

# C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

设计模式简介

# “设计模式”简介

## 什么是设计模式

- 设计模式被用来在特定场景下，描述解决一般设计问题的方案
- 每一个模式描述了一个不断重复发生的问题，以及解决该问题方案的核心
- 提取出的设计模式，帮助我们多次使用该方案而不必做重复劳动。

## 例：建筑工程中的设计模式



徽派建筑



哥特式建筑

# 软件设计模式的历史

## 时间

1990s初

## 人物

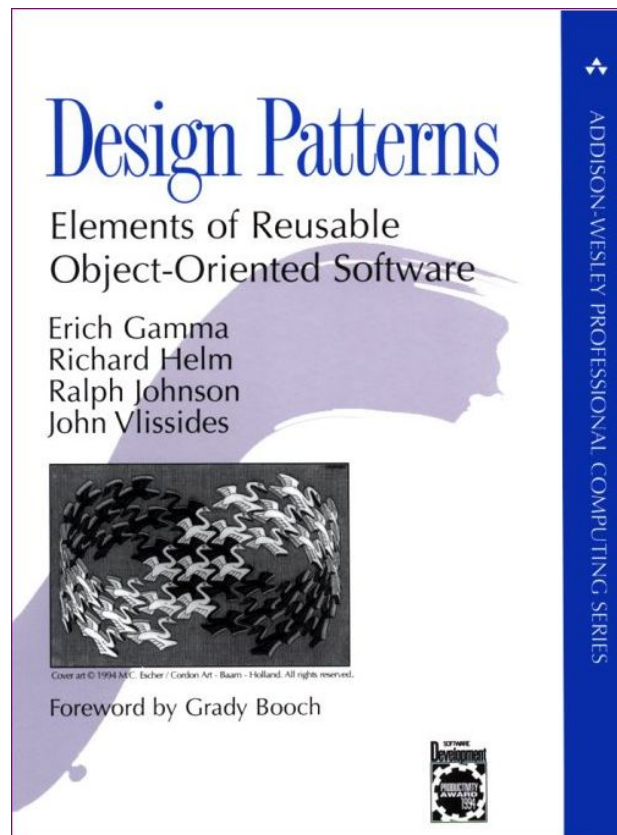
四人帮 ( Gang of Four, GoF )

## 事件

在软件设计中引入设计模式

## 内容

- 将设计模式的概念应用于软件设计
- 基于设计模式的OO策略和方法
- 对设计模式进行编目和描述
- 收录了23种设计模式

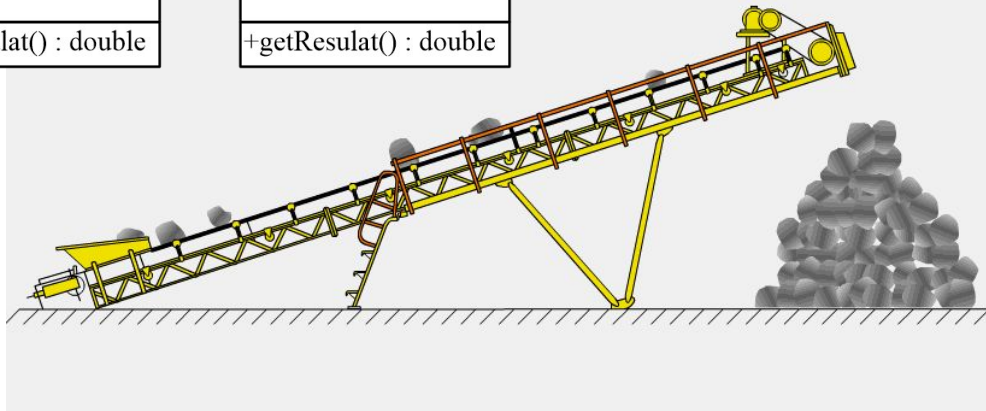
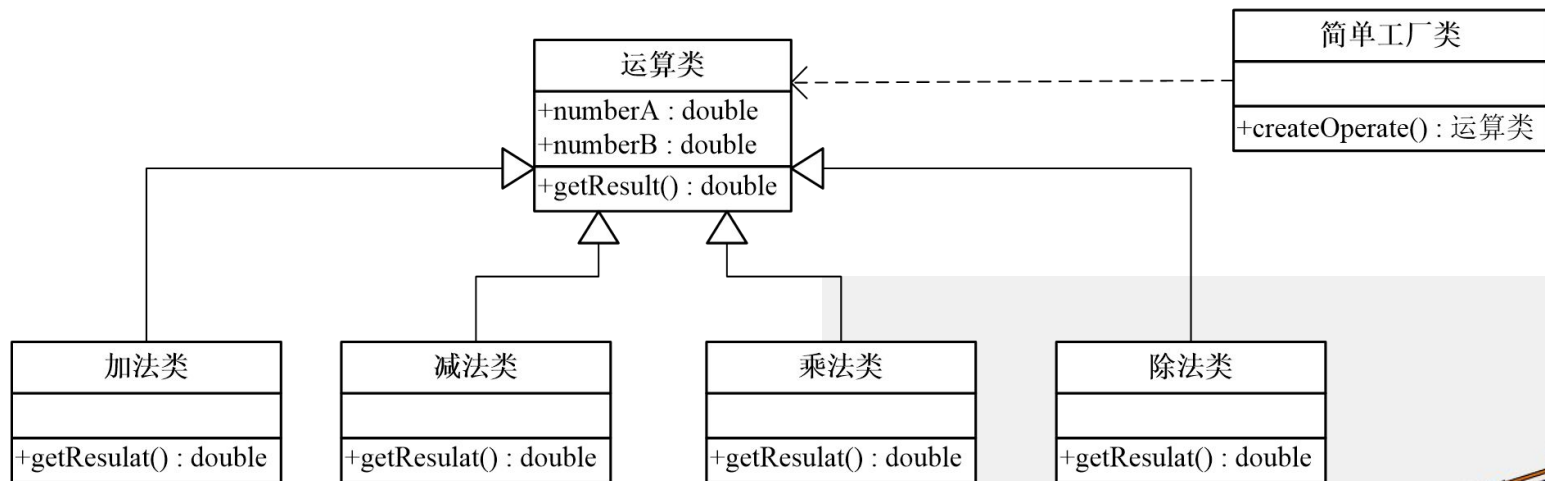


# 面向对向的软件设计模式

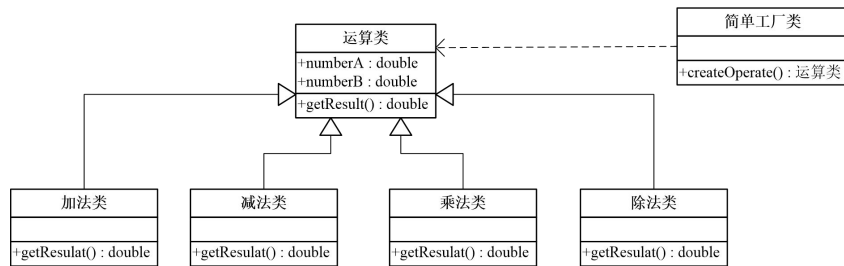
范围	目的		
	创建型	结构型	行为型
类	Factory Method	Adapter	Interpreter Template Method
对象	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor



## 示例：工厂模式



# 工厂模式的实现



```
class COperation
{
public:
    double numberA;
    double numberB;
    virtual double getResult() = 0;
};
```

//加法

```
class AddOperation: public COperation{
public:
    virtual double getResult() {
        return numberA+numberB;
    }
};
```

//减法

```
class SubOperation: public COperation{
public:
    virtual double getResult() {
        return numberA-numberB;
    }
};
```

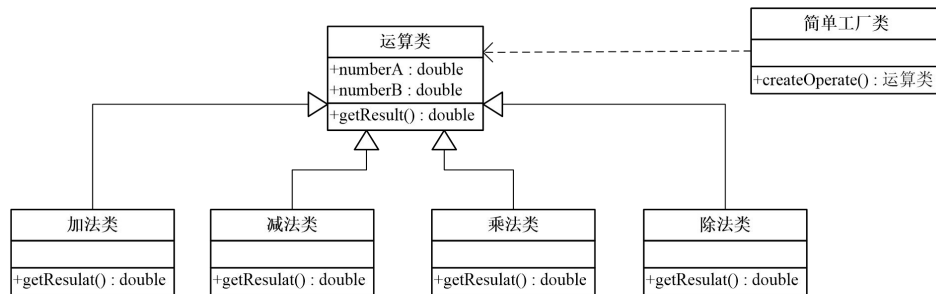
//乘法

```
class MultiOperation: public COperation {
public:
    virtual double getResult() {
        return numberA*numberB;
    }
};
```

//除法

```
class DivOperation: public COperation{
public:
    virtual double getResult() {
        return numberA/numberB;
    }
};
```

# 用“工厂”计算



```
int main()
```

```
{
```

```
    double a,b;
```

```
    char ch;
```

```
    cin>>a>>ch>>b;
```

```
    COperation * op=CCalculatorFactory::Create(ch);
```

```
    op->numberA=a;
```

```
    op->numberB=b;
```

```
    cout<<op->getResult()<<endl;
```

```
    return 0;
```

```
}
```

```
//工厂类
```

```
class CCalculatorFactory{
```

```
public:
```

```
    static COperation* Create(char cOperator);
```

```
};
```



```
COperation* CCalculatorFactory::Create(char cOperator)
```

```
{
```

```
    COperation *oper;
```

```
    switch (cOperator)
```

```
    {
```

```
        case '+':
```

```
            oper=new AddOperation(); break;
```

```
        case '-':
```

```
            oper=new SubOperation(); break;
```

```
        case '*':
```

```
            oper=new MultiOperation(); break;
```

```
        case '/':
```

```
            oper=new DivOperation(); break;
```

```
        default:
```

```
            oper=new AddOperation();
```

```
            break;
```

```
    }
```

```
    return oper;
```

```
}
```



# THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：[www.csdn.net](http://www.csdn.net)  
企业服务：<http://ems.csdn.net/>  
人才服务：<http://job.csdn.net/>  
CTO俱乐部：<http://cto.csdn.net/>  
高校俱乐部：<http://student.csdn.net/>  
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>  
项目外包：<http://www.csto.com/>  
CSDN博客：<http://blog.csdn.net/>  
CSDN论坛：<http://bbs.csdn.net/>  
CSDN下载：<http://download.csdn.net/>

