



C++. Уровень 3

Урок 8

Механизм исключительных ситуаций

Возбуждение и обработка ситуаций.
Свертка стека. Стандартные классы
исключительных ситуаций.

План урока

- Введение в исключения.
- Обработка исключений.
- Генерация исключений за пределами блока try.
- Обработчик catch-all.
- Класс-исключение.



-
- Исключения и наследования.
 - `std::exception`.
 - Повторная генерация исключений.
 - Функциональный try-блок.
 - Недостатки и опасности использования исключений.



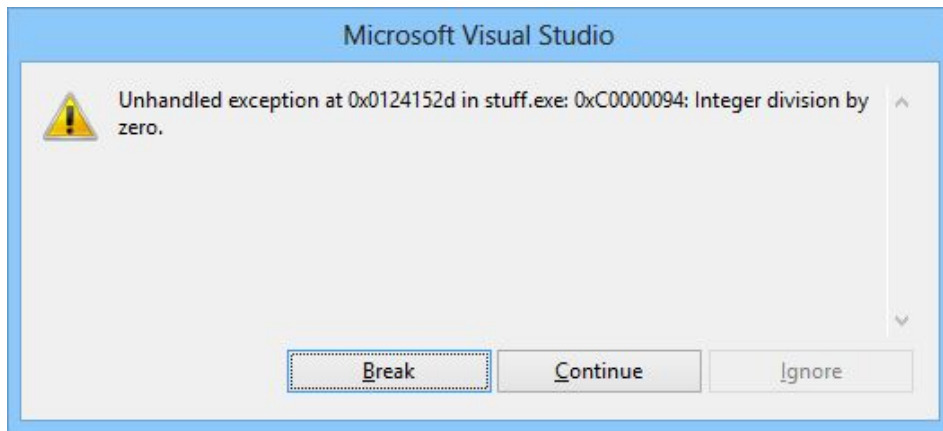
A photograph of a light blue ceramic cup filled with coffee, sitting on a burlap surface surrounded by scattered dark coffee beans. The image is dimmed to serve as a background for the title.

Введение в исключения

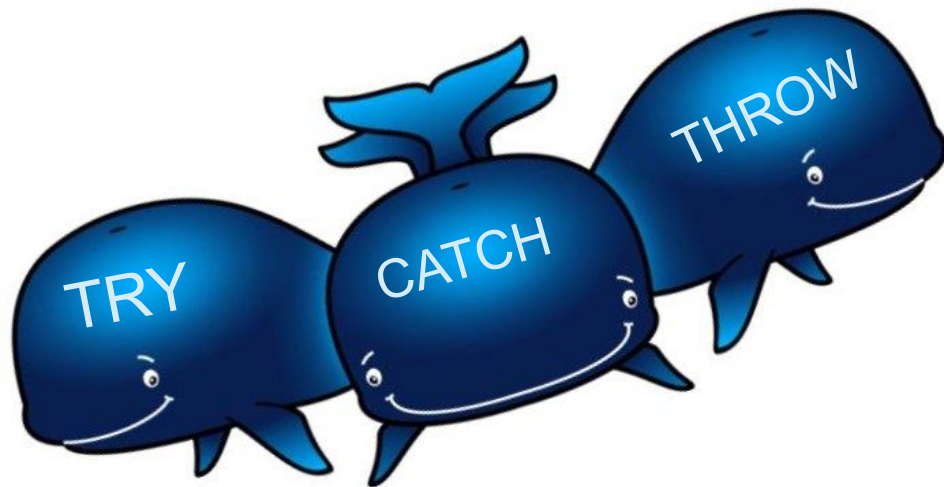


Введение в исключения

Исключение — это реакция на нестандартную ситуацию, возникшую на время выполнения программы.



Ключевые слова исключений



Выбрасывание исключений

Оператор throw

используется, чтобы
сигнализировать о
возникновении исключения
или ошибки.



Ищем исключения



В блоке **try** должны
генерироваться
исключения.



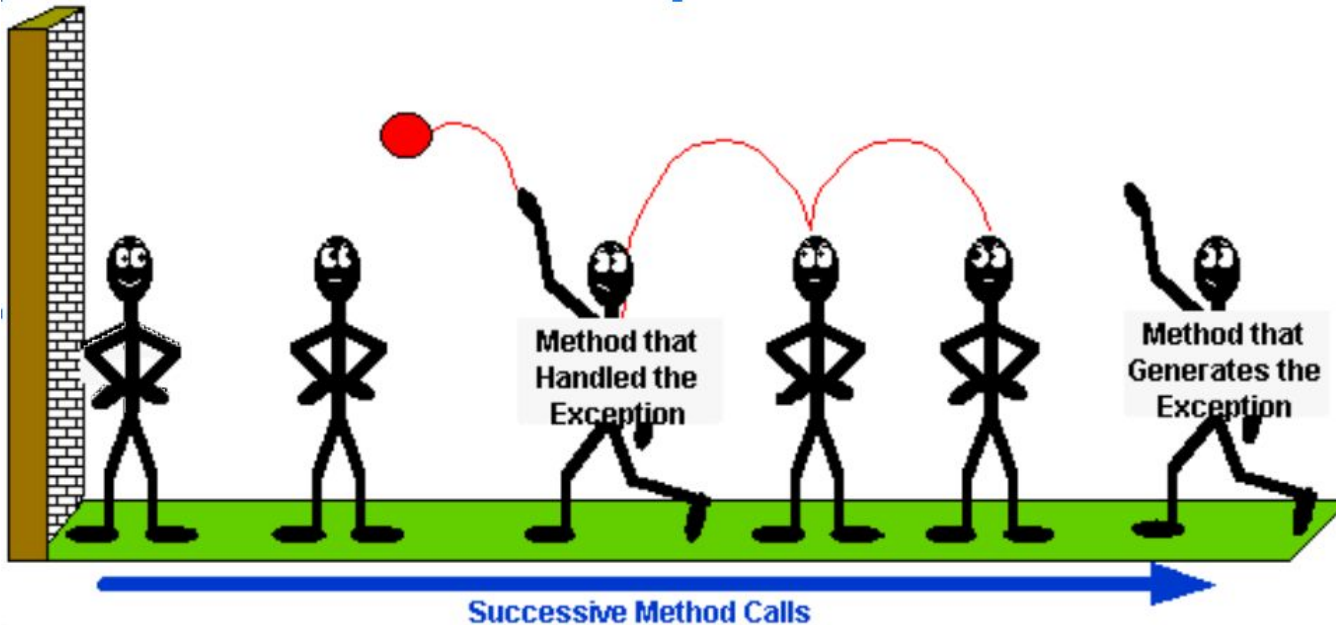
Обработка исключений



Ключевое слово **catch** используется для определения блока кода, который обрабатывает исключения определенного типа данных.



Генерация исключений за пределами блока try



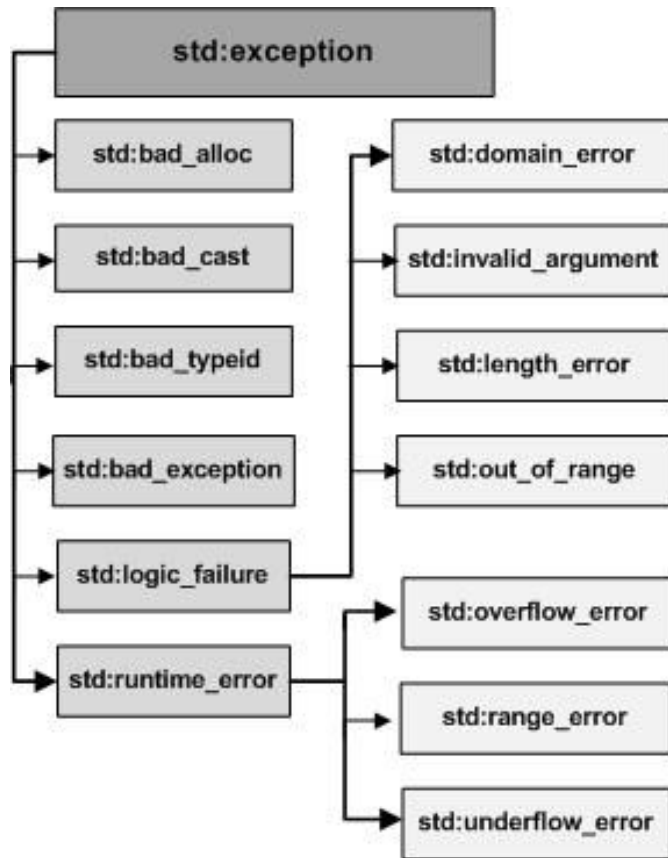
Обработчик catch-all



Класс-исключение

Весь список исключений:

en.cppreference.com/w/cpp/error/exception



Функциональный блок try

Функциональные try-блоки позволяют:

- записывать ошибки в лог-файл перед их передачей на уровень выше в стеке вызовов;
- изменять тип выбрасываемого исключения.

```
Child(int age) try : Parent(age)
```



Недостатки использования исключений

- Очистка памяти.
- Исключения и деструкторы.
- Проблемы с производительностью.





Решите задачи



Что будет выведено на экран?

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

int main() {
    int x;
    try {
        try {
            try {
                throw x;
            } catch(double) {cout << "catch(double)";}
        } catch(int) {cout << "catch(int)";}
    } catch(char a) {cout << "catch(char)"; }
    return 0;
}
```



Что будет выведено на экран?

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

struct A {
    ~A() {
        cout << "A";
    }
};

struct B : public A {
    B() {
        throw 1;
    }
    ~B() {
        cout << "B";
    }
};

int main() {
    try {
        B b;
    }
    catch(...) { }
```



Вопросы участников

