When and How to Reduce a Factory Designed by the Factory Method Pattern to Another Factory That Works as a Simple Factory Pattern

Chen Junhua, Chen Yiyang, Xu Bo

The usage scenario of the **simple factory pattern** is as clear as his naming. When an object needs to be manufactured, the user does not need to care about the way it is used. All he needs is to use the object. There are many stores in a city. They sell many of the same things. Creating a **factory** to produce these goods is obviously the most **efficient way** of doing things. The owner of a store does not need to be too concerned about anything other than sales, and the factory only needs to do its best to provide all the products he can produce.

Simple factory model is a better choice when some product types are **limited** and consumers do not have much personal needs. As a computer student busy with programming, we only require the school canteen to provide hamburgers for lunch, but we **don't care about** the brand of hamburgers, or we insist on eating only KFC or McDonald's hamburgers. In this case, the simple factory model can well meet our needs. We will not be able to whimper and eat all kinds of hamburgers in Burger King, and it is not realistic in the real world.

But for a customer who is very **critical** of Hamburg, the simple factory model is far from meeting the standards in his mind. He is familiar with all kinds of hamburgers in the world and even has his own opinion about hamburger production. If a hamburger shop wants to attract such customers, he can't just go to a large scale hamburger factory to buy hamburgers on a large scale. He needs to build **his own factory**, preferably one with high flexibility. This factory of its own can mass produce hamburgers that customers often buy, but leave room for adding new elements to **existing manufacturing processes**.

In conclusion, when we can simplify the factory method depends on **the**

characteristics of the goods and the needs of the users. If the quantity of this product is **not large** and will **not change** much in the future, we might as well simplify it into a simple factory model. When a student enters a university and faces the choice of subjects, the categories of majors are complicated. We need a factory method model to produce ”majors”. But for a student who is determined to learn computer-related majors, his choice is limited and will not change much in the short term. We can design a simple factory model for him. **In this case, we can even use the computer-related majors designed by the ”sub-factory” under the factory method pattern directly for the simple factory pattern**.