

Исключения

№ урока: 8 **Курс:** C++ Essential

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать необходимость использования исключений как отдельный вид обработки ошибок, в каких случаях можно и нужно использовать исключения, в каких нежелательно. Научить студентов создавать собственный класс исключений, ознакомить с перечнем стандартных.

Изучив материал данного занятия, учащийся сможет

- Понимать, как работают исключения.
- Реализовать собственный класс исключения.
- Понимать, почему не нужно использовать исключение в деструкторах.
- Понимать специфику работы разворачивания стека при исключении.
- Понимать, чем может помочь использование исключения в конструкторе.

Содержание урока

1. Что такое исключение?
2. Механизм «перехвата» исключений.
3. Стандартные классы, представляющие исключения.
4. Создание своего исключения.

Резюме

- `try {}` – блок, в котором подразумевается возможный выброс исключения.
- `catch(...){}` – конструкция, которая занимается обработкой любого исключения, которое было сгенерировано в соответствующем блоке `try`.
- `catch(MyException& ex){}` – отлавливание исключения типа `MyException`.
- В программе не может быть обработано одновременно 2 и более исключений. В результате генерации 2 исключений будет вызван `std::terminate`.
- Старайтесь придерживаться единообразия стиля написания кода, не смешивайте обычную обработку ошибок с помощью возвращаемого значения и исключений в одном месте.
- Исключения могут обойтись дорого, потратив 5-20% производительности от всей программы.
- Не все системы и проекты поддерживают исключения.

Закрепление материала

- В чем заключается механизм обработки исключений?
- Что такое разворачивание стека? Как оно происходит?
- Может ли к одному блоку `try` прикрепляться несколько блоков `catch` или только один?
- Каким образом можно перекинуть исключение на верхний уровень?
- Можно ли использовать механизм исключений в конструкторе, деструкторе?

Дополнительное задание

Задание

Почитайте описание `std::exception_ptr` (C++11): как ключевое слово для функции и как оператор.

Изучите поведение `std::exception_ptr` и к нему прилегающих терминов. В чем преимущество его использования перед старыми стандартами?

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Оцените ранее написанный вами код, какие места являются сомнительными и могут бросить исключения? Оберните эти места блоками `try-catch`, создайте свой класс по обработке исключений. Какую полезную информацию для пользователя вы бы вынесли в этот класс?

Задание 3

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы» описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

<http://www.acodersjourney.com/2016/08/top-15-c-exception-handling-mistakes-avoid/>

<https://ru.cppreference.com/w/cpp/error/exception>

<https://msdn.microsoft.com/ru-ru/library/6dekhhbbc.aspx>