

- 多项式复合计算
 - 问题描述
 - 你的任务
 - 背景知识
 - 接口说明
 - 调用示例
 - 数据说明
 - 注意事项

多项式复合计算

问题描述

小明最近正在用C++学习机器学习，但不懂得利用第三方库的他每次做作业都需要从最底层开始实现，十分苦恼。他请你帮助他实现一个好用的计算引擎，能够通过C++的运算符自动处理函数的运算。你决定先实现一个最简单的版本，能够完成多项式函数的复合和运算。

你的任务

建立一个名为**Polynomial**的类，以支持多项式函数的运算。

多项式函数是形如 $f(x) = \sum_{i=0}^n w_i x^i = w_0 + w_1 x + w_2 x^2 + \dots + w_n x^n$ 的函数，其中参数 w_i 会在构造时给定。

Polynomial类要求编写**构造函数**、**拷贝构造函数**，重载 $+$ 运算符、 $*$ 运算符和 $()$ 运算符，另外实现 `derivative()` 函数实现求导的功能。

其中 $+$ 运算符完成两个多项式函数的相加，返回一个新的函数 $h(x)$ ，即 $h(x) = f(x) + g(x)$

$*$ 运算符完成两个函数的相乘，即 $h(x) = f(x)g(x)$

$()$ 运算符根据参数的不同，有着不同的功能：

1. 当参数为另一个Polynomial对象时，它完成多项式函数的复合，即 $h(x) = f(g(x))$ 。
2. 当参数为一个实值，它将该实值代入函数 $f(x)$ ，并计算函数值。

注意，函数可能进行多次运算和复合。

`derivative()` 函数则求出函数 $f(x)$ 在给定的点 $x = x_0$ 处对 x 的导数值。

背景知识

1. 一个函数 $f(x)$ 的导数 $f'(x)$ 是 $f(x)$ 在自变量 x 处发生微小变化时，函数值的增量与自变量增量的比值。
2. 幂函数的导数：若 $f(x) = ax^n$ ，则它的导数 $f'(x) = nax^{n-1}$

接口说明

你需要在自己定义的Polynomial类中至少实现以下接口

```
//Polynomial类的默认构造函数
Polynomial::Polynomial();

//Polynomial类的拷贝构造函数
Polynomial::Polynomial(const Polynomial& p);

//输入参数向量w构造多项式，n为w的长度，等于最高次幂+1，w[i]是x的i次幂的参数
//如f(x)=1+x+x^2+x^3时，w={1,1,1,1},n=4
Polynomial::Polynomial(const double w[],const int n);

//重载+运算符
Polynomial Polynomial::operator+(const Polynomial& p)const;

//重载*运算符
Polynomial Polynomial::operator*(const Polynomial& p)const;

//重载()运算符，实现函数复合
Polynomial Polynomial::operator()(const Polynomial& p)const;

//重载()运算符，实现函数值计算
double Polynomial::operator()(double x) const;

//求函数在x处的导数
double Polynomial::derivative(double x) const;
```

调用示例

Polynomial调用示例

```
Polynomial p(w,n);//创建一个Polynomial对象，参数向量为w，最大幂次为n
```

```
Polynomial p3=p1+p2;//计算两个多项式的和
```

```
Polynomial p4=p3*p2;//计算两个多项式的积
```

```
Polynomial p5=p1(p4);//计算多项式的复合
```

```
double result=p5(1.5);//求多项式p5在x=1.5处的函数值
```

```
double d=p5.derivative(1.5);//求多项式p5在x=1.5处的导数
```

数据说明

1. 多项式函数的最大幂次不超过100
2. 测试用例保证所有的运算都不会溢出
3. 计算结果精确到小数点后6位

注意事项

1. 你需要完成一个类，并创建以下**两个文件(注意文件名小写)**，将其打包为zip压缩包上传；

```
xxxx.zip
|
|--polynomial.h
|--polynomial.cpp
```

2. 你可以在类中增加新的函数，但不要改变类的命名及接口，否则无法通过测试；
3. 注意内存安全，避免内存泄漏；
4. 注意文件编码格式为utf-8；
5. 注意不要在提交的源代码中包含main函数；