

Object-Oriented Versus Functional Programming; The Visitor Pattern

April 15, 2011

Functional vs OO

```
interface Animal {  
    String happyNoise();  
    String excitedNoise();  
}
```

```
class Dog implements Animal {  
    String happyNoise() { return "pant pant"; }  
    String excitedNoise() { return "bark"; }  
}
```

```
class Cat implements Animal {  
    String happyNoise() { return "purrrrrr"; }  
    String excitedNoise() { return "meow"; }  
}
```

Functional vs OO

```
class Chicken implements Animal {  
    String happyNoise() { return "cluck cluck"; }  
    String excitedNoise() { return "cockadoodledoo"; }  
}
```

Functional vs OO

```
interface Animal {
    String happyNoise();
    String excitedNoise();
    String angryNoise();
}

class Dog implements Animal {
    String happyNoise() { return "pant pant"; }
    String excitedNoise() { return "bark"; }
    String angryNoise() { return "grrrrr"; }
}

class Cat implements Animal {
    String happyNoise() { return "purrrrr"; }
    String excitedNoise() { return "meow"; }
    String angryNoise() { return "hissss"; }
}
```

Functional vs OO

```
datatype Animal = Dog | Cat ;
```

```
fun happyNoise(Dog) = "pant pant"  
  | happyNoise(Cat) = "purrrrr"
```

```
fun excitedNoise(Dog) = "bark"  
  | excitedNoise(Cat) = "meow"
```

Functional vs OO

```
fun angryNoise(Dog) = "grrrrr"  
  | angryNoise(Cat) = "hisssss"
```

Functional vs OO

```
datatype Animal = Dog | Cat | Chicken;
```

```
fun happyNoise(Dog) = "pant pant"  
  | happyNoise(Cat) = "purrrrr"  
  | happyNoise(Chicken) = "cluck cluck";
```

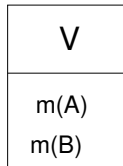
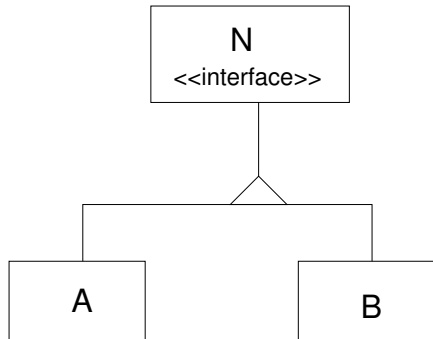
```
fun excitedNoise(Dog) = "bark"  
  | excitedNoise(Cat) = "meow"  
  | excitedNoise(Chicken) = "cockadoodledoo";
```

```
fun angryNoise(Dog) = "grrrrrr"  
  | angryNoise(Cat) = "hisssss"  
  | angryNoise(Chicken) = "squaaaack";
```

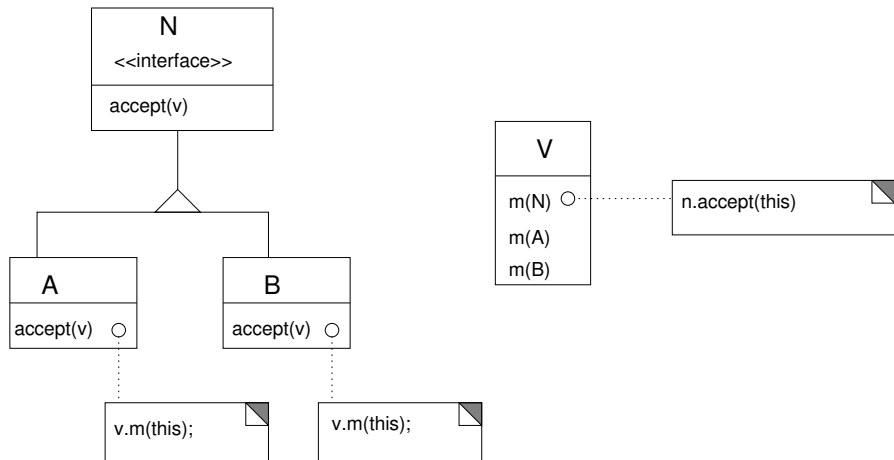
Functional vs OO

	Dog	Cat	Chicken
happyNoise	pant pant	purrrr	cluck cluck
excitedNoise	bark	meow	cockadoodledoo
angryNoise	grrrrr	hisssss	squaaaaack

Visitor: General problem



Visitor: Structure



Subtying the Visitors

