Facade Design Pattern

Bryan Hansen

@bh5k

Facade

Concepts

Easier API usage

Reduce external dependencies

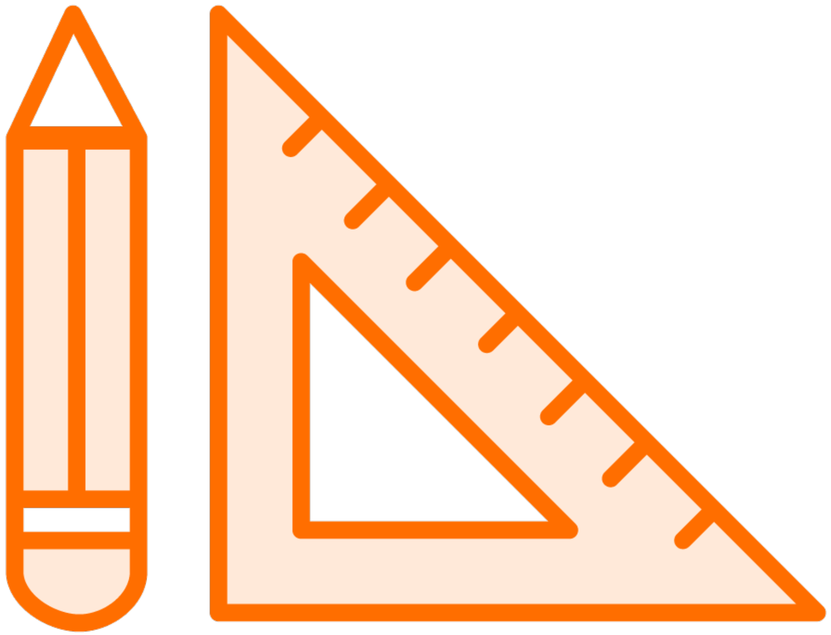
Simplified interface

Typically refactoring pattern

Examples:

java.net.URL

Design

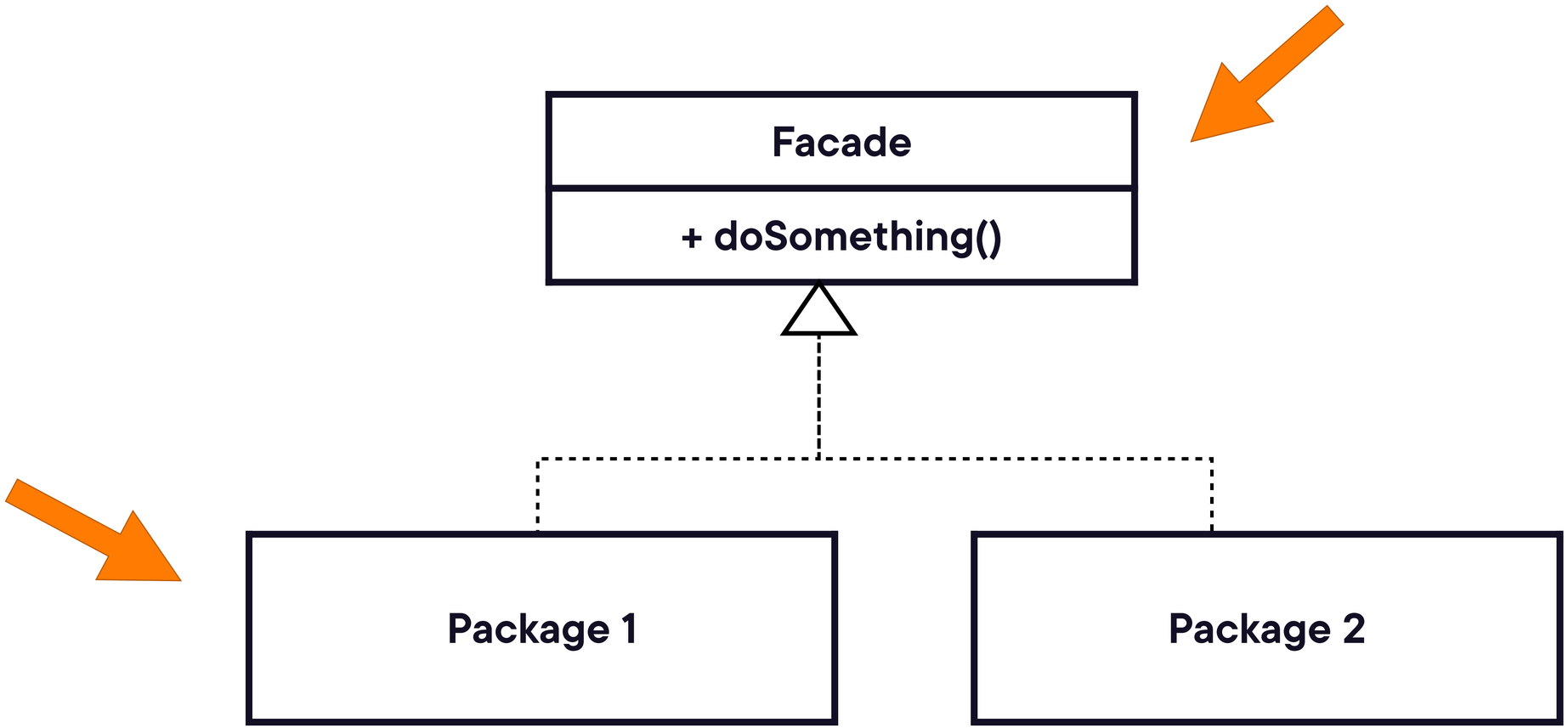


Class that utilizes composition

Shouldn’t need inheritance

Typically encompasses full lifecycle

UML



Everyday Example - InputStream

```
URL url = new URI("https://app.pluralsight.com/profile/
author/bryan-hansen").toURL();

BufferedReader in = new BufferedReader(
  new InputStreamReader(url.openStream()));

String inputLine;

while ((inputLine = in.readLine()) != null) {
  System.out.println(inputLine);
}
```

Exercise Facade

Complex Client

Client, Facade, JDBC

Simplified Client Code

Pitfalls

Typically used to clean up code

Should think about API design

Flat problem/structure

The “Singleton” of Structural Patterns

Contrast

Facade	VS	Adapter
Simplifies Interface		Also a refactoring pattern
Works with composites		Modifies behavior (adds)
Cleaner API		Provides a different interface

Facade Summary

Simplifies Client Interface

Easy Pattern to implement

Refactoring Pattern