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 С++»**

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:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

std::cout **<<** t.arcCos\_(-2);

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

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