МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСТИТЕТ)

ЛАБОРАТОРНАЯ РАБОТА №02

по курсу объектно-ориентированное программирование I семестр, 2021/22 уч. год

Студент *Маринин Иван Сергеевич, группа М80-208Б-20*

Преподаватель *Дорохов Евгений Павлович*

Условие

**Создать класс Angle** для работы с углами на плоскости, задаваемыми величиной в градусах и минутах. Обязательно должны быть реализованы: перевод в радианы, приведение к диапазону 0–360, сложение и вычитание углов, деление углов, получение значений тригонометрических функций, сравнение углов.

Операции над объектами реализовать в виде перегрузки операторов.

Описание программы

Исходный код лежит в 3 файлах:

1. main.cpp - исполняемый код.

2. angle.h - специальный файл .h, содержащий прототипы используемых мною функций.

3. angle.cpp - реализация функций для моего задания.

4. Makefile

**Дневник отладки**

Во время выполнения лабораторной работы программа не нуждалась в отладке, все ошибки компиляции были исправлены с первой попытки. После их исправления программа работала так, как было задумано изначально.

ivanmarinin@MacBook-Air-Ivan lab\_01 % ./a.out

На вход программе дается два угла,

градусы и минуты пишутся через пробел например,

60 30 (60 градусов и 30 мин)

90 60 (90 градусов и 60 мин)

60 10

90 30

Первый угол: 60 deg 10 min

Второй угол: 90 deg 30 min

Первый угол в радианах: 1.05011 rad

Второй угол в радианах: 1.57952 rad

Приведение к диапазону 0–360 первого угла: 60 deg 10 min

Приведение к диапазону 0–360 второго угла: 90 deg 30 min

Сумма двух углов: 150deg 40min

Разность двух углов: -31 deg 80 min

Деление двух углов: 0.664825 rad

Значения тригонометрических функций первого угла:

cos = 0.497479

sin = 0.867476

tan = 1.74375

Значения тригонометрических функций второго угла:

cos = -0.00872654

sin = 0.999962

tan = -114.589

**Недочёты**  
Недочётов не было обнаружено.

**Выводы**

Лабораторная работа №2 - это, по сути, та же самая лабораторная №1, только предусматривающая возможность перегрузки операторов. Лабораторная была выполнена успешно, в ее процессе были еще раз осознаны основные принципы ООП и перегрузки операторов.

**Исходный код**

angle.h

#ifndef ANGLE\_H

#define ANGLE\_H

#include <iostream>

#include <cmath>

using namespace std;

const double pi = 3.141592653589793;

class Angle {

public:

Angle(int deg1, int min1);

Angle(const Angle &a);

Angle();

double Rad(Angle a); // Перевод в радианы

void Reduction(Angle a); // Округление 0-360

void Trig(Angle a); // Тригонометрические функции

Angle operator+(const Angle &a1); // Сумма углов

Angle operator-(const Angle &a1); // Разность углов

Angle operator/(Angle aa); // Частное углов

friend ostream& operator<<(ostream &out, const Angle &a);

friend istream& operator>>(istream &in, Angle &a);

private:

int deg;

int min;

int rad;

};

#endif

angle.cpp

#include "angle.h"

Angle::Angle(int deg1, int min1) {

deg = deg1;

min = min1;

}

Angle::Angle(const Angle &a) {

deg = a.deg;

min = a.min;

}

Angle::Angle() {

deg = 0;

min = 0;

}

double Angle::Rad(Angle a) {

return (deg \* pi / 180 + min \* pi / (180 \* 60));

}

void Angle::Reduction(Angle a) {

cout << deg % 360 << " deg " << min << " min " << endl;

}

Angle Angle::operator+(const Angle& a1){

int sum1 = deg + a1.deg;

int sum2 = min + a1.min;

sum1 += sum2 / 100;

sum2 %= 100;

Angle ans(sum1, sum2);

return ans;

}

Angle Angle::operator-(const Angle& a1){

int diff1 = deg - a1.deg;

int diff2 = min - a1.min;

Angle ans(diff1, diff2);

if (diff2 > 0) return ans;

else {

diff2 = 100 + diff2;

diff1 -= 1;

return ans;

}

}

Angle Angle::operator/(Angle aa) {

double A1 = deg \* pi / 180 + min \* pi / (180 \* 60);

double A2 = aa.deg \* pi / 180 + aa.min \* pi / (180 \* 60);

double A = (A1 / A2) \* 180 / pi;

aa.deg = A;

return aa;

}

void Angle::Trig(Angle a) {

cout << "cos = " << cos(Rad(a)) << endl;

cout << "sin = " << sin(Rad(a)) << endl;

cout << "tan = " << tan(Rad(a)) << endl;

}

istream& operator>>(istream &in, Angle &a) {

in >> a.deg;

in >> a.min;

return in;

}

ostream& operator<<(ostream &out, const Angle &a){

out << a.deg << " deg " << a.min << " min ";

return out;

}

main.cpp

#include "angle.h"

int main() {

cout << "\nНа вход программе дается два угла,\nградусы и минуты пишутся через пробел например,\n60 30 (60 градусов и 30 мин)\n90 60 (90 градусов и 60 мин)" << endl;

Angle a, b;

cin >> a >> b;

cout << "\nПервый угол: " << a << endl;

cout << "Второй угол: " << b << endl;

cout << "Первый угол в радианах: " << a.Rad(a) << " rad" << endl;

cout << "Второй угол в радианах: " << b.Rad(b) << " rad" << endl;

cout << endl;

cout << "Приведение к диапазону 0–360 первого угла: ";

a.Reduction(a);

cout << "Приведение к диапазону 0–360 второго угла: ";

b.Reduction(b);

cout << endl;

cout << "Сумма двух углов: " << a + b << endl;;

cout << "Разность двух углов: " << a - b << endl;

cout << "Деление двух углов: " << a / b << endl;

cout << endl;

cout << "Значения тригонометрических функций первого угла:" << endl;

a.Trig(a);

cout << endl;

cout << "Значения тригонометрических функций второго угла:" << endl;

b.Trig(b);

return 0;

}