Множественное наследование

№ урока: 4 **Курс:** C++ Essential

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать и применять на практике множественное наследование, виртуальное наследование, приведение полиморфных типов с помощью dynamic_cast.

Изучив материал данного занятия, учащийся сможет

- Понимать, для чего нужно множественное и виртуальное наследование, знать синтаксис при наследовании.
- Уметь объяснить необходимость использования dynamic_cast, знать все необходимые условия использования данного приведения типов.
- Уметь решать проблемы, которые могут возникнуть при использовании множественного наследования.
- Различать восходящее и нисходящее приведение типов, приведение к самому производному типу (most derived).

Содержание урока

1. Механизм множественного наследования
2. Вызов конструкторов базовых классов
3. Проблемы множественного наследования и их решение (совпадение имен методов, полей)
4. Для чего нужно виртуальное наследование
5. Приведение типов (dynamic_cast)

Резюме

- class Pedigreed : public virtual Animal – синтаксис виртуального наследования, класс Породистый является Животным и наследуется виртуально.
- class Cat : public Domestic, public Pedigreed – синтаксис множественного наследования, класс Кошка является Домашним и Породистым животным
- auto ped = dynamic_cast<Pedigreed*>(animal) – приведение указателя на базовый класс Animal к производному классу Pedigreed через динамическое преобразование dynamic_cast
- auto cat = dynamic_cast<Cat*>(ped) – приведение указателя на базовый класс Pedigreed к производному классу Cat через динамическое преобразование dynamic_cast
- if(dynamic_cast<void*>(animal) == dynamic_cast<void*>(ptr_ped)) – проверка на то, указывают ли оба указателя на самом деле на один и тот же объект

Закрепление материала

- Что такое множественное наследование? Чем оно помогает, чем мешает разработчику?
- Расскажите о проблеме ромбовидного наследования. Как можно ее решить?
- Как работает приведение типов dynamic_cast? На каком этапе оно производится?
- Какой порядок вызовов конструкторов при множественном наследовании?

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- Нужно ли делать базовый класс полиморфным, если я хочу сделать восходящее приведение типов? В случае нисходящего?

Дополнительное задание

Задание
Разобрать принцип работы виртуального наследования через vptr, vtable.
Приведите пример реорганизации архитектуры классов таким образом, чтобы можно было избежать использования dynamic_cast.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Вновь пересмотрите медицинскую карточку, которую вы рассматривали на прошлом уроке. Внесите поправки, внедрите ромбовидное наследование. Рассмотрите реальный случай, когда можно обойтись без виртуального наследования (например, двойной вызов конструктора базового класса реально необходим и имеет смысл).

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

C++, dynamic_cast
https://docs.microsoft.com/en-us/cpp/cpp/dynamic-cast-operator?view=vs-2017
https://docs.microsoft.com/en-us/cpp/cpp/casting?view=vs-2017

CyberBionic systematics
CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com