Refactoring for Inheritance

Stocks Again

Let's revisit the StockModel and StockView classes:

```
>>> model = StockModel('AAPL', 176.18, 177.09, 154718, 2505047)
>>> view = StockView()
>>> view.render(model)
'AAPL: $177.09 (Bearish)'
```

Let's subclass to create views for other output formats.

JSON View

We may want to serve this data through an API endpoint.

Let's make a view that will render a JSON response body:

```
import json
class StockJSONView(StockView):
    def render(self, model):
        params = self.params(model)
        return json.dumps(params)
```

```
>>> model = StockModel('AAPL', 159.29, 163.05, 44035531, 22509937)
>>> view = StockJSONView()
>>> view.render(model)
'{"name": "AAPL", "price": 163.05, "sentiment": "Bullish"}'
```

HTML View

```
STOCK_HTML_TEMPLATE = '''
<html>
  <title>Stock Report for {name}</title>
  <body>
     <dl><dt>Name:</dt><dd>{name}</dd>
         <dt>Closing price:</dt><dd>{price}</dd>
         <dt>Assessment:</dt><dd>{sentiment}</dd>
     </dl></body></html>
'''.strip()

class StockHTMLView(StockView):
    def __init__(self, template):
        self.template = template

    def render(self, model):
        params = self.params(model)
        return self.template.format_map(params)
```

HTML View

```
>>> model = StockModel('FB', 172.06, 183.37, 76_670_183, 25219450)
>>> view = StockHTMLView(STOCK_HTML_TEMPLATE)
>>> print(view.render(model))
<html>
  <title>Stock Report for FB</title>
  <body>
     <dl><dt>Name:</dt><dd>FB</dd>
         <dt>Closing price:</dt><dd>183.37</dd>
     <dt>Assessment:</dt><dd>Bullish<dd></dd>
</dl></body</html>
```

Polymorphism

The `render()` method produces different output, depending on whether you invoke it on an instance of `StockView`, `StockHTMLView` or `StockJSONView`.

But the signature and kind of result is the same.

This is called polymorphism. It means the same operation (the `render()` method) is available in each, customized to the type.

Class Diagram