Выполнено студентом группы ВМК-22 Александровым Данилом

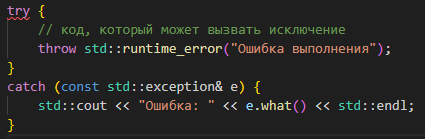
**Обработка исключительных ситуаций. Несколько catch, catch(...), указание типов бросаемых исключений в функции.**

**Основы обработки исключений**

В C++ обработка исключений осуществляется с помощью блоков try, catch и throw. Исключения используются для обработки ошибок, которые могут возникнуть в процессе выполнения программы. Блоки try и catch позволяют перехватывать и обрабатывать эти ошибки.

* try — блок кода, в котором могут возникнуть исключительные ситуации. Этот код "пытается" выполнить действия.
* catch — блок кода, который перехватывает исключения, выброшенные в блоке try.

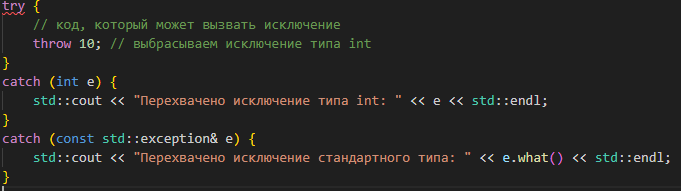
Пример базовой обработки исключений:



**Несколько catch блоков**

В C++ можно использовать несколько блоков catch, чтобы перехватывать различные типы исключений. Это позволяет точно обработать разные виды ошибок.

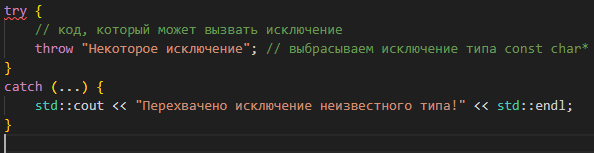
Пример:



**catch(...) — универсальный обработчик**

Блок catch(...) используется для перехвата всех исключений, включая те, которые не были явно указаны в других блоках catch. Это универсальный блок, который перехватывает любые исключения.

Пример:

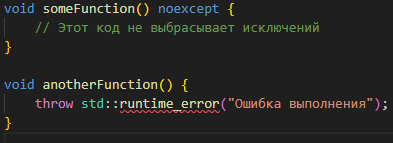


**Указание типов исключений в функциях**

В C++ можно указывать, какие типы исключений может выбрасывать функция, с помощью спецификатора noexcept или путем указания типов исключений в сигнатуре функции. Однако, стоит отметить, что в C++11 и позже использование спецификатора throw() для указания типов исключений было устаревшим и не рекомендуется.

* noexcept — спецификатор, который указывает, что функция не выбрасывает исключений.
* throw() — устаревший способ указания типов исключений.

Пример использования noexcept:



**Пример с указанием типов исключений в функции (устаревший подход)**

В старых версиях C++ можно было использовать спецификатор throw() для указания типов исключений, которые функция может выбросить. Однако, этот подход был признан устаревшим в C++11, и теперь рекомендуется использовать noexcept.

Пример (устаревший синтаксис):

