Объектно-ориентированное программирование на языке C++

Обработка исключений C++

Базовые понятия

Исключение (англ. exception) – ошибка в программе, приводящая к невозможности (бессмысленности) дальнейшего выполнения алгоритма.

Обработка исключений (англ. exception handling) – механизм языка, предназначенный для описания реакции программы на исключения

Синтаксис

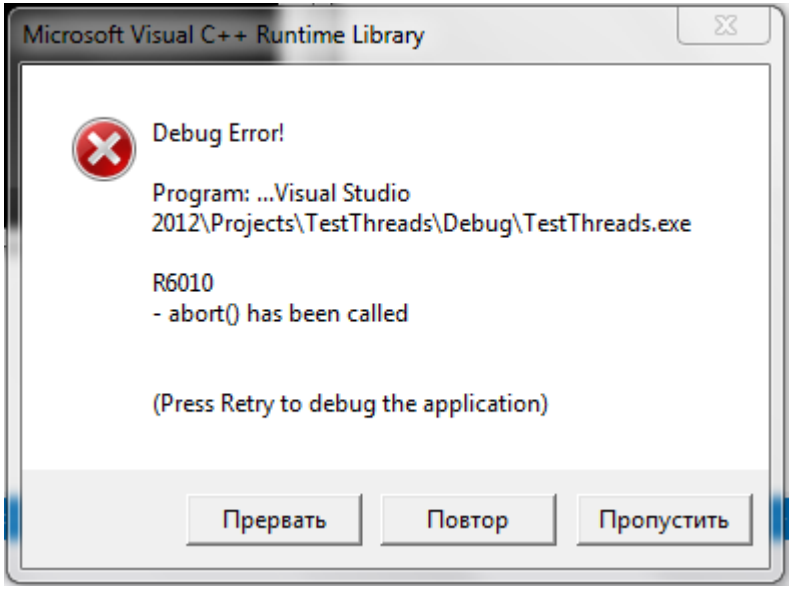
```
try {
   // Код, который может содержать ошибку
}
catch(тип_исключения имя_переменной) {
   // Код обработчика исключения
}
catch(тип_исключения2 имя_переменной2) {
   // Код обработчика исключения
}
catch(…) {
   // Код обработчика исключения
}
```

Вызов исключения:

```
throw(параметр);
```

Внимание

Если сгенерировать исключение, но не перехватить его, программа аварийно завершится



main.cpp

(Global Scope) main()

```cpp
#include <iostream>
#include <clocale>
using namespace std;

void func(int *mas, int size){
    try {
        for (int i = 0; i < size; i++){
            if (mas[i] < 0) throw 0;
            if (mas[i] > 7) throw 1;
            mas[i] += -1;
        }
    } catch (int ex){
        if (ex == 0) std::cout << "Какое-то из элементов массива меньше нуля\n";
        if (ex == 1) std::cout << "Какое-то из элементов массива больше семи\n";
    }
}

void func_without_handler(int *mas, int size){
    for (int i = 0; i < size; i++){
        if (mas[i] < 0) throw 'a';
        if (mas[i] > 7) throw 'b';
        mas[i] *= -1;
    }
}

int main(){
    setlocale(LC_ALL, "");
    int m[10] = {1,4,7,3,5,9,3,5,6,0};
    for (int i = 0; i < 10; i++) cout << m[i] << " "; cout << endl;
    func(m, 10);
    for (int i = 0; i < 10; i++) cout << m[i] << " "; cout << endl;
    try {
        func_without_handler(m, 10);
        for (int i = 0; i < 10; i++) cout << m[i] << " "; cout << endl;
    }
    catch(char ex) {
        if (ex == 'a') std::cout << "Какое-то из элементов массива меньше нуля\n";
        if (ex == 'b') std::cout << "Какое-то из элементов массива больше семи\n";
    }
    return 0;
}
```

std::exception

```
#include <exception>
```

`std::exception` – базовый класс

`const char *std::exception::what()`
возвращает детали исключения

`std::runtime_error` – информация об ошибке во время выполнения программы

```
if (something_wrong)
throw std::runtime_error("Упс");
```

Стандартные исключения

Модуль <stdexcept>

1. `std::length_error` – генерируется, когда размер контейнера превышает максимально допустимый
2. `std::invalid_argument` – генерируется, когда на вход контейнерам подаются неправильные данные
3. `std::out_of_range` – генерируется при выходе за границы контейнера

Задание:

1. Создать класс, использующий массив целых чисел. Функции: добавление и удаление элементов.
2. Создать классы исключения, отнаследованные от класса std::exception.
   Генерировать исключения в следующих случаях:
   – Если добавляемое число меньше нуля
   – Если при вызове функции удаления массив пуст

Домашнее задание

1. Создать класс, описывающий треугольник, как массив из трех точек. Функции: Установка значения точек и вычисление периметра.
2. Создать классы исключения, отнаследованные от класса std::exception.
   Генерировать исключения в следующих случаях:
   – Если не выполняется неравенство треугольника
   – Если три точки лежат на одной прямой.