**1、静态代理**

用户通过代理对象要求目标对象进行一些行动。静态代理在使用时,需要定义接口或者父类,被代理对象与代理对象一起实现相同的接口或者是继承相同父类.



**2、简单工厂模式：生产子类对象**

public class Operation {

private double numberA=0;

private double numberB=0;

public double getNumberA() {

return numberA;

}

public void setNumberA(double numberA) {

this.numberA = numberA;

}

public double getNumberB() {

return numberB;

}

public void setNumberB(double numberB) {

this.numberB = numberB;

}

public double getResult() {

double result=0;

return result;

}

}

class **OperationAdd** extends Operation{

@Override

public double getResult() {

double result=getNumberA()+getNumberB();

return result;

}

}

class **OperationSub** extends Operation{

@Override

public double getResult() {

double result=getNumberA()-getNumberB();

return result;

}

}

class **OperationMul** extends Operation{

@Override

public double getResult() {

double result=getNumberA()\*getNumberB();

return result;

}

}

class **OperationDiv** extends Operation{

@Override

public double getResult() {

double result=0;

if (getNumberB()== 0) {

System.out.println("分母不能为0");

} else {

result=(double)getNumberA()/getNumberB();

}

return result;

}

}

**核心：**

**public class OperationFactory {**

**public static Operation creatOperation(String operation) {**

**Operation oper=null;**

**switch (operation) {**

**case "+":**

**oper=new OperationAdd();**

**break;**

**case "-":**

**oper=new OperationSub();**

**break;**

**case "\*":**

**oper=new OperationMul();**

**break;**

**case "/":**

**oper=new OperationDiv();**

**break;**

**}**

**return oper;**

**}**

**}**

**应用：**

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

System.out.println("请输入数字A:");

double numberA = sc.nextDouble();

System.out.println("请选择运算符号+、-、\*、/");

String strOperate = sc.next();

System.out.println("请输入数字B:");

double numberB = sc.nextDouble();

Operation operation=OperationFactory.creatOperation(strOperate);

operation.setNumberA(numberA);

operation.setNumberB(numberB);

System.out.println("结果为：" +operation.getResult());

} catch (Exception e) {

System.out.println("输入有误");

}

sc.close();

}

}