

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: ООП

тема: «Исключительные ситуации в C++»

Выполнил: ст. группы ПВ202

Аладиб язан

Проверил:

Буханов Д.Г.

Белгород 2022

**Цель работы:** Получение теоретических знаний об исключительных ситуациях в C++. Получение практических навыков при работе с исключениями в C++.

**Задание:**

1. Изучить теоретические сведения об исключениях в C++.
2. Изучить самостоятельно стандартные классы для исключений предусмотренных в C++.
3. Разработать программу в соответствии с заданным вариантом задания.
4. Оформить отчет.

**Задание Варианта :**

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).

## Выполнение работы:

```
#include <iostream>
#include <locale.h>
#include <math.h>
#include <cstdlib>
#include <windows.h>

using namespace std;

class EquatException {
protected:
    string str;
public:
    EquatException(string mess) :
        str(mess) {};
    string error() {
        return (" !!!" + str);
    };
};

class FirstNull : public EquatException {
public:
    FirstNull(string) :
        EquatException(" Equation is not quadratic! ") {};
};

class NegDiscr : public EquatException {
public:
    NegDiscr(string) :
        EquatException(" Discriminant is negative ") {};
};

class TrigODZ : public EquatException {
public:
    TrigODZ(string) :
        EquatException(" does not satisfy the domain: [-1;1] ") {};
};

class Equations {
public:
    virtual void print() = 0;
    virtual void write() = 0;
    virtual void root() = 0;
    virtual double findDiscrim() = 0;
    virtual void print_solution() = 0;
    virtual void solve_equation() = 0;
};
```

```

class Square_Trig_equation : public Equations {
private:
    double a, b, c, x1, x2;
    int m;
public:
    void print() {
        cout << "a = " << a << " b = " << b << " c = " << c;
    };

    void write() {
        cout << "Enter the coefficients a, b, c of the trigonometric quadratic
equation: " << endl;
        cin >> a;
        cin >> b;
        cin >> c;
        cout << "Enter the type of equation you want to solve sin - 1, cos - 2,
tg - 3 " << endl;
        cin >> m;
    };

    void test_null_a() {
        if (a == 0.0)
            throw FirstNull("");
    }

    void test_discr(double discr) {
        if (discr < 0.0) {
            cout << "D = " << discr << endl;
            throw NegDiscr("");
        }
    }

    void test_trig(double x) {
        if (fabs(x) > 1.0) {
            cout << "Root " << x;
            throw TrigODZ("");
        }
    }

    void root() {
        double d = findDiscrim();
        test_null_a();
        x1 = (-b + (sqrt(d))) / (2 * a);
        x2 = (-b - (sqrt(d))) / (2 * a);
    }

    double findDiscrim() {
        double d;

```

```

        d = (b * b) - (4 * a * c);
        test_discr(d);
        return d;
    }

    void print_solution() {
        cout << "Roots:" << endl;
        switch (m) {
            case 1:
                cout << "x1=(-1)^n*arcsin(" << x1 << ") + pi*n" << endl;
                cout << "x2=(-1)^n*arcsin(" << x2 << ") + pi*n" << endl;
                test_trig(x1);
                test_trig(x2);
                break;
            case 2:
                cout << "x1=+-arccos(" << x1 << ") + 2pi*n" << endl;
                cout << "x2=+-arccos(" << x2 << ") + 2pi*n" << endl;
                test_trig(x1);
                test_trig(x2);
                break;
            default:
                cout << "x1=arctg" << x1 << " + pi*n" << endl;
                cout << "x2=arctg" << x2 << " + pi*n" << endl;
                break;
        }
    }
}

void solve_equation() {
    write();
    root();
    print_solution();
}

};

int main() {
    SetConsoleOutputCP(CP_UTF8); // Подключение русского языка
    Square_Trig_equation eq;
    try {
        eq.solve_equation();
    }
    catch (EquatException& r) {
        cout << r.error() << endl;
        exit(0);
    }
    return 0;
}

```