Tutorial 8 – Factory Patterns

Exercise 1

Refer to the Pizza case study in chapter 4 of "Head First Design Patterns". You are required to implement patterns for a similar scenario

A Café in NZ and another one in OZ sell coffee. Select two types of coffee you would like to implement from Cappuccino, Latte, Flat White or Mocha for the two cafes .
Since there is virtually no real behavior to implement, define only `Brew()` and `Pour()` as methods inside the `Cafe` class, make these methods return strings.

(a). The BREW() for a Cappuccino style coffee should return
NZ Café : "I am brewing coffee for making a Cappuccino using NZ chocolate crema"
OZ Café : "I am brewing Aussie coffee for making a Cappuccino"
Similarly, declare BREW() methods for Latte, Flat White and Mocha.
(b). The `Pour()` method for all coffee types return " I am pouring the coffee"

Name the classes you have made for:

a. Product: _____

b. ConcreteProduct: _____

c. Creator: _____

d. ConcreteCreator: _____

Below are the class diagrams from the referenced book chapter:

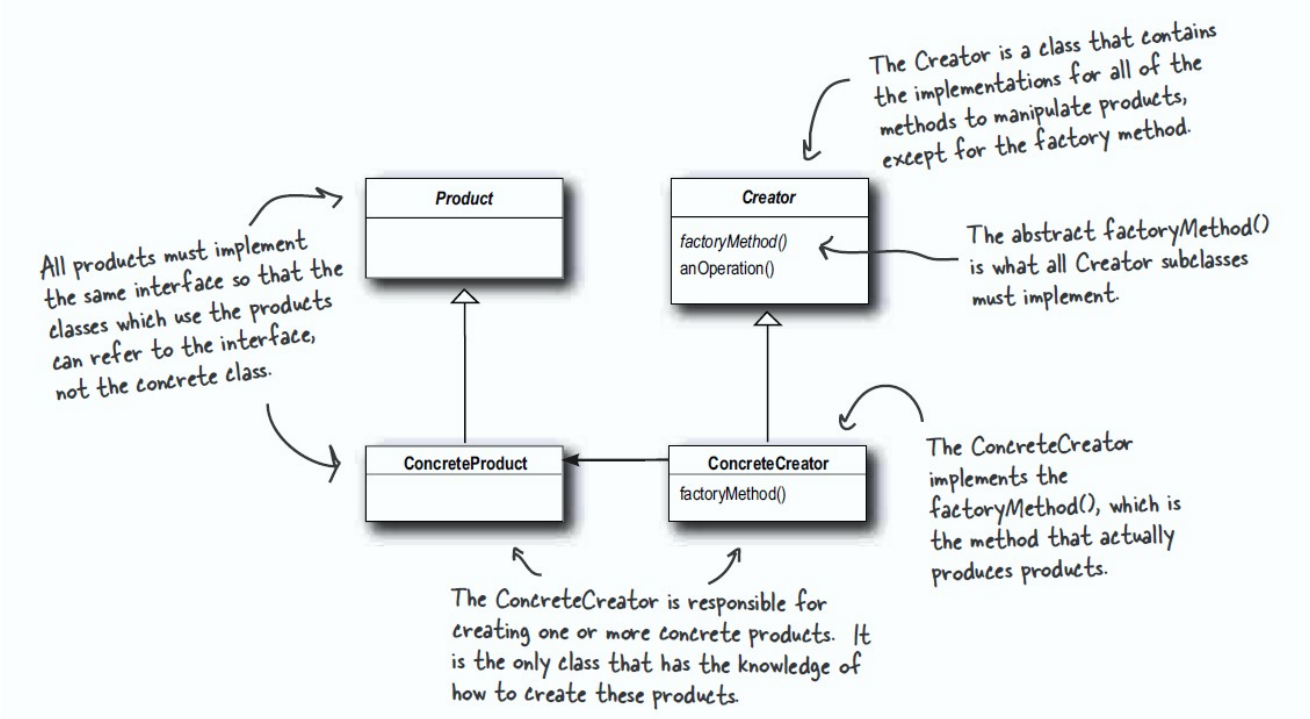

Image from "Head First Design Patterns" by Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra

The Pizza example from the book is given as reference:

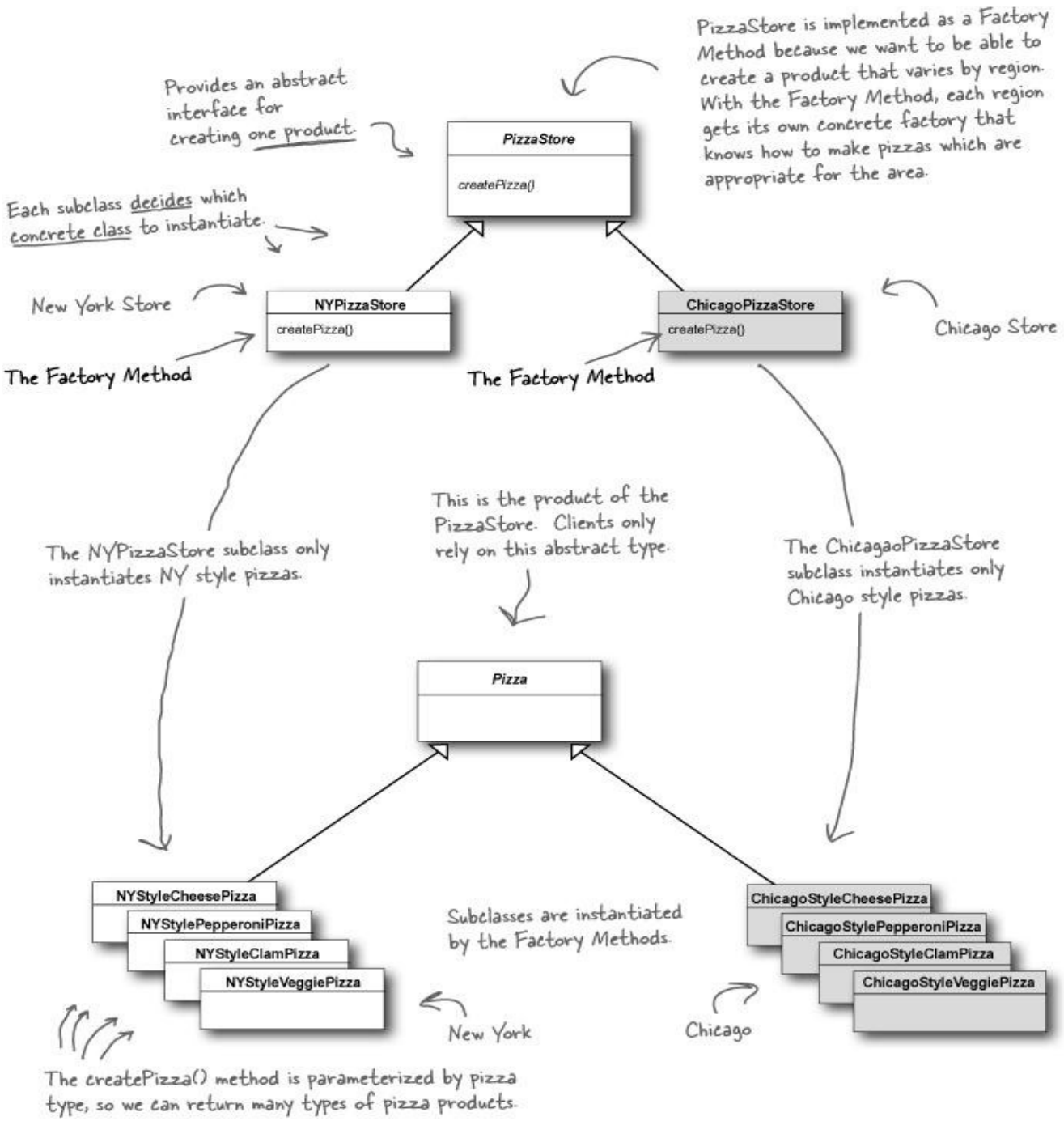

Image from “Head First Design Patterns” by Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra

Exercise 2

Your task now is to extend exercise 1 and implement the **Abstract Factory Pattern** based on the example in the book (refer to pages 144-156 of the book). One of the key extensions to the previous project which you will have to implement is an additional factory (abstract factory) responsible for creating families of ingredients objects. Again to simplify, in the ingredients factory work only with two types of ingredients for the coffee – `Milk` and `CoffeeBeans`. They will be fields of the abstract `Coffee` class. For each of the `Milk` and `CoffeeBeans` ingredients, create two types of concrete classes i.e.

(a). `IMIlk (has classes LowFatMilk and FullCreamMilk)`

(b). `ICoffeeBeans` has classes (`RegularCoffeeBeans and DecaffeinatedCoffeeBeans)`

Further,

(a). NZ Café : uses ingredients `DecaffeinatedCoffeeBeans` and `FullCreamMilk`

(b). OZ Café : uses ingredients `RegularCoffeeBeans` and `LowFatMilk`

Therefore,

(a). The BREW( ) `for` a Cappuccino style coffee should return

NZ Café : "Brewing coffee for making a Cappuccino using decaffeinated coffee beans and full cream milk"

OZ Café : "Brewing coffee for making a Cappuccino using regular coffee beans and low cream milk"

Similarly, declare BREW( ) methods for Latte, Flat White and Mocha.

(b). The `Pour( )` method for all returns " I am pouring the coffee"

How many classes have you made: _____. Name them below

AbstractFactory: _____

ConcreteFactory: _____

AbstractProduct: _____

ConcreteProduct: _____

Client: _____

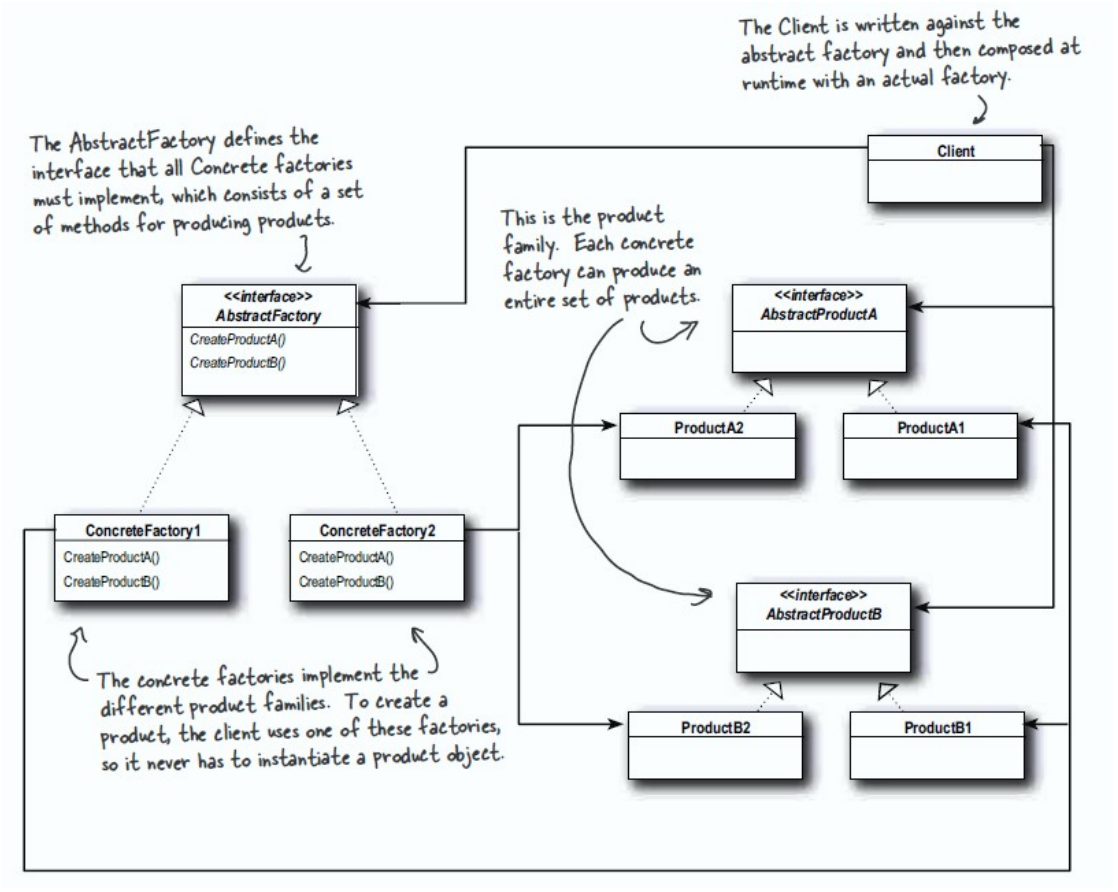

Image from "Head First Design Patterns" by Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra

The Pizza example from the book is given as reference:

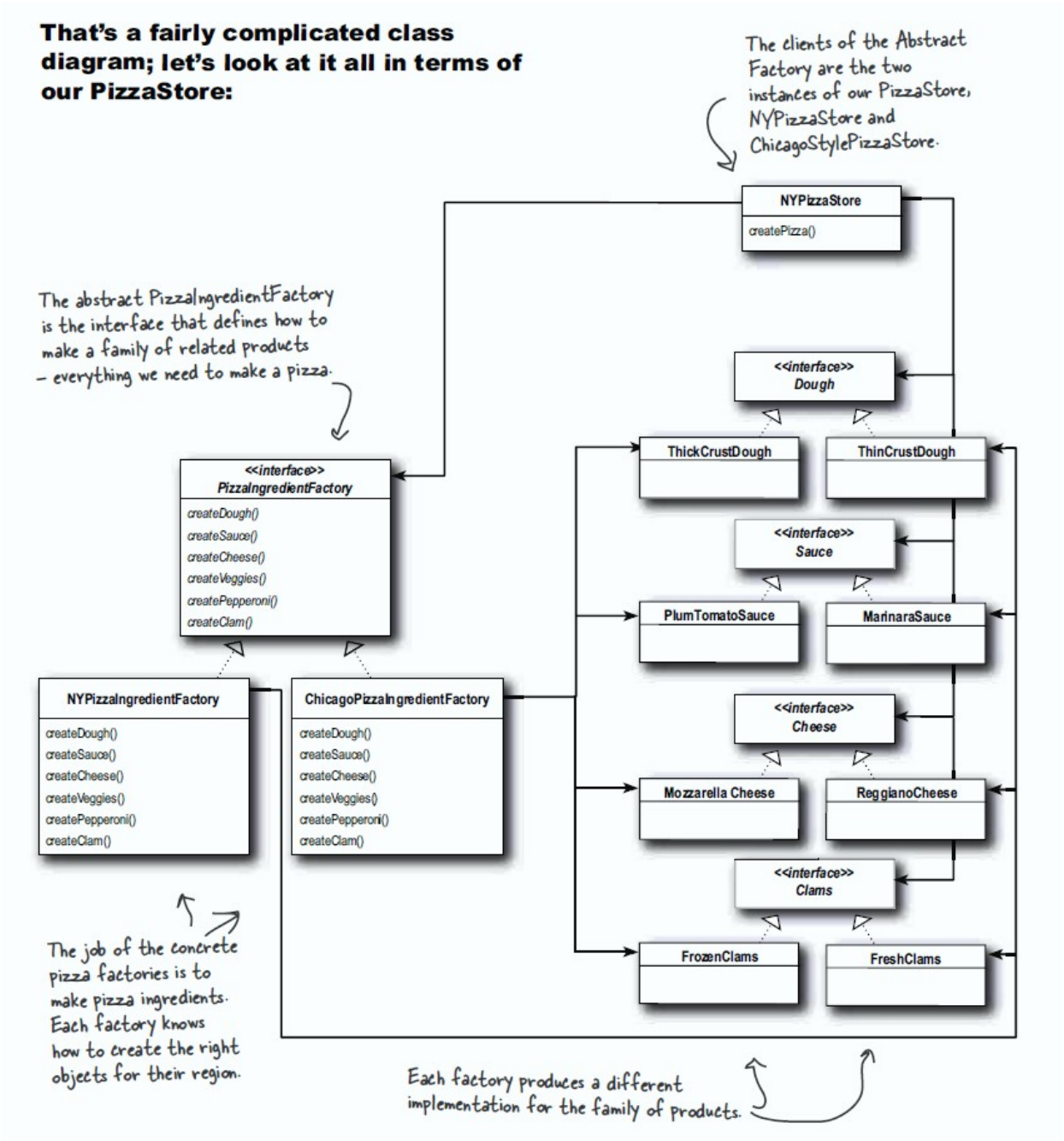

Image from "Head First Design Patterns" by Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra

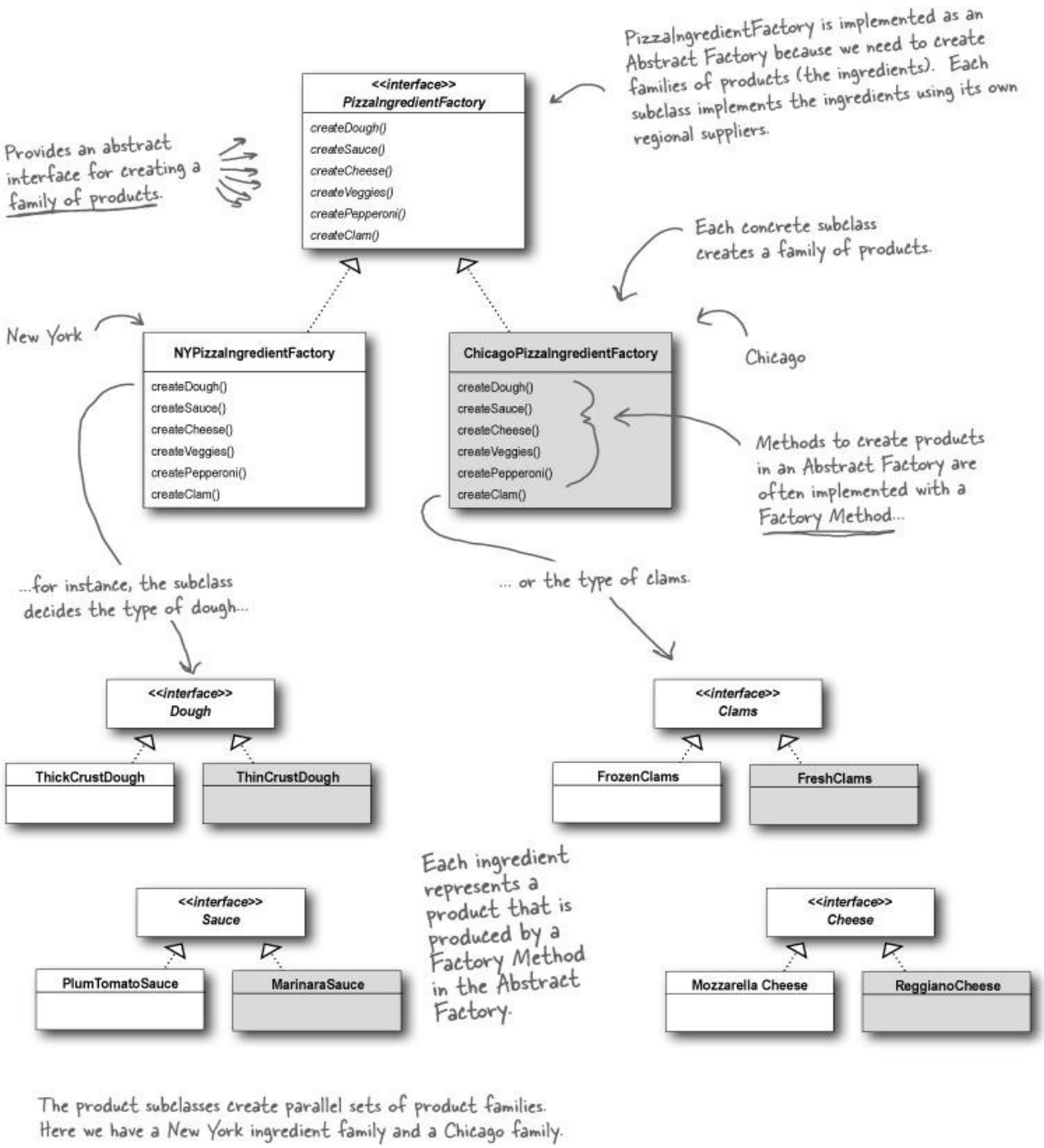

Image from “Head First Design Patterns” by Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra