CSDN 学院

# C++语言基础

迁者 - 贺利坚

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

http://edu.csdn.net

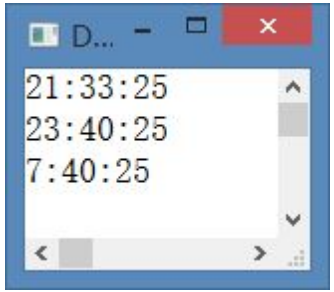
本节主题：

案例：Time类的设计

# "老式"的时间类

```
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;
};
```


```
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;
}
```



# 新设计!!!

```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute; // 分
   unsigned short int second; // 秒
public:
   CTime(int h=0,int m=0,int s=0);
   void setTime(int h,int m,int s);
   //(1)输入输出运算的重载
   //(2)比较运算符(二目)的重载
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   //(5)赋值运算符的重载
};
```

```
//(1)输入输出运算的重载
friend istream &operator>>(istream &in,CTime &t);
friend ostream &operator<<(ostream &out,const CTime &t);
```

```
//(2)比较运算符(二目)的重载
bool operator > (CTime &t);
bool operator < (CTime &t);
bool operator >= (CTime &t);
bool operator <= (CTime &t);
bool operator == (CTime &t);
bool operator != (CTime &t);
```

```
//(3)二目运算符的重载
CTime operator+(CTime &c);
CTime operator-(CTime &c);
CTime operator+(int s);
CTime operator-(int s);
```

```
//(4)一目运算符的重载
CTime operator++(int);
CTime operator++();
CTime operator--(int);
CTime operator--();
```

```
//(5)赋值运算符的重载
CTime operator+=(CTime &c);
CTime operator-=(CTime &c);
CTime operator+=(int s);
CTime operator-=(int s);
```

# 现在出发!

```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute; // 分
   unsigned short int second; // 秒
public:
   CTime(int h=0,int m=0,int s=0);
   void setTime(int h,int m,int s);
   //(1)输入输出运算的重载
   //(2)比较运算符(二目)的重载
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   //(5)赋值运算符的重载
};
```

```
//构造函数
CTime::CTime(int h,int m,int s):hour(h),minute(m),second(s) {}

// 设置时间
void CTime::setTime(int h,int m,int s)
{
   hour=h;
   minute=m;
   second=s;
}
```

# 输入输出

```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute; // 分
   unsigned short int second; // 秒
public:
   //(1)输入输出运算的重载
   friend istream &operator>>(istream &in,CTime &t);
   friend ostream &operator<<(ostream &out,const CTime &t);
   //(2)比较运算符(二目)
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   //(5)赋值运算符的重载
};
```

```
// 重载输入运算符>>
istream &operator>>(istream &in,CTime &t){
   char ch1,ch2;
   while(1) {
      cout<<"请输入时间(hh:mm:ss) ";
      cin>>t.hour>>ch1>>t.minute>>ch2>>t.second;
      if (ch1==':' && ch2==':')
         if (t.hour>-1 && t.hour<24 && t.minute>-1
            && t.minute<60 && t.second>-1 && t.second<60) break;
      cerr<<"时间格式不正确! 请重新输入\n";
   }
   return in;
}
```

```
// 重载输出运算符<<
ostream &operator<<(ostream &out, const CTime &t)
{
   out<<t.hour<<':'<<t.minute<<':'<<t.second;
   return out;
}
```

```
int main()
{
   CTime t1(12,34,28),t2,t;
   cin>>t2;
   cout<<"t1: "<<t1<<endl;
   cout<<"t2: "<<t2<<endl;
   return 0;
}
```

# 比较运算符重载

```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute;  // 分
   unsigned short int second;  // 秒
public:
   //(1)输入输出运算的重载
  //(2)比较运算符(二目)的重载
   bool operator > (CTime &t);
   bool operator < (CTime &t);
   bool operator >= (CTime &t);
   bool operator <= (CTime &t);
   bool operator == (CTime &t);
   bool operator != (CTime &t);
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   //(5)赋值运算符的重载
};
```

```
bool CTime::operator > (CTime &t)
{
   if (hour>t.hour) return true;
   if (hour<t.hour) return false;
   if (minute>t.minute) return true;
   if (minute<t.minute) return false;
   if (second>t.second) return true;
   return false;
}

bool CTime::operator < (CTime &t)
{
   if (hour<t.hour) return true;
   if (hour>t.hour) return false;
   if (minute<t.minute) return true;
   if (minute>t.minute) return false;
   if (second<t.second) return true;
   return false;
}
```

```
bool CTime::operator == (CTime &t)
{
   if (*this<t || *this>t) return false;
   return true;
}

bool CTime::operator != (CTime &t)
{
   if (*this==t) return false;
   return true;
}

bool CTime::operator >= (CTime &t)
{
   if (*this<t) return false;
   return true;
}

bool CTime::operator <= (CTime &t)
{
   if (*this>t) return false;
   return true;
}
```

# 二目运算符的重载

```
class CTime
{
private:
    unsigned short int hour;   // 时
    unsigned short int minute; // 分
    unsigned short int second; // 秒
public:
    //(1)输入输出运算的重载
   //(2)比较运算符(二目)的重载
   //(3)二目运算符的重载
    CTime operator+(CTime &c);
    CTime operator-(CTime &c);
    CTime operator+(int s);
    CTime operator-(int s);
    //(4)一目运算符的重载
    //(5)赋值运算符的重载
};
```

```
// 计算时间之和
CTime CTime::operator + (CTime &t)
{
    int h,m,s;
    s=second+t.second;
    m=minute+t.minute;
    h=hour+t.hour;
    if (s>59) {
        s-=60;
        m++;
    }
    if (m>59){
        m-=60;
        h++;
    }
    while (h>23) h-=24;
    CTime t0(h,m,s);
    return t0;
}
```

```
//返回s秒后的时间
CTime CTime::operator+(int s){
    int ss=s%60;
    int mm=(s/60)%60;
    int hh=s/3600;
    CTime t0(hh,mm,ss);
    return *this+t0;
}
```

```
// 计算时间之差
CTime CTime::operator - (CTime &t){
    int h,m,s;
    s=second-t.second;
    m=minute-t.minute;
    h=hour-t.hour;
    if (s<0) {
        s+=60;
        m--;
    }
    if (m<0) {
        m+=60;
        h--;
    }
    while (h<0) h+=24;
    CTime t0(h,m,s);
    return t0;
}
```

```
//返回s秒前的时间
CTime CTime::operator-(int s){
    int ss=s%60;
    int mm=(s/60)%60;
    int hh=s/3600;
    CTime t0(hh,mm,ss);
    return *this-t0;
}
```

# 一目运算符的重载

```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute; // 分
   unsigned short int second; // 秒
public:
    //(1)输入输出运算的重载
   //(2)比较运算符(二目)的重载
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   CTime operator++(int);
   CTime operator++();
   CTime operator--(int);
   CTime operator--();
   //(5)赋值运算符的重载
};
```

```
//后置++
CTime CTime::operator++(int)
{
   CTime t=*this;
   *this=*this+1;
   return t;
}

//前置++
CTime CTime::operator++()
{
   *this=*this+1;
   return *this;
}
```

```
//后置--
CTime CTime::operator--(int)
{
   CTime t=*this;
   *this=*this-1;
   return t;
}

//前置--
CTime CTime::operator--()
{
   *this=*this-1;
   return *this;
}
```

# 赋值运算符的重载

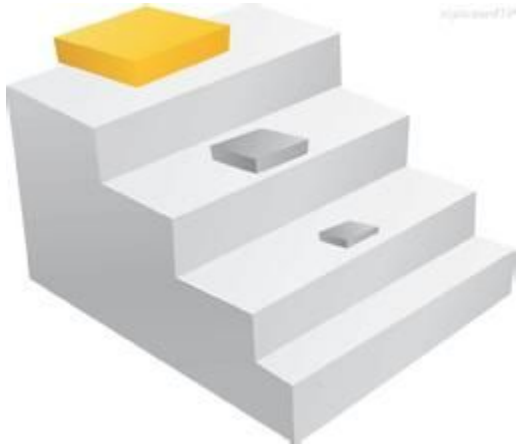
```
class CTime
{
private:
   unsigned short int hour;   // 时
   unsigned short int minute; // 分
   unsigned short int second; // 秒
public:
    //(1)输入输出运算的重载
   //(2)比较运算符(二目)的重载
   //(3)二目运算符的重载
   //(4)一目运算符的重载
   //(5)赋值运算符的重载
   CTime operator+=(CTime &c);
   CTime operator-=(CTime &c);
   CTime operator+=(int s);
   CTime operator-=(int s);
};
```

```
//两个时间加起来
CTime CTime::operator+=(CTime &c)
{
   *this=*this+c;
   return *this;
}

//两个时间差几时几分几秒
CTime CTime::operator-=(CTime &c)
{
   *this=*this-c;
   return *this;
}
```

```
//返回s秒后的时间
CTime CTime::operator+=(int s)
{
   *this=*this+s;
   return *this;
}

//返回s秒前的时间
CTime CTime::operator-=(int s)
{
   *this=*this-s;
   return *this;
}
```

# 终局

- 整体设计
- 迭代式实现和测试



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/