**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Почечура Артемий Андреевич

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. **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.

1. **Описание программы**

Реализуется класс Angle для работы с углами. Созданы две поля – градусы и минуты. Реализовано получение значения каждого поля путём вызова функции, изменение значения поля (таким же путём), переопределены операторы заведения переменной, сложения, вычитания, деления на константу, два оператора сравнения (третий не нужен, т.к. его можно выразить из двух других), операторы ввода и вывода. Также реализовано вычисление тригонометрических функций. Некорректный ввод обрабатывается с помощью конструкции try, catch, throw. Все полученные значения переводятся в диапазон 0-360.

Файлы: main.cpp.

1. **Набор тестов и результаты их исполнения**

В самом начале вводим количество тестов:

Select the number of tests:

5

Тест 1: 60 40 50 20 5 (первый угол: 60 градусов 40 минут; второй угол: 50 градусов, 20 минут; затем следует значение константы, которая используется в вычислениях):

Select the values of first angle:

60 40

Select the values of second angle:

50 20

Select constant:

5

Shifting the first angle in range 0-360:

60 40

Shifting the first angle in radian:

1.05883

Sum and difference of angles:

111 0

10 20

Division of first angle and constant:

12 8

Trigonometric functions of first angle:

0.871784 0.48989 1.77955 0.561939

Comparison of angles:

First angle larger than second angle

Тест 2: 200 75 456 20 10

Select the values of first angle:

200 75

Select the values of second angle:

456 20

Select constant:

10

Shifting the first angle in range 0-360:

201 15

Shifting the first angle in radian:

3.51248

Sum and difference of angles:

297 35

104 55

Division of first angle and constant:

20.1 1.5

Trigonometric functions of first angle:

-0.362438 -0.932008 0.388879 2.5715

Comparison of angles:

First angle larger than second angle

Тест 3: aaaaa

Select the values of first angle:

aaaaa

Error: Wrong values

1. **Листинг программы**

// Почечура Артемий Андреевич

// М8О-206Б-20

// Лабораторная работа №2

// 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.

#include <iostream>

#include <math.h>

#include <string>

#include <vector>

#include <algorithm>

#include <map>

#include <queue>

#include <stack>

#include <set>

using namespace std;

const double pi=atan(1.0)\*4;

class Angle{

private:

double degree, minute;

public:

Angle(){

degree=0;

minute=0;

}

Angle(double new\_degree, double new\_minute){

degree=new\_degree;

minute=new\_minute;

}

double get\_degree() const{

return degree;

}

double get\_minute() const{

return minute;

}

void set\_degree(double a){

degree=a;

}

void set\_minute(double a){

minute=a;

}

double degree\_to\_radian(){

double b;

if(get\_degree()>0){

b=get\_degree()+get\_minute()/60;

} else {

b=get\_degree()-get\_minute()/60;

}

return pi\*b/180;

}

void bringing\_to\_0\_360(){

Angle b;

int q,c,l;

if(get\_degree()<0){

set\_minute(get\_minute()\*(-1));

}

q=get\_minute()/60;

set\_degree(get\_degree()+q); //получаем чистые градусы

c=get\_degree()/360;

b.set\_minute(get\_minute()-q\*60); //получаем чистые минуты

if(get\_degree()>=0){

b.set\_degree(get\_degree()-c\*360);

} else {

b.set\_degree(get\_degree()+(c\*(-1)+1)\*360); //т.к. числа отрицательные в данном случае, делаем ещё один лишний круг

if(b.get\_minute()!=0){

b.set\_minute(60+b.get\_minute());

b.set\_degree(b.get\_degree()-1);

}

}

set\_degree(b.get\_degree());

set\_minute(b.get\_minute());

}

double Sin(){

return sin(degree\_to\_radian());

}

double Cos(){

return cos(degree\_to\_radian());

}

double Tan(){

return Sin()/Cos();

}

double Arctan(){

return Cos()/Sin();

}

};

Angle operator+(Angle a, Angle b){

if(a.get\_degree()<0){

a.set\_minute(a.get\_minute()\*(-1));

}

if(b.get\_degree()<0){

b.set\_minute(b.get\_minute()\*(-1));

}

double c=a.get\_degree()\*60+a.get\_minute()+b.get\_degree()\*60+b.get\_minute(); // переводим всё в минуты

int C=c/60; // минуты в градусы

double m=c-C\*60; // остаток минут

Angle g((c-m)/60,m);

if(g.get\_degree()<0){

g.set\_minute(g.get\_minute()\*(-1));

}

return g;

}

Angle operator-(Angle a, Angle b){

if(a.get\_degree()<0){

a.set\_minute(a.get\_minute()\*(-1));

}

if(b.get\_degree()<0){

b.set\_minute(b.get\_minute()\*(-1));

}

double c=a.get\_degree()\*60+a.get\_minute()-b.get\_degree()\*60-b.get\_minute();

int C=c/60;

double m=c-C\*60;

Angle g((c-m)/60,m);

if(g.get\_degree()<0){

g.set\_minute(g.get\_minute()\*(-1));

}

return g;

}

Angle operator/(Angle a, int b){

if(b!=0){

if(a.get\_degree()<0){

a.set\_minute(a.get\_minute()\*(-1));

}

int y=(a.get\_minute())/60;

Angle c((a.get\_degree()+y)/b, (a.get\_minute()-y\*60)/b);

return c;

} else {

throw invalid\_argument("Error: Division on zero\n");

}

}

bool operator>(Angle a, Angle b){

double A=a.degree\_to\_radian();

double B=b.degree\_to\_radian();

if(A>B){

return true;

}

return false;

}

bool operator==(Angle a, Angle b){

double A=a.degree\_to\_radian();

double B=b.degree\_to\_radian();

if(A==B){

return true;

}

return false;

}

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

cout << a.get\_degree() << " " << a.get\_minute();

return stream;

}

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

double b,c;

if(!(cin >> b >> c)){

throw invalid\_argument("Error: Wrong values");

} else {

a.set\_degree(b);

a.set\_minute(c);

}

return stream;

}

// минуты не могут быть отрицательными и зависят от знака градусов

int main() {

ios::sync\_with\_stdio(false);

cin.tie(nullptr); cout.tie(nullptr);

Angle a1,a2;

double b,c,d,e,f;

int N;

cout << "Select the number of tests:\n";

if(!(cin >> N)){

cout << "Error: Wrong values\n";

}

for(int i=0;i<N;i++){

try{

cout << "\nSelect the values of first angle:\n";

cin >> a1;

cout << "Select the values of second angle:\n";

cin >> a2;

cout << "Select constant:\n";

if(!(cin >> f)){

cout << "Wrong values";

continue;

}

Angle A,B;

cout << "Shifting the first angle in range 0-360:\n";

A=a1;

A.bringing\_to\_0\_360();

cout << A << "\n";

cout << "Shifting the first angle in radian:\n";

A=a1;

cout << A.degree\_to\_radian() << "\n";

cout << "Sum and difference of angles:\n";

A=a1+a2;

B=a1-a2;

A.bringing\_to\_0\_360();

B.bringing\_to\_0\_360();

cout << A << "\n";

cout << B << "\n";

cout << "Division of first angle and constant:\n";

A=a1/f;

A.bringing\_to\_0\_360();

cout << A << "\n";

cout << "Trigonometric functions of first angle:\n";

cout << a1.Sin() << " " << a1.Cos() << " " << a1.Tan() << " " << a1.Arctan() << "\n";

cout << "Comparison of angles:\n";

a1.bringing\_to\_0\_360();

a2.bringing\_to\_0\_360();

if(a1>a2){

cout << "First angle larger than second angle\n";

} else if (a1==a2){

cout << "Angles is equal\n";

} else {

cout << "First angle smaller than second angle\n";

}

} catch(invalid\_argument const& arg){

cout << arg.what() << "\n";

return -1;

}

}

return 0;

}

1. **Вывод**

Во время работы над поставленной задачей научился полноценно работать с перегрузкой всевозможных операторов, также потренировался в применении утилит try, catch, throw.

**ЛИТЕРАТУРА**

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