

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:	<b>30.09.19</b>

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:
    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);
};

int operator "" _deg (unsigned long long deg);
int operator "" _min (unsigned long long min);

#endif
```

**angle.cpp**

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

#include <iostream>

int operator "" _deg (unsigned long long deg){
    return (deg / 60) % 360;
}

int operator "" _min (unsigned long long min){
    return min % 60;
}

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;
}

```

## 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 << "20593 minutes is:\n";
    std::cout << 20593_deg << " " << 20593_min << std::endl;
    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 << "SUB:\n";
    (a - b).awrite(std::cout);
    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](https://github.com/GitGood2000/oop_exercise_02)

**4. Набор testcases:**

**test\_00.test**

30 16

20 47

**test\_00.result**

20593 minutes is:

343 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**

20593 minutes is:

343 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**

20593 minutes is:

343 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**

20593 minutes is:

343 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**

20593 minutes is:

343 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) конвертирует их из углов/минут в минуты
- 2) Программа переводит значение угла в минутах (20593) в градусы и минуты (343 13)
- 3) Программа выводит значение угла a в радианах
- 4) Программа выводит значение угла a в диапазоне от 0 до 360 градусов в градусах/минутах
- 5) Программа выводит результат сложения углов a и b в градусах/минутах с помощью перегрузки операторов
- 6) Программа выводит результат вычитания углов a и b в градусах/минутах с помощью перегрузки операторов
- 7) Программа выводит результат деления углов a и b с помощью перегрузки операторов
- 8) Программа выводит синус, косинус и тангенс угла a
- 9) 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.