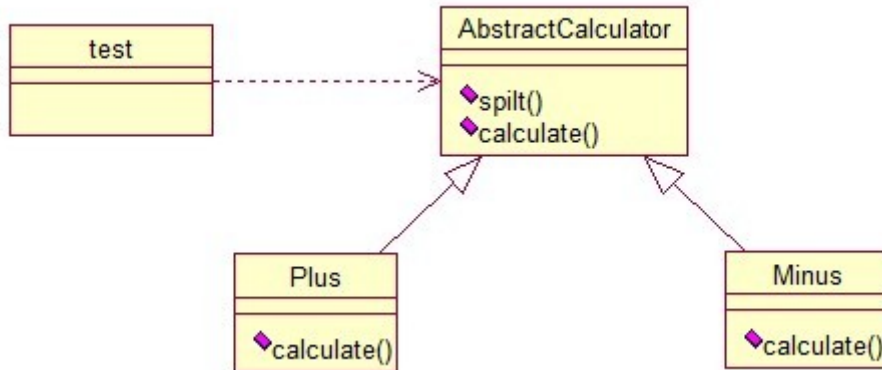


解释一下模板方法模式，就是指：一个抽象类中，有一个主方法，再定义1...n个方法，可以是抽象的，也可以是实际的方法，定义一个类，继承该抽象类，重写抽象方法，通过调用抽象类，实现对子类的调用，先看个关系图：



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](#)

```
public abstract class AbstractCalculator {

    /*主方法，实现对本类其它方法的调用*/

    public final int calculate(String exp,String opt){

        int array[] = split(exp,opt);

        return calculate(array[0],array[1]);

    }

    /*被子类重写的方法*/

    abstract public int calculate(int num1,int num2);

    public int[] split(String exp,String opt){

        String array[] = exp.split(opt);

        int arrayInt[] = new int[2];

        arrayInt[0] = Integer.parseInt(array[0]);

        arrayInt[1] = Integer.parseInt(array[1]);

    }

}
```

```

        return arrayInt;
    }
}

```

[java] [view plaincopy](#)

```

public class Plus extends AbstractCalculator {

    @Override

    public int calculate(int num1,int num2) {

        return num1 + num2;

    }

}

```

测试类：

[java] [view plaincopy](#)

```

public class StrategyTest {

    public static void main(String[] args) {

        String exp = "8+8";

        AbstractCalculator cal = new Plus();

        int result = cal.calculate(exp, "\\+");

        System.out.println(result);

    }

}

```

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。