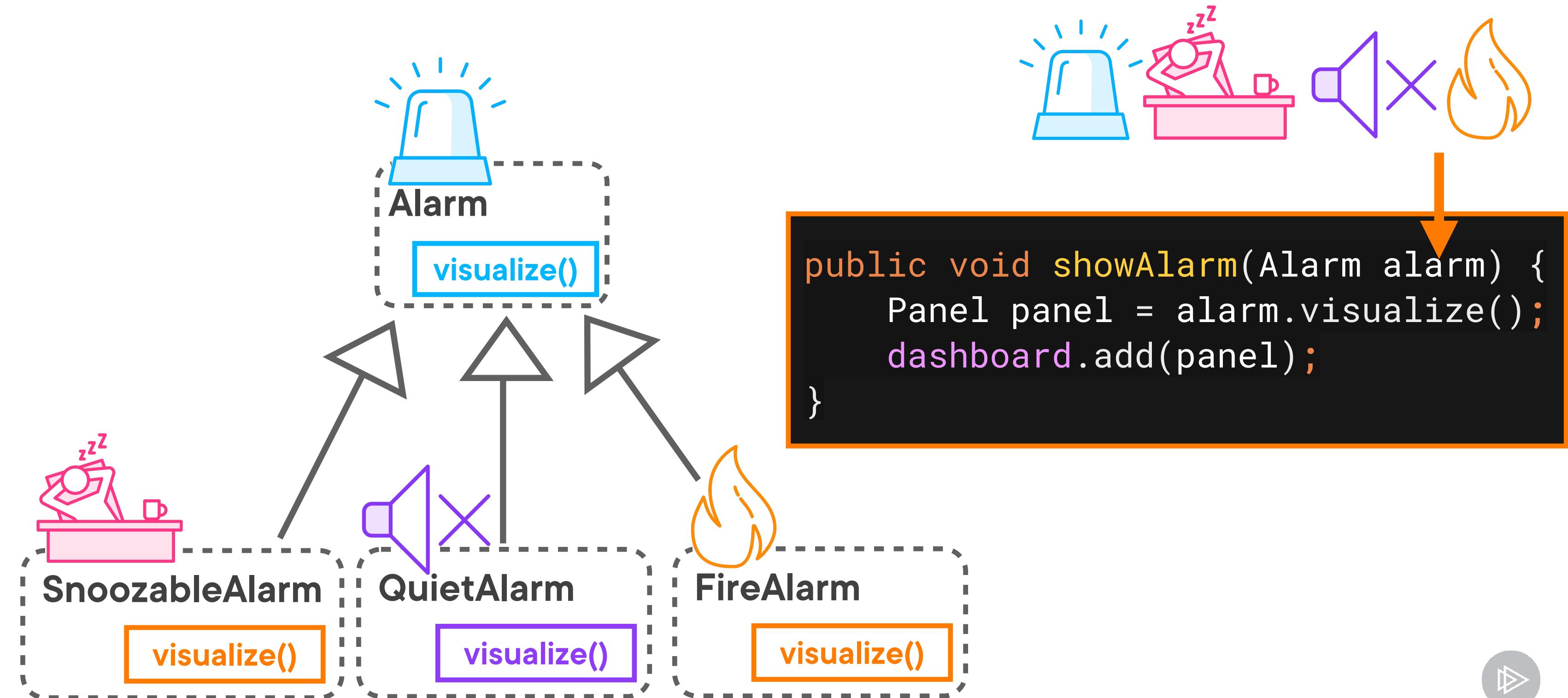
snooze()



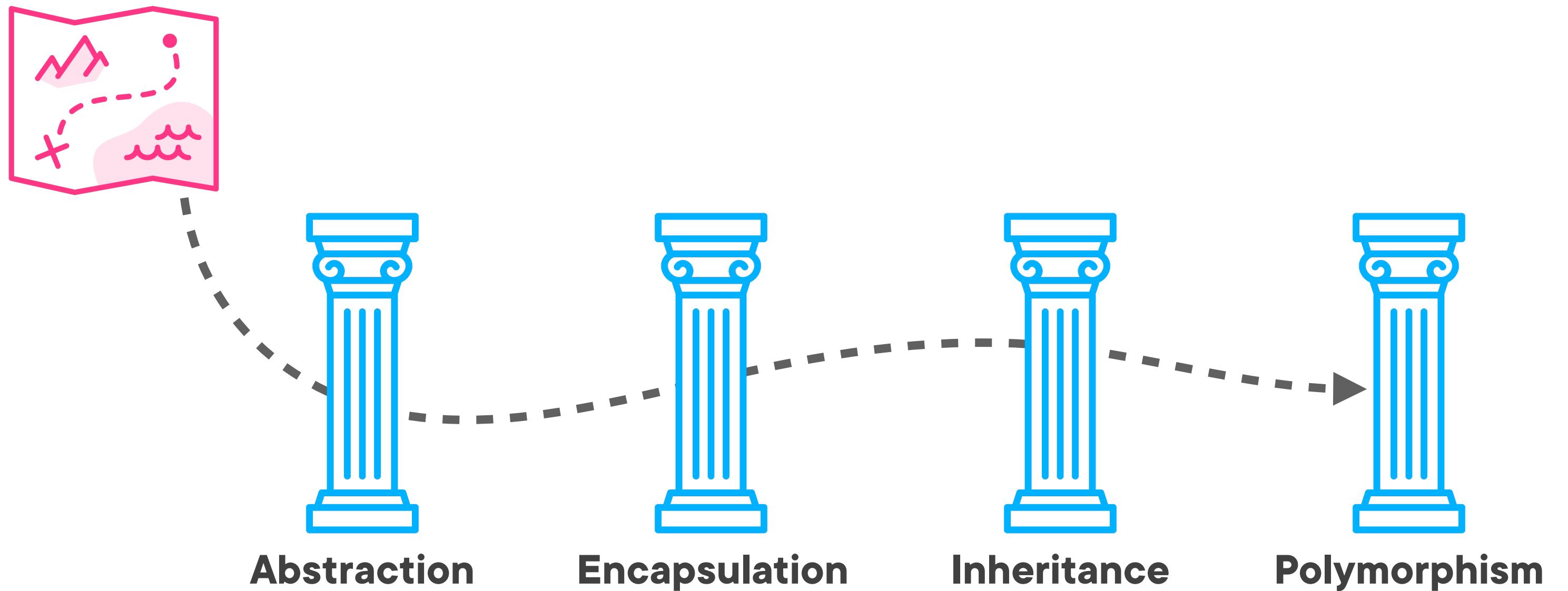
# Subclasses



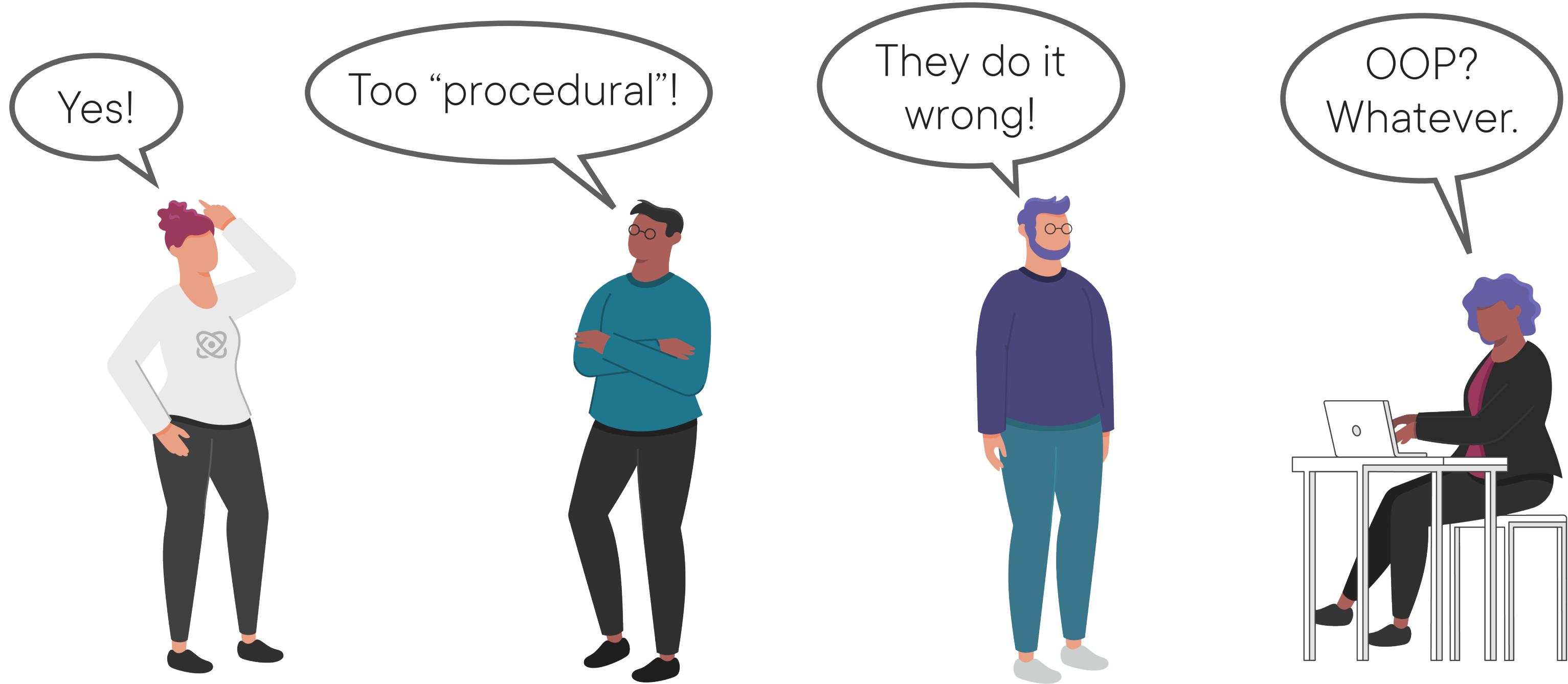
# Polymorphism



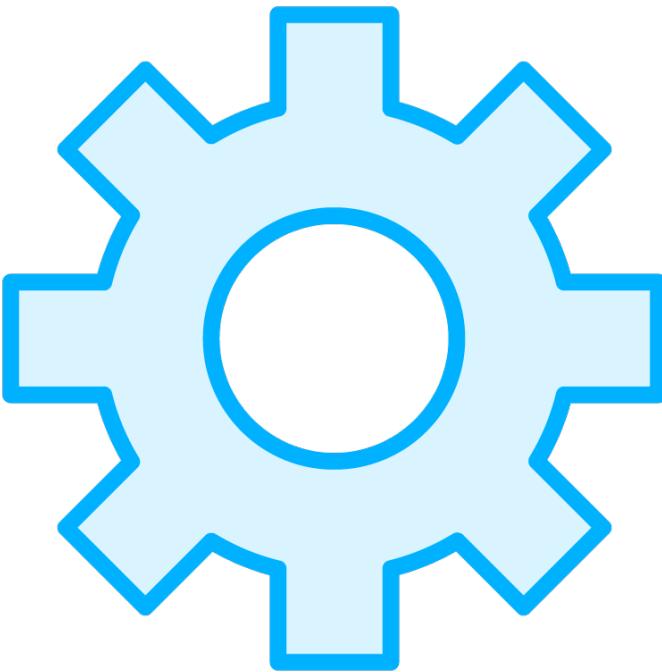
# The Pillars of OOP



# Is Java's OOP Good?



# Two Sides of Object-oriented Programming



## Mechanics

**How the object-oriented features of Java work**



## Design

**How to write object-oriented code that is flexible and easy to maintain**



# Introduction

**Course Overview**

**Approaching Object-oriented Programming**



# Abstraction and Encapsulation

Working with Objects

Defining Your Own Classes

Hiding Information

Designing with Abstraction and Encapsulation



# Inheritance and Polymorphism

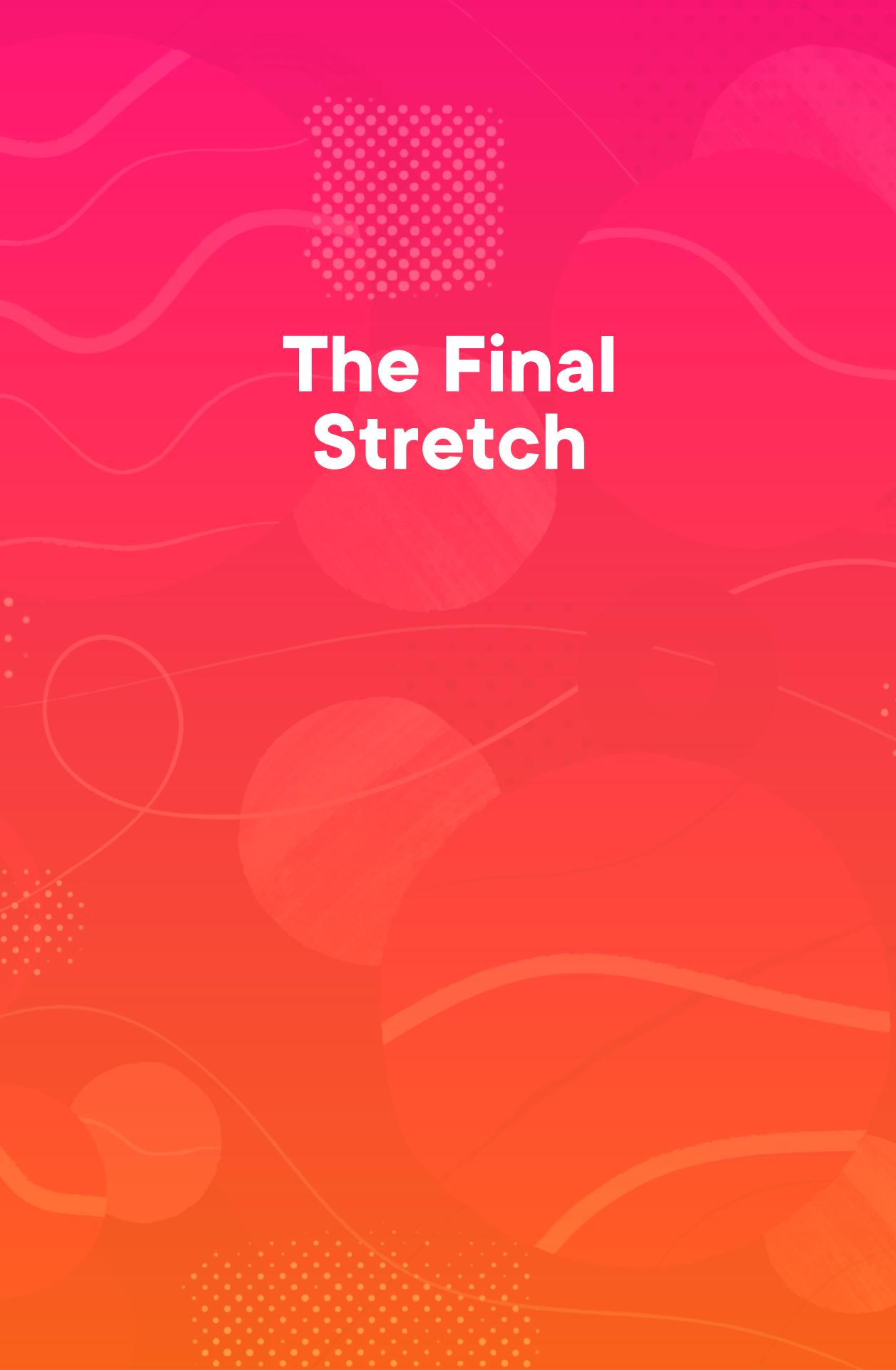
**Inheriting from Another Class**

**Understanding Polymorphism**

**Talking to Interfaces**

**Designing with Inheritance and Polymorphism**





# The Final Stretch

**Using the *static* Keyword**  
**Wrapping Up this Training**



**Up Next:**

# **Working with Objects**

---

