

# Наследование, полиморфизм

**№ урока:** 3    **Курс:** C++ Essential

**Средства обучения:** Qt Creator

## Обзор, цель и назначение урока

Научить студентов понимать и применять на практике принципы ООП: наследование и полиморфизм.

## Изучив материал данного занятия, учащийся сможет

- Понимать, для чего нужно наследование, знать синтаксис наследования.
- Уметь объяснить разницу каждого модификатора доступа при наследовании и определить итоговый доступ.
- Понимать надобность полиморфизма.
- Знать отличие виртуальных функций от обычных, отличие от простого перекрытия.

## Содержание урока

1. Механизм наследования (одиночное наследование)
2. Определение дочернего класса
3. Модификатор доступа: protected
4. Конструктор базового класса, указатель this
5. Идея полиморфизма
6. Виртуальные методы
7. Виртуальный деструктор

## Резюме

- Наследование – принцип ООП, который позволяет переносить свойства и поведение от одного класса к другому.
- Дочерний класс (он же производный, класс-потомок, derived class) – класс, который наследует свойства и поведение от базового (родительского) класса.
- Полиморфизм – принцип ООП, который позволяет переопределять поведение (интерфейс, функцию, оператор) с одной и той же сигнатурой (описание входных и выходных параметров) и наименованием, так, чтобы алгоритм работы был разным при каждом проявлении данного поведения у разных классов.
- class Mammal – класс, который описывает млекопитающих
- class Cat: public Mammal – дочерний класс, производный от класса Mammal, наследуется публично
- protected – защищенный модификатор доступа (только при наследовании имеет смысл)
- virtual void foo() – виртуальный метод
- virtual ~Mammal() – виртуальный деструктор
- override – ключевое слово для проверки корректности переопределения виртуальной функции в производном классе

## Закрепление материала

- Что такое наследование? Полиморфизм?
- Какие модификаторы доступа знаете?
- Принцип работы виртуальных методов.
- Для чего нужен виртуальный деструктор? Всегда ли нужно его писать?
- Может ли быть виртуальный конструктор? Если да, то как он работает, если нет, то почему?

## Дополнительное задание

Задание

Разобрать принцип работы виртуальных методов через `vpitr`, `vtable`.

Подумать над тем, как можно реализовать виртуальный конструктор.

Для чего необходимо приватное наследование и использование приватных виртуальных методов?

## Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Пересмотрите медицинскую карточку, которую вы создали на прошлом уроке. Какую иерархию данных вы могли бы сделать? Сделайте цепочку наследования из 3 уровней. Добавьте минимум 3 виртуальных метода. Сделайте три различных наследования, чтобы было уместно использовать публичное, приватное и защищенное наследование. Почему вы сделали именно такой выбор?

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++, virtual functions

<https://msdn.microsoft.com/en-us/library/0y01k918.aspx>