CSDN 学院

C++语言基础

迁者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

友元——友元函数和友元类

问题的提出

- 一个类中的成员
  - 公用的(public)成员：在类外可以访问
  - 私有的(private)成员：只有本类中的函数可以访问
- 友元(friend)
  - 友元可以**访问与其有好友关系的类中的私有成员**——有限制的共享。
- 友元包括友元函数和友元类
  - 友元函数：如果在本类以外的其他地方定义的函数，在类体中用friend进行声明，此函数就称为**本类的友元函数**，友元函数可以访问这个类中的私有成员。
  - 友元类：类A将另一个类B声明为其**友元类**，友元类B中的所有函数都是A类的友元函数，可以访问A类中的所有成员。

普通的友元函数

display是Time类的friend函数

display不是Time类的成员函数，但可以引用Time中的私有成员

```
class Time
{
public:
   Time(int,int,int);
   friend void display(Time &);
private:
   int hour;
   int minute;
   int sec;
};
Time::Time(int h,int m,int s)
{
   hour=h;
   minute=m;
   sec=s;
}
```

```
void display(Time& t)
{
   cout<<t.hour<<":"<<t.minute<<":"<<t.sec<<endl;
}

int main( )
{
   Time t1(10,13,56);
   display(t1);
   return 0;
}
```

友元成员函数——另一类中的成员函数作友元

```
class Date;
//要对Date类进行提前引用声明
class Time{
public:
      Time(int,int,int);
      void display(Date &);
private:
      int hour;……
};
void Time::display(Date &d){
  cout<<d.year<<<<d.month<<d.day<<endl;
      cout<<hour<<…;
}
```

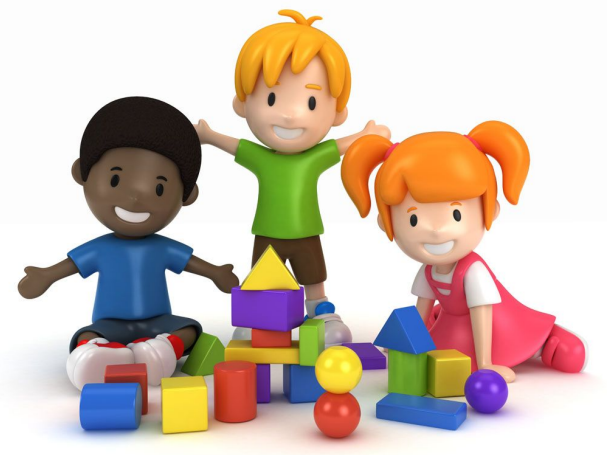
Time::display不是Date类的成员函数，但可以引用Date中的私有成员

```
class Date{
public:
      Date(int,int,int);
      friend void  Time::display(Date &);
private:
      int year, month, day;
};
int main( ){
      Time t1(10,13,56);
      Date d1(12,25,2004);
      t1.display(d1);
      return 0;
}
```

Time::display是Date类的friend函数

友元函数小结

- 如果在本类以外的其他地方定义了一个函数在类体中用 friend 对其进行声明，此函数就称为本类的友元函数。
- 友元函数可以访问这个类中的私有成员。
- 友元函数可以是不属于任何类的非成员函数，也可以是其他类的成员函数。
- 一个函数(包括普通函数和成员函数)可以被多个类声明为“朋友”，这样就可以引用多个类中的私有数据。



友元类

```
class Date;  ///对Date类的提前引用声明
class Time{
public:
    Time(int,int,int);
    void add(Date &);
    void display(Date &);
 private:
    int hour;……
};
void Time::add(Date &d){
   if(d.day>30) …
   if(d.month>12) …
 }
```

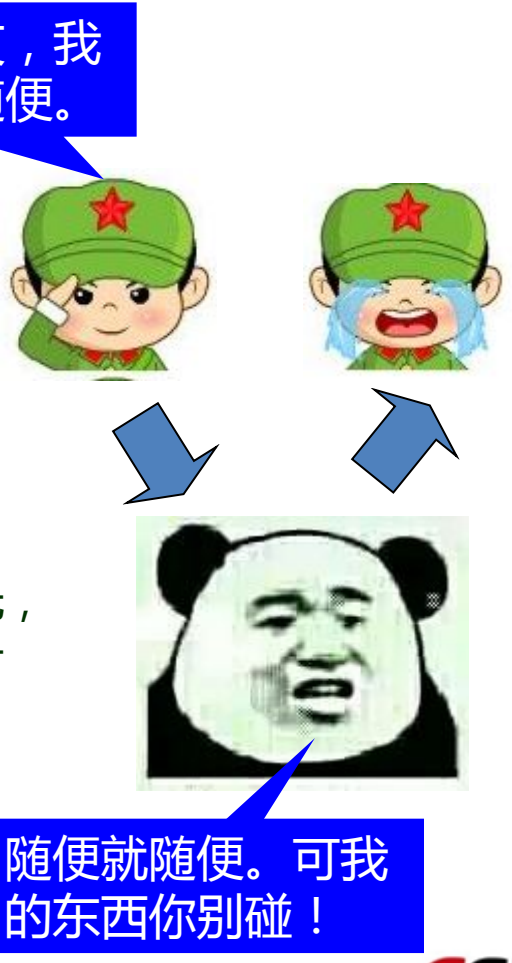
Time说：那我就不客气，直接访问Date中的私有成员

```
class Date{
public:
      Date(int,int,int);
      friend class Time;
private:
      int month;……
};
int main( ){
      Time t1(23,59,32);
      Date d1(12,25,2004);
      t1.add(d1);
      t1.display(d1);
      return 0;
}
```

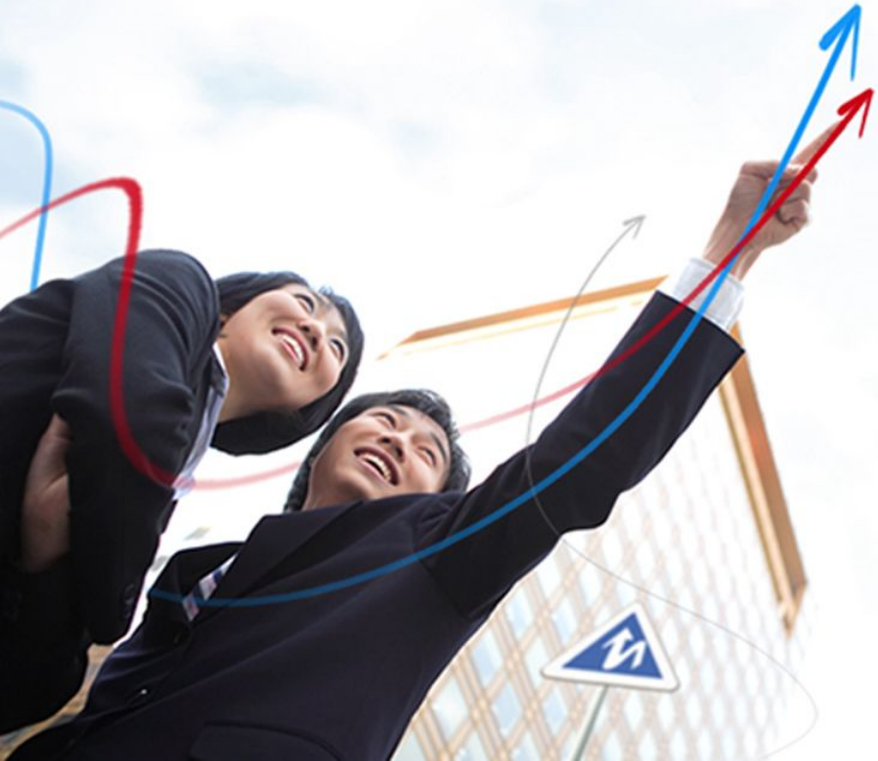
Date说：Time类是我的friend类

友元的性质与利用

- 性质
  - 友元的关系是单向的而不是双向的。
  - 友元的关系不能传递。
- 除非确有必要，一般并不把整个类声明为友元类
  - 优：有助于数据共享，能提高程序的效率；
  - 劣：友元访问其他类中的私有成员，是对封装原则的破坏，不利于信息隐藏。
- 在使用友元时，要注意到它的副作用，不要过多地使用友元，只有在使用它能使程序精炼，并能大大提高程序的效率时才用友元。
- 大多问题，用公共成员函数可以解决；但有些问题，必须靠友元机制。



CSDN 学院



THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/