САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Практическая работа №9

«Обработка исключительных операций»

Выполнил: Мороз И.О.

Группа К3120

Проверил: Осипов Н.А.

Санкт-Петербург

2018 г.

# Упражнение 1.

1. В программе с предыдущего занятия был реализован обработчик исключительных операций. Для этого в классе Student был создан класс ExScore. В его конструктор передаются две переменны: функция, в которой произошла ошибка, и неверное значение. Потенциальную опасность представляет функция, в которой записываются оценки студента. Если оценка больше 5, то срабатывает обработчик исключительных операций

class ExScore

{

public:

string origin;

int iValue;

ExScore(string or1, int sc)

{

origin = or1;

iValue = sc;

}

};

//…

void set\_scores(int student\_scores[])

{

for (int i = 0; i < 5; ++i) {

if (student\_scores[i] > 5)

{

throw ExScore("в функции set\_scores()", student\_scores[i]);

}

scores[i] = student\_scores[i];

}

}

//…

try

{

student01.set\_scores(scores);

double average\_score = sum / 5.0;

student01.set\_average\_score(average\_score);

cout << "Average ball for " << student01.get\_name() << " " << student01.get\_last\_name() << " is " << student01.get\_average\_score() << endl;

}

catch (Student::ExScore ex)

{

cout << "\nОшибка инициализации " << ex.origin;

cout << "\nВведённое значение оценки " << ex.iValue << " является недопустимым\n";

}

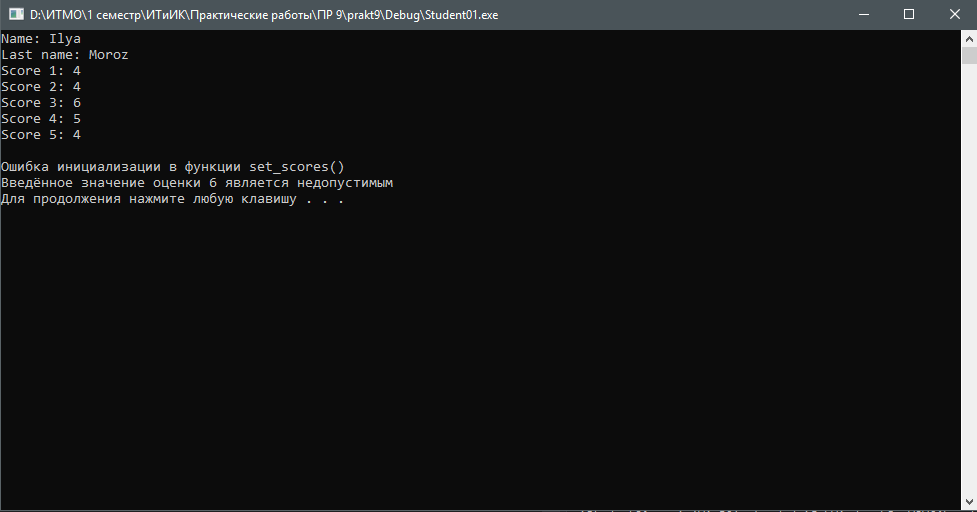


Рисунок 1 – обработчик исключительных операций обработал исключение

# Упражнение 2.

1. Был создан класс Triangle. Был создан метод, возвращающий значение площади треугольника с известными сторонами

#include "Triangle.h"

#include <iostream>

#include <math.h>

using namespace std;

Triangle::Triangle() {}

Triangle::Triangle(int a, int b, int c)

{

Triangle::SetSides(a, b, c);

}

void Triangle::SetSides(int a, int b, int c)

{

Triangle::set\_a(a);

Triangle::set\_b(b);

Triangle::set\_c(c);

}

double Triangle::GetArea()

{

double p = (a + b + c) / 2;

return sqrt(p \* (p - a)\*(p - b)\*(p - c));

}

void Triangle::ShowTriangle()

{

cout << "\nSide a: " << a << "; side b: " << b << "; side c: " << c << ";\n";

}

void Triangle::set\_a(int side)

{

a = side;

}

void Triangle::set\_b(int side)

{

b = side;

}

void Triangle::set\_c(int side)

{

c = side;

}

int Triangle::get\_a()

{

return a;

}

int Triangle::get\_b()

{

return b;

}

int Triangle::get\_c()

{

return c;

}

Triangle::~Triangle()

{

}

1. В методе main был создан объект Triangle. Был реализован ввод трёх сторон треугольника и вывод его площади в консоль

Triangle \*tr1 = new Triangle();

int a, b, c;

cout << "Side a: ";

cin >> a;

cout << "Side b: ";

cin >> b;

cout << "Side c: ";

cin >> c;

tr1->SetSides(a, b, c);

cout << "Area: " << tr1->GetArea() << endl;

system("pause");

delete tr1;

return 0;

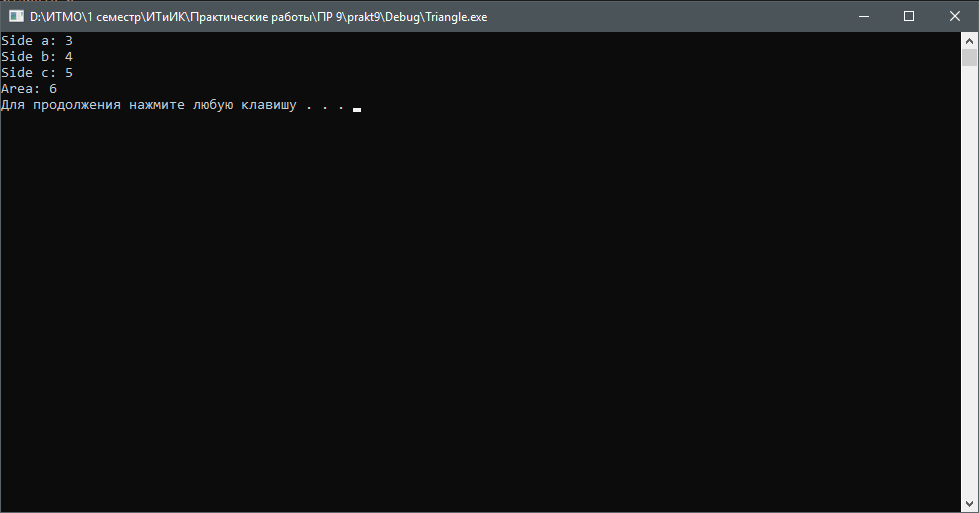
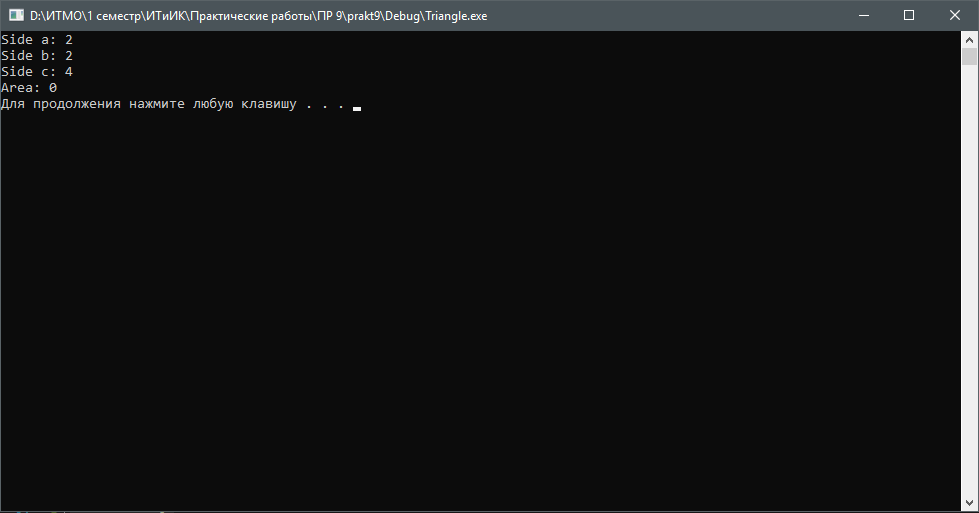


Рисунок 2 – вывод площади в консоль

  
Рисунок 3 – результат работы программы, если ввести стороны несуществующего треугольника

1. Для обработки исключений был создан класс ExScore. В методе присваивания значений сторонам треугольника была написана проверка на корректность введённых значений

#include "ExScore.h"

#include <string>

ExScore::ExScore(string or1, int sc)

{

origin = or1;

iValue = sc;

}

//…

void Triangle::SetSides(int a, int b, int c)

{

if (a + b <= c)

{

throw ExScore("в функции SetSides a + b <= c", c);

}

if (a + c <= b)

{

throw ExScore("в функции SetSides a + c <= b", b);

}

if (b + c <= a)

{

throw ExScore("в функции SetSides b + c <= a", a);

}

Triangle::set\_a(a);

Triangle::set\_b(b);

Triangle::set\_c(c);

}

//…

try

{

cout << "Side a: ";

cin >> a;

cout << "Side b: ";

cin >> b;

cout << "Side c: ";

cin >> c;

tr1->SetSides(a, b, c);

cout << "Area: " << tr1->GetArea() << endl;

}

catch (ExScore ex)

{

cout << "\nОшибка инициализации " << ex.origin;

cout << "\nЗначение " << ex.iValue << " является недопустимым\n";

}

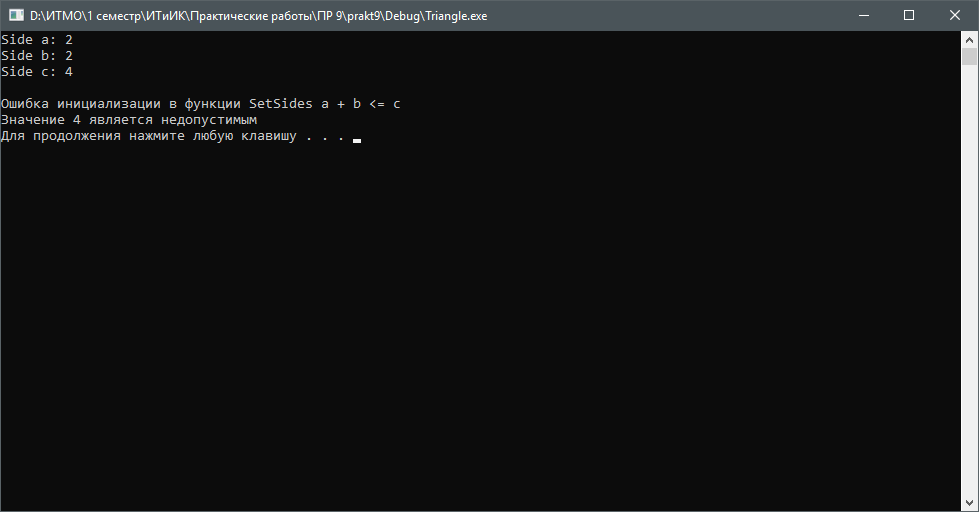


Рисунок 4 – пример обработки исключения

# Вывод

Был изучен обработчик исключительных функций. В созданом ранее классе Student данный обработчик был использован для проверки введённых оценок на их корректность. Для закрепления полученных знаний был создан класс Triangle, в котором обработчик исключений проверял корректность введённых сторон (сумма двух сторон треугольника всегда болше третьей).

# Контрольные вопросы

1. Как работает конструкция try-catch?

Throw возвращает объект класса ExScore сообщая, что произошло исключение. Catch обрабатывает полученный объект.

1. Возможно ли создать данную конструкцию без catch?

Нет