

# Наследование C++

## 1. Базовое наследование

Создайте класс `Vehicle` с методами `start()` и `stop()`. От него наследуйте классы `Car` и `Bicycle`, добавив каждому уникальные методы (`honk()` для машины, `ringBell()` для велосипеда). Продемонстрируйте работу методов.

## 2. Переопределение методов

Создайте класс `Shape` с виртуальным методом `draw()`. Наследуйте от него классы `Circle`, `Square` и `Triangle`, переопределив метод `draw()` для каждого. Создайте массив указателей на `Shape` и продемонстрируйте полиморфизм.

## 3. Абстрактный класс

Создайте абстрактный класс `Animal` с чистой виртуальной функцией `makeSound()`. Реализуйте классы `Dog`, `Cat` и `Cow`, наследующие от `Animal`. Создайте функцию, принимающую вектор указателей на `Animal` и вызывающую `makeSound()` для каждого.

## 4. Многоуровневое наследование

Постройте иерархию: `Person` → `Employee` → `Manager`. Добавьте уникальные поля и методы на каждом уровне. Продемонстрируйте доступ к методам разных уровней.

## 5. Множественное наследование

Создайте классы `Printer` (с методом `print()`) и `Scanner` (с методом `scan()`). На их основе создайте класс `Copier`, который может и печатать, и сканировать. Продемонстрируйте работу всех методов.

## 6. Виртуальный деструктор

Создайте класс `Base` с неvirtуальным деструктором и класс `Derived`, наследующий от `Base`. Продемонстрируйте утечку памяти при удалении объекта `Derived` через указатель на `Base`. Затем сделайте деструктор виртуальным и покажите разницу.

## 7. Интерфейсы (чисто абстрактные классы)

Создайте интерфейс `Drawable` с чисто виртуальными методами `draw()` и `getArea()`. Реализуйте его в классах `Rectangle` и `Circle`. Создайте функцию, которая принимает `Drawable` и выводит информацию о фигуре.

## 8. Защищённое наследование

Создайте класс Base с public, protected и private членами. Наследуйте от него класс Derived с разными типами наследования (public, protected, private). Продемонстрируйте различия в доступе к членам.

## 9. Композиция vs наследование

Создайте класс Engine. Реализуйте класс Car двумя способами: через наследование и через композицию (содержащий объект Engine). Обсудите плюсы и минусы каждого подхода.

## 10. Финальные классы и методы

Создайте класс Base с виртуальным методом foo() и финальным методом bar(). Попытайтесь переопределить оба метода в классе Derived. Объясните результат. Затем пометьте весь класс Derived как final и попытайтесь от него наследовать.