We all scream for Ice cream!

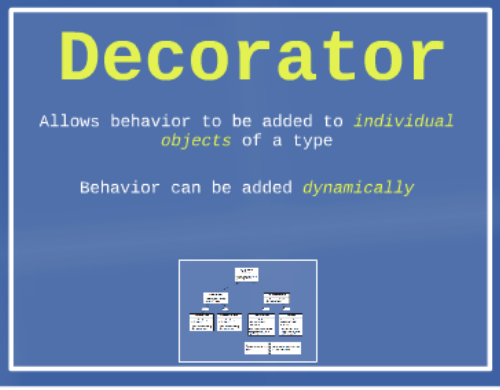
Choose base *flavor* & add one or more *mix-ins*

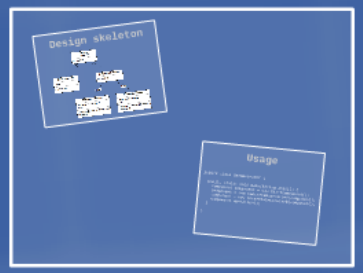
Different *flavors*/*mix-ins* can have different *costs*

Design software to produce full *description* & calculate final *cost*

Design options?

- Single ice-cream class
- Combinations hierarchy

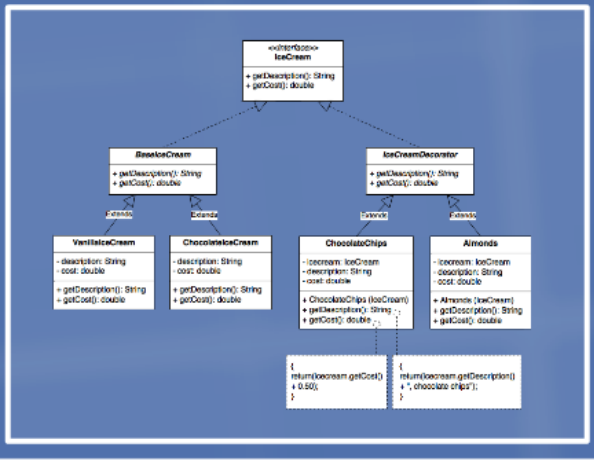


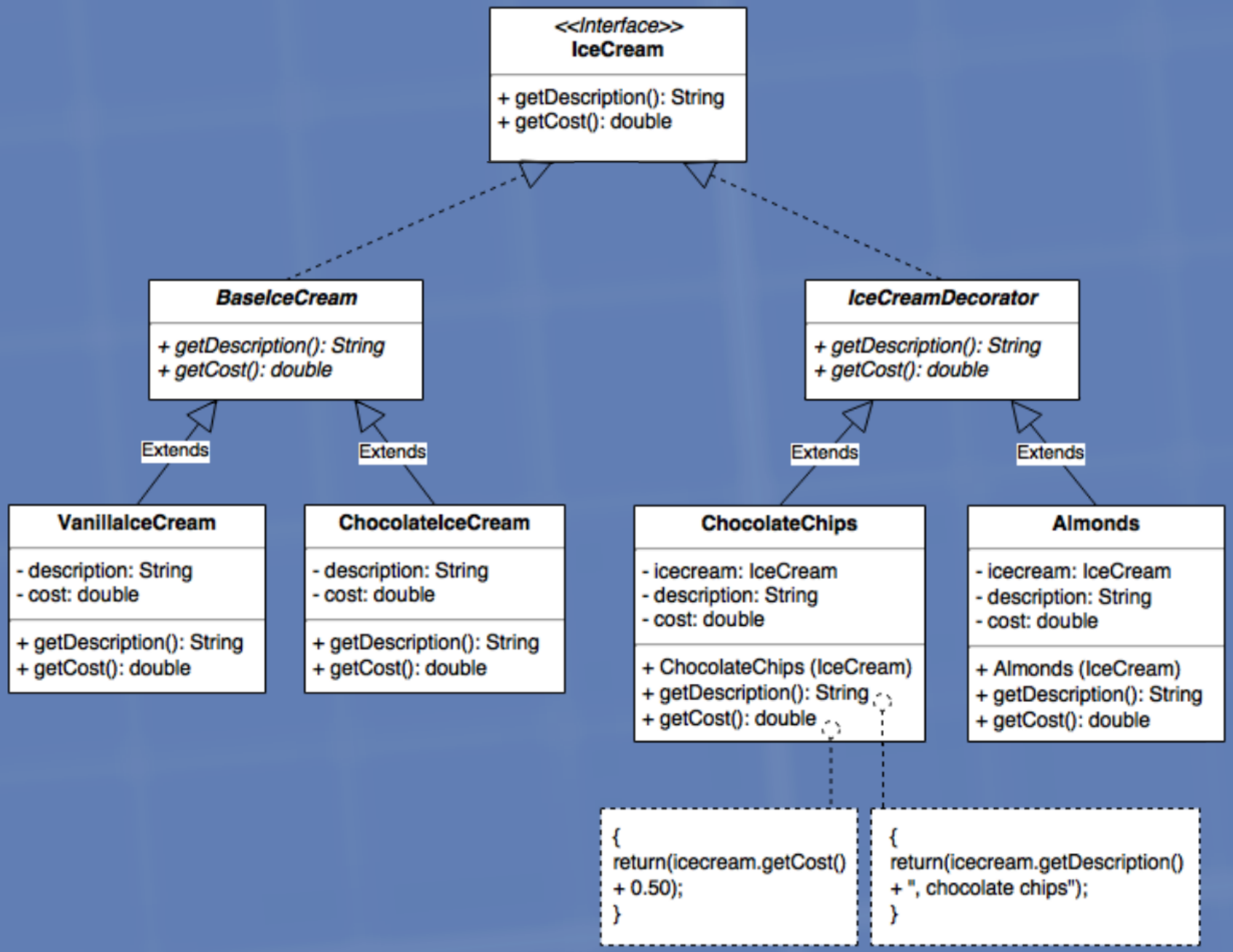


Decorator

Allows behavior to be added to *individual objects* of a type

Behavior can be added *dynamically*




<<Interface>>
IceCream
+ getDescription(): String
+ getCost(): double
BaseIceCream
+ getDescription(): String
+ getCost(): double
IceCreamDecorator
+ getDescription(): String
+ getCost(): double
Extends
Extends
Extends
Extends
VanillaIceCream
- description: String
- cost: double
+ getDescription(): String
+ getCost(): double
ChocolateIceCream
- description: String
- cost: double
+ getDescription(): String
+ getCost(): double
ChocolateChips
- icecream: IceCream
- description: String
- cost: double
+ ChocolateChips (IceCream)
+ getDescription(): String
+ getCost(): double
Almonds
- icecream: IceCream
- description: String
- cost: double
+ Almonds (IceCream)
+ getDescription(): String
+ getCost(): double
{
return(icecream.getCost()
+ 0.50);
}
{
return(icecream.getDescription()
+ ", chocolate chips");
}

We all scream for Ice cream!

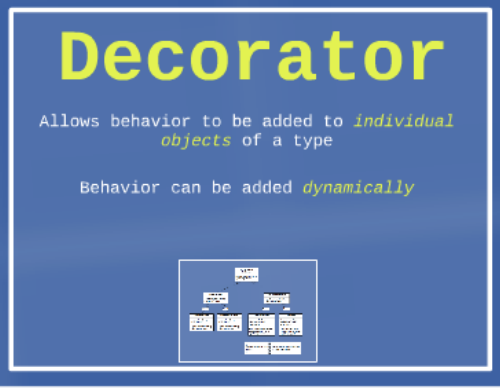
Choose base *flavor* & add one or more *mix-ins*

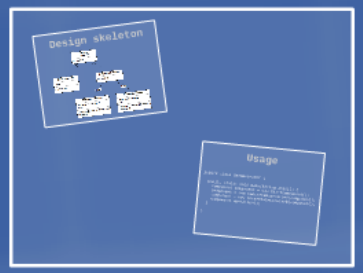
Different *flavors*/*mix-ins* can have different *costs*

Design software to produce full *description* & calculate final *cost*

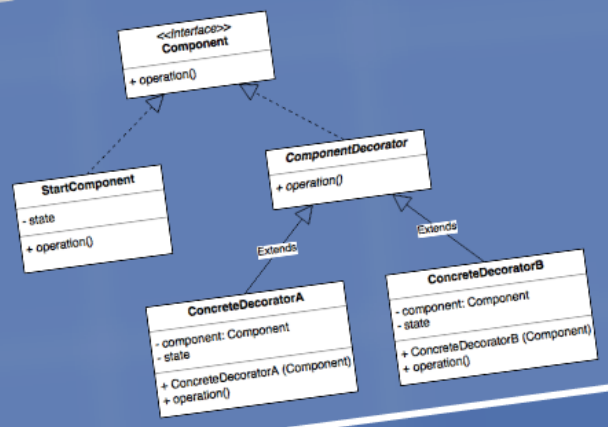
Design options?

- Single ice-cream class
- Combinations hierarchy





Design skeleton



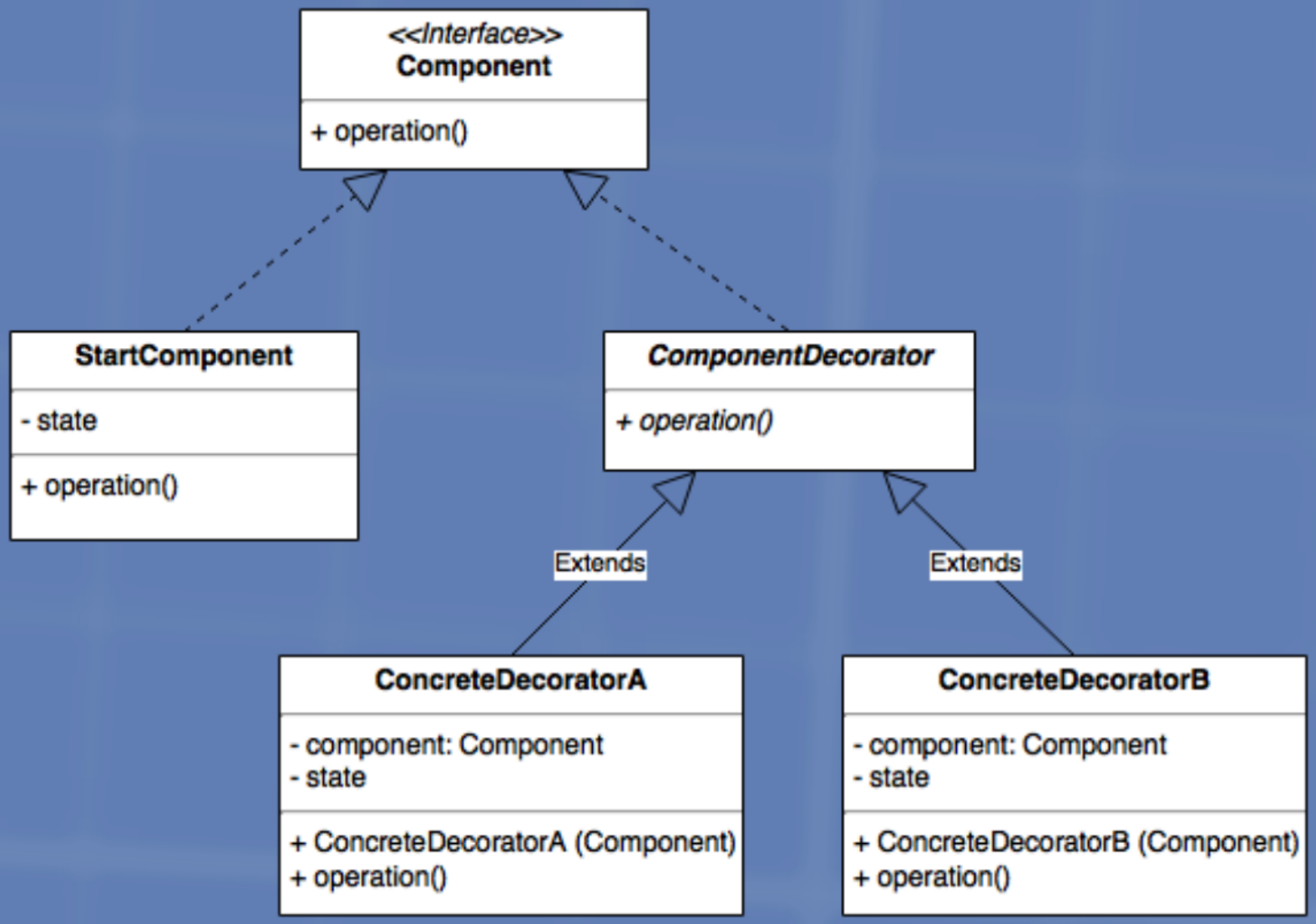
Usage

```
public class UseDecorator {

  public static void main(String args[]) {
    Component component = new StartComponent();
    component = new ConcreteDecoratorA(component);
    component = new ConcreteDecoratorB(component);
    component.operation();
  }

}
```

Design skeleton



Usage

```
public class UseDecorator {

  public static void main(String args[]) {
    Component component = new StartComponent();
    component = new ConcreteDecoratorA(component);
    component = new ConcreteDecoratorB(component);
    component.operation();
  }

}
```