

Hiding Complexity with the Façade Pattern



Gerald Britton

IT SOLUTIONS DESIGNER

@GeraldBritton www.linkedin.com/in/geraldbritton

Motivation



Access a database

Get employee records

Ask DBA which database to use

Import the right Python modules

Instantiate a control object

Build a connection string

Connect to the database

Run a query and process the results

Close the connection

Demo



Retrieve and print employee records

Use free AdventureWorks database

- Created by Microsoft
- Provided by Redgate
- Hosted in Microsoft Azure
- See <https://goo.gl/QCgJBW>

Façade

Classification: Structural

Unified interface to a set of interfaces

Higher-level interface

Makes the set of interfaces easier to use

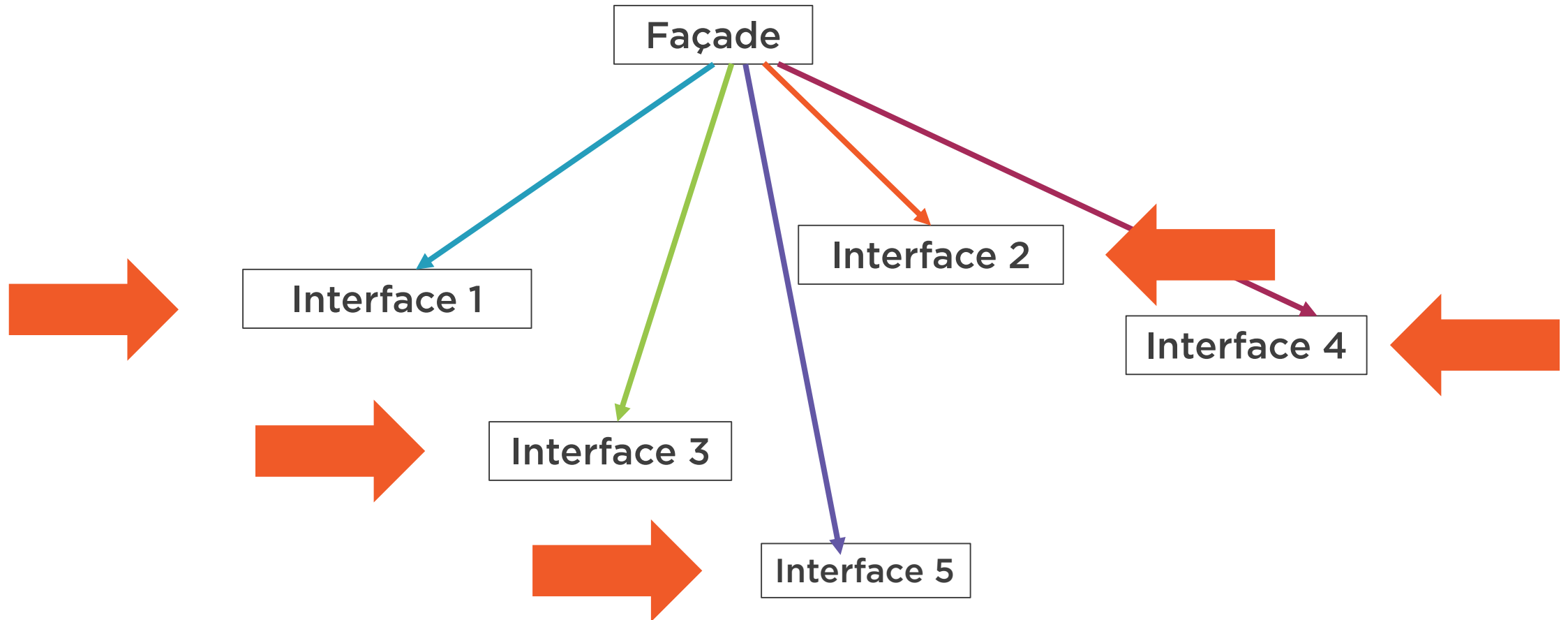
Reduces complexity

Example has five interfaces:

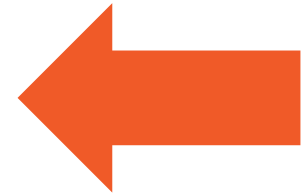
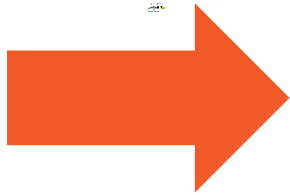
- connect, cursor, execute, commit and close

Implementing façade will result in just one

Façade Structure



Façade Structure



Demo



Summary



Shields clients from subsystem details

Reduces the objects to handle

Promotes weak coupling

Vary the subsystem

No change to client code

Clients can still use subsystems directly