

State Pattern

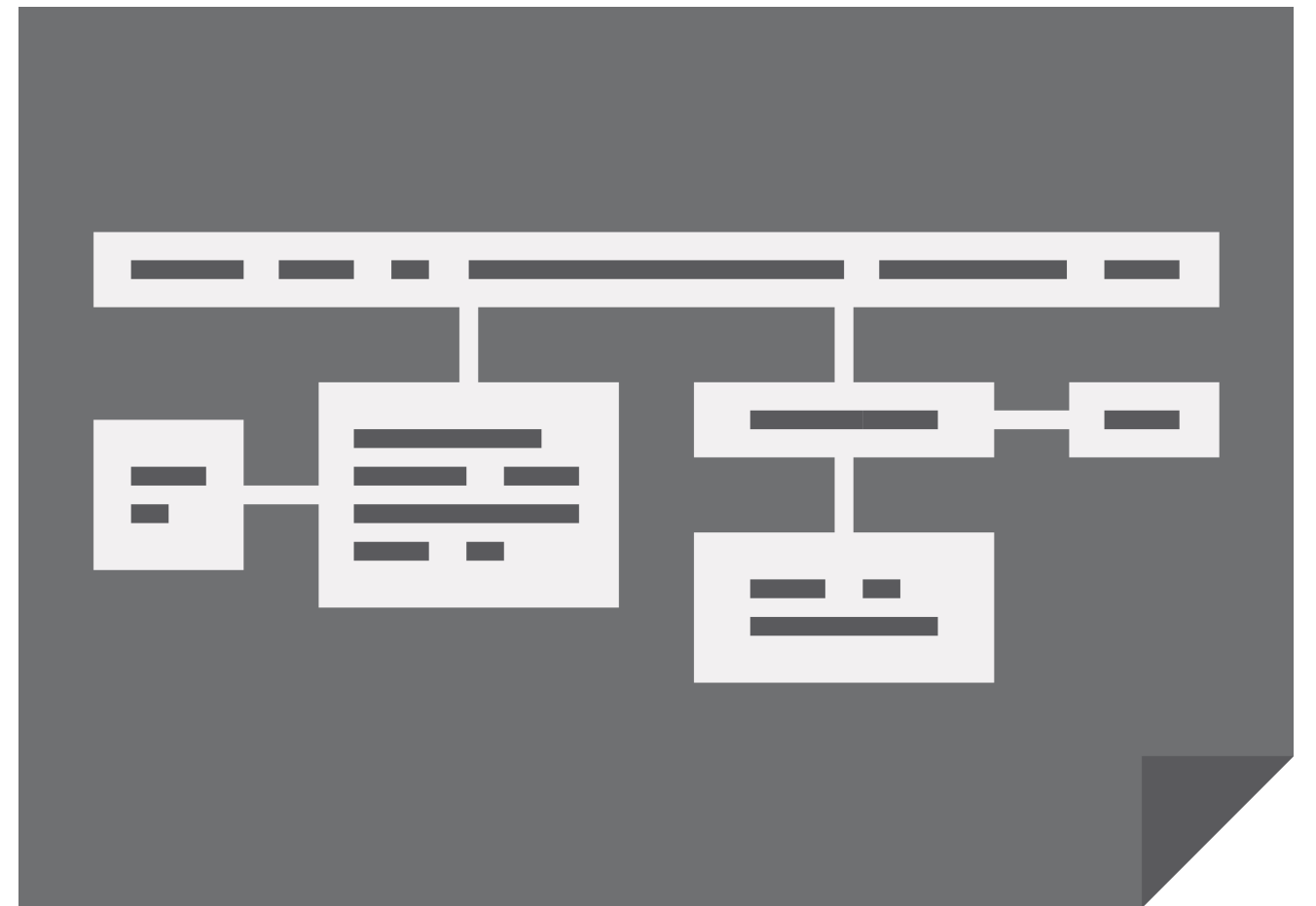


Bryan Hansen

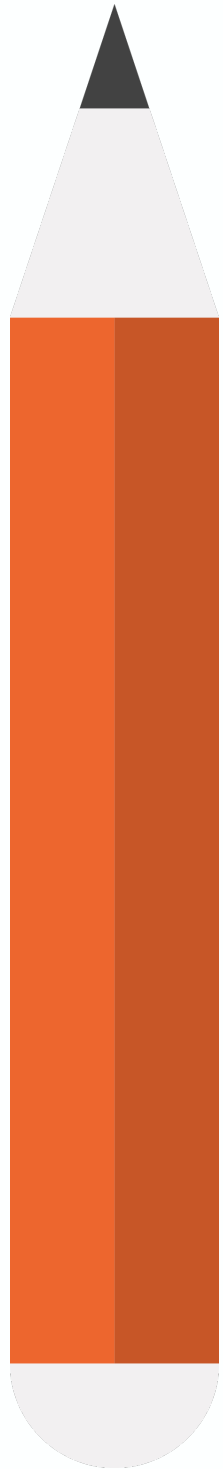
twitter: bh5k | <http://www.linkedin.com/in/hansenbryan>

Concepts

- Localize state behavior
- State Object
- Separates What from Where
- OCP
- Examples:
 - None!
 - JSF
 - Iterator



Design



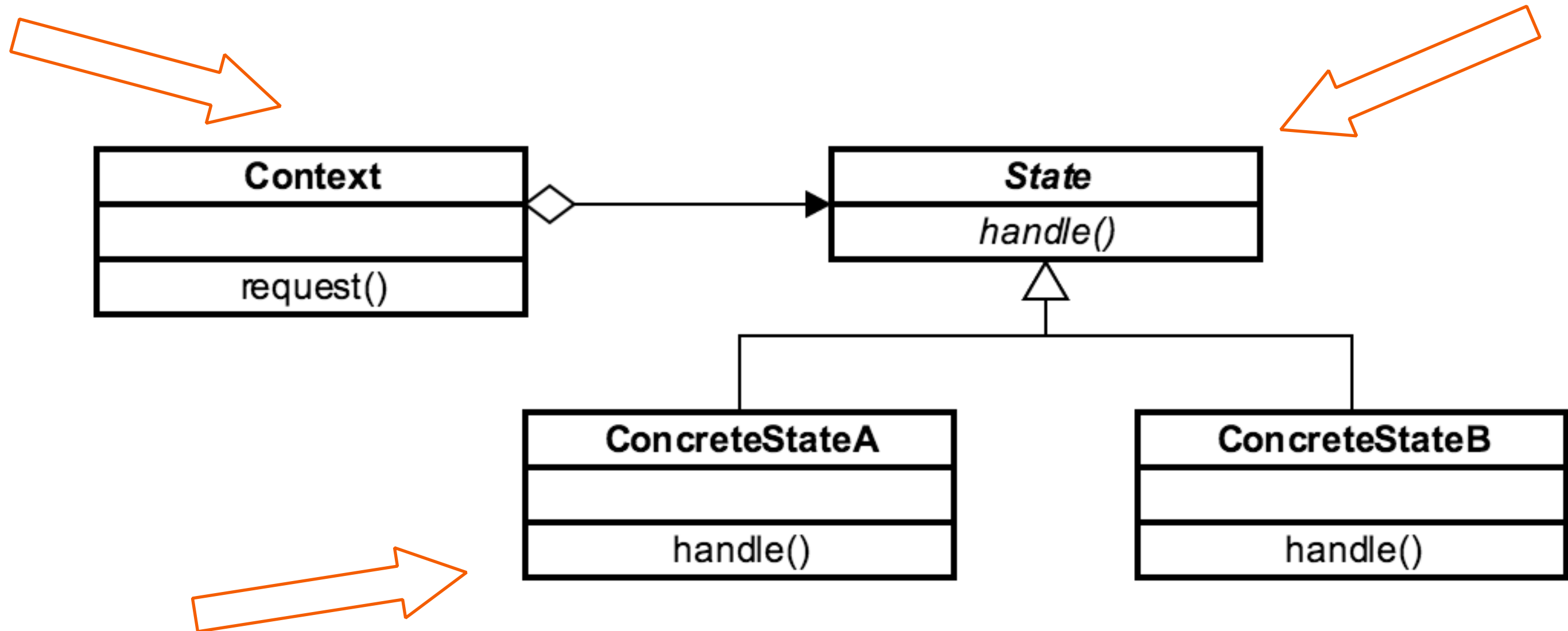
Abstract Class / Interface

Class based

Context unaware

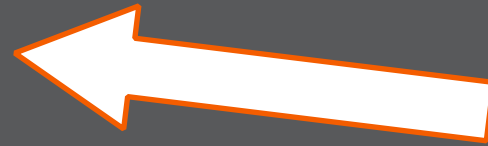
Context, State, ConcreteState

UML



Everyday Example - if/else

```
final static int ON = 0;  
final static int OFF = 1;  
int state = OFF;
```



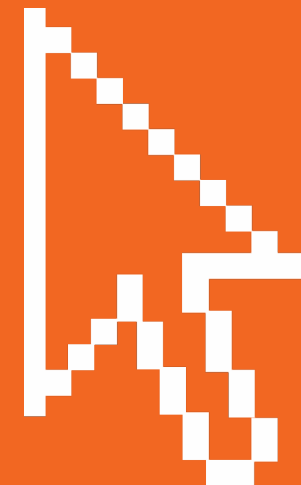
```
public void pullChain() {  
    if(state == ON) {  
        System.out.println("Fan is already on");  
    }  
    else if (state == OFF) {  
        System.out.println("Turning Fan on.");  
        state = ON;  
    }  
}
```

Exercise State

State

ConcreteState

Conditional Logic



Pitfalls

- Know your States
- More classes
- Keep logic out of Context
- State change triggers



Contrast

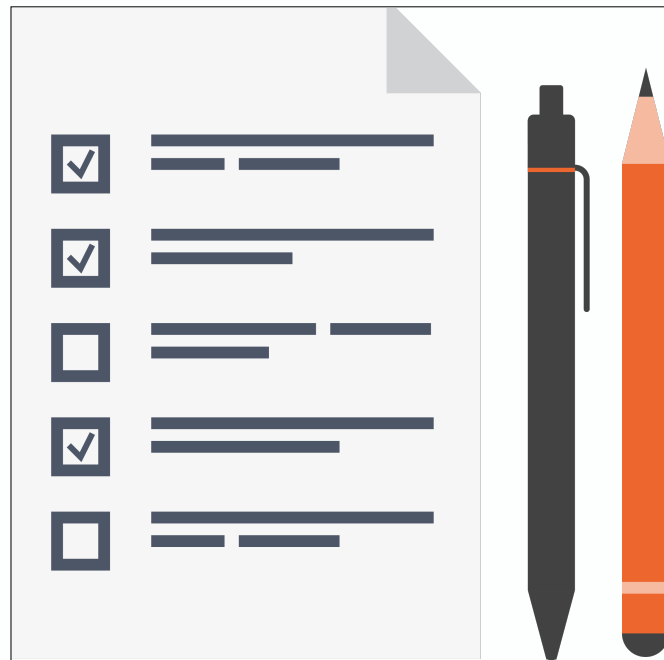
State

- Interface based
- Transitions
- Class per State

Strategy

- Interface based
- Algorithms are Independent
- Class per Algorithm

State Summary



- Simplifies cyclomatic complexity
- Adding additional states made easier
- More classes
- Similar implementation to Strategy