Builder Pattern

Yunrui Huang
12/01/2024

Why we need Builder

When we build a complex object, the construction process often becomes difficult to manage. The Builder pattern would help us to separates the construction process in an easy way.

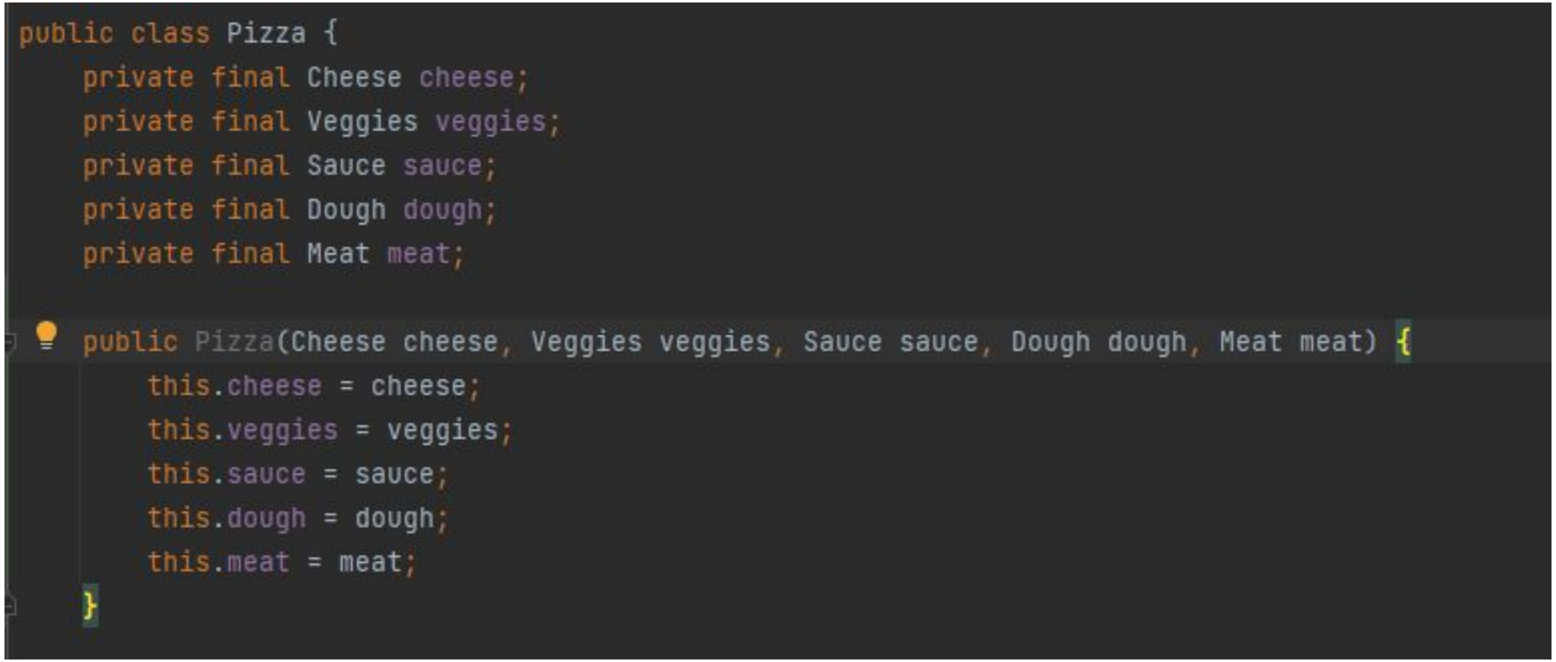
What do we want to achieve

- Simplify the Object Construction
  - Break down the construction process in small step
- Improve Parameter Management
  - Allow parameter be mandatory and optional
  - Easily to add or change
- Reduce Unused Parameter
  - Avoid to write null for unused parameter

Builder Pattern

The Builder design pattern in Java, a fundamental creational pattern, allows for the step-by-step construction of complex objects. It separates the construction of a complex object from its representation so that the same construction process can create different representations

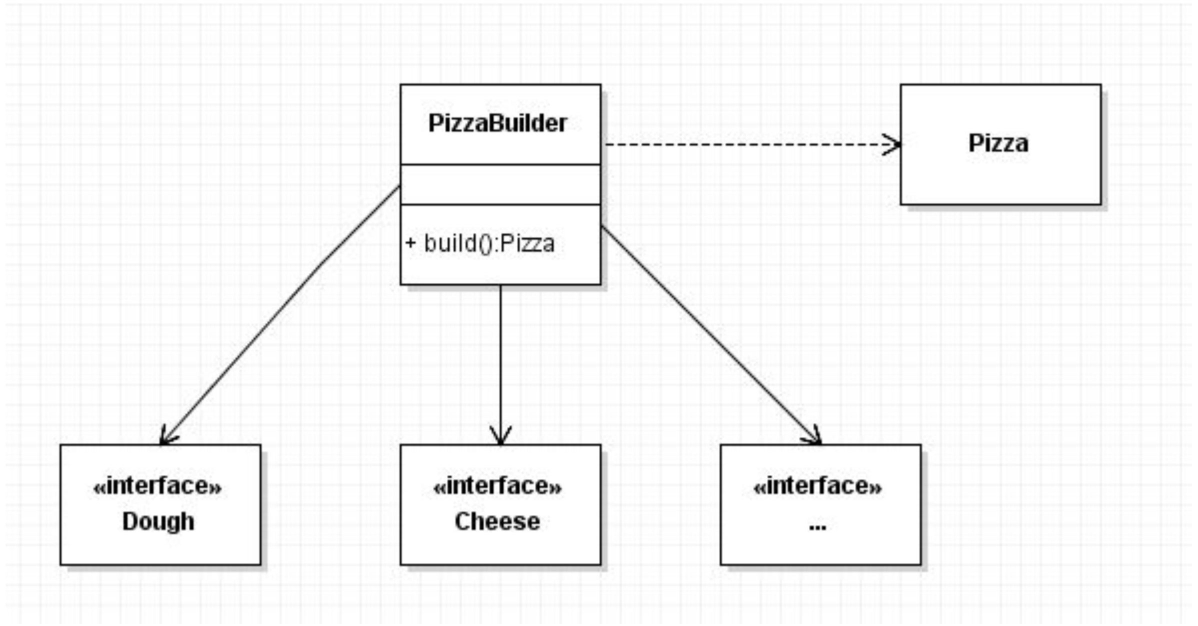
Example



```
public class Pizza {
    private final Cheese cheese;
    private final Veggies veggies;
    private final Sauce sauce;
    private final Dough dough;
    private final Meat meat;

    public Pizza(Cheese cheese, Veggies veggies, Sauce sauce, Dough dough, Meat meat) {
        this.cheese = cheese;
        this.veggies = veggies;
        this.sauce = sauce;
        this.dough = dough;
        this.meat = meat;
    }
```

Builder Pattern UML



Show Code

When to Use

- Builder Pattern is ideal for complex object creation
- The algorithm for creating a complex object should be independent of the parts that make up the object and how they're assembled
- The construction process must allow different representations for the object that's constructed
- It's particularly useful when a product requires a lot of steps to be created and when these steps need to be executed in a specific sequence

Pros and Cons

Pros of the Builder Pattern

- Simplifies complex object creation
- Improves code readability
- Handles optional parameters

Pros and Cons

Cons of the Builder Pattern

- May overcomplicate simple scenarios
- Builder class increase complex
- Less flexible to construct the object

Related Pattern

- Abstract Factory
- Template Method

Thanks