

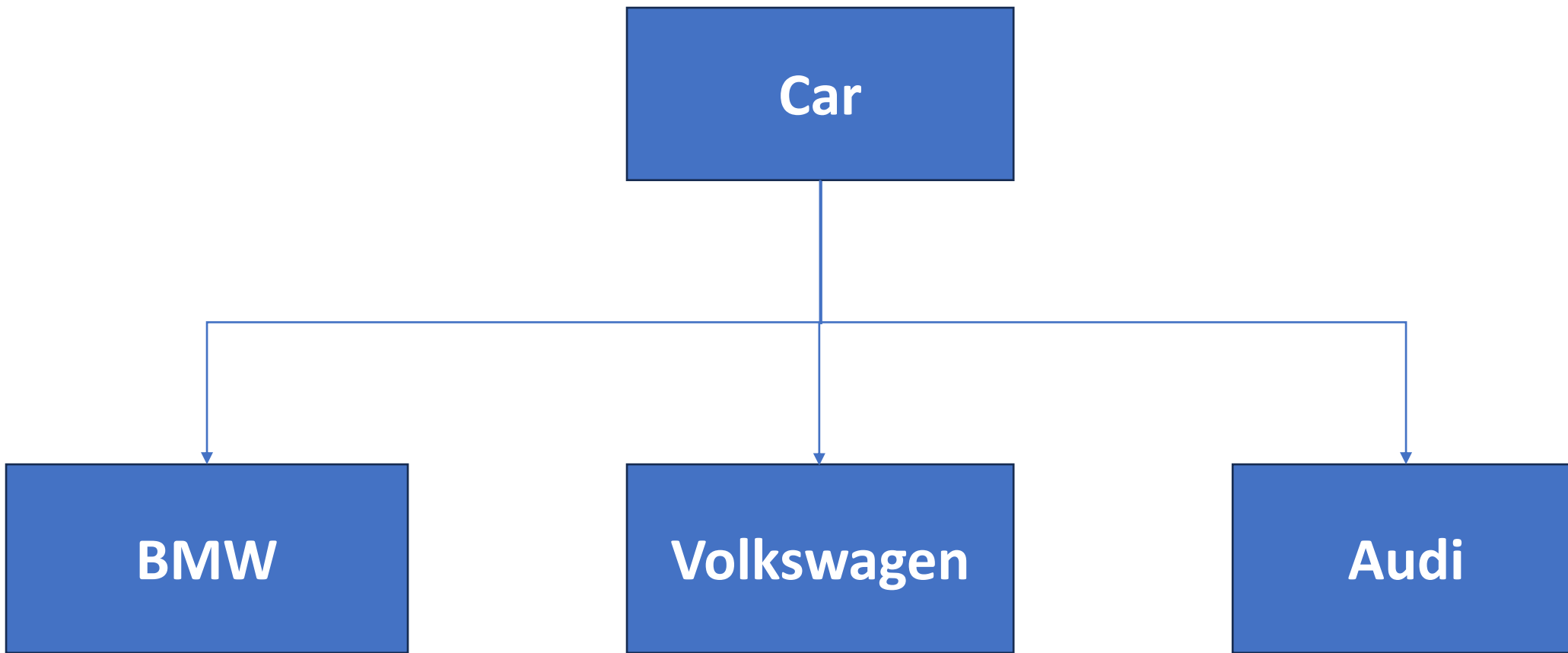


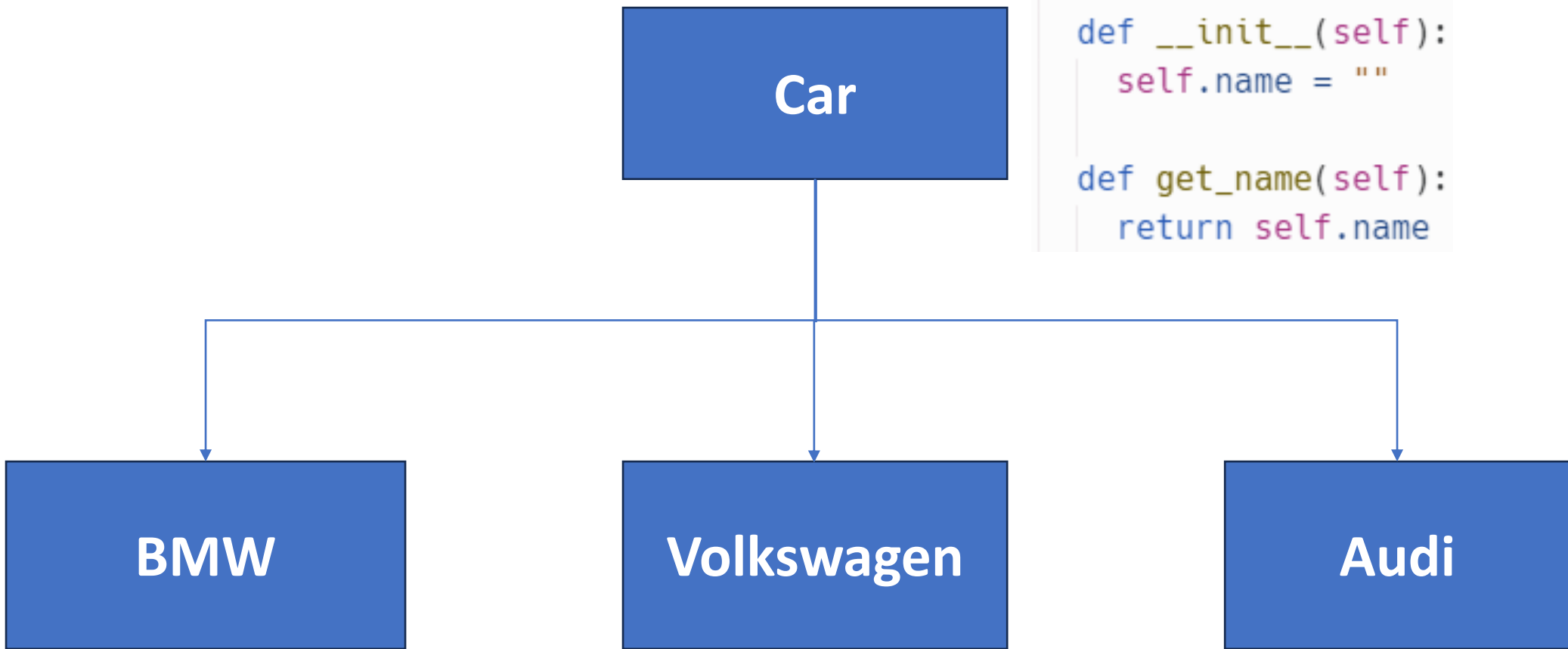
# Programming for Social Scientists

---

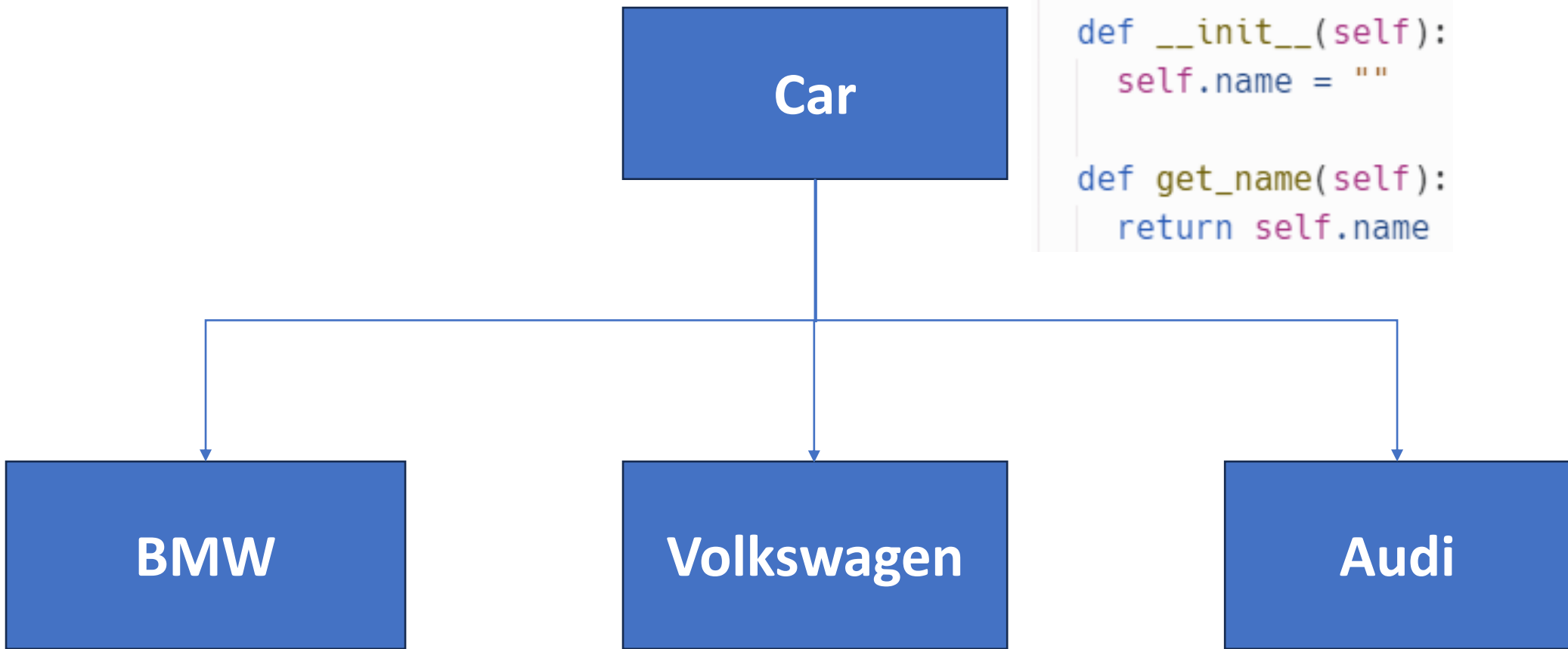
Johan A. Dornschneider-Elkink

# Inheritance





```
class Car:  
  
    def __init__(self):  
        self.name = ""  
  
    def get_name(self):  
        return self.name
```

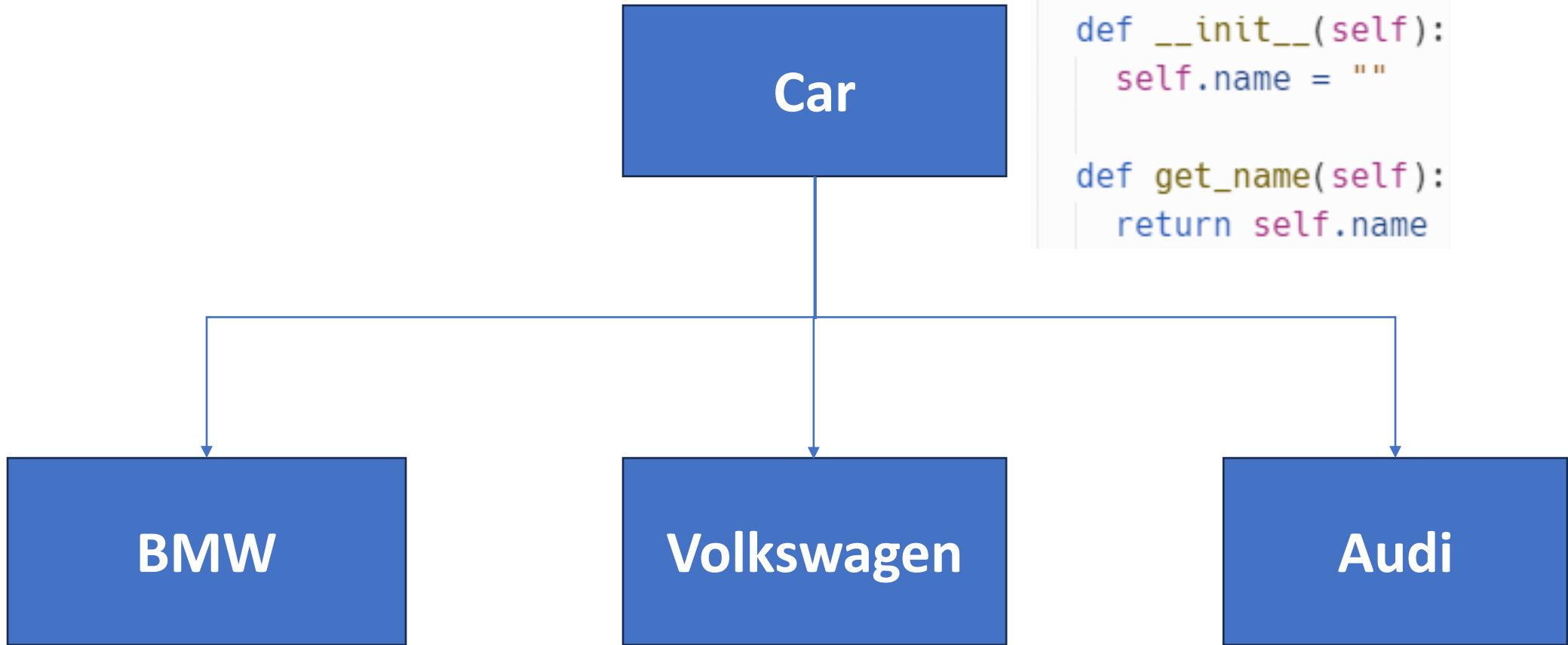


```
class Car:  
  
    def __init__(self):  
        self.name = ""  
  
    def get_name(self):  
        return self.name
```

```
class Volkswagen(Car):  
  
    def __init__(self):  
        self.name = "Volkswagen"
```

```
car = Volkswagen()  
print("This is a " + car.get_name())
```

```
class Car:  
  
    def __init__(self):  
        self.name = ""  
  
    def get_name(self):  
        return self.name
```



```
class Volkswagen(Car):  
  
    def __init__(self):  
        self.name = "Volkswagen"
```

```
class Car:

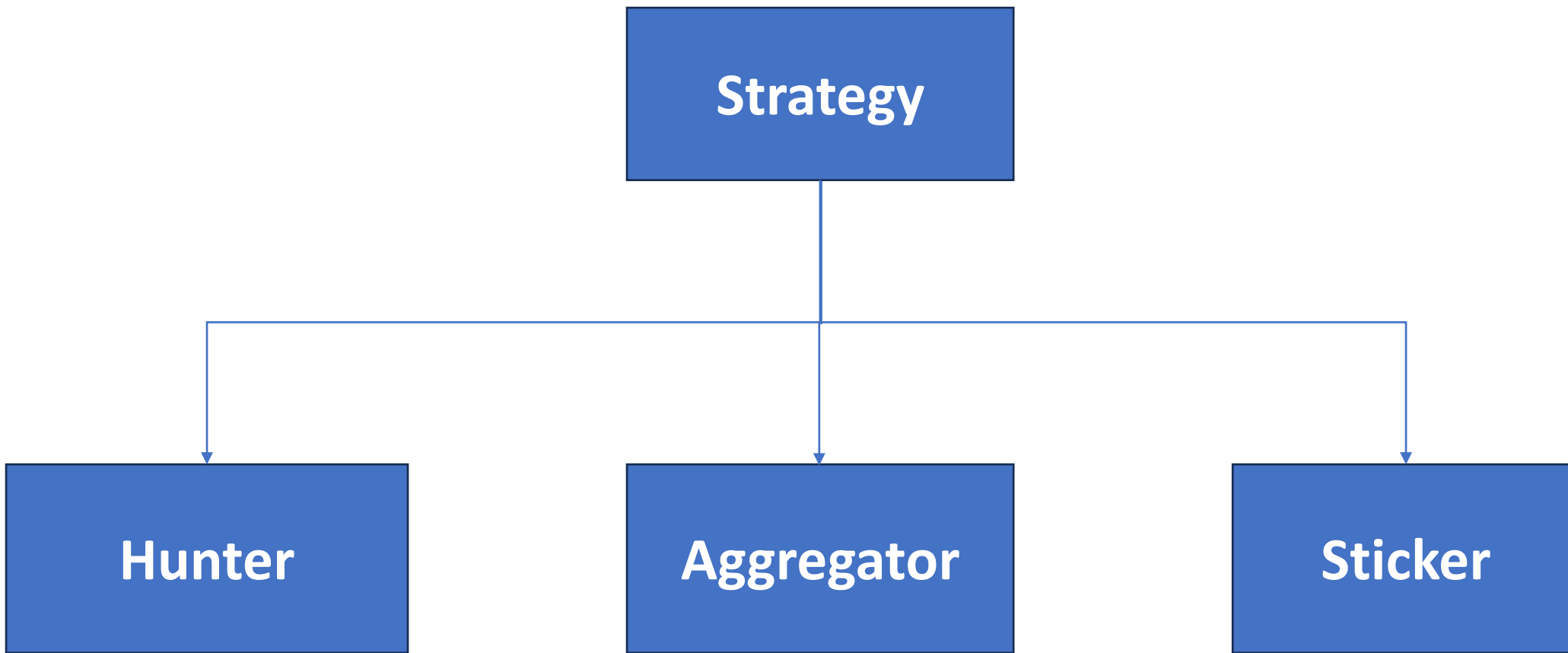
    def __init__(self):
        self.name = ""

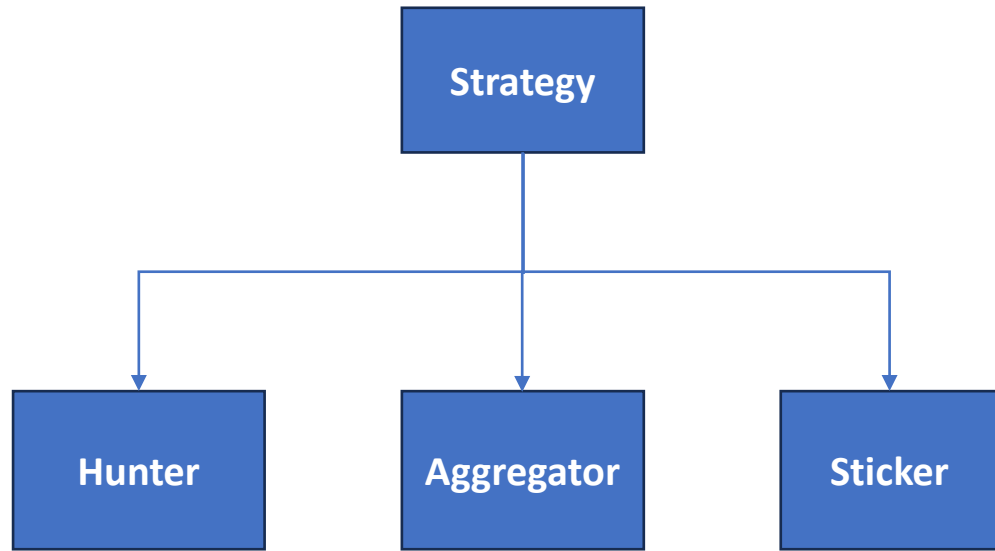
    def get_name(self):
        return self.name

class Volkswagen(Car):

    def __init__(self):
        super().__init__()
        self.name = "Volkswagen"
```

super() refers to the class from which you inherit, e.g. to call its constructor.





- Common interface (same methods for all strategies)
- Can implement strategies in separate files
- Can pass strategies as objects to Party objects
- Easy to add additional strategies later