

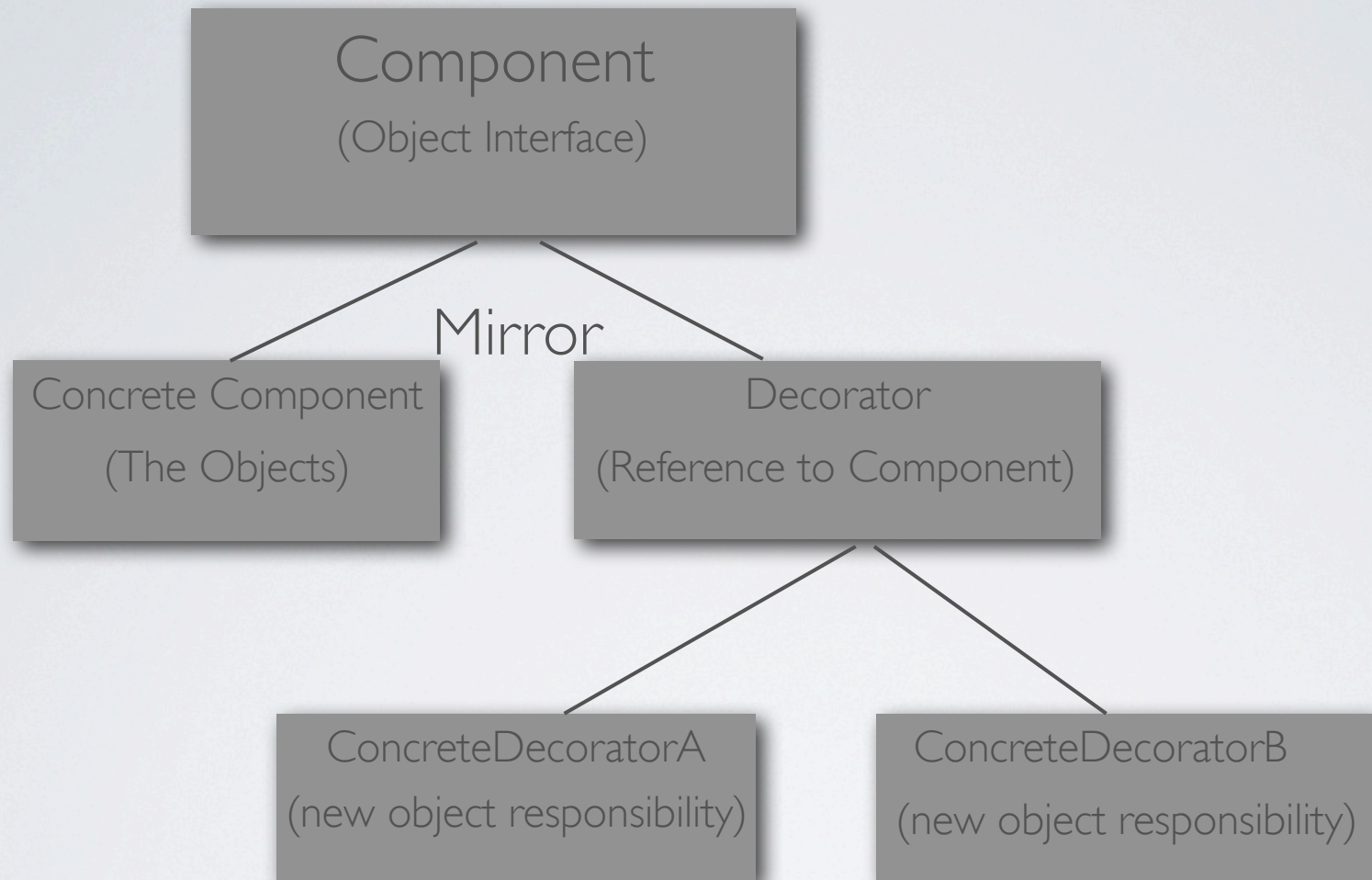
DECORATOR

Design Pattern

Dray Brown
January 19, 2011

WHAT IS THE DECORATOR PATTERN?

- A design pattern that allows additional responsibilities to be added to a single object dynamically.
- New functionality can be added to an object without modifying similar objects.
 - Done by adding a transparent layer (also known as a “Wrapper”).



Car



I am a car.

Car



I am a car.

I am sporty!

Car



I am a car.
I am expensive.

Car

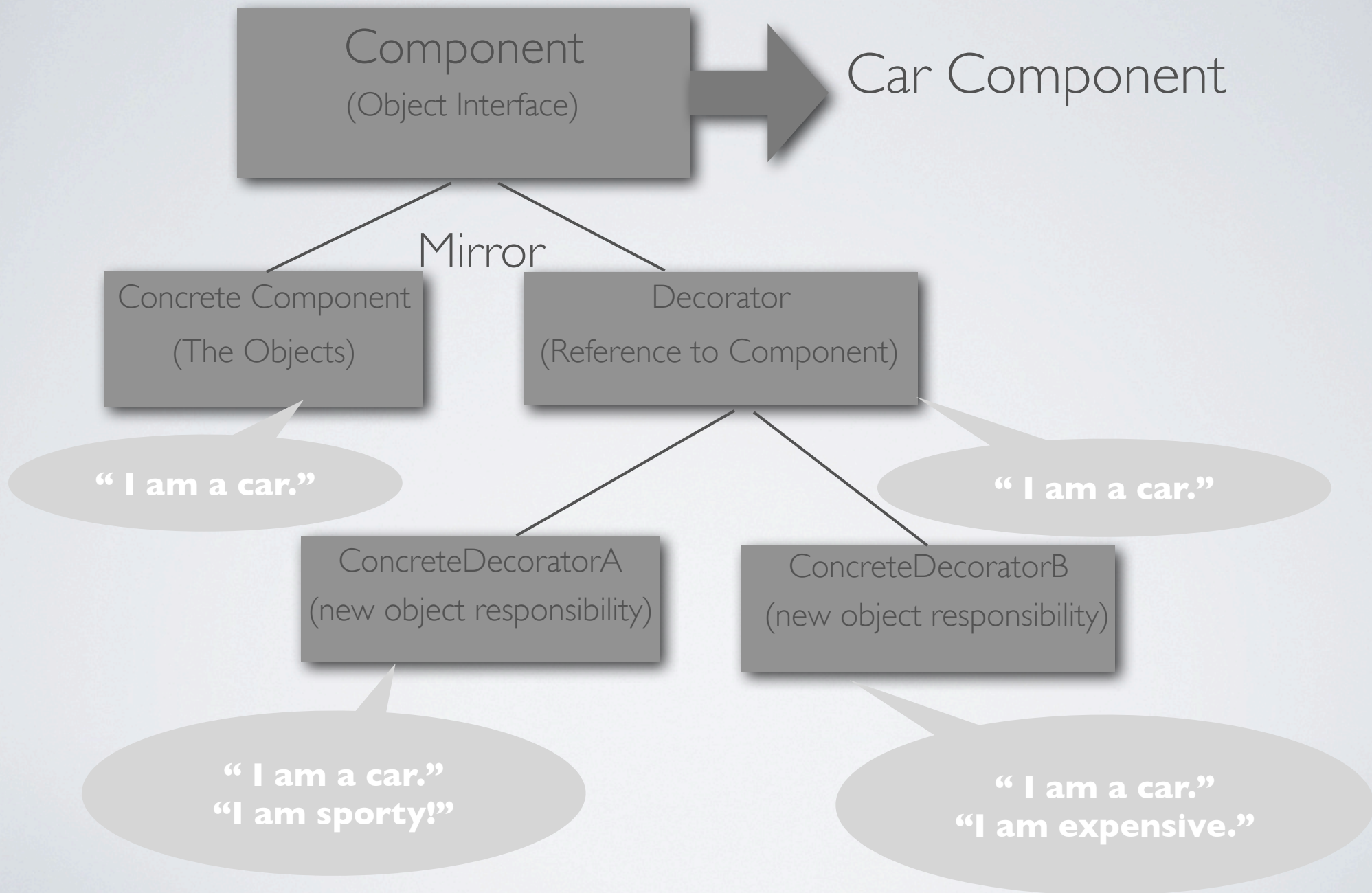


I am a car.
I am expensive.
I am sporty!

CAR DESCRIPTIONS

Car
“I am a car”

- Standard car statement - “I am a car.”
- Sporty car statement - “I am sporty!”
- Expensive car statement - “I am expensive.”
- Expensive & Sporty car statement - “I am expensive. I am sporty!”



IMPLEMENTATION USING JAVA

```
public class Car {  
    public Car(){  
    }  
  
    public String statement(){  
        return "I am a car.";  
    }  
}  
  
public abstract class Decorator extends Car{  
    public abstract String statement();  
}  
  
public class Sporty extends Decorator{  
    Car auto;  
  
    public Sporty (Car sport){  
        auto = sport;  
    }  
    public String statement(){  
        return auto.statement() + " I am sporty.";  
    }  
}
```



```
public class Expensive extends Car {  
    Car auto;  
  
    public Expensive (Car das){  
        auto = das;  
    }  
    public String statement(){  
        return auto.statement() + " I am expensive.";  
    }  
}
```

```
public class DecorateMe { //Tester  
    public static void main (String [] args){  
  
        Car mercedes = new Car();  
        mercedes = new Expensive(mercedes);  
  
        Car honda = new Car();  
  
        Car mitsubishi = new Car();  
        mitsubishi = new Sporty(mitsubishi);  
  
        Car porsche = new Car();  
        porsche = new Sporty(porsche);  
        porsche = new Expensive(porsche);  
  
        System.out.println("I am a Honda. " +honda.statement());  
        System.out.println("I am a Mitsubishi, " +mitsubishi.statement());  
        System.out.println("I am a Mercedes. " +mercedes.statement());  
        System.out.println("I am a Porsche. " +porsche.statement());  
    }  
}
```

REAL WORLD EXAMPLE

McMaster-Carr website

<http://www.mcmaster.com/#ball-valves/=anitas>