



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Информационная безопасность»

ОТЧЕТ

по лабораторной работе № 6

по учебной дисциплине «Алгоритмические языки»

на тему: «Исключения»

Вариант 9 (18)

Выполнил:

Студент 1 курса, гр. ИУ8-24

Ожогин Михаил

2024 г.

Цель работы:

На примере класса «Треугольник» изучить исключения, производные классы от исключений и работу с ними.

Задачи работы:

- Изучить необходимые учебные материалы, посвященные исключениям в языке Си++
- Разработать программу на языке Си++ для решения заданного варианта задания
- Отладить программу
- Представить результаты работы программы
- Подготовить отчет по лабораторной работе

Условие задачи:

1. Описание класса Triangle:

- Приватные поля класса:
 1. Длины сторон a, b, c
- Публичные поля класса:
 1. Конструктор с заполнением всех полей
 2. Функция для печати длин сторон
 3. Функция для проверки валидности треугольника

2. Описание класса TriangleException:

- Производный от класса `std::invalid_argument`
- Приватные поля класса:
 1. Длины сторон a, b, c
- Публичные поля класса:
 1. Конструктор с заполнением всех полей
 2. Функция для печати длин сторон

3. Отработать примеры несуществующих и существующих треугольников с помощью try/catch блока

Выполнение работы:

Заголовочный файл класса Triangle и TriangleException – **triangle.hpp**:

```
#ifndef TRIANGLE_HPP_
#define TRIANGLE_HPP_

#include <exception>

class TriangleException : public std::invalid_argument {
public:
    explicit TriangleException(const double&, const double&, const double&);
    void print() const;
private:
    double _a;
    double _b;
    double _c;
};

class Triangle {
public:
    explicit Triangle(const double&, const double&, const double&);
    void print() const;
    bool isValid() const;
private:
    double _a;
    double _b;
    double _c;
};

TriangleException::TriangleException(const double& a, const double& b,
const double& c)
```

```

        : std::invalid_argument("Triangle doesn't exists"), _a(a),
        _b(b), _c(c) {}

void TriangleException::print() const {
    std::cout << "Triangle sides: " << _a << ", "
                << _b << ", "
                << _c << std::endl;
}

Triangle::Triangle(const double& a, const double& b, const double& c) :
_a(a), _b(b), _c(c) {
    if(!isValid()) {
        throw TriangleException(_a, _b, _c);
    }
}

void Triangle::print() const {
    std::cout << "Triangle sides: " << _a << ", "
                << _b << ", "
                << _c << std::endl;
}

bool Triangle::isValid() const {
    if((_a + _b > _c) && (_a + _c > _b) && (_b + _c > _a)) {
        return true;
    }
    return false;
}

#endif // TRIANGLE_HPP_

```

Файл реализации – **main.cpp**:

```

#include <iostream>

#include "../include/triangle.hpp"

```

```
int main() {  
    try {  
        Triangle validTriangle(3, 4, 5);  
        validTriangle.print();  
    } catch (const TriangleException& ex) {  
        std::cout << "Exception caught: " << ex.what() << std::endl;  
        ex.print();  
    }  
  
    try {  
        Triangle invalidTriangle(1, 2, 4);  
        invalidTriangle.print();  
    } catch (const TriangleException& ex) {  
        std::cout << "Exception caught: " << ex.what() << std::endl;  
        ex.print();  
    }  
  
    return 0;  
}
```

Вывод:

В ходе выполнения лабораторной работы было успешно реализовано использование исключений в собственных классах, создание производного класса от исключения, использование блока try/catch. Это позволило познакомиться с применением исключений в практических задачах программирования.