实验三 类和对象

1. 实验目的

掌握类和对象的定义方法、成员函数的实现与调用方法，深刻领会类与对象的区别、类实现数据隐藏与封装的原理等。掌握类的构造函数的定义方法、构造函数的重载、析构函数的定义方法、掌握对象数据成员的初始化方法。

1. 实验内容
2. 定义一个Data类，包括year、month、day三个数据成员，通过set\_data( )成员函数输入出生日期数据，通过disp\_date( )成员函数将其输出，并设计出相应的程序测试类功能。

# include <iostream>

using namespace std;

class Data

{

public:

Data()

{

year=0;

month=0;

day=0;

}

void set\_data( );

void disp\_data( );

private:

int year;

int month;

int day;

};

void Data::set\_data()

{

cin>>year;

cin>>month;

cin>>day;

}

void Data::disp\_data()

{

cout<<year<<"年"<<month<<"月"<<day<<"日"<<endl;

}

int main()

{

Data d1;

Data d2;

cout<<"请输入出生日期"<<endl;

cout<<"年，月，日"<<endl;

d1.set\_data();

d1.disp\_data();

d2.disp\_data();

return 0;

}

1. 定义一个距离类，包含数据成员两个点的坐标，通过接口函数输入两点的坐标值，再通过Cal\_Distance( )成员函数计算出两点之间的距离，并输出结果。同时设计出相应的程序测试类的功能。

# include <iostream>

# include <math.h>

using namespace std;

class DISTANCE

{

public:

DISTANCE()

{

};

double Cal\_Distance();

void get\_p1();

void get\_p2();

private:

double x1;

double x2;

double y1;

double y2;

double Distance;

};

void DISTANCE::get\_p1()

{

cin>>x1;

cin>>y1;

}

void DISTANCE::get\_p2()

{

cin>>x2;

cin>>y2;

}

double DISTANCE::Cal\_Distance()

{

double x,y;

x=x1-x2;

y=y1-y2;

Distance=sqrt(x\*x+y\*y);

cout<<Distance<<endl;

return 0;

}

int main()

{

DISTANCE d1;

cout<<"请输入第一个点的横纵坐标:"<<endl;

d1.get\_p1();

cout<<"请输入第二个点的横纵坐标:"<<endl;

d1.get\_p2();

cout<<"两点间的距离:"<<endl;

d1.Cal\_Distance();

return 0;

}

法二：

#include <iostream>

#include <math.h>

#include "Point.h"

#include "Distance.h"

using namespace std;

class Point

{

public:

Point();

virtual ~Point();

public:

Point(int a,int b);

void get\_x();

void get\_y();

void put();

public:

int x;

int y;

};

Point::Point(int a,int b)

{

x=a;

y=b;

}

void Point::get\_x()

{

cout<<"输入横坐标:"<<endl;

cin>>x;

}

void Point::get\_y()

{

cout<<"输入纵坐标:"<<endl;

cin>>y;

}

void Point::put()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

class Distance

{

public:

Distance();

virtual ~Distance();

public:

Distance(Point a,Point b);

void Cal\_Distance(Point a,Point b);

public:

Point p1;

Point p2;

double d;

};

Distance::Distance(Point a,Point b)

{

p1=a;

p2=b;

}

void Distance::Cal\_Distance(Point a,Point b)

{

int i,j;

i=a.x-b.x;

j=a.y-b.y;

d=sqrt(i\*i+j\*j);

cout<<"距离为:"<<d<<endl;

}

void main()

{

Point p1,p2;

Distance c;

p1.get\_x();

p1.get\_y();

p1.put();

p2.get\_x();

p2.get\_y();

p2.put();

c.Cal\_Distance(p1,p2);

}

1. 自定义一个类，并在其构造函数和析构函数中编写一段输出代码能表明该函数被调用，在主函数中定义一个类的对象，然后通过成员函数输出字符串“This is a C++ Program.”，在主程序中完成字符串输出功能，并测试构造函数和析构函数何时被调用。

# include <iostream>

# include <math.h>

using namespace std;

class Sentence

{

public:

Sentence(int sz)

{

size=sz;

p=new int[sz];

cout<<"Call the constructor"<<endl;

}

void disp\_sentence();

~Sentence()

{

cout<<"Call destructor"<<endl;

delete []p;

}

private:

int size;

int \*p;

};

void Sentence::disp\_sentence()

{

cout<<"This is a C++ Program."<<endl;

}

int main()

{

Sentence s(20);

s.disp\_sentence();

return 0;

}

1. 以下是产品类Product的定义，但没有类的实现部分，请根据注释中类的定义编写类的实现部分的代码，并编写相应的对所定义的类进行测试的程序。

Class Product

{

Private:

string name; //产品名称

int price; //产品单价

int quantity; //产品剩余量

public：

Product(string a, int b, int c); //构造函数

void purchase(int money); //购买产品

void remain() //显示剩余产品数量

~Product(); //析构函数

}；

# include <iostream>

# include <math.h>

using namespace std;

class Product

{

private:

string name;

int price;

int quantity;

public:

int quantity1;

Product(string a, int b, int c)

{

name=a;

price=b;

quantity1=c;

quantity=100;

quantity=quantity-quantity1;

};

void purchase(int money);

void remain();

~Product()

{

cout<<"Call destructor"<<endl;

}

};

void Product::purchase(int money)

{

money=price\*quantity1;

cout<<"总价为"<<money<<endl;

}

void Product::remain()

{

cout<<"产品剩余量为"<<quantity<<endl;

}

int main()

{

Product p("a",5,30);

p.purchase(30);

p.remain();

return 0;

}

3、实验小结