**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 11**

**«ОБРАБОТКА ИСКЛЮЧЕНИЙ»**

Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Прикладное программирование»

|  |  |
| --- | --- |
| Преподаватель:  Сорокин Д.Ю.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2019г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2335  Котлицкий С.А. |

Санкт-Петербург

2018/2019

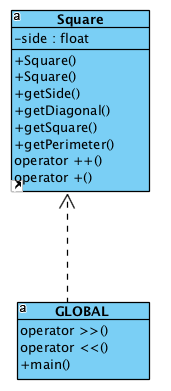
**Цель работы:** продемонстрировать работу механизма обработки исключительных ситуаций

**Практическое задание:** продемонстрировать обработку исключительных ситуаций в конструкторе класса из 8 лабораторной работы, применив условия на значения параметров.

**Индивидуальное задание:**

1. Квадрат.

Диаграмма классов программы представлена ниже (Рисунок 1):



*Рисунок 1 Диаграмма классов программы*

Код программы:

#include <iostream>

#include <math.h>

using namespace std;

class Square {

float side;

public:

Square(){

side = 1;

}

Square(float side){

this -> side = side;

}

float getSide(){

return side;

}

float getDiagonal(){

return sqrt(2 \* side \* side);

}

float getSquare(){

return side \* side;

}

float getPerimeter(){

return side \* 4;

}

Square&operator++(int){

side++;

return \*this;

}

Square&operator+(Square&C) {

Square T(0);

T.side = this -> side + C.side;

return T;

}

friend istream&operator>>(istream&i, Square&C);

friend ostream&operator<<(ostream&o, Square&C);

};

istream&operator>>(istream&i, Square&C) {

cout << "Введите длину стороны" << endl;

i >> C.side;

return i;

}

ostream&operator<<(ostream&o, Square&C) {

o << "side = " << C.side << endl;

return o;

}

void tryCatch(Square \*B) {

try{

cin>>\*B;

if(B -> getSide() == 0)

throw 0;

}

catch(int code) {

cout << "Error! Code " << code << endl;

tryCatch(B);

}

};

int main(int argc, const char \* argv[]) {

Square \*A = new Square(5), \*B = new Square();

tryCatch(B);

Square T = \*A+\*B;

T++;

cout << "Side A " << A -> getSide() << "\nSide B " << B -> getSide() << endl;

cout << "Perimeter A " << A -> getPerimeter() << "\nPerimeter B " << B -> getPerimeter() << endl;

cout << "Diagonal A " << A -> getDiagonal() << "\nDiagonal B " << B -> getDiagonal() << endl;

cout << "Square A " << A -> getSquare() << "\nSquare B " << B -> getSquare() << endl;

cout << "T side " << T.getSide() << endl;

cout<<\*A;

delete A;

delete B;

return 0;

}

Протокол программы:

Введите длину стороны

0

Error! Code 0

Введите длину стороны

4

Side A 5

Side B 4

Perimeter A 20

Perimeter B 16

Diagonal A 7.07107

Diagonal B 5.65685

Square A 25

Square B 16

T side 10

side = 5

Program ended with exit code: 0