

中山大学本科生期中考试

考试科目：《程序设计II实验》

学年学期：2024-2025 学年第 2 学期

开课单位：计算机学院

考试方式：上机闭卷

考试时间：100 分钟

【A6】重载角度类

题目描述

定义一个角度类 `Angle`，请重载它们之间的 `+`，`-`，`>`，`<`，`=` 运算符和输入输出 `>>`，`<<`。

Angle 类

包括 3 个私有变量和上述公有运算符

```
class Angle {
private:
    int degree;    // 度
    int minute;    // 分
    int second;    // 秒
public:
    Angle(): degree(0), minute(0), second(0) {}
    Angle(int deg, int min, int sec): degree(deg), minute(min), second(sec) {}
    friend Angle operator+(const Angle& a, const Angle& b);
    friend Angle operator-(const Angle& a, const Angle& b);
    friend istream& operator>>(istream& is, Angle& ang);
    friend ostream& operator<<(ostream& os, const Angle& ang);
    bool operator>(const Angle& ang);
    bool operator<(const Angle& ang);
    bool operator==(const Angle& ang);
};
```

- `+` 和 `-` 运算符实现角度的加减法，注意单位换算（ $1^\circ=60'$ ， $1'=60''$ ）；结果为负时，所有分量 `degree`, `minute`, `second` 均不为正
- `>>` 和 `<<` 表示输入输出运算符，输入到 `degree`, `minute`, `second` 三个变量，输出按 `a°b'c''` 的格式进行输出

3. `>`、`=` 和 `<` 运算符比较角度之间的大小，比较度、分、秒的总和即可。
4. 注意角度值规范化（秒和分应在 0-59 范围内）

样例输入

```
9 44 45
11 22 23
```

样例输出

```
0
0
1
21°7'8''
-1°-37'-38''
```

具体操作见 `main.cpp`。

`main.cpp`：

```
#include "Angle.hpp"
using namespace std;

int main() {
    Angle A,B;
    cin >> A >> B;
    cout << (A>B) << endl;
    cout << (A==B) << endl;
    cout << (A<B) << endl;
    cout << (A+B) << endl;
    cout << (A-B) << endl;
    return 0;
}
```

`Angle.hpp`：

```
#ifndef ANGLE_HPP
#define ANGLE
#include <iostream>
using namespace std;
```

```

class Angle {
private:
    int degree;    // 度
    int minute;    // 分
    int second;    // 秒
public:
    Angle(): degree(0), minute(0), second(0) {}
    Angle(int deg, int min, int sec): degree(deg), minute(min), second(sec) {}
    friend Angle operator+(const Angle& a, const Angle& b);
    friend Angle operator-(const Angle& a, const Angle& b);
    friend istream& operator>>(istream& is, Angle& ang);
    friend ostream& operator<<(ostream& os, const Angle& ang);
    bool operator>(const Angle& ang);
    bool operator<(const Angle& ang);
    bool operator==(const Angle& ang);
};
#endif

```

Angle.cpp : (答案，仅供参考)

编译检查 20%，标准测试 20%，随机测试 60%

```

#include "Angle.hpp"

Angle operator+(const Angle& a, const Angle& b) {
    int n_degree = a.degree+b.degree;
    int n_minute = a.minute+b.minute;
    int n_second = a.second+b.second;

    if (n_second > 60) {
        n_minute ++;
        n_second -= 60;
    }
    if (n_minute > 60) {
        n_degree ++;
        n_minute -= 60;
    }

    return Angle(n_degree, n_minute, n_second);
}

Angle operator-(const Angle& a, const Angle& b) {
    // 先将两个角度转换为总秒数进行计算
    int total_a = a.degree * 3600 + a.minute * 60 + a.second;
    int total_b = b.degree * 3600 + b.minute * 60 + b.second;
    int total_diff = total_a - total_b;

```

```

// 处理结果为负的情况
bool is_negative = false;
if (total_diff < 0) {
    is_negative = true;
    total_diff = -total_diff;
}

// 将总秒数转换回度分秒
int n_degree = total_diff / 3600;
int remaining = total_diff % 3600;
int n_minute = remaining / 60;
int n_second = remaining % 60;

// 如果结果是负的，所有分量都设为负
if (is_negative) {
    n_degree = -n_degree;
    n_minute = -n_minute;
    n_second = -n_second;
}

return Angle(n_degree, n_minute, n_second);
}

istream& operator>>(istream& is, Angle& ang) {
    is >> ang.degree >> ang.minute >> ang.second;
    return is;
}

ostream& operator<<(ostream& os, const Angle& ang) {
    os << ang.degree << "°" << ang.minute << "'" << ang.second << "'";
    return os;
}

bool Angle::operator>(const Angle& ang) {
    if (degree > ang.degree)
        return 1;
    else if (degree < ang.degree)
        return 0;
    else {
        if (minute > ang.minute)
            return 1;
        else if (minute < ang.minute)
            return 0;
        else {
            if (second > ang.second)
                return 1;
            else if (second < ang.second)
                return 0;
        }
    }
}

```

```

    }
}

bool Angle::operator<(const Angle& ang) {
    if (degree < ang.degree)
        return 1;
    else if (degree > ang.degree)
        return 0;
    else {
        if (minute < ang.minute)
            return 1;
        else if (minute > ang.minute)
            return 0;
        else {
            if (second < ang.second)
                return 1;
            else if (second > ang.second)
                return 0;
        }
    }
}

bool Angle::operator==(const Angle& ang) {
    return (degree == ang.degree && minute == ang.minute && second ==
ang.second);
}

```