

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Отчёт по Лабораторной работе №2
“Операторы, Литералы“
по курсу “Объектно-Объективное Программирование“
III Семестр

Студент:	Катермин В.С.
Группа:	М8О-208Б-18
Преподаватель:	Журавлёв А.А.
Оценка:	
Дата:	30.09.19

1. **Тема:** Операторы, Литералы в C++.

2. **Код программы:**

angle.h

```
#ifndef D_ANGLE_H
#define D_ANGLE_H

#include <iostream>

struct Angle{
private:
    int deg;
    int min;

public:
    Angle();
    Angle(int d, int m);
    void read(std::istream& is);
    void awrite(std::ostream& os) const;
    int mntsonly();

    double deg_to_rad();

    int to360();

    //Angle& operator+=(const Angle& o);
    Angle operator+(const Angle& o); // operator[] - перегрузка операторов
    Angle operator-(const Angle& o);
    double operator/(const Angle& o);

    double asin();
    double acos();
    double atan();

    int operator>(const Angle& o);
    friend std::istream& operator >> (std::istream& is, Angle& a);
    friend std::ostream& operator << (std::ostream& os, const Angle& a);
};

Angle operator "" _deg (unsigned long long deg);
Angle operator "" _min (unsigned long long min);
std::istream& operator >> (std::istream& is, Angle& a);
std::ostream& operator << (std::ostream& os, const Angle& a);
#endif
```

angle.cpp

```
#include "angle.h"
#include <cmath>

#include <iostream>

Angle::Angle()
: deg(0)
, min(0) {}

Angle::Angle(int d, int m)
: deg(d)
, min(m) {}

Angle operator "" _deg (unsigned long long deg){
    Angle a(deg, 0);
    return a;
}
```

```

Angle operator "" _min (unsigned long long min){
    Angle a(0, min);
    return a;
}

void Angle::read(std::istream& is) {
    is >> deg >> min;
}

void Angle::awrite(std::ostream& os) const {
    os << deg << ' ' << min << '\n';
}

int Angle::mntsonly(){
    if (deg < 0){
        return 60 - abs(min);
    } else {
        return min;
    }
}

double Angle::deg_to_rad(){
    double rad = ((double)deg + ((double)min / 60)) / 180 * M_PI; //M_PI - Пи в cmath
    //(double)angle - перевод angle (заданный в int) из int в double
    return rad;
}

int Angle::to360(){
    if (deg < 0) {
        return 360 + (deg % 360);
    } else {
        return deg % 360;
    }
}

Angle Angle::operator+(const Angle& rhs) {
    Angle result;
    result.deg = deg + rhs.deg;
    result.min = min + rhs.min;
    if (result.min >= 60) {
        result.deg += 1;
        result.min -= 60;
    } else if (result.min <= -60) {
        result.deg -= 1;
        result.min += 60;
    }
    return result;
}

Angle Angle::operator-(const Angle& rhs) {
    Angle result;
    result.deg = deg - rhs.deg;
    result.min = min - rhs.min;
    if (result.min < 0) {
        result.deg -= 1;
        result.min += 60;
    } else if ((result.deg < 0) && (result.min > 0)) {
        result.deg += 1;
        result.min -= 60;
    }
    return result;
}

```

```

double Angle::operator/(const Angle& rhs) {
    double result = (((double)deg + ((double)min / 60)) / ((double)rhs.deg + ((double)rhs.min / 60)));
    return result;
}

double Angle::asin() {
    return sin((((double)deg + ((double)min / 60)) / 180 * M_PI);
}

double Angle::acos() {
    return cos((((double)deg + ((double)min / 60)) / 180 * M_PI);
}

double Angle::atan(){
    return tan((((double)deg + ((double)min / 60)) / 180 * M_PI);
}

int Angle::operator>(const Angle& rhs) {
    int result = 0;
    if (deg > rhs.deg){
        result = 1;
    } else if (deg < rhs.deg) {
        result = -1;
    } else if (deg == rhs.deg) {
        if (min > rhs.min) {
            result = 1;
        } else if (min < rhs.min) {
            result = -1;
        }
    }
    return result;
}

std::istream& operator >> (std::istream& is, Angle& a){
    a.read(is);
    return is;
}

std::ostream& operator << (std::ostream& os, const Angle& a){
    a.awrite(os);
    return os;
}

```

lab2.cpp

```

#include <iostream>

#include "angle.h"

int main(){
    Angle a;
    Angle b;
    //std::cout << "Enter numbers:\n[degrees] and [minutes] (-59...59)" << std::endl;
    //a.read(std::cin);
    //b.read(std::cin);
    std::cout << 17_deg + 13_min;
    std::cin >> a;
    std::cin >> b;
    std::cout << "RADS:\n";
    std::cout << a.deg_to_rad() << std::endl;
    std::cout << "0-360 DEG:\n";
    std::cout << a.to360() << " " << a.mntsonly() << std::endl;
    std::cout << "SUM:\n";
}

```

```

    //(a + b).awrite(std::cout);
    std::cout << (a + b);
    std::cout << "SUB:\n";
    //(a - b).awrite(std::cout);
    std::cout << (a - b);
    std::cout << "DIVIDE:\n";
    std::cout << (a / b) << std::endl;
    std::cout << "SIN:\n";
    std::cout << a.asin() << std::endl;
    std::cout << "COS:\n";
    std::cout << a.acos() << std::endl;
    std::cout << "TAN:\n";
    std::cout << a.atan() << std::endl;

    switch (a > b) {
    case 0:
        std::cout << "A = B" << std::endl;
        break;

    case 1:
        std::cout << "A > B" << std::endl;
        break;

    case -1:
        std::cout << "A < B" << std::endl;
        break;
    }

    return 0;
}

```

3. Ссылка на репозиторий:

https://github.com/GitGood2000/oop_exercise_02

4. Набор testcases:

test_00.test

30 16

20 47

test_00.result

17 13

RADS:

0.528253

0-360 DEG:

30 16

SUM:

51 3

SUB:

9 29

DIVIDE:

1.4563

SIN:

0.504025

COS:

0.863689

TAN:

0.583573

A > B

test_01.test

361 23

89 48

test_01.result

17 13
RADS:
6.30733
0-360 DEG:
1 23
SUM:
451 11
SUB:
271 35
DIVIDE:
4.02431
SIN:
0.0241414
COS:
0.999709
TAN:
0.0241484
 $A > B$

test_02.test

30 13
60 43

test_02.result

17 13
RADS:
0.52738
0-360 DEG:
30 13
SUM:
90 56
SUB:
-31 30
DIVIDE:
0.497667
SIN:
0.503271
COS:
0.864128
TAN:
0.582403
 $A < B$

test_03.test

60 3
60 3

test_03.result

17 13
RADS:
1.04807
0-360 DEG:
60 3
SUM:
120 6
SUB:
0 0
DIVIDE:
1
SIN:
0.866461
COS:
0.499244

TAN:
1.73555
A = B

test_04.test
-293 -43
-142 -51

test_04.result
17 13
RADS:
-5.12632
0-360 DEG:
67 17
SUM:
-436 -34
SUB:
-150 -52
DIVIDE:
2.05612
SIN:
0.915546
COS:
0.402214
TAN:
2.27626
A < B

5. Результаты выполнения тестов:

```
user@PSB133S01ZFH:~/3sem_projects/oop_lab2/oop_exercise_02/tests$ bash test.sh ../build/lab2
Test test_00.test: SUCCESS
Test test_01.test: SUCCESS
Test test_02.test: SUCCESS
Test test_03.test: SUCCESS
Test test_04.test: SUCCESS
```

6. Объяснение результатов работы программы:

- 1) Программа вводит данные двух углов (a, b)
- 2) Программа выводит значение угла a в радианах
- 3) Программа выводит значение угла a в диапазоне от 0 до 360 градусов в градусах/минутах
- 4) Программа выводит результат сложения углов a и b в градусах/минутах с помощью перегрузки операторов
- 5) Программа выводит результат вычитания углов a и b в градусах/минутах с помощью перегрузки операторов
- 6) Программа выводит результат деления углов a и b с помощью перегрузки операторов
- 7) Программа выводит синус, косинус и тангенс угла a
- 8) Программа выводит результат проверки отношения равенства углов a и b с помощью перегрузки операторов.

7. **Вывод:** 1) Ознакомились с операторами и литералами в C++ и усвоили навык работы с ними; 2) Написана программа, производящая операции с помощью пользовательских литералов и метода перегрузки операторов.