

3.23 Понятие исключений. Синтаксис использования throw и конструкции try... catch. Что является и что не является исключениями. Пример ошибки времени выполнения, не являющейся исключением.

Определение *Исключение* — нестандартная ситуация, возникающая в ходе выполнения программы.

В плюсах можно бросать что угодно. Например, int

```
int f(int x, int y) {  
    if (y == 0)  
        throw 1;  
    return x / y;  
}
```

Try...catch

```
try {  
    f(1, 0);  
} catch (int x) {  
  
}
```

Try может поймать только то, что с помощью оператора throw брошено (только тот вид ошибки, который мы указали, только на том уровне, на котором мы хотим) Например, если мы вызвали из try ... catch в main какую-то функцию, эта функция запустила еще функцию, а та еще десять, и в какой-то из них сработало исключение, это исключение полетит вверх до main, и если оно не поймалось, выдаст ошибку

Если ошибка не смогла скастоваться к тому виду ошибок, которые мы ловим, она не поймается :(

Разница между runtime ошибками и исключениями

Не любой runtime error является исключением. Через try ... catch можно отловить только то, что брошено throw, и только это

- × Деление на ноль - не исключение
- × UB - не исключение
- × delete по плохому указателю - не исключение
- × Битая ссылка - не исключение
- × Выход за границу массива - не исключение

Выход за границу массива

```
std::vector<int> v(10);  
try {  
    v[1000000] = 1;  
} catch (...) {  
    std::cout << "!!!";  
}
```

Выдаст seg fault

Конструкция `catch(...)` означает, что поймается что угодно