

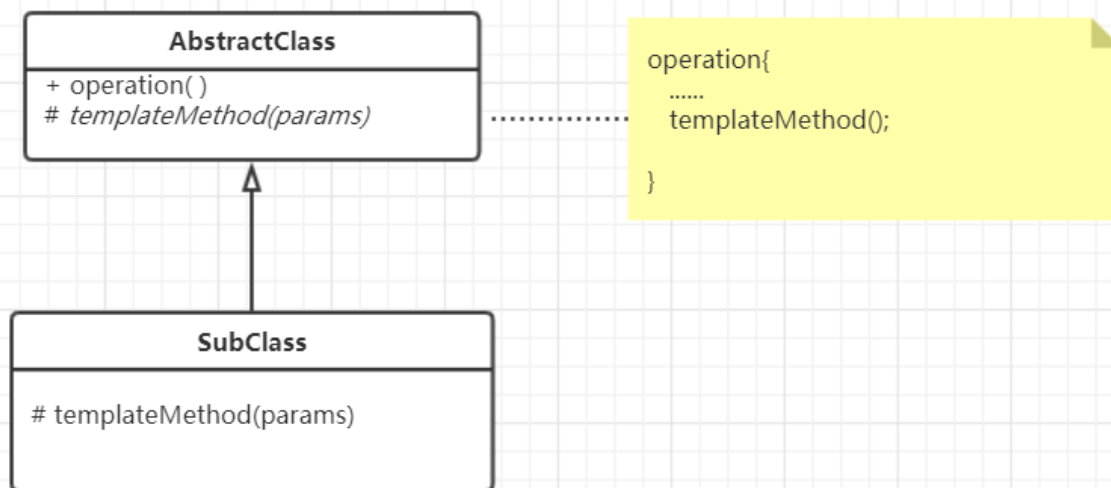
# 模板方法模式

## Template Method

郭嘉

模式定义：

定义一个操作的算法骨架，而将一些步骤延迟到子类中。Template Method 使得子类可以不改变一个算法的结构即可重定义该算法的某些特定步骤



## 案例

```
1 package com.tuling.designpattern.templateMethod;
2
3 /**
4  * @author 腾讯课堂-图灵学院 郭嘉
5  * @Slogan 致敬大师，致敬未来的你
6  */
7 public class TemplateMethodTest {
8     public static void main(String[] args) {
9         AbstractClass abstractClass=new SubClass1();
10        abstractClass.operation();
11    }
12 }
13
14
15 abstract class AbstractClass{
16
17     public void operation(){
18         // open
19         System.out.println(" pre ... ");
20
21         System.out.println(" step1 ");
22
23         System.out.println(" step2 ");
24
25         templateMethod();
```

```

26  // ....
27
28
29  }
30  abstract protected void templateMethod();
31
32  }
33
34  class SubClass extends AbstractClass{
35
36      @Override
37      protected void templateMethod() {
38          System.out.println("SubClass executed . ");
39      }
40  }
41
42  class SubClass1 extends AbstractClass{
43
44      @Override
45      protected void templateMethod() {
46          System.out.println("SubClass1 executed . ");
47      }
48  }

```

## 应用场景

1. 当你想让客户端只扩展算法的特定步骤，而不是整个算法或其结构时，请使用Template Method模式。
2. 当你有几个类包含几乎相同的算法，但有一些细微的差异时，请使用此模式。

## 优点：

1. 你可以让客户端只覆盖大型算法的某些部分，从而减少算法其他部分发生的更改对它们的影响。

2. 你可以将重复的代码拖放到超类中。

## Servlet Api & Spring 中的应用

```
1  
2 javax.servlet.http.HttpServlet  
3 org.springframework.web.servlet.mvc.AbstractController
```