

实验二 构造函数和析构函数

以下 1、2、3 三题是实验一中做过的，但这次要求数据的初始化用构造函数实现，并且改正实验一中存在的问题

1、定义一个长方体类，其中数据成员包括 length、width、height；有成员函数如下：构造函数（数据成员的初始化），cal(计算长方体体积)、show(输出长方体的体积)。然后编写一个主函数，求 3 个长方体的体积。

```
#include<iostream>
using namespace std;
class Cuboid
{
public:
    Cuboid(int l=0,int w=0,int h=0):length(l),width(w),height(h){ };
    int cal()
    {
        return length * width * height;
    }
    void show()
    {
        cout << cal()<< endl;
    }
private:
    int length,width,height;
};
int main()
{
    Cuboid c1;
    Cuboid c2(1,2,3);
    Cuboid c3(4,5,6);
    c1.show();
    c2.show();
    c3.show();
    return 0;
}
```

2、设计一个日期类，由构造函数进行初始日期值的设定、能实现日期数据的输出、加一天的操作。

```
#include<iostream>
using namespace std;
int d[13]={0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
class Date
{private:
    int year,month,day;
    bool isLeapYear();
public:
```

```

        Date(int y = 2015, int m = 1, int d = 1);
        void show();
        Date add();
};
Date::Date(int y, int m, int d) : year(y), month(m), day(d) { }
void Date::show()
{
    cout<<year<<"-"<<month<<"-"<<day<<endl;
}
bool Date::isLeapYear()
{
    return (year%4==0 && year%100!=0) || (year%400==0);
}
Date Date::add()
{
    if(month==2)
        d[month]++;
    if(day!=d[month])
        day++;
    else
    {
        if(day==d[month])
        {
            day=1;
            if(month==12)
            {
                month=1;year++;
            }
            else
                month++;
        }
    }
    return *this;
}
int main()
{
    Date date(2015, 12, 31);
    date.add();
    date.show();
    return 0;
}

```

3、定义一个素数类 Prime，可以求出给定范围的素数。写出主函数，输入范围，显示该范围的所有素数。

```

#include<iostream>
using namespace std;
class Prime
{public:
    void SetRange(int nLow, int nHigh)
    {
        low = nLow;
        high = nHigh;
    }
}

```

```

        void CalculatePrimenumber()
        {
            int i;
            for(i=low;i<=high;i++)
                if(IsPrimenumber(i))
                    cout<<i<<" ";
            cout<<'\\n';
        }
private:
    bool IsPrimenumber(int n)
    {
        int i;
        if(n==1) return false;
        for(i=2;i<n;i++)
            if(n%i==0)
                return false;
        return true;
    }
    int low, high;
};

int main()
{
    Prime num;
    num.SetRange(435, 1000);
    num.CalculatePrimenumber();
    return 0;
}

```

4、编写一个类 CAL，计算出 $\text{fact}=1\times 2\times 3\times \cdots \times n$ 值。类中包含数据成员 n、fact，成员函数包括初始化 n 的构造函数，求 fact 值的 process 函数，输出 n、fact 值的 show 函数。写出主函数，输入 n 的值，并输出结果。

```
#include<iostream>
using namespace std;
class CAL
{
    int n;
    double fact;
public:
    CAL(int x){n=x;}
    void process()
    {
        int i;
        double p=1;
        for(i=1;i<=n;i++)
            p=p*i;
        fact=p;
    }
    void show()
    {
        cout<<fact<<endl;
    }
};
main()
{
    CAL c(10);
    c.process ();
    c.show();
}
```

5、定义一个点类，由构造函数进行数据成员初始化，定义一个成员函数或一般函数，实现求二点之间的距离。

```
#include <iostream>
#include <cmath>
using namespace std;
class Point
{
public:
    Point(int xx,int yy):x(xx),y(yy){}
    friend double dis(Point p,Point q);
private:
    double x,y;
```

```

};
double dis(Point p,Point q)
{
    return sqrt((p.x-q.x)*(p.x-q.x)+(p.y-q.y)*(p.y-q.y));
}
main()
{
    Point p1(20,20),p2(30,30);
    cout<<dis(p1,p2)<<endl;
}

```

6、有两个矩阵 a 和 b，均为 3 行 5 列。求两个矩阵之和。

```

#include<iostream>
using namespace std;
class Matrix
{
public:
    Matrix()
    {
        for(int i=0;i<3;i++)
            for(int j=0;j<5;j++)
                a[i][j]=0;
    }
    void in()
    {
        cout<<"请输入 3*5 的数据"<<endl;
        for(int i=0;i<3;i++)
            for(int j=0;j<5;j++)
                cin>>a[i][j];
    }
    void display()
    {
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<5;j++)
                cout<<a[i][j]<<" ";
            cout<<endl;
        }
    }
    Matrix add(Matrix b)
    {
        Matrix c;
        for(int i=0;i<3;i++)
            for(int j=0;j<5;j++)
            {

```

```
        c.a[i][j]=a[i][j]+b.a[i][j];
    }
    return c;
}
private:
    double a[3][5];
};
int main()
{
    Matrix a,b,c;
    a.in();b.in();
    c=a.add(b);
    cout<<"a+b 的结果如下: "<<endl;
    c.display ();
    return 0;
}
```