程设第十三次作业

20305055 熊明

- 1. 将第六题程序做如下修改:
 - 将main函数第2行改为const Student stud(101, 78.5)报错
 - o 在上面基础上修改,使之能正常运行,用change函数修改数据成员num和score的值

```
1 #include<iostream>
   using namespace std;
 3
   class Student {
 4 public:
 5
        Student(int n, float s) :num(n), score(s) {}
        void change(int n, float s)const { num = n; score = s; }
 6
 7
        void display()const { cout << num << " " << score << endl; }</pre>
 8
   private:
 9
        mutable int num;
        mutable float score;
10
11
    };
12
   int main() {
13
        const Student stud(101, 78.5);
        stud.display();
14
        stud.change(101, 80.5);
15
16
        stud.display();
        return 0;
17
18 }
```

运行结果如下:

101 78.5 101 80.5

○ 修改mian函数:

```
1 #include<iostream>
 2
   using namespace std;
 3 class Student {
 4
    public:
        Student(int n, float s) :num(n), score(s) {}
 5
        void change(int n, float s) { num = n; score = s; }
 6
        void display() { cout << num << " " << score << endl; }</pre>
 7
 8
    private:
 9
        int num;
10
        float score;
11
   }:
    int main() {
12
13
        Student stud(101, 78.5);
        Student* p = &stud;
14
15
        p->display();
16
        p->change(101, 80.5);
```

运行结果如下:

```
101 78.5
101 80.5
```

o 在 (2) 的基础上修改main函数第三行前加const

```
1 #include<iostream>
 2
   using namespace std;
 3 class Student {
 4
   public:
 5
        Student(int n, float s) :num(n), score(s) {}
        void change(int n, float s) { num = n; score = s; }
 6
 7
        void display()const { cout << num << " " << score << endl; }</pre>
 8
   private:
 9
        int num;
        float score;
10
11 };
   int main() {
12
13
        Student stud(101, 78.5);
14
        const Student* p = &stud;
15
        p->display();
16
        stud.change(101, 80.5);
17
        p->display();
18
        return 0;
19 }
```

运行结果如下:

```
101 78.5
101 80.5
```

○ 再将main函数第三行改为Student* const p = &stud;

```
1 #include<iostream>
 2 using namespace std;
 3 class Student {
   public:
 4
 5
        Student(int n, float s) :num(n), score(s) {}
 6
        void change(int n, float s) { num = n; score = s; }
        void display(){ cout << num << " " << score << endl; }</pre>
 7
 8
    private:
 9
        int num;
10
        float score;
11
   };
12
    int main() {
        Student stud(101, 78.5);
13
14
        Student* const p = &stud;
15
        p->display();
        p->change(101, 80.5);
16
17
        p->display();
```

```
18 return 0;
19 }
```

运行结果如下:

```
101 78.5
101 80.5
```

2. 将例9.13 程序中的 display 函数不放在 Time 类中,而作为类外的普通函数,然后分别在 Time 和 Date 类中将 display 声明为友元函数。在主函数中调用 display 函数display 函数分别引用Time和 Date 两个类的对象的私有数据输出年月和时分秒请读者完成并上机调试。

```
1 #include <iostream>
 2
   using namespace std;
    class Date;
 3
    class Time
 5
   {
 6
    public:
 7
         Time(int, int, int);
 8
         friend void display(const Date&, const Time&);
 9
    private:
10
         int hour;
11
         int minute;
12
        int sec;
13
   };
14
15
    Time::Time(int h, int m, int s)
16
17
         hour = h;
18
         minute = m;
19
         sec = s;
20
21
22
   class Date
23
24
    public:
25
         Date(int, int, int);
26
         friend void display(const Date&, const Time&);
27
    private:
28
         int month;
29
         int day;
30
         int year;
31
   };
32
33
   Date::Date(int m, int d, int y)
34
    {
35
         month = m;
36
         day = d;
37
         year = y;
38
39
40
    void display(const Date& d, const Time& t)
41
         \operatorname{cout} << \operatorname{d.month} << "/" << \operatorname{d.day} << "/" << \operatorname{d.year} << \operatorname{endl};
42
         cout << t.hour << ":" << t.minute << ":" << t.sec << endl;</pre>
43
```

```
44
45
46
   int main()
47
48
49
        Time t1(10, 13, 56);
        Date d1(12, 25, 2004);
50
51
        display(d1, t1);
        return 0;
52
53
    }
```

运行结果:

12/25/2004 10:13:56

3. 将例 9.14 改写为在类模板外定义各成员函数的

```
1 #include <iostream>
   using namespace std;
    template<class numtype>
 4
   class Compare
 5
    {public:
       Compare(numtype a, numtype b);
 6
 7
       numtype max();
 8
      numtype min();
 9
      private:
10
       numtype x,y;
11
     };
12
    template <class numtype>
    Compare<numtype>::Compare(numtype a, numtype b)
13
14
     \{x=a;y=b;\}
15
    template <class numtype>
16
   numtype Compare<numtype>::max()
17
     {return (x>y)?x:y;}
18
   template <class numtype>
19
    numtype Compare<numtype>::min()
20
     {return (x<y)?x:y;}
21
22
    int main()
23
    {Compare<int> cmp1(3,7);
24
    cout<<cmp1.max()<<" is the Maximum of two integer numbers."<<endl;</pre>
25
     cout<<cmp1.min()<<" is the Minimum of two integer numbers."</pre>
    <<endl<<endl;
26
     Compare<float> cmp2(45.78,93.6);
27
     cout<<cmp2.max()<<" is the Maximum of two float numbers."<<endl;</pre>
     cout<<cmp2.min()<<" is the Minimum of two float numbers."<<endl<<endl;</pre>
28
29
     Compare<char> cmp3('a', 'A');
     cout<<cmp3.max()<<" is the Maximum of two characters."<<endl;</pre>
30
     cout<<cmp3.min()<<" is the Minimum of two characters."<<endl;</pre>
31
     return 0;
32
33
```

7 is the Maximum of two integer numbers. 3 is the Minimum of two integer numbers.

93.6 is the Maximum of two float numbers.45.78 is the Minimum of two float numbers.

a is the Maximum of two characters.

A is the Minimum of two characters.