第7章　数据的抽象与封装

1、实体、对象与类的概念	2、类的定义	3、对象声明与引用	**4、私有、公有与保护**
5、日期类的设计	6、两种程序设计思想	7、汽车类的设计	8、几何图形圆类的设计
9、构造函数的定义	10、重载构造函数	11、析构函数的定义	12、整数翻译函数
13、实际意义的析构函数	14、Person类的设计	15、对象与指针	16、this指针

公有、私有与保护成员的定义

- 在public:后面定义的数据成员和函数成员都称作公有成员
  - 例如：public: char name[10]; int min(int a, int b);
  - 字符数组name和函数min()都是公有成员
- 在private:后面定义的数据成员和函数成员都称作私有成员
  - 例如：private: int age; int max(int a, int b);
  - 整数变量age和函数max()都是私有成员
- 在protected:后面定义的数据成员和函数成员都称作保护成员
  - 例如：protected: float price; int abs(int a);
  - 浮点变量price和函数abs()都是保护成员

公有成员

- 外界(类外)能够直接访问该成员，通过.运算符
  - 一般函数成员被定义为公有成员
  - 通过调用公有函数成员实施规定的操作
  - 外界与类之间起着接口的作用
- 例如：钟表类中的设置函数set(int, int, int, float)就定义成公有成员

```
class Clock
{public:
        void set(int h,int m,int s,float p)
        {       //对小时、分钟、秒钟的值进行有效性检验后，再赋值
                Hour=h>=0&&h<=24?h:0;
                Minute=m>=0&&m<=60?m:0;
                Second=s>=0&&s<=60?m:0;
                Price=p;
        }
};  //分号别忘
Clock  XJTU;
XJTU.set(11,23,25,1000); √
```

私有成员

- 外界(类外)不能够直接访问该成员
  - 一般数据成员被定义为私有成员
  - 使得成员被封装隐藏起来，外界不能随便修改对象的私有数据成员
  - 只有通过类中公有函数对数据进行修改，达到数据的安全性
- 例如：钟表类的时分秒定义为私有成员，通过set()函数才能修改

```
class Clock
{
private:
        int Hour,Minute,Second;
        float Price;
public:
        void set(int h,int m,int s,float p);
        ......
};
Clock  XJTU;
XJTU.Hour=10; ×
```

保护成员

- 外界部分区域能够访问该成员
  - 换句话说某些数据或函数成员在类外被有限制的访问
  - 私有是对外界完全封闭，公有是完全开放，保护是介于两者之间
- 例如：下面有两个类的定义，一个是时间类，一个是日期类

```
class time
{
private:
        int hour,minute;
protected:
        int second;
……
};
```

```
class date:public time
{
private:
        int year,month,day;
public:
    void show_date_time()
    {
        cout<<year<<'-'<<month<<'-'<<day<<'\t';
        cout<<hour<<':'<<minute<<':'<<second<<endl;
    }
……
};
```

分数类的抽象描述

- 数据成员：分子与分母
  - 都是整型变量：int a,b;
  - 分子、分母定义成公有、私有、保护成员，理论上都可以
  - 但仔细分析，分母不能为0，应该定义成私有成员，以便数据安全控制
  - 分子是任意整数都可以，又不希望外界任意读写，所以定义成保护成员
- 函数成员：设置数据、输出分数、分数相加、求最大公因数
  - void set(int aa,int bb);//设置分子分母
  - void show();//显示分数
  - Fraction add(Fraction u);//加一个分数
  - int divisor(int p,int q);//求最大公约数
  - 分析一下什么四个函数应定义成什么种类的成员呢？
  - 前三个函数应该定义为公有成员
  - 最后一个函数可以任意定义，本类定义为私有成员

分数类的定义

```
class Fraction
{
protected:
        int a;//分子可以定义为保护成员
private:
        int b;//分母应该定义为私有成员，因为要防止分母为0
        int divisor(int p,int q);//求最大公约数
public:
        void set(int aa,int bb);//设置分子分母
        void show();//显示分数
        Fraction add(Fraction u);//加一个分数
};
```

类外定义设置与显示成员函数

```
//设置分子、分母
void Fraction:: set(int aa,int bb)
{
        a=aa;
        if (bb!=0) //分母有效性检验
                b=bb;
        else
        {       a=0;
                b=1;
        }
}
//显示分数
void Fraction::show()
{
        cout<<a<<"/"<<b;
}
```

类外定义相加成员函数

```
//分数相加，本类对象加u
Fraction Fraction::add(Fraction u)
{
        int tmp;
        Fraction v;

        v.a=a*u.b+b*u.a;   //分子
        v.b=b*u.b;//分母
        tmp=divisor(v.a ,v.b);//计算分子、分母的公约数
        v.a=v.a/tmp;//约去公约数
        v.b=v.b/tmp;//约去公约数
        return v;          //返回结果
}
```

再定义一个新类：实数类

```
class Real:public Fraction
{
public:
        void show_real()
        {
                cout<<a<<'/'<<b<<'='<<a/(double)b<<endl;
        }
};
```

在新的实数类中引用分数类的私有成员b，不允许

感谢收看！