## 第4章 类与对象

**习题4-8**

#include<iostream>

using namespace std;

class Dog

{

private:

    /\* data \*/

    int age;

    int weight;

public:

    Dog(int age, int weight);

    int getAge();

    int getWeight();

    void setAge(int age);

    void setWeight(int weight);

};

Dog::Dog(int age, int weight)

{

    this->age = age;

    this->weight = weight;

}

int Dog::getAge()

{

    return this->age;

}

int Dog::getWeight()

{

    return this->weight;

}

void Dog::setAge(int age)

{

    this->age = age;

}

void Dog::setWeight(int weight)

{

    this->weight = weight;

}

int main()

{

    Dog dog1(2, 6);

    Dog dog2(3, 10);

    cout<<"The dog1: "<<"age = "<<dog1.getAge()<<" weight = "<<dog1.getWeight()<<endl;

    cout<<"The dog2: "<<"age = "<<dog2.getAge()<<" weight = "<<dog2.getWeight()<<endl;

    return 0;

}

**习题4-9**

#include<iostream>

using namespace std;

class Rectangle{

private:

    int leftx, bottomy;

    int rightx, topy;

public:

    Rectangle(int leftx, int bottomy, int rightx, int topy);

    int getleftx();

    int getbottomy();

    int getrightx();

    int gettopy();

    int area();

};

Rectangle::Rectangle(int leftx, int bottomy, int rightx, int topy)

{

    this->leftx = leftx;

    this->bottomy = bottomy;

    this->rightx = rightx;

    this->topy = topy;

}

int Rectangle::getleftx()

{

    return leftx;

}

int Rectangle::getbottomy()

{

    return bottomy;

}

int Rectangle::getrightx()

{

    return rightx;

}

int Rectangle::gettopy()

{

    return topy;

}

int Rectangle::area()

{

    return (rightx - leftx)\*(topy - bottomy);

}

int main()

{

    Rectangle rect1(10, 20, 30, 40);

    cout<<"The rect's area is "<<rect1.area()<<endl;

    return 0;

}

**习题4-10**

/\*

习题4-10设计一个用于人事管理的“人员”类。

由于考虑到通用性，这里只抽象出所有类型人员都具有的属性：编号、性别、出生日期、身份证号等。

其中“出生日期”声明为一个“日期”类内嵌子对象。

用成员函数实现对人员信息的录入和显示。

要求包括：构造函数和析构函数、赋值构造函数、带默认形参值的成员函数、类的组合。

\*/

#include<iostream>

#include<cstring>

using namespace std;

class Date{

public:

    Date(){}

    Date(int y, int m, int d);

    Date(Date& obj);

    ~Date();

    void show();

private:

    int year;

    int month;

    int day;

};

Date::Date(int y,int m, int d)

:year(y), month(m), day(d)

{

}

Date::Date(Date& obj)

:year(obj.year), month(obj.month), day(obj.day)

{

}

Date::~Date()

{

}

void Date::show()

{

    cout<<year<<": "<<month<<": "<<day<<endl;

}

class Person{

public:

    Person(){}

    Person(int nm, bool gd, Date& bd, string& id1);

    Person(Person& obj);

    ~Person();

    void input();

    void show();

private:

    int num;

    bool gender;

    Date birthday;

    string id;

};

Person::Person(int nm, bool gd, Date& bd, string& id1)

:num(nm), gender(gd), birthday(bd), id(id1)

{

}

Person::Person(Person& obj)

:num(obj.num), gender(obj.gender), birthday(obj.birthday), id(obj.id)

{

}

Person::~Person()

{

}

void Person::input()

{

    int n;

    string sex;

    int y, m, d;

    string id1;

    cout<<"Please enter a num: ";

    cin>>n;

    cout<<"Please enter the sex(female/male):";

    cin>>sex;

    cout<<"Please enter the birthday(year month day):";

    cin>>y>>m>>d;

    cout<<"Please enter the id:";

    cin>>id1;

    num = n;

    if(sex=="female")

        gender = true;

    else

        gender = false;

    birthday = Date(y, m, d);

    id = id1;

}

void Person::show()

{

    cout<<"Num: "<<num<<endl;

    if(gender)

        cout<<"Gender: female"<<endl;

    else

        cout<<"Gender: male"<<endl;

    cout<<"Birthday: ";

    birthday.show();

    cout<<"ID: "<<id<<endl;

}

int main()

{

    Person p1;

    p1.input();

    p1.show();

    return 0;

}

/\*

习题4-11 定义并实现一个矩形类，有长、宽两个属性，由成员函数计算矩形的面积。

\*/

#include<iostream>

using namespace std;

class Rectangle

{

public:

    Rectangle(int l, int w);

    Rectangle(Rectangle& obj);

    ~Rectangle();

    int area();

private:

    int length;

    int width;

};

Rectangle::Rectangle(int l, int w)

:length(l), width(w)

{

}

Rectangle::Rectangle(Rectangle& obj)

:length(obj.length), width(obj.width)

{

}

Rectangle::~Rectangle()

{

}

int Rectangle::area()

{

    return length\*width;

}

int main()

{

    int len, wid;

    cout<<"Please enter length and width of the rectangle separated by space: ";

    cin>>len>>wid;

    Rectangle rect(len, wid);

    cout<<"The area of Rectangle is "<<rect.area()<<endl;

    return 0;

}

/\*

习题4-12 定义一个DataType（数据类型）类，能处理包含字符型、整型、浮点型3种类型的数据，给出其构造函数。

\*/

#include<iostream>

using namespace std;

class DataType

{

public:

    DataType(char c1);

    DataType(int n1);

    DataType(float rl);

    ~DataType(){};

    char getChar();

    int getInt();

    float getFloat();

private:

    char ch;

    int num;

    float real;

};

DataType::DataType(char c1)

:ch(c1)

{

}

DataType::DataType(int n1)

:num(n1)

{

}

DataType::DataType(float r1)

:real(r1)

{

}

char DataType::getChar()

{

    return ch;

}

int DataType::getInt()

{

    return num;

}

float DataType::getFloat()

{

    return real;

}

int main()

{

    DataType charVar('a');

    DataType intVar(4);

    DataType floatVar(static\_cast<float>(4.5));

    cout<<charVar.getChar()<<endl;

    cout<<intVar.getInt()<<endl;

    cout<<floatVar.getFloat()<<endl;

    return 0;

}

/\*

习题4-13 定义一个Circle类，有数据成员radius（半径），成员函数getArea()，计算圆的面积，构造一个Circle的对象进行测试。

\*/

#include<iostream>

using namespace std;

class Circle

{

public:

    Circle(int r):radius(r){};

    float area()

    {

        return 3.1415926\*radius\*radius;

    }

private:

    int radius;

};

int main()

{

    int r;

    cout<<"Please enter the radius of cirlce:";

    cin>>r;

    Circle c1(r);

    cout<<"The area of circle is "<<c1.area()<<endl;

}

/\*

习题4-14 定义一个Tree（树）类，有成员ages（树龄），成员函数grow(int years)对ages加上years，age()显示Tree对象的ages值。

\*/

#include<iostream>

using namespace std;

class Tree

{

public:

    Tree(int a):ages(a){}

    void grow(int years);

    int age();

private:

    int ages;

};

void Tree::grow(int years)

{

    ages += years;

}

int Tree::age()

{

    return ages;

}

int main()

{

    int a;

    cout<<"Please enter the ages of tree:";

    cin>>a;

    Tree tr(a);

    cout<<"The ages of tree is "<<tr.age()<<endl;

    int g = 2;

    tr.grow(g);

    cout<<g<<" years later, the ages of tree is "<<tr.age()<<endl;

    return 0;

}

**习题4-20**

class Complex

{

private:

double real;

double image;

public:

Complex();

Complex(double r, double i);

Complex(double r);

Complex(const Complex& obj);

Complex add(const Complex& c2);

void show();

};

#include "Complex.h"

#include <iostream>

Complex::Complex()

:real(0), image(0)

{

}

Complex::Complex(double r, double i)

: real(r), image(i)

{

}

Complex::Complex(double r)

: real(r), image(0)

{

}

Complex::Complex(const Complex& obj)

: real(obj.real), image(obj.image)

{

}

Complex Complex::add(const Complex& c2)

{

real = real + c2.real;

image = image + c2.image;

return \*this;

}

void Complex::show()

{

std::cout << real << "+" << image << " i" << std::endl;

}

#include "complex.h"

int main()

{

Complex c1(3, 5);

Complex c2 = 4.5;

c1.add(c2);

c1.show();

return 0;

}