

Assignment 4

一、概念题

1.1、简述Demeter法则的基本思想；过度使用Demeter法则会带来什么问题。

1.2、简述为什么要对操作符重载进行重载；操作符重载会带来什么问题。

1.3、简述操作符重载的两种形式；这两种形式有什么区别。

二、编程题

2.1、定义一个日期类Date，其实例为year年month月day日；定义一个时间类Time，其实例为hour时minute分second秒；定义一个日期时间类Datetime，其实例为某日/某时。如下：

```
1  class Date{
2      private:
3          int _year, _month, _day;
4      public:
5          Date(int year, int month, int day);
6  }
7  class Time{
8      private:
9          int _hour, _minute, _second;
10     public:
11         Time(int hour, int minute, int second);
12 }
13 class Datetime{
14     private:
15         Date _date;
16         Time _time;
17     public:
18         Datetime(const Date & data, const Time & time);
19         std::string to_string() const;
20         // 可以不实现，帮助调试，格式如下
21         // year.month.day/hour.minute.second.
22 }
```

```

23     Datetime operator+(long) const; // 日期时间加秒数
24     ...
25     bool operator<(const Datetime &) const; // 比较日期时间大小
26     bool operator==(const Datetime &) const; // 比较日期时间是否相等
27 }

```

要求：需要重载日期时间Datetime类以下操作符

- **operator+**
 - 日期时间加秒数，如 $1999.12.31/23.59.59 + 86401s = 2000.1.2/0.0.0$ 。
(86401秒为1天零1秒)
- **operator-**
 - 日期时间减秒数，如 $2000.1.1/0.0.0 - 1s = 1999.12.31/23.59.59$ 。
 - 日期时间减日期时间，如 $2000.1.1/0.0.0 - 1999.12.31/23.59.59 = 1s$ 。
- **operator++**
 - 日期时间自增一秒，去往未来， $++ 2000.1.1/0.0.0 = 2000.1.1/0.0.1$ 。
- **operator--**
 - 日期时间自减一秒，回到过去， $-- 2000.1.1/0.0.0 = 1999.12.31/23.59.59$ 。
- **operator<**
- **operator==**

注意 ⚠：需要考虑闰年

1. 公元年份非4的倍数，为平年，或
2. 公元年份为4的倍数但非100的倍数，为闰年，或
3. 公元年份为100的倍数但非400的倍数，为平年，或
4. 公元年份为400的倍数为闰年。

<https://zh.wikipedia.org/wiki/闰年>