链滴

设计模式之建造者模式

作者：sologxl

原文链接：https://ld246.com/article/1541745493839

来源网站：链滴

今天来搞一搞建造者模式，本猿这次手动笔记本写代码，讲解一下所谓的建造者模式，主要是表达一对该模式的喜爱。希望大家也能爱上这个模式。

我们用代码来举个例子：

比如我们去麦当劳，需要点个套餐。我们可以这么做

```
//我们先创建一个套餐类
public class Meal{

  private String foods;
  private String drink;

  public void setfoods(String foods){
    this.foods = foods;
  }
  public void setdrink(String drink){
    this.drink = drink;
  }
  public void getdrink(){
    this.foods = foods;
  }
  public void getdrink(){
    this.drink = drink;
  }
}

//然后我们继续创建一个套餐构造类

public abstract class MealBuilder{
  Meal meal = new Meal();

  public abstract void buildfoods();

  public abstract void builddrink();

  public Meal getMeal(){
    return meal;
  }
}

//接下来我们要实现套餐构造类
public class MealA extends MeadBuilder{
  @Overrion
  public void buildfoods(){
    meal.setfoods("薯条")
  }
  @Overrion
  public void builddrink(){
    meal.setdrink("可乐")
  }
}
```

原文链接：设计模式之建造者模式

```
//最后我们创建一个建造者
//建造者模式名称就是这么来的
public class Builder{
  private MealBuilder mealBuilder;

  public void setMealBuilder(MeanlBuilder mealbuilder){
    this.mealBuilder = mealbuilder;
  }

  public Meal getMeal(){
     mealBuilder.setfoods();
     mealBuilder.setdrink();

     return mealBuilder.getMeal();
  }

}

//测试使用建造者
public static void main（String[] args）{

    MealA meala = new MealA();

    Waiter waiter = new Waiter();

    waiter.setMealBuilder(meala);

    Meal meal = waiter.getMeal();

    System.out.println("foods:" + meal.getfoods);
}
```

从测试例子可以看出，开发者最后只需要定义套餐和服务员（即建造者），然后让建造者去建造套餐可，全程无需知道底层操作，只需调用和命令做事即可。

其实很多时候，我们在开发就是要这种封装思维，无论是工厂模式还是建造者模式，都是为了方便高调用，因为这样就无需知道具体底层而提高开发效率和代码结构清晰