

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

**Лабораторная работа №7**

по дисциплине: Объектно-ориентированное программирование

тема: **«Исключительные ситуации в C++»**

Выполнил: ст. группы ПВ-233

Ситников Алексей Павлович

Проверил:

Белгород 2025 г.

## Вариант 3 (13)

### Задание:

Разработать абстрактный класс решения уравнений. Реализовать класс для решения тригонометрических уравнений (потомок от абстрактного класса), предусмотреть обработку исключений при решении тригонометрических уравнений (ОДЗ тригонометрических функций).

### Заголовочный файл:

```
#include <iostream>
#include <stdexcept>

class SolveEquation{
public:
    virtual double sum(double x, double y) = 0;
    virtual double dif(double x, double y) = 0;
    virtual double mul(double x, double y) = 0;
    virtual double div(double x, double y) = 0;
    virtual double pow_(double x, double y) = 0;
};

class SolveTrigonometric : public SolveEquation{
public:
    double sin_(double x);
    double cos_(double x);
    double tg_(double x);
    double ctg_(double x);
    double arcCos_(double x);
    double arcSin_(double x);
    double arcCtg_(double x);
    double arcTg_(double x);
    double sum(double x, double y);
    double dif(double x, double y);
    double mul(double x, double y);
    double div(double x, double y);
    double pow_(double x, double y);
};
```

### Реализация:

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

#define e 1e-15
```

```

double compare(double x) {
    return fabs(x) < e ? 0 : x;
}

double SolveTrigonometric::sin_(double x) {
    x = x * M_PI/180;
    return compare(sin(x));
}

double SolveTrigonometric::cos_(double x) {
    x = x * M_PI/180;
    return compare(cos(x));
}

double SolveTrigonometric::tg_(double x) {
    if(cos_(x) == 0) {
        throw std::runtime_error("tg(x) doesn't exist");
    }
    return compare(div(sin_(x), cos_(x)));
}

double SolveTrigonometric::ctg_(double x) {
    if(sin_(x) == 0) {
        throw std::runtime_error("ctg(x) doesn't exist");
    }
    return compare(div(cos_(x), sin_(x)));
}

double SolveTrigonometric::arcCos_(double x) {
    if(x < -1 || x > 1) {
        throw std::runtime_error("x out of range");
    }
    return compare(acos((double)x) * 180/M_PI);
}

double SolveTrigonometric::arcSin_(double x) {
    if(x < -1 || x > 1) {
        throw std::runtime_error("x out of range");
    }
    return compare(asin((double)x) * 180/M_PI);
}

double SolveTrigonometric::arcCtg_(double x) {
    return compare(atan(1/x) * 180/M_PI);
}

double SolveTrigonometric::arcTg_(double x) {
    return compare(atan((double)x) * 180/M_PI);
}

double SolveTrigonometric::sum(double x, double y) {
    return compare(x+y);
}

double SolveTrigonometric::dif(double x, double y) {
    return compare(x-y);
}

```

```
double SolveTrigonometric::mul(double x, double y){
    return compare(x*y);
}

double SolveTrigonometric::div(double x, double y){
    if(y == 0){
        throw std::runtime_error("ZeroDivisionError");
    }
    return compare(x/y);
}

double SolveTrigonometric::pow_(double x, double y){
    if(x < 0 && y != std::floor(y)){
        throw std::runtime_error("ComplexResultError");
    }
    return compare(pow(x, y));
}
```

main

```
#include <iostream>
#include "classes.h"

int main() {
    SolveTrigonometric t;

    try{
        std::cout << t.sum(t.tg_(0), t.tg_(90));
    }catch(const std::exception &ex) {
        std::cout << ex.what();
    }
    return 0;
}
```

Вывод:

```
C:\Users\admin\CLionProjects\OOP\cmake-build-debug\OOP.exe
tg(x) doesn't exist
Process finished with exit code 0
```

```
std::cout << t.arcCos_(-2);
```

```
C:\Users\admin\CLionProjects\OOP\cmake-build-debug\OOP.exe
x out of range
Process finished with exit code 0
```

**Вывод:** в ходе проделанной работы я научился использовать исключения.