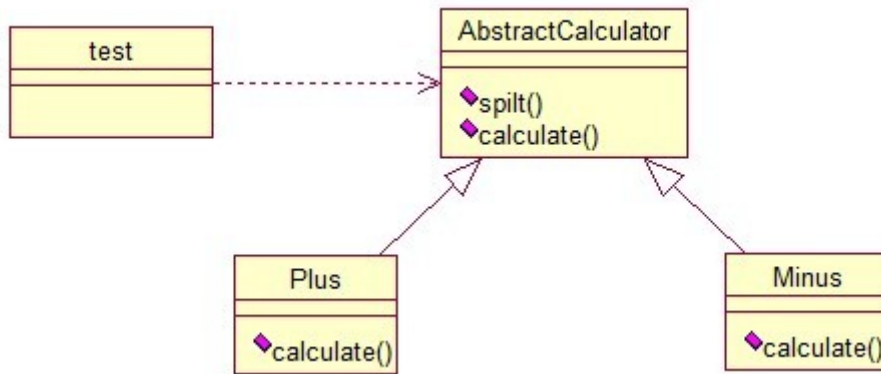


模板方法模式 (Template Method)

解释一下模板方法模式，就是指：一个抽象类中，有一个主方法，再定义1...n个方法，可以是抽象的，也可以是实际的方法，定义一个类，继承该抽象类，重写抽象方法，通过调用抽象类，实现对子类的调用，先看个关系图：



就是在AbstractCalculator类中定义一个主方法calculate，calculate()调用spilt()等，Plus和Minus分别继承AbstractCalculator类，通过对AbstractCalculator的调用实现对子类的调用，看下面的例子：

[java] [view plaincopy](#)

```
1. public abstract class AbstractCalculator {
2.
3.     /*主方法，实现对本类其它方法的调用*/
4.     public final int calculate(String exp,String opt){
5.         int array[] = split(exp,opt);
6.         return calculate(array[0],array[1]);
7.     }
8.
9.     /*被子类重写的方法*/
10.    abstract public int calculate(int num1,int num2);
11.
12.    public int[] split(String exp,String opt){
13.        String array[] = exp.split(opt);
14.        int arrayInt[] = new int[2];
15.        arrayInt[0] = Integer.parseInt(array[0]);
16.        arrayInt[1] = Integer.parseInt(array[1]);
17.        return arrayInt;
18.    }
19. }
```

[java] [view plaincopy](#)

```
1. public class Plus extends AbstractCalculator {
2.
3.     @Override
```

```
4.     public int calculate(int num1,int num2) {  
5.         return num1 + num2;  
6.     }  
7. }
```

测试类:

[java] [view plaincopy](#)

```
1. public class StrategyTest {  
2.       
3.     public static void main(String[] args) {  
4.         String exp = "8+8";  
5.         AbstractCalculator cal = new Plus();  
6.         int result = cal.calculate(exp, "\\+");  
7.         System.out.println(result);  
8.     }  
9. }
```

我跟踪下这个小程序的执行过程：首先将exp和"\\+"做参数，调用AbstractCalculator类里的calculate(String,String)方法，在calculate(String,String)里调用同类的split()，之后再调用calculate(int ,int)方法，从这个方法进入到子类中，执行完return num1 + num2后，将值返回到AbstractCalculator类，赋给result，打印出来。正好验证了我们开头的思路。