

What

Best Practices Problems Solutions

Singleton

MVC

Civil Engineering

Gang of four

Eric Gamma

Richard Helm

Ralph Johnson

John Vlissides

Solution

Problem

Context

Why

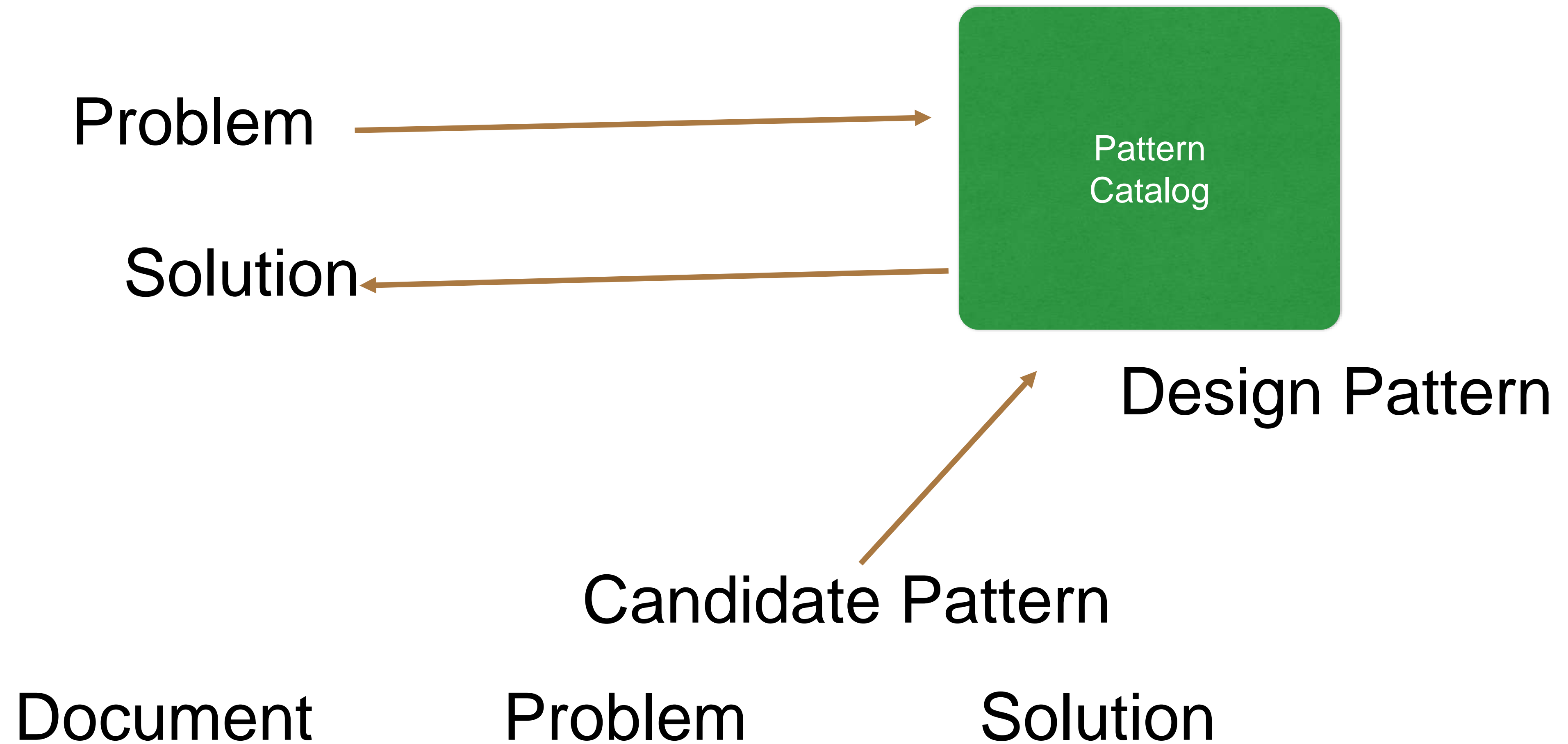
Capture design experience

Promote reuse without having to reinvent the wheel

Define the system structure better

Provide a common design language

How



Pattern Catalog

GOF Pattern Catalog

JEE Pattern Catalog

GOF Pattern Catalog

Creational Patterns

Structural Patterns

Behavioral Patterns

Creational Patterns

Singleton

Factory

Abstract Factory

Builder

Prototype

Structural Patterns

Adapter

Bridge

Flyweight

Decorator

Proxy

Behavioral Patterns

Command

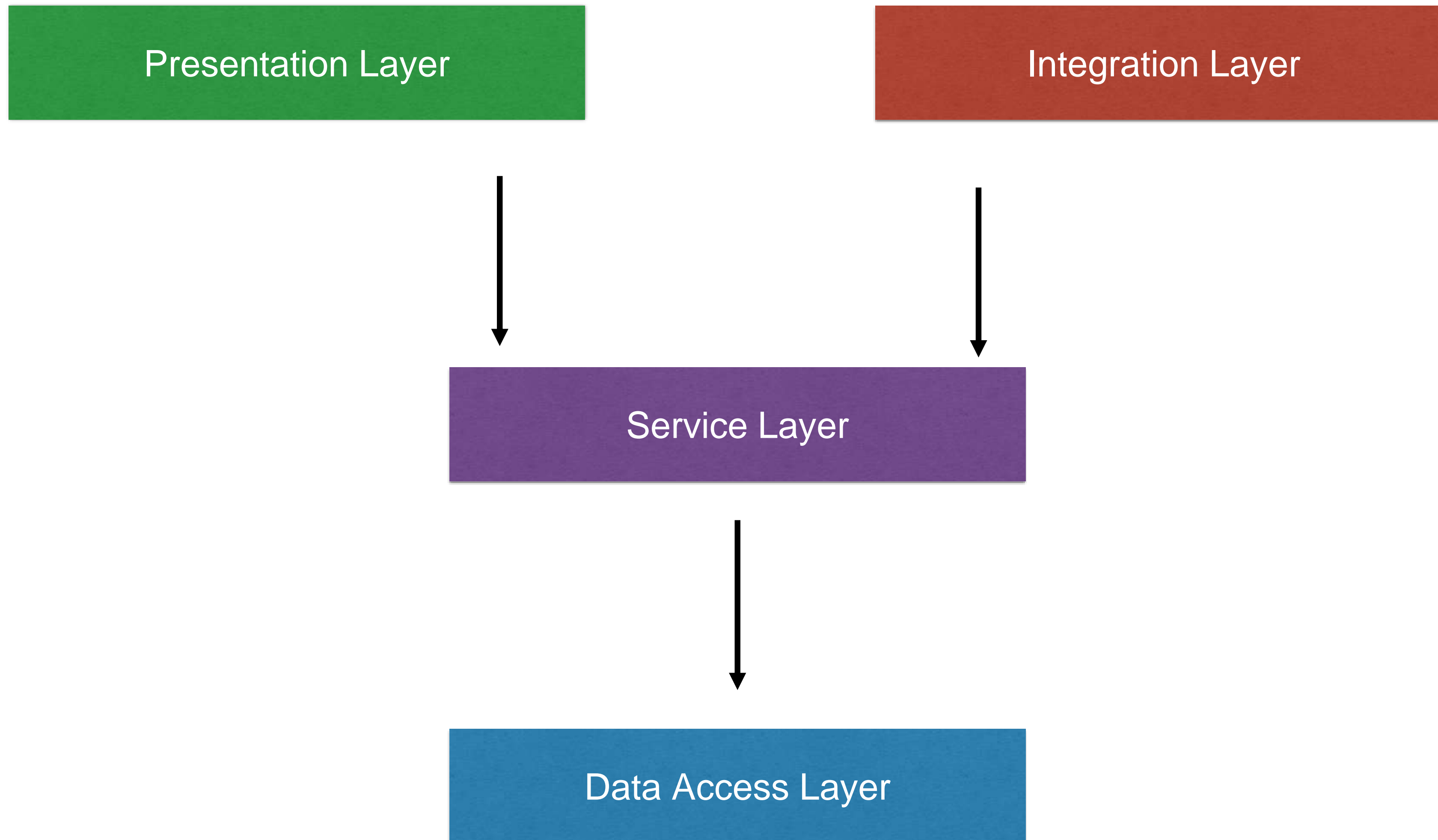
Interpreter

Template Method

Observer

Visitor

Java EE Pattern Catalog



Presentation Layer

Intercepting Filter

Front Controller

MVC

Context Object

Business Layer

Business Delegate

Transer Object

Session Facade

Service Locator

Data Access Layer

Data Access Object

Integration Layer

Service Activator

Web Service Broker

Abstract Factory

Factory of Factories

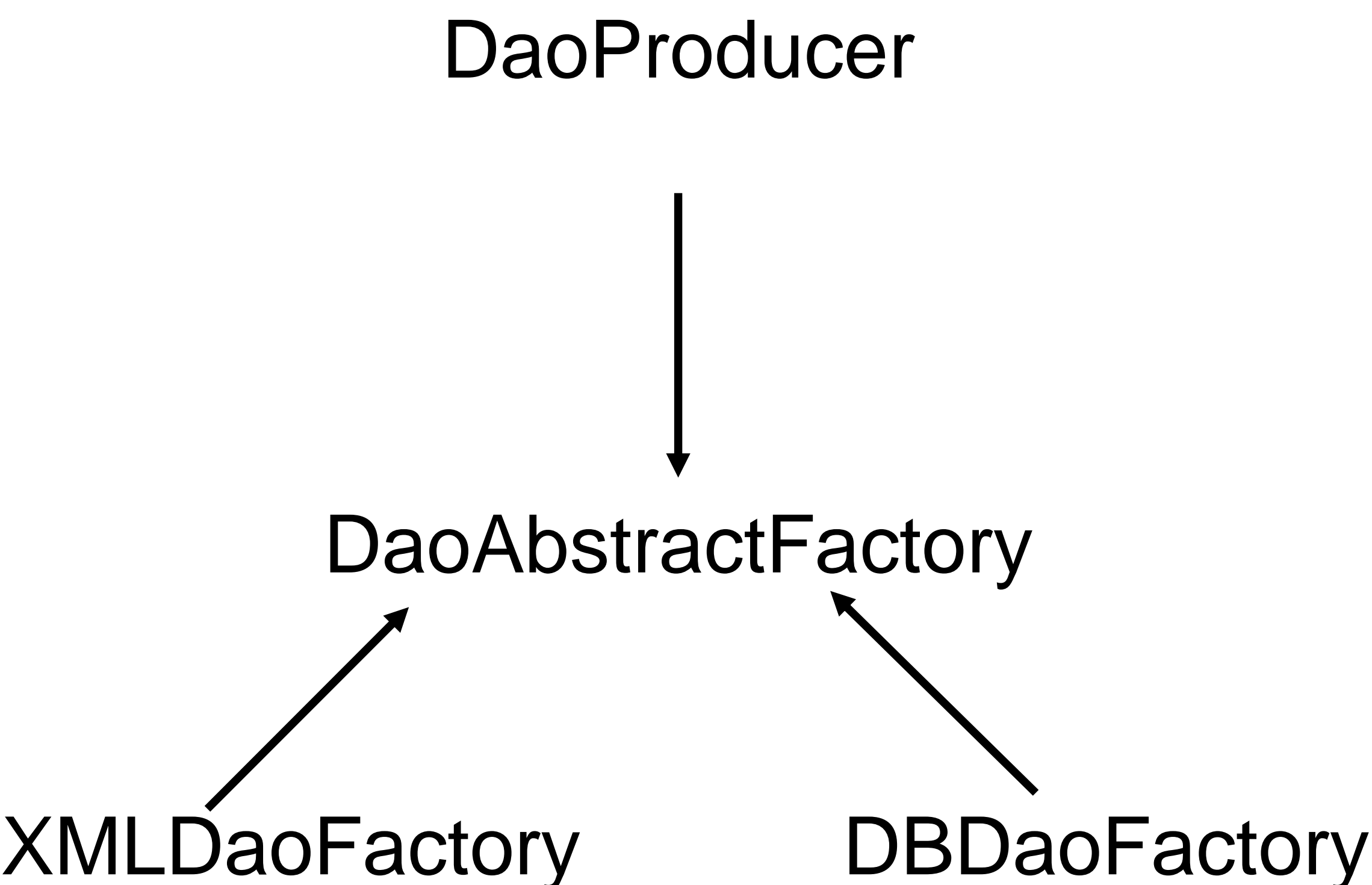
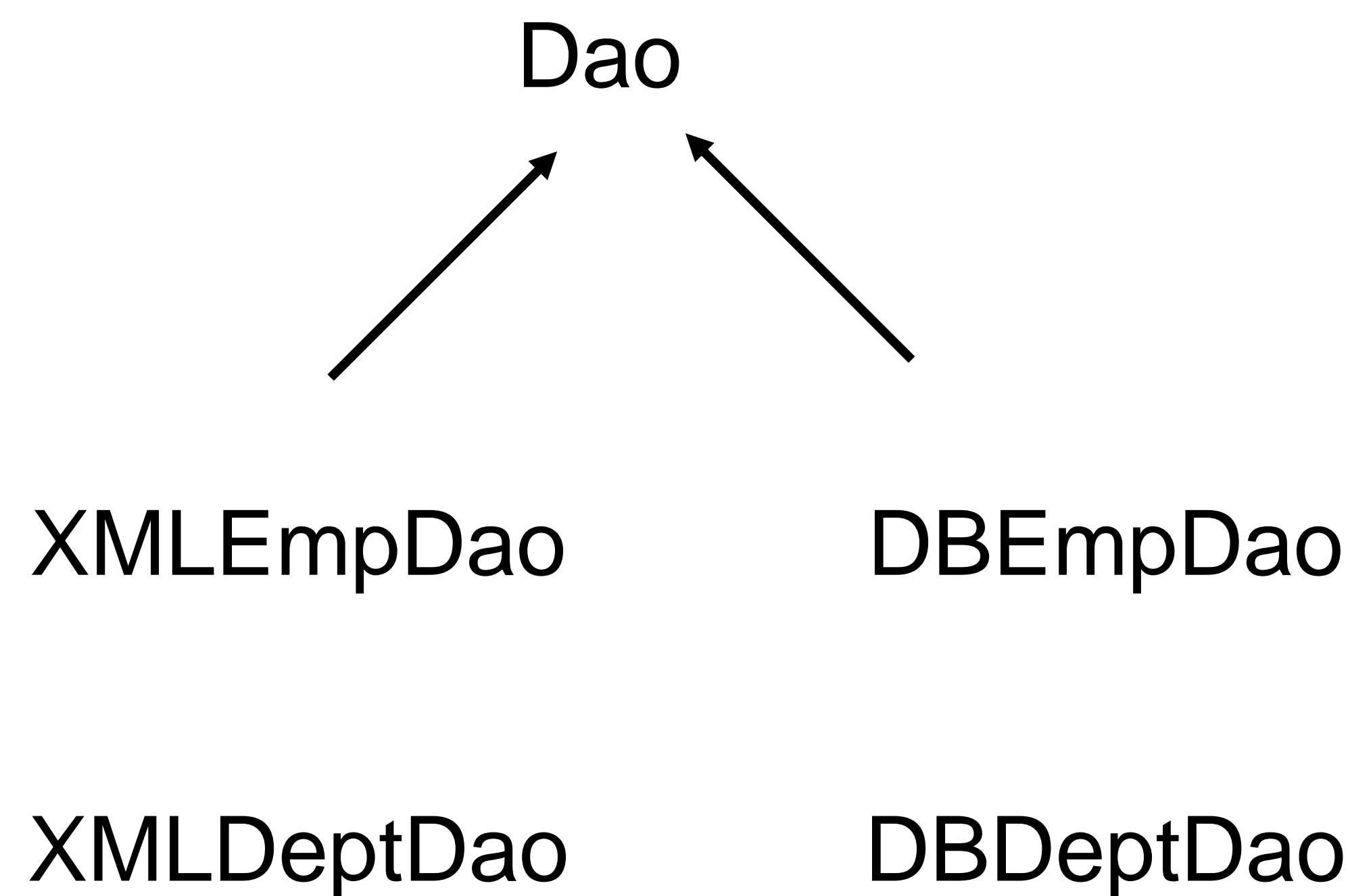
JAXP

DocumentBuilderFactory

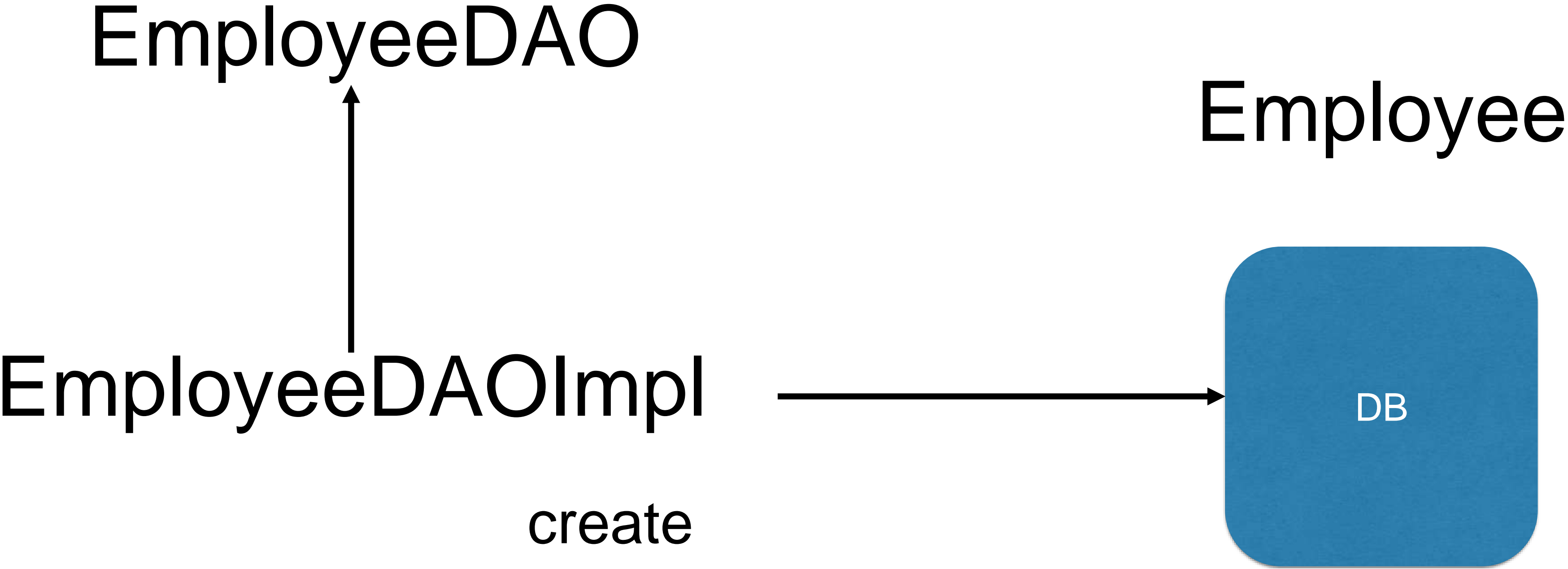
DocumentBuilder

Document

DAO Factory

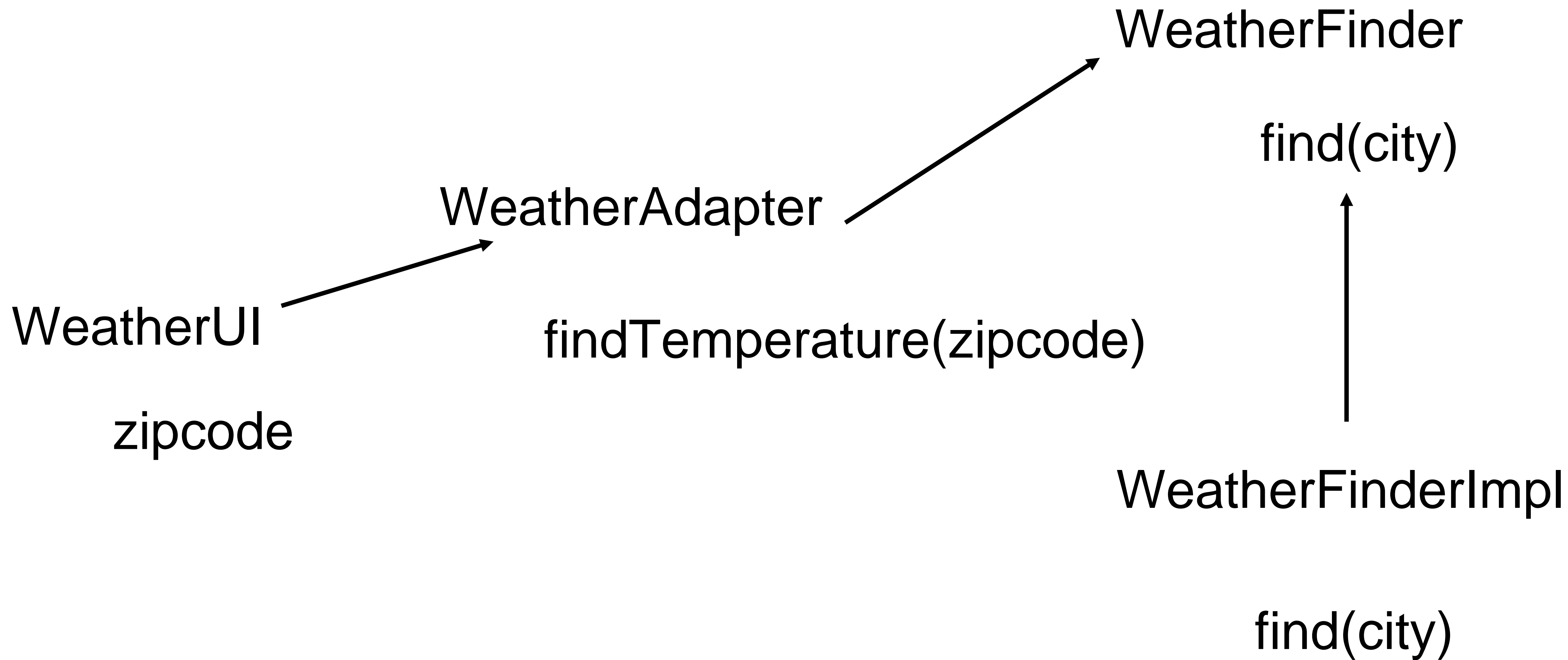


DAO



Adapter





Behavioral Pattern

Client



Invoker



Command

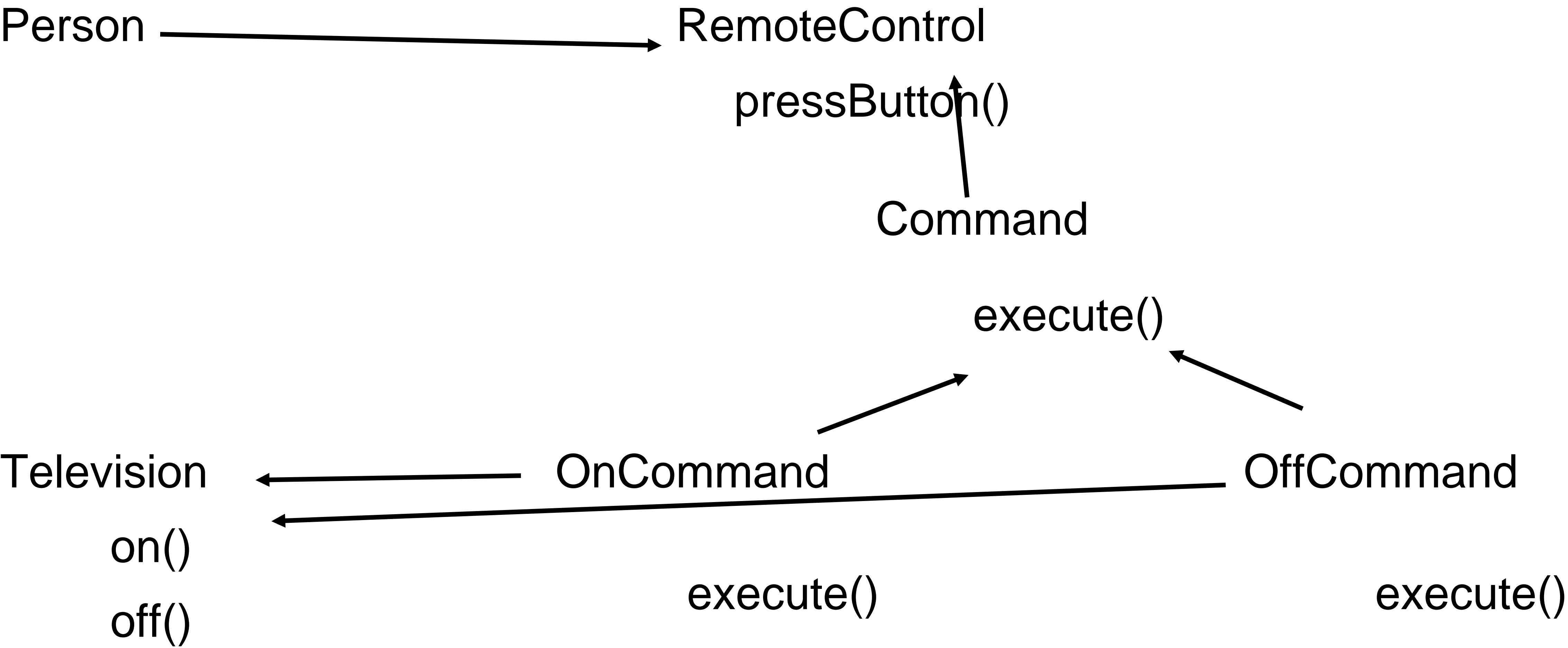


Receiver



ConcreteCommand

Command



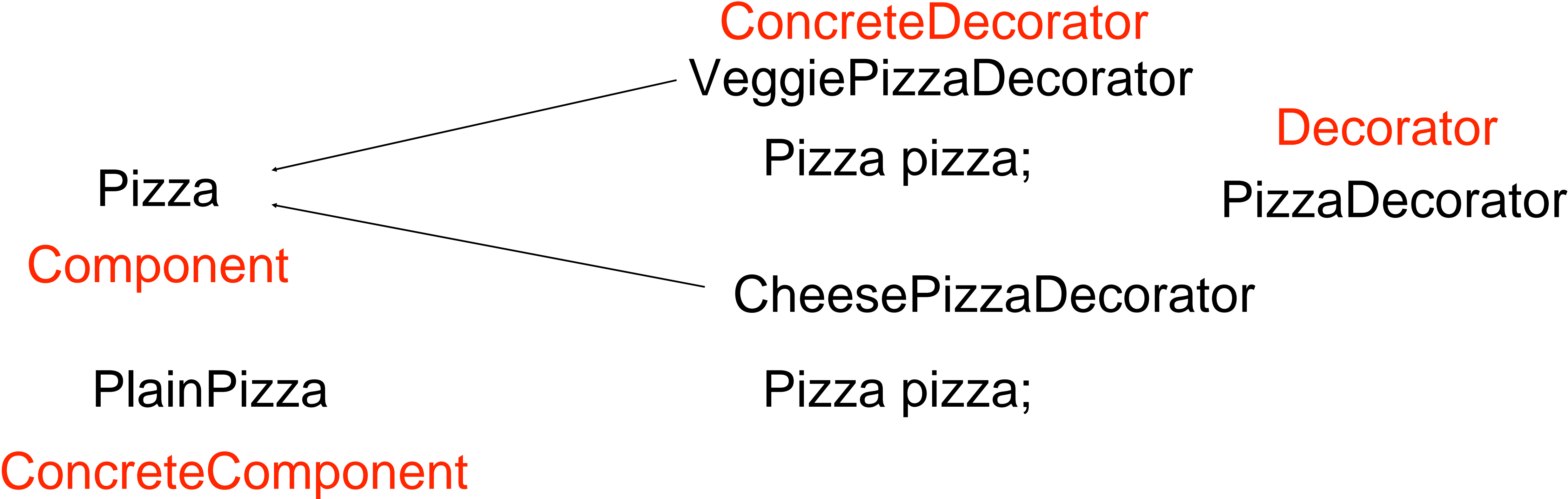
Decorator

Functionality at runtime

```
new BufferedReader(new FileReader(..));
```

Reader

Client



Factory

Hides the object creation process

Car Factory

Chocolate Factory

Toy Factory

Connection connection =

DriverManager.getConnection(String connectionString)

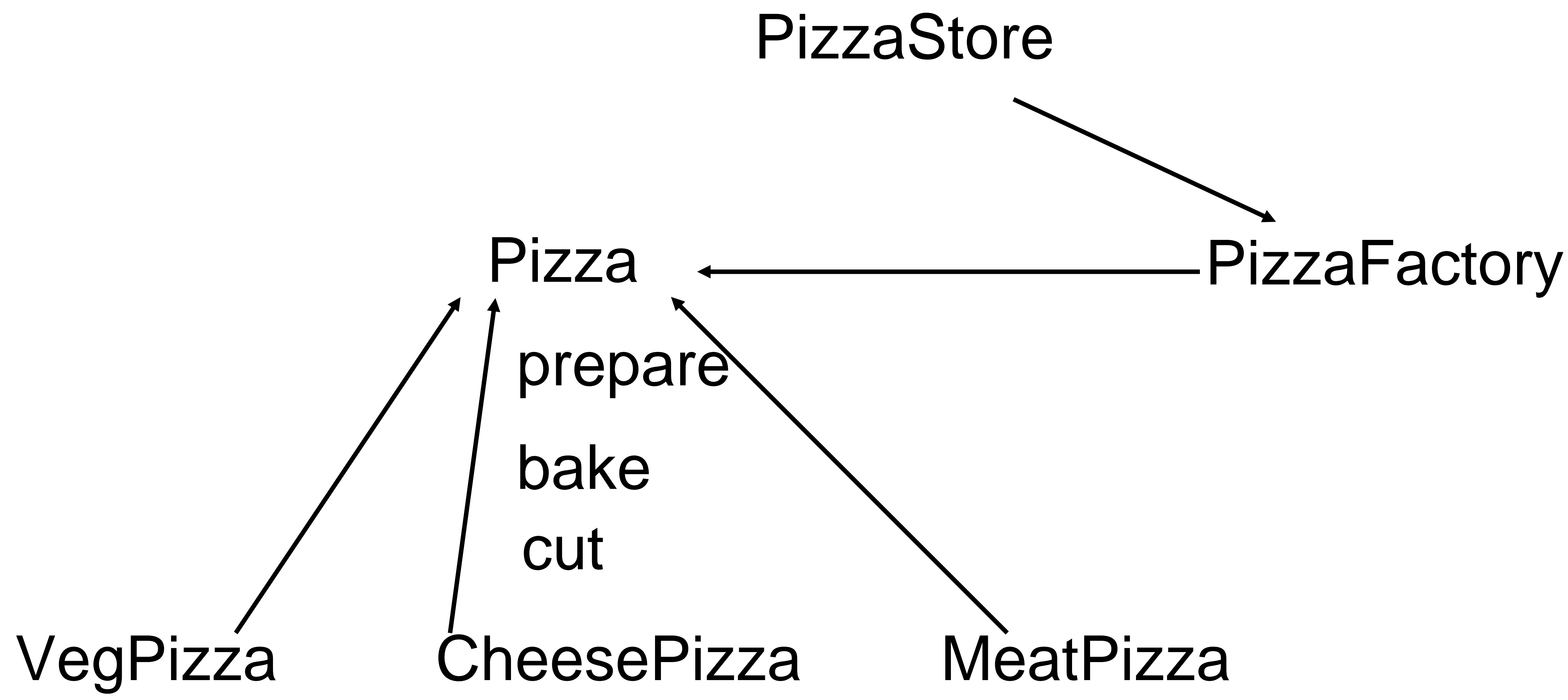
Oracle

Driver

MySql

Sql Server

Pizza Store

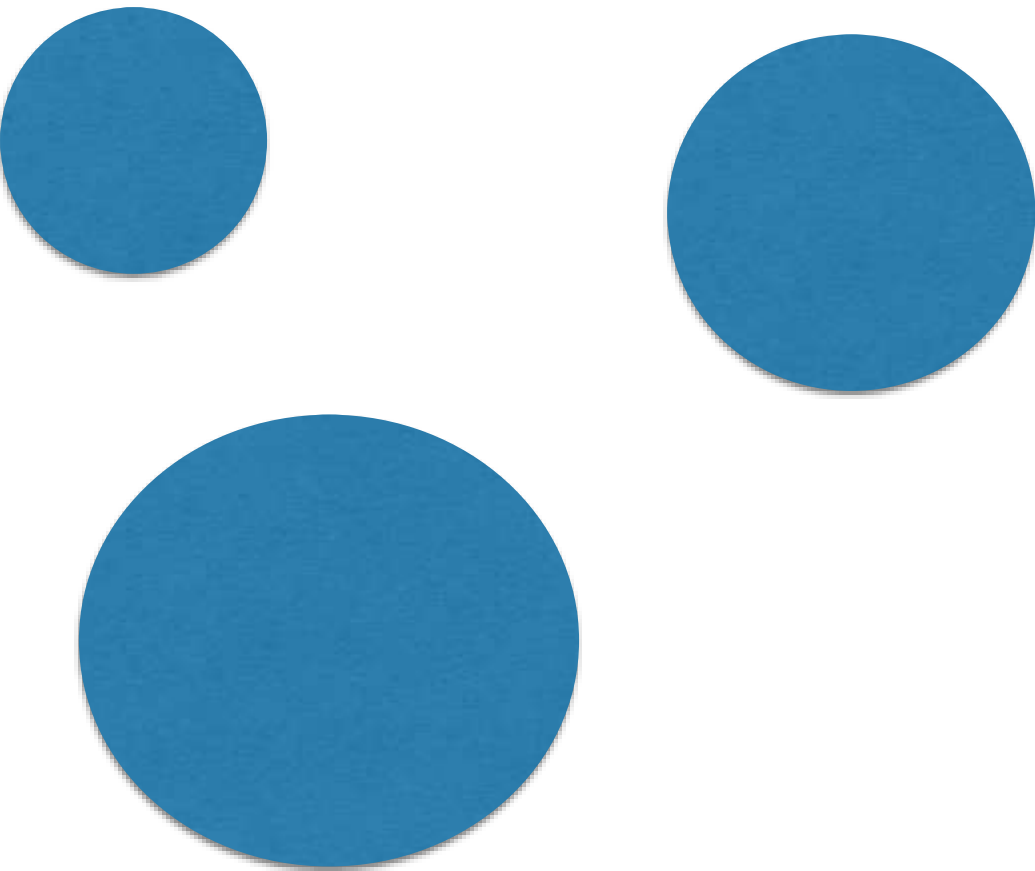


Flyweight

Save Memory

Paint App

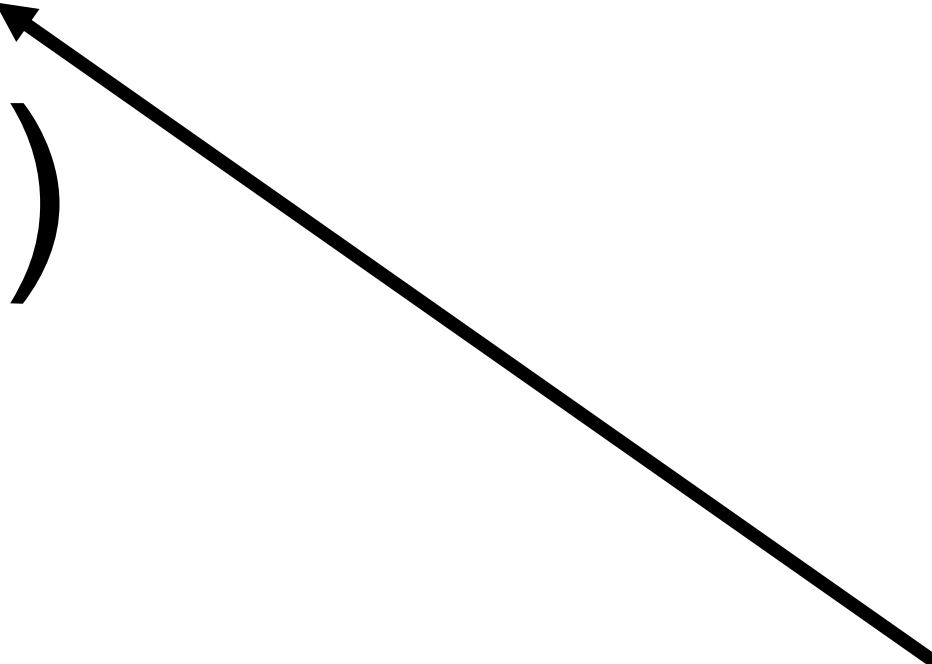
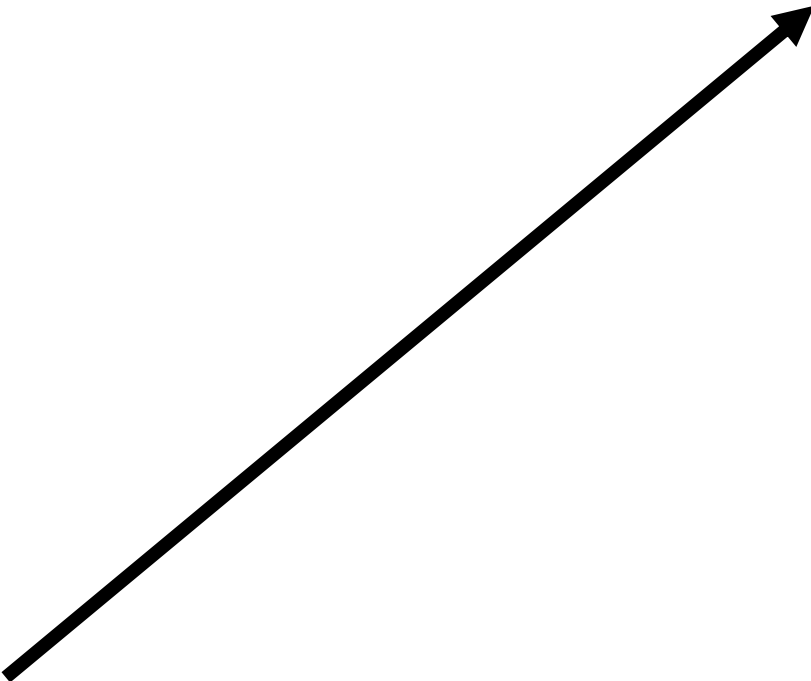
Shape
draw()



Circle

radius

draw()



Rectangle

length

breadth

draw()



Separate the Extrinsic State

Pass them as parameters

Create a factory class

Key Points

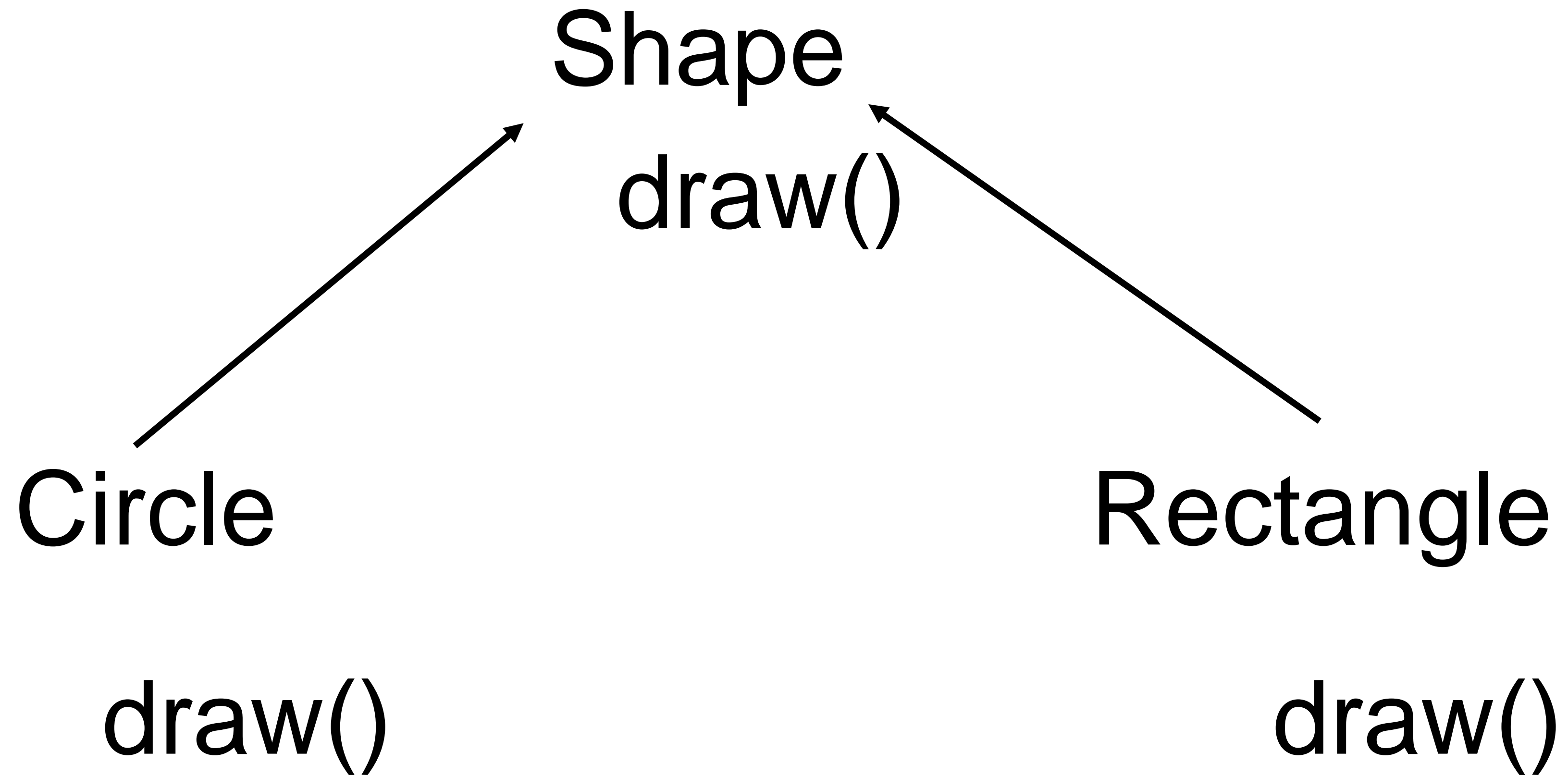
Memory Foot Print

Non Sharable state as parameters

Create a factory class

Make the Clients Use it

Paint App



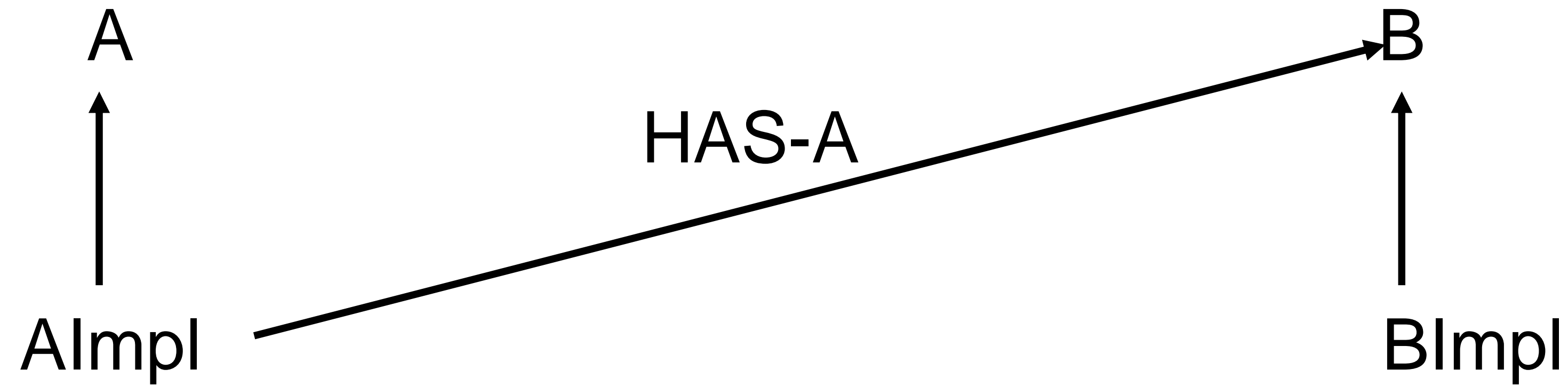
Separate the Extrinsic State

Pass them as parameters

Create a factory class

Dependency Injection

Inversion Of Control



```
B b = new BImpl();
```

Spring Container

Presentation Layer



Service Layer



Data Access Layer

Interceptor Filter

Front Controller

MVC

Business Delegate

Business Object

Data Access Object

Singleton

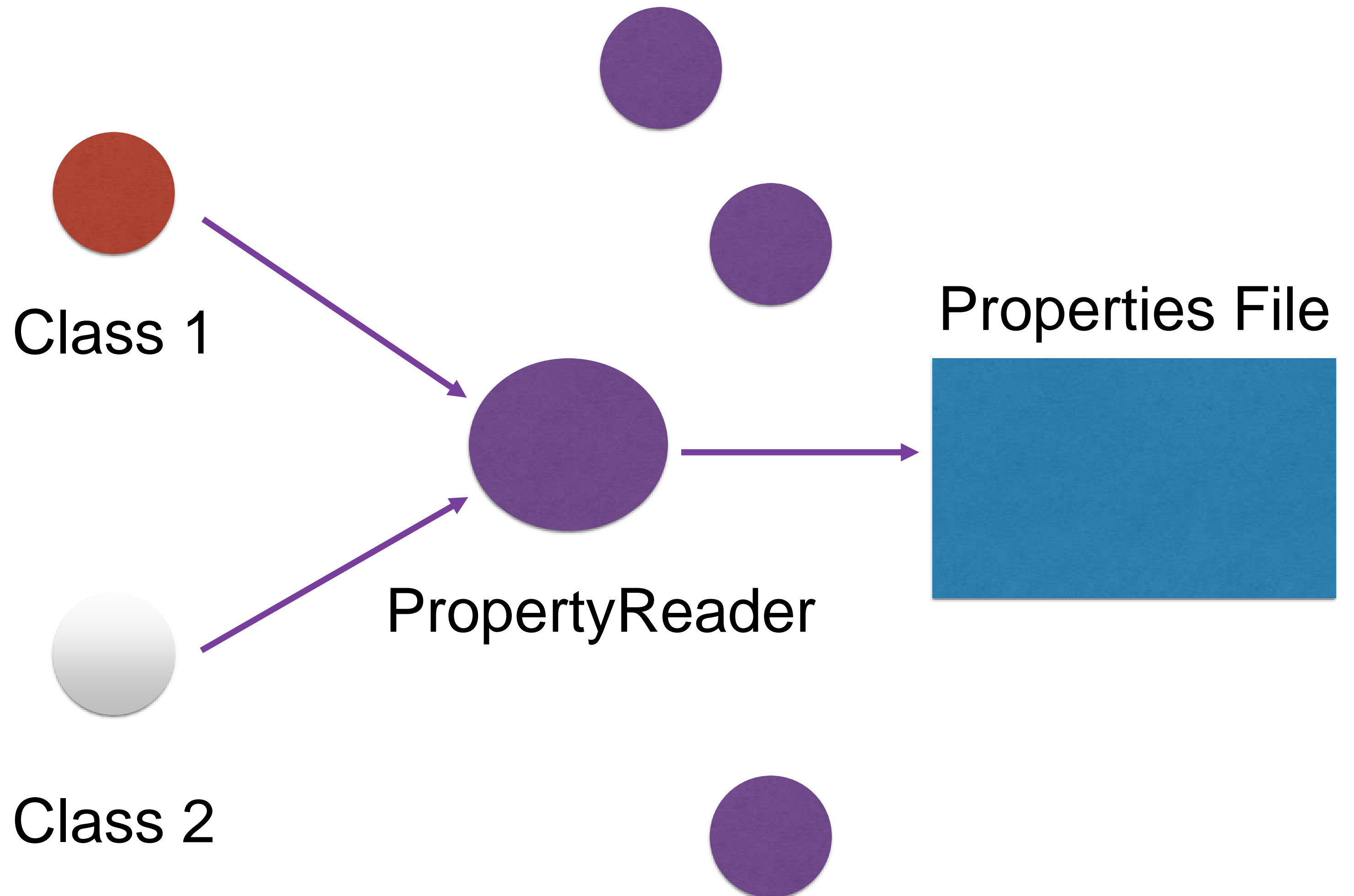
ONE AND ONLY ONE

Logger

error

debug

info



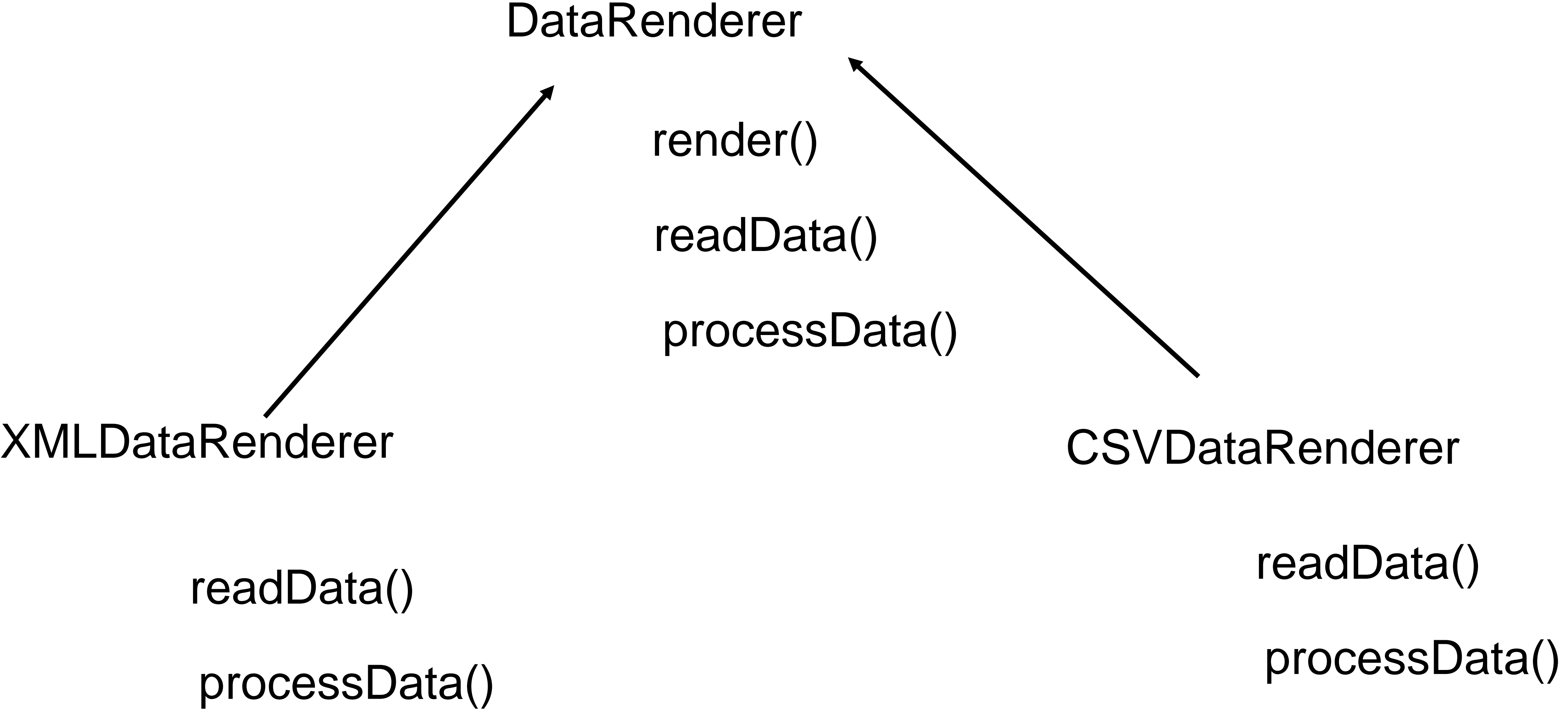
Steps

Declare the constructor of the class as private

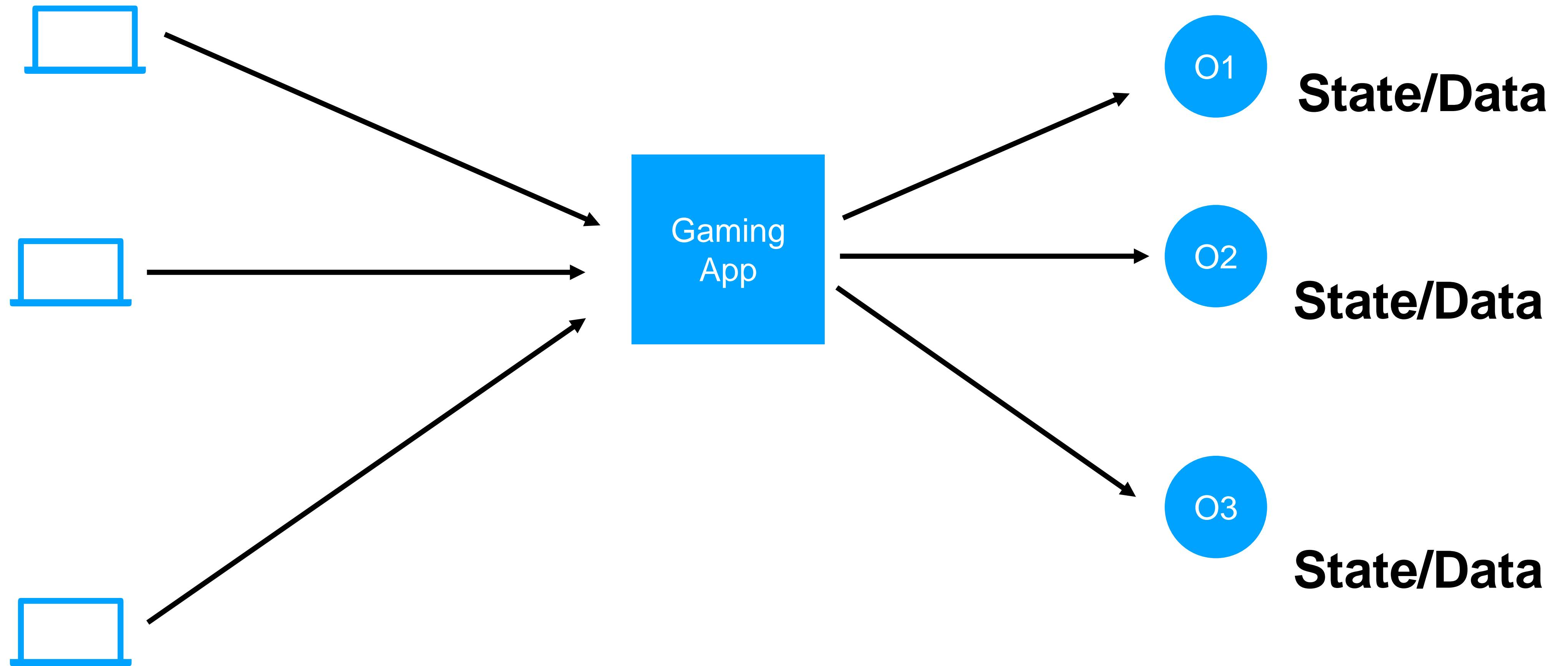
Declare a static method

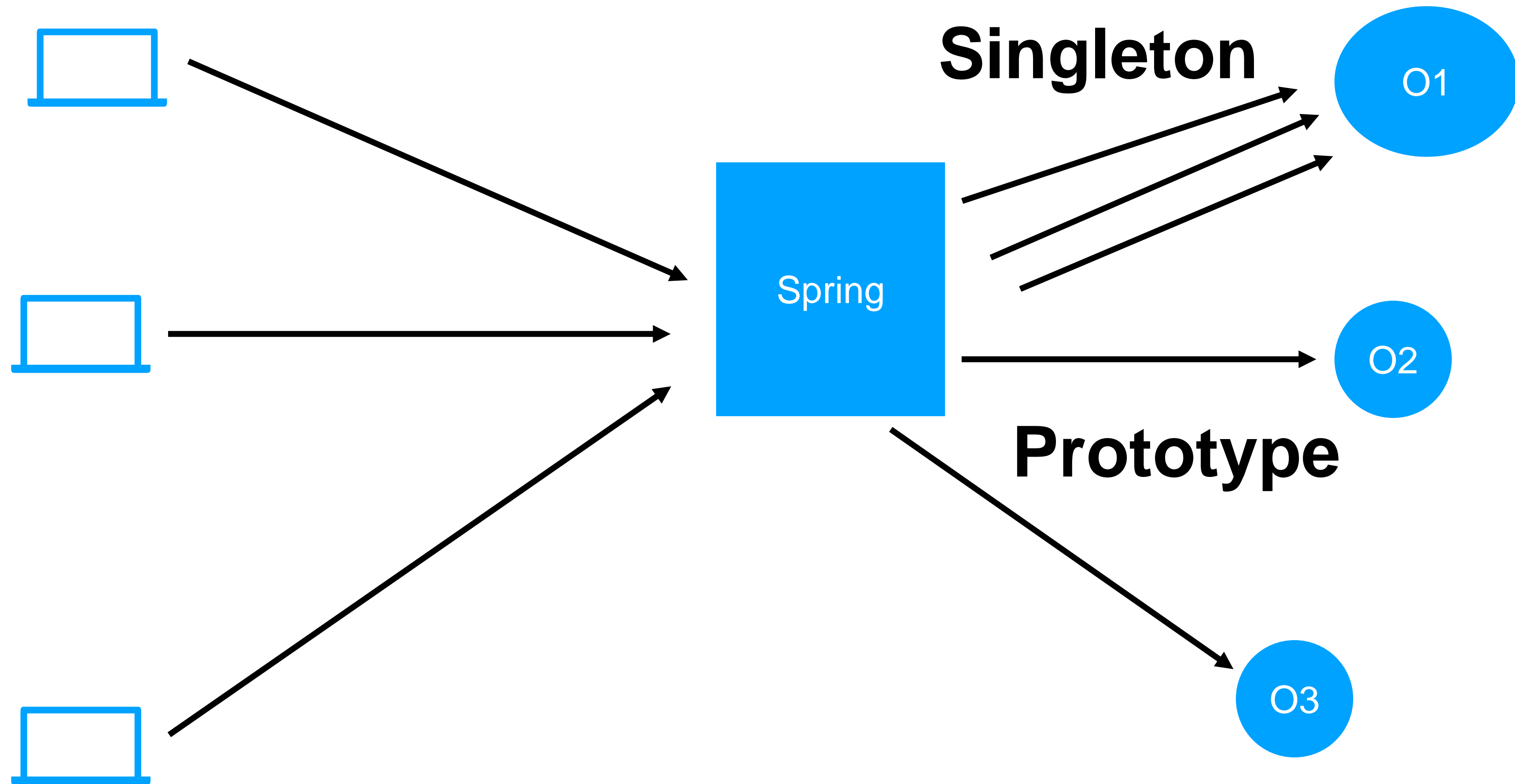
Declare a static member of the same class
type in the class

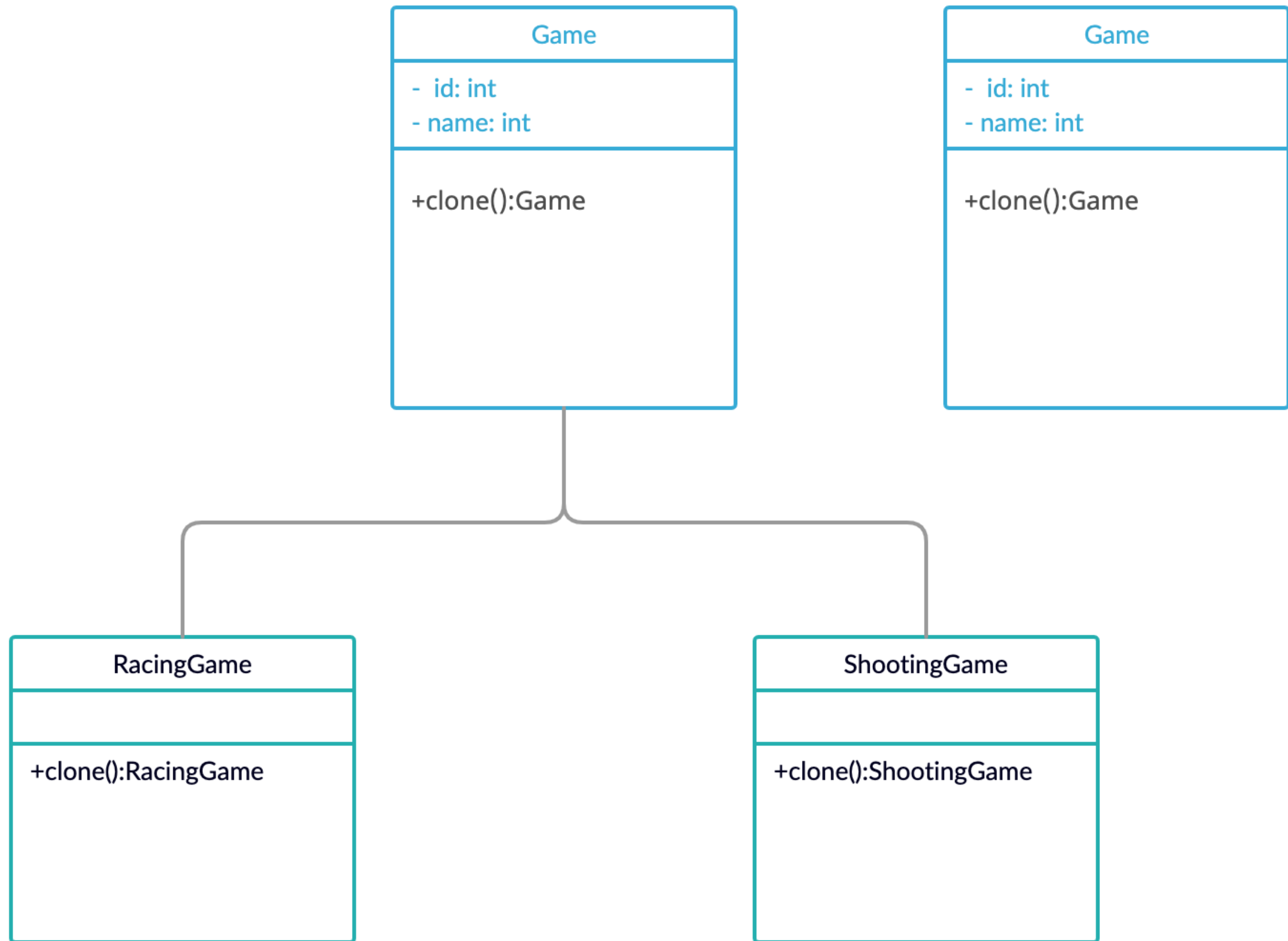
Template Method



Prototype







Facade

