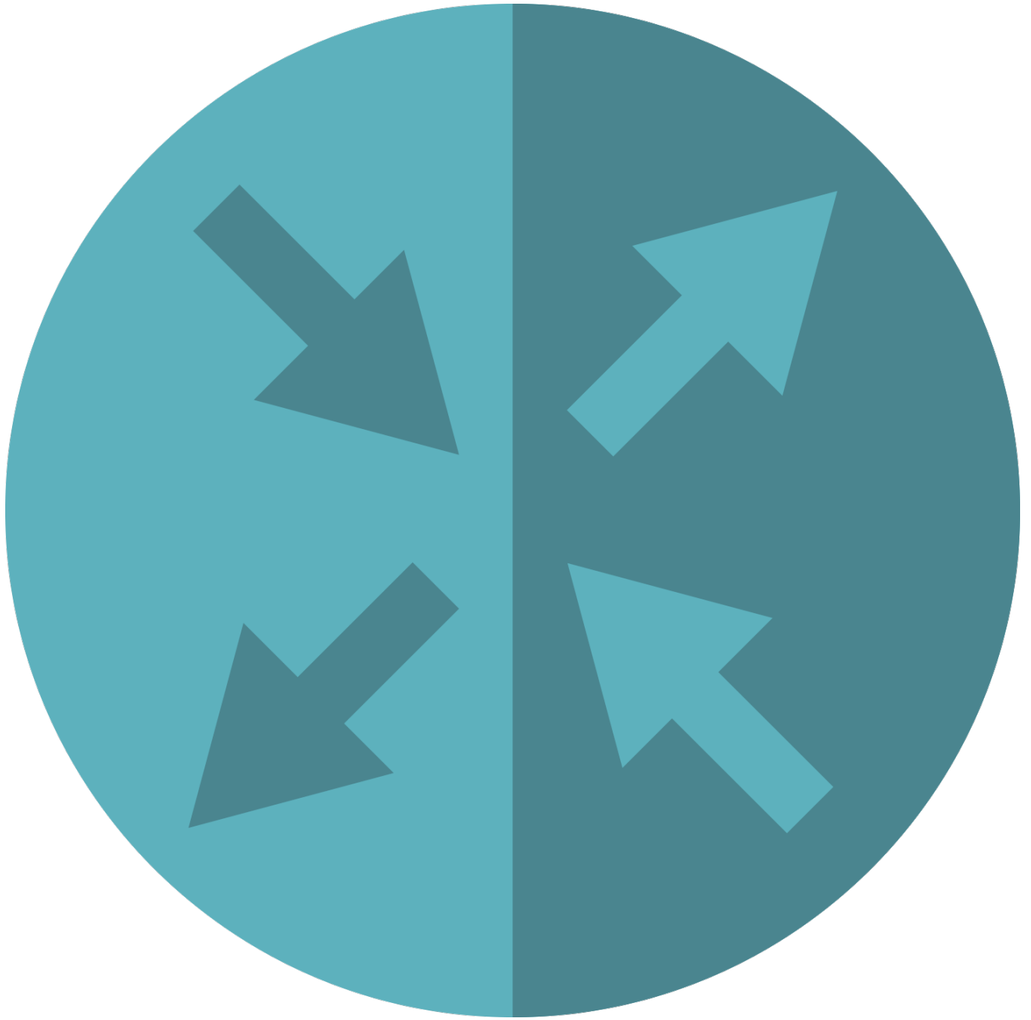
Facade Pattern

Bryan Hansen

twitter: bh5k | http://www.linkedin.com/in/hansenbryan

Concepts

- Make an API easier to use
- Reduce dependencies on outside code
- Simplify the interface or client usage
- Usually a refactoring pattern
- Examples:
  - java.net.URL
  - javax.faces.context.FacesContext



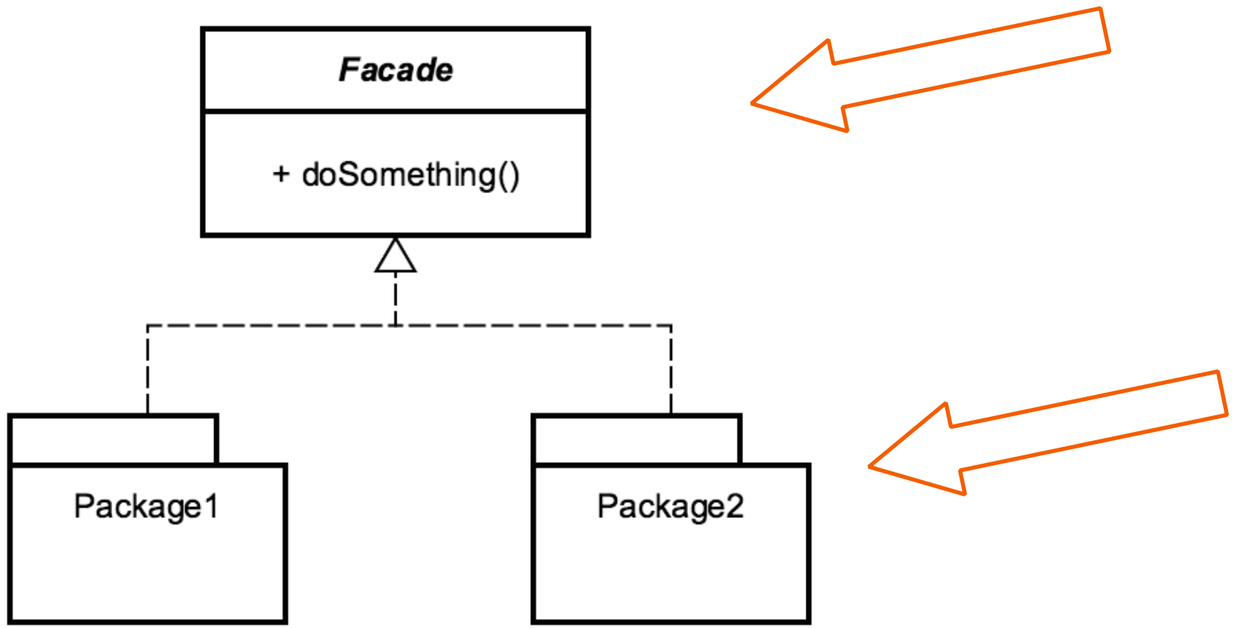
Design

Class that utilizes composition

Shouldn’t have a need for inheritance

Typically encompasses full lifecycle

UML



Everyday Example - URL

```
URL url = new URL("http", "www.pluralsight.com", 80,
   "/author/bryan-hansen");

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 Pattern

Contrast

Facade

- Simplifies Interface
- Works with composites
- Cleaner API

Adapter

- Also a refactoring pattern
- Modifies behavior (adds)
- Provides a different interface
- Single Object

Facade Summary

- Simplifies Client Interface
- Easy Pattern to implement
- Refactoring Pattern