**Материал подготовил:** Снежко Лев

**Источники:**

Б. Страуструп «Принципы и практика использования С++», п. 5.6, с.188

<https://purecodecpp.com/archives/2044>

**Вопрос 42. Исключения: основы. Обработка исключений. Try-catch, catch(…)**

Как и в большинстве языков программирования, в языке С++ существует

механизм обработки ошибок: исключения. Основная идея исключений

состоит в отделении выявления ошибки (что можно сделать в вызываемой

функции) от ее обработки (что можно сделать в вызывающей функции),

чтобы гарантировать, что ни одна выявленная ошибка не останется

необработанной. Иначе говоря, исключения предоставляют механизм,

позволяющий сочетать наилучшие подходы к обработке ошибок, исследованные нами до сих пор. Какой бы легкой ни была обработка ошибок,

исключения сделают ее еще легче.

Основная идея заключается в следующем: если функция обнаруживает

ошибку, которую не может обработать, она не выполняет

оператор return, как обычно. а генерирует исключение с помо-

щью инструкции throw, показывая, что произошло нечто неправильное.

Любая функция , прямо или косвенно вызывающая данную функцию, может

перехватить созданное исключение с помощью конструкции catch, т.е.

указать. что следует делать, если вызываемый код вызвал и нструкцию

throw. Функция выражает свою заинтересованность в перехвате исключений

с помощью блока try, перечисляя

виды исключений. которые она планирует обрабатывать в своих разделах

catch блока try. Если ни одна из вызывающих функций не

перехватила исключение, то программа прекращает работу.

Пример:

*vector<int> v ; // Вектор целых чисел*

*for (int i ; cin>>i ; )*

*v . push \_ back ( i ) ; // Чтение в контейнер*

*for ( int i = О ; i<=v . size ( ) ; ++i ) // Вывод значений*

*cout << "v [ " << i << " ] == " << v [ i ] << endl ;*

*-* этот код содержит ошибку, которая не обрабатывается try-catch

*int main ( )*

*try {*

*vector<int> v ; // Вектор зна чений типа i n t*

*for ( int х ; cin>>x; )*

*v . push\_back (x) ; // Чтение значений в вектор*

*for ( int i = О ; i<=v . size ( ) ; ++i) // Вывод зна чений*

*cout << "v [ " << i <<" ] == " << v [ i ] << ' \n ' ;*

*catch (out\_of\_ranqe) {*

*cerr << "Ошибка диапазоиа ! \n" ;*

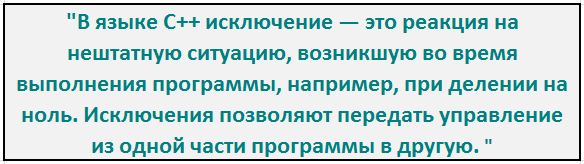
*return 1 ;*

*catch ( . . . ) { // Перехват всех остальных исключений*

*cerr << "Исключеиие : что-то пошло ие так\n " ;*

*return 2;*

*}*



Пример 2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | #include <iostream>;  using namespace std;    int main()  {  setlocale(LC\_ALL, "rus");    int num1;  int num2;    int var = 3; // управляющая переменная для while  while (var != 0)  {  cout << "Введите число num1: ";  cin >> num1;  cout << "Введите число num2: ";  cin >> num2;    cout << "num1 + num2 = " << num1 + num2 << endl;  cout << "num1 / num2 = " << num1 / num2 << endl;  cout << "num1 - num2 = " << num1 - num2 << endl;  cout << "=================================" << endl << endl;    var--;  }    cout << "Конец работы программы!" << endl << endl;    return 0;  } |

*-* здесь происходит следующее. Так как переменная **var**равна 3, цикл **while**, в нормальной ситуации, должен отработать три раза. С каждым шагом цикла **var**уменьшается на единицу с помощью [декремента](https://purecodecpp.com/archives/453). Но так как мы сразу внесли значение 0 в переменную **num2**, программа не пройдет до конца даже первый шаг цикла. Она прервется.

Как работает исключение? – Программист прописывает в коде (в **try**-блоке) конкретное условие, что если переменная **num2**станет равна 0, то в таком случае необходимо генерировать исключение в **throw**. Далее, то что сгенерировал **throw**, перехватывает  **catch**-блок (в виде параметра функции) и программа выполнит тот код, который прописан в этом блоке.

Вот так:

*#include <iostream>;*

*using namespace std;*

*int main()*

*{*

*setlocale(LC\_ALL, "rus");*

*int num1;*

*int num2;*

*int var = 3;*

*while (var != 0)*

*{*

*cout << "Введите число num1: ";*

*cin >> num1;*

*cout << "Введите число num2: ";*

*cin >> num2;*

*cout << "num1 + num2 = " << num1 + num2 << endl;*

*cout << "num1 / num2 = ";*

*try // тут располагается код, который потенциально может вызвать ошибку*

*{*

*if (num2 == 0)*

*{*

*throw 999; // генерировать целое число 999*

*}*

*cout << num1 / num2 << endl;*

*}*

*catch (int thr)// сюда передастся число, которое сгенерировал throw*

*{*

*cout << "Ошибка №" << thr << " - деление на 0!!!" << endl;*

*}*

*cout << "num1 - num2 = " << num1 - num2 << endl;*

*cout << "=================================" << endl << endl;*

*var--;*

*}*

*cout << "Программа завершила работу!" << endl << endl;*

*return 0;*

*}*

*cout << "Программа завершила работу!" << endl << endl;*

*return 0;*

*}*

Определения:

Исключение

Вопросы: