

# C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





本节主题：

案例：Time类的设计

# 要做一个电子表!

## ❏ 定义时间类

❏ 数据：时、分、秒

❏ 动作：设置、调整、显示时间

## ❏ 原型

```
class Time
{
public:

private:
    int hour;
    int minute;
    int sec;
};
```

```
Time();
Time(int, int, int);
void set_time( );
void show_time( );
void add_a_sec(); //增加1秒钟
void add_a_minute(); //增加1分钟
void add_an_hour(); //增加1小时
```

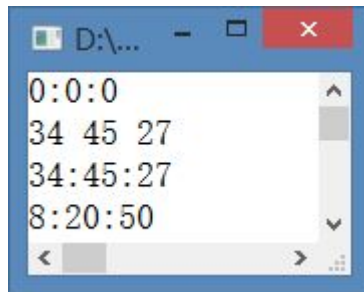


# Time类的初步实现与测试

```
class Time
{
public:
    Time():hour(0), minute(0), sec(0){};
    Time(int h, int m, int s):hour(h), minute(m), sec(s){};
    void set_time( );
    void show_time( );
    void add_a_sec(); //增加1秒钟
    void add_a_minute(); //增加1分钟
    void add_an_hour(); //增加1小时
private:
    int hour;
    int minute;
    int sec;
};
```

```
void Time::show_time( )
{
    cout<<hour<<":"<<minute<<":"<<sec<<endl;
}
```

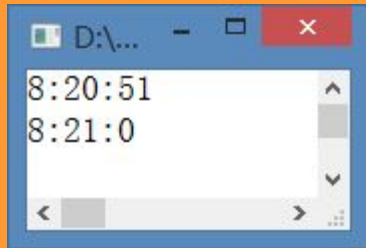
```
void Time::set_time( )
{
    cin>>hour>>minute>>sec;
}
```



```
int main( )
{
    Time t1;
    t1.show_time();
    t1.set_time( );
    t1.show_time( );
    Time t2(8,20,50);
    t2.show_time( );
    return 0;
}
```

# Time中时间调整功能的实现与测试

```
class Time
{
public:
    Time():hour(0), minute(0), sec(0){};
    Time(int h, int m, int s):hour(h), minute(m), sec(s){};
    void set_time( );
    void show_time( );
    void add_a_sec(); //增加1秒钟
    void add_a_minute(); //增加1分钟
    void add_an_hour(); //增加1小时
private:
    int hour;
    int minute;
    int sec;
};
```



```
void Time::add_an_hour()
{
    ++hour;
    if (hour>23)
        hour=0;
}
```

```
void Time::add_a_minute()
{
    ++minute;
    if (minute>59)
    {
        minute=0;
        add_an_hour();
    }
}
```

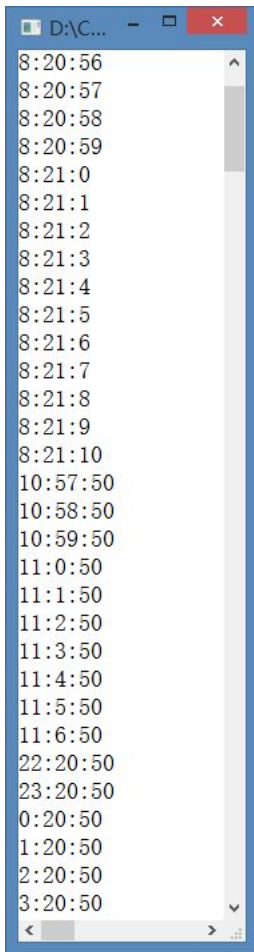
```
void Time::add_a_sec()
{
    ++sec;
    if (sec>59)
    {
        sec=0;
        add_a_minute();
    }
}
```

```
int main( )
{
    Time t1(8,20,50);
    t1.add_a_sec();
    t1.show_time();
    Time t2(8,20,59);
    t2.add_a_sec();
    t2.show_time();
    return 0;
}
```

## 这样测试更有效!

```
class Time
{
public:
    Time():hour(0), minute(0), sec(0){};
    Time(int h, int m, int s):hour(h), minute(m), sec(s){};
    void set_time( );
    void show_time( );
    void add_a_sec(); //增加1秒钟
    void add_a_minute(); //增加1分钟
    void add_an_hour(); //增加1小时
private:
    int hour;
    int minute;
    int sec;
};
```

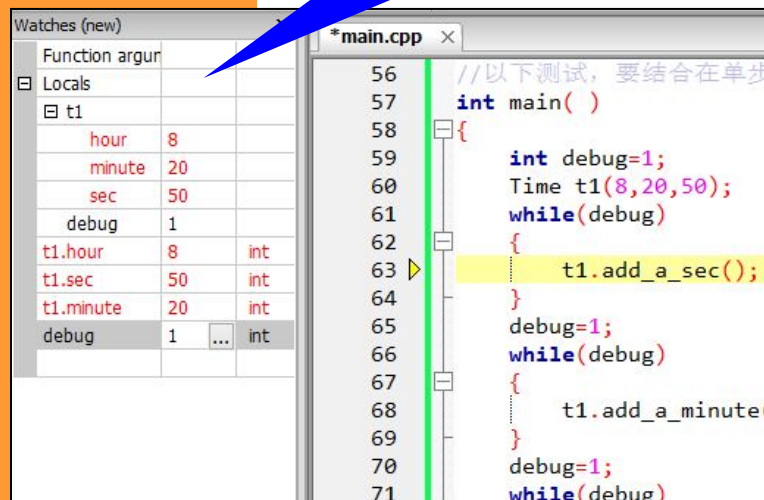
```
int main( )
{
    int i;
    Time t1(8,20,50);
    for(i=0; i<20; i++)
    {
        t1.add_a_sec();
        t1.show_time();
    }
    Time t2(10,56,50);
    for(i=0; i<10; i++)
    {
        t2.add_a_minute();
        t2.show_time();
    }
    Time t3(21,20,50);
    for(i=0; i<10; i++)
    {
        t3.add_an_hour();
        t3.show_time();
    }
    return 0;
}
```



## 用单步更灵活的测试!

```
class Time
{
public:
    Time():hour(0), minute(0), sec(0){};
    Time(int h, int m, int s):hour(h), minute(m), sec(s){};
    void set_time( );
    void show_time( );
    void add_a_sec(); //增加1秒钟
    void add_a_minute(); //增加1分钟
    void add_an_hour(); //增加1小时
private:
    int hour;
    int minute;
    int sec;
};
```

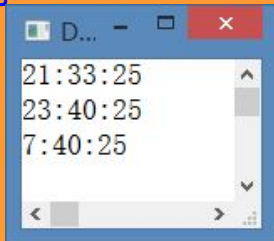
结合在单步执行中观察窗口修改变量的值进行。



```
int main( )
{
    int debug=1;
    Time t1(8,20,50);
    while(debug)
    {
        t1.add_a_sec();
    }
    debug=1;
    while(debug)
    {
        t1.add_a_minute();
    }
    debug=1;
    while(debug)
    {
        t1.add_an_hour();
    }
    return 0;
}
```

## Time类的完善：增加调整时间的方式

```
class Time
{
public:
    Time(): hour(0), minute(0), sec(0) {}
    Time(int h, int m, int s):hour(h), minute(m), sec(s) {}
    void set_time( );
    void show_time( );
    void add_a_sec(); //增加1秒钟
    void add_a_minute(); //增加1分钟
    void add_an_hour(); //增加1小时
    void add_seconds(int); //增加n秒钟
    void add_minutes(int); //增加n分钟
    void add_hours(int); //增加n小时
private:
    int hour;
    int minute;
    int sec;
};
```



```
void Time::add_hours(int n)
{
    hour+=n;
    if (hour>23)
        hour%=24;
}
```

```
int main( )
{
    Time t1(21, 32, 45);
    t1.add_seconds(40);
    t1.show_time( );
    t1.add_minutes(127);
    t1.show_time( );
    t1.add_hours(8);
    t1.show_time( );
    return 0;
}
```

```
void Time::add_minutes(int n)
{
    minute+=n;
    if (minute>59)
    {
        add_hours(minute/60);
        minute%=60;
    }
}
```

```
void Time::add_seconds(int n)
{
    sec+=n;
    if (sec>59)
    {
        add_minutes(sec/60);
        sec%=60;
    }
}
```



## Time类的完善：录入中的格式和数据成员数值限制

```
class Time
{
public:
    Time(): hour(0), minute(0), sec(0) {}
    Time(int h, int m, int s):hour(h), minute(m), sec(s) {}
```

```
    void set_time( );
```

```
    void show_time( );
```

```
    void add_a_sec(); //增加1秒钟
```

```
    void add_a_minute(); //增加1分钟
```

```
    void add_an_hour(); //增加1小时
```

```
    void add_seconds(int); //增加n秒钟
```

```
    void add_minutes(int); //增加n分钟
```

```
    void add_hours(int); //增加n小时
```

```
private:
```

```
    bool is_time(int, int, int);
```

```
    int hour;
```

```
    int minute;
```

```
    int sec;
```

```
};
```

```
bool Time::is_time(int h,int m, int s)
{
    if (h<0 || h>24 || m<0 || m>60 || s<0 || s>60)
        return false;
    return true;
}
```

```
int main( )
{
    Time t1;
    t1.set_time();
    t1.show_time( );
    return 0;
}
```

```
void Time::set_time( )
```

```
{
```

```
    char c1,c2;
```

```
    cout<<"请输入时间(格式hh:mm:ss)";
```

```
    while(1)
```

```
{
```

```
    cin>>hour>>c1>>minute>>c2>>sec;
```

```
    if(c1!=':' || c2!=':')
```

```
        cout<<"格式不正确，请重新输入"<<endl;
```

```
    else if (!is_time(hour,minute,sec))
```

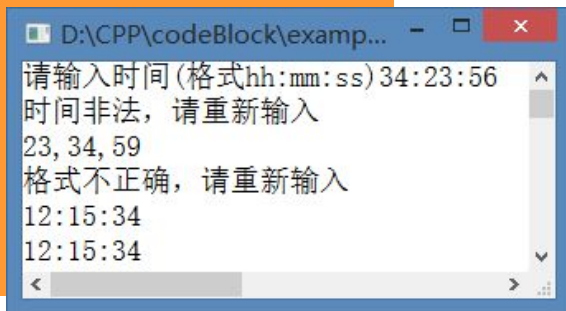
```
        cout<<"时间非法，请重新输入"<<endl;
```

```
    else
```

```
        break;
```

```
}
```

```
}
```



# THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：[www.csdn.net](http://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/>