**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Попов Илья Павлович

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

# Постановка задачи

Вариант 10:

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.

# Описание программы

Был создан класс Angle - класс углов на плоскости, задаваемых величиной в градусах и минутах.

Реализованы такие операции над этим классом, как перевод в радианы, приведение к диапазону 0–360, сложение и вычитание углов, деление углов, получение значений тригонометрических функций, сравнение углов.

## Методы класса Angle:

Angle() : d(0), m(0), rad(0), otr(false) {} - создание Angle без аргументов

Angle(int degrees, int minutes) : d(degrees), m(minutes), otr(false) {

rad = (degrees + (minutes/60)) \* (M\_PI/180);

} - создание Angle с заданными аргументами

int degree() const { return d; } - геттер градусной меры угла

int minute() const { return m; } - геттер минутной меры угла

double radian() const { return rad; } - геттер радианной меры угла

bool otric() const { return otr; } - геттер отрицательности угла(используется когда градусная составляющая угла равна 0, а минутная задается всегда положительно, и не может сигнализировать о том, что угол отрицательный)

bool is\_equal(const Angle& a) const { - проверка углов на равенство

return d == a.degree() && m == a.minute();

}

## Перегружены операторы

1. >>, << - ввод и вывод Angle
2. + сложение Angle
3. - вычитание Angle
4. \* умножение Angle на число
5. / деление Angle на число

В программе присутствует проверка корректности введенных значений, как при создании класса, так и при обращениям к его методам.

# Набор тестов

**Тест № 1:** *//этот тест демонстрирует работоспособность программы на обычных числах(градусные меры углов лежат в границах 0-360, углы положительные)*

угол № 1 = 55 градусов 8 минут

угол № 2 = 279 градусов 35 минут

число для деления = 5

**Тест № 2:** *//этот тест демонстрирует работоспособность программы на отрицательных углах, на углах с кол-вом минут > 60, на углах, которые лежат вне промежутка 0-360)*

угол № 1 = -55 градусов 66 минут

угол № 2 = 789 градусов 18 минут

число для деления = -5

**Тест № 3:** *//этот тест демонстрирует работоспособность программы на углах с нулевым количеством градусов или секунд*

угол № 1 = -0 градусов 66 минут

угол № 2 = 14 градусов 0 минут

число для деления = 2

**Тест № 4:** *//этот тест демонстрирует работоспособность программы на углах с нулевым углом*

угол № 1 = -55 градусов 66 минут

угол № 2 = 0 градусов 0 минут

число для деления = 12

**Тесты №5, №6 и №7** *//направлены на демонстрацию устойчивости программы при вводе пользователем некорректных входных данных.*

# Результаты выполнения тестов

## **Тест №1**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

55.8

Введите значение угла(состоящее из градусов и минут) в формате x.y:

279.35

Введите число для демонстрации функций сложения и умножения

5

---------------------------------------------------

Ваши углы:

a1 = 55.8

a2 = 279.35

num: 5

Приведение к диапазону 0-360:

a1 = 55.8

a2 = 279.35

Сравнение углов:

<

a1 + a2 = 334.43

a1 - a2 = -224.27

a1 \* num = 275.40

a1 / num = 11.1

Перевод углов в радианы:

a1 = 0.962258

a2 = 4.87965

Синус угла a1 = 0.819152

Косинус угла a1 = 0.573576

Тангенс угла a1 = 1.42815

Котангенс угла a1 = 0.700208

## **Тест №2**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

-55.66

Введите значение угла(состоящее из градусов и минут) в формате x.y:

789.18

Введите число для демонстрации функций сложения и умножения

-5

---------------------------------------------------

Ваши углы:

a1 = -55.66

a2 = 789.18

num: -5

Приведение к диапазону 0-360:

a1 = -56.6

a2 = 69.18

Сравнение углов:

<

a1 + a2 = 13.12

a1 - a2 = -125.24

a1 \* num = 280.30

a1 / num = 11.13

Перевод углов в радианы:

a1 = -0.97913

a2 = 13.7759

Синус угла a1 = -0.809017

Косинус угла a1 = 0.587785

Тангенс угла a1 = -1.37638

Котангенс угла a1 = -0.726543

## **Тест №3**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

-0.66

Введите значение угла(состоящее из градусов и минут) в формате x.y:

14.0

Введите число для демонстрации функций сложения и умножения

2

---------------------------------------------------

Ваши углы:

a1 = -0.66

a2 = 14.0

num: 2

Приведение к диапазону 0-360:

a1 = -1.6

a2 = 14.0

Сравнение углов:

<

a1 + a2 = 12.54

a1 - a2 = -15.6

a1 \* num = -2.12

a1 / num = 0.33

Перевод углов в радианы:

a1 = -0.0191986

a2 = 0.244346

Синус угла a1 = 0.0174524

Косинус угла a1 = 0.999848

Тангенс угла a1 = 0.0174551

Котангенс угла a1 = 57.29

## Тест №4

Введите значение угла(состоящее из градусов и минут) в формате x.y:

-55.66

Введите значение угла(состоящее из градусов и минут) в формате x.y:

0.0

Введите число для демонстрации функций сложения и умножения

12

---------------------------------------------------

Ваши углы:

a1 = -55.66

a2 = 0.0

num: 12

Приведение к диапазону 0-360:

a1 = -56.6

a2 = 0.0

Сравнение углов:

<

a1 + a2 = -56.6

a1 - a2 = -56.6

a1 \* num = -313.12

a1 / num = -4.5

Перевод углов в радианы:

a1 = -0.97913

a2 = 0

Синус угла a1 = -0.809017

Косинус угла a1 = 0.587785

Тангенс угла a1 = -1.37638

Котангенс угла a1 = -0.726543

## **Тест №5**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

iii.7

Ошибка! Некорректный ввод.

## **Тест №6**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

-5.-6

Ошибка! Некорректный ввод.

Минутная мера должна быть >0!

## **Тест №7**

Введите значение угла(состоящее из градусов и минут) в формате x.y:

6.6.6

Ошибка! Некорректный ввод.

Проверьте, правильно ли расставлены точки!

# Листинг программы

/\*

Попов Илья М80-206Б-20

Лабораторная работа №1

Вариант 10.

Это второй вариант лабораторной работы с перегрузкой операторов

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.

\*/

#define MINUTE\_IN\_DEGREE 60

#define DEGREE\_IN\_CIRCLE 360

#define M\_PI 3.14159265358979323846

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

class Angle {

public:

Angle() : d(0), m(0), rad(0), otr(false) {}

Angle(int degrees, int minutes) : d(degrees), m(minutes), otr(false) {

rad = (degrees + (minutes/60)) \* (M\_PI/180);

}

int degree() const { return d; }

int minute() const { return m; }

double radian() const { return rad; }

bool otric() const { return otr; }

bool is\_equal(const Angle& a) const {

return d == a.degree() && m == a.minute();

}

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

private:

int d, m;

double rad;

bool otr;

};

int is\_number(const string& s) {//проверка входных данных

bool otr = false;

for (int i = 0; i < s.length(); ++i) {

if (s[i] == '-' && i == 0) {

otr = true;

continue;

}

else if (s[i] < '0' || s[i] > '9') {

return 0;

}

}

if (otr) { return 1; }

else { return 2; }

}

//----------------Ввод вывод--------------------------------

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

cout << "Введите значение угла(состоящее из градусов и минут) в формате x.y:" << endl;

string str, str\_d, str\_m;

cin >> str;

bool point = false;

for (int i = 0; i < str.length(); ++i) {

if (str[i] == '.' && (i == 0 || i == str.length() - 1 || point == true)) {

cout << "Ошибка! Некорректный ввод.\nПроверьте, правильно ли расставлены точки!" << endl;

exit(4);

}

if (str[i] == '.' && i != 0 && i != str.length() - 1 && point == false) {

point = true;

}

if (!point) {

str\_d.push\_back(str[i]);

}

else if (point && str[i] != '.') {

str\_m.push\_back(str[i]);

}

}

if (point == false) {

cout << "Ошибка! Некорректный ввод.\nОтсутствует минутная мера угла!" << endl;

exit(5);

}

if (!is\_number(str\_d) || !is\_number(str\_m)) {

cout << "Ошибка! Некорректный ввод." << endl;

exit(1);

}

if (is\_number(str\_d) == 1) { a.otr = true; }

a.d = stoi(str\_d); a.m = stoi(str\_m);

if (a.m < 0) {

cout << "Ошибка! Некорректный ввод.\nМинутная мера должна быть >0!" << endl;

exit(2);

}

a.rad = (a.d + (a.m / 60)) \* (M\_PI / 180);

return in;

}

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

if (a.degree() == 0 && a.otric() == true) {

cout << "-";

}

cout << a.degree() << "." << a.minute() << endl;

return out;

}

//----------------ПЕРЕВОДЫ--------------------------------

int degrees\_to\_min(const Angle& a1) {

if (a1.otric()) {

return a1.degree() \* 60 - a1.minute();

}

return a1.degree() \* 60 + a1.minute();

}

Angle min\_to\_degrees(int min) {

int new\_d = min / 60;

int new\_m = min % 60;

if (new\_d < 0) {

new\_m \*= (-1);

}

Angle cur(new\_d, new\_m);

return cur;

}

Angle rangeDegrees(const Angle& a1) {//приведение к диапазону 0–360

int new\_d = a1.degree();

int new\_m = a1.minute();

while (new\_m > MINUTE\_IN\_DEGREE && !a1.otric()) {

new\_m -= MINUTE\_IN\_DEGREE;

new\_d++;

}

while (new\_m > MINUTE\_IN\_DEGREE && a1.otric()) {

new\_m -= MINUTE\_IN\_DEGREE;

new\_d--;

}

if (abs(new\_d) >= DEGREE\_IN\_CIRCLE) {

if (a1.otric()) {

new\_d = DEGREE\_IN\_CIRCLE + new\_d % DEGREE\_IN\_CIRCLE;

if (new\_m != 0) {

new\_m = MINUTE\_IN\_DEGREE - new\_m;

}

}

else if (!a1.otric()) {

new\_d = new\_d % DEGREE\_IN\_CIRCLE;

}

}

Angle cur(new\_d, new\_m);

return cur;

}

//----------------МАТЕМАТИКА--------------------------------

double degrees\_to\_radians(Angle a1) {//перевод углов в радианы

Angle cur = rangeDegrees(a1);

double rad = degrees\_to\_min(a1) \* (M\_PI / (180 \* 60));

return rad;

}

bool cmpDegrees(const Angle& a1, const Angle& a2) { //сравнение углов(первый больше второго?)

Angle cur1 = rangeDegrees(a1);

Angle cur2 = rangeDegrees(a2);

if (cur1.degree() > cur2.degree()) {

return true;

}

else if (cur1.degree() == cur2.degree() && cur1.minute() > cur2.minute()) {

return true;

}

else {

return false;

}

}

Angle operator+ (const Angle& a1, const Angle& a2) {

int res = degrees\_to\_min(a1) + degrees\_to\_min(a2);

Angle cur = rangeDegrees(min\_to\_degrees(res));

return cur;

}

Angle operator- (const Angle& a1, const Angle& a2) {

int res = degrees\_to\_min(a1) - degrees\_to\_min(a2);

Angle cur = rangeDegrees(min\_to\_degrees(res));

return cur;

}

Angle operator\* (const Angle& a1, int multipli) {

int res = degrees\_to\_min(a1) \* multipli;

Angle cur = rangeDegrees(min\_to\_degrees(res));

return cur;

}

Angle operator/ (const Angle& a1, int div) {

if (div == 0) {

cout << "На 0 делить нельзя!" << endl;

exit(3);

}

int new\_d = a1.degree() / div;

if (div < 0) {

div \*= -1;

}

int new\_m = a1.minute() / div;

Angle res(new\_d, new\_m);

Angle cur = rangeDegrees(res);

return cur;

}

//----------------ТРИГОНОМЕТРИЯ--------------------------------

double sinDegrees(const Angle& a1) {//синус угла

return sin(a1.radian());

}

double cosDegrees(const Angle& a1) {//косинус угла

return cos(a1.radian());

}

double tgDegrees(const Angle& a1) {//тангенс угла

return sinDegrees(a1) / cosDegrees(a1);

}

double ctgDegrees(const Angle& a1) {//котангенс угла

return 1 / tgDegrees(a1);

}

//////////////////////////////////////////////////////////////////////////

int main() {

setlocale(LC\_ALL, "rus");

Angle a1, a2;

cin >> a1 >> a2;

cout << "Введите число для демонстрации функций сложения и умножения" << endl;

int num;

cin >> num;

cout << "---------------------------------------------------" << endl;

cout << endl << "Ваши углы:" << endl;

cout << "a1 = " << a1 << "a2 = " << a2;

cout << "num: " << num << endl ;

cout << endl << "Приведение к диапазону 0–360: " << endl;

cout << "a1 = " << rangeDegrees(a1);

cout << "a2 = " << rangeDegrees(a2) << endl;

cout << "Сравнение углов:" << endl;

if (a1.is\_equal(a2)) {

cout << "Углы равны" << endl;

}

else {

cmpDegrees(a1, a2) ? cout << " > " << endl : cout << " < " << endl;

}

cout << "a1 + a2 = " << a1 + a2;

cout << "a1 - a2 = " << a1 - a2;

cout << "a1 \* num = " << a1 \* num;

cout << "a1 / num = " << a1 / num;

cout << endl << "Перевод углов в радианы:" << endl;

cout << "a1 = " << degrees\_to\_radians(a1) << endl;

cout << "a2 = " << degrees\_to\_radians(a2) << endl;

cout << endl << "Синус угла a1 = " << sinDegrees(a1) << endl;

cout << endl << "Косинус угла a1 = " << cosDegrees(a1) << endl;

cout << endl << "Тангенс угла a1 = " << tgDegrees(a1) << endl;

cout << endl << "Котангенс угла a1 = " << ctgDegrees(a1) << endl;

}

# Выводы:

В процессе выполнения этой лабораторной работы, а познакомился с такими понятиями, как класс, его методы и данные. Научился с ними правильно взаимодействовать и использовать в своих функциях.

# ЛИТЕРАТУРА

1. Уроки программирования на языке С++ [Электронный ресурс]. URL: https://ravesli.com/uroki-cpp (дата обращения 24.09.2021)