В С++ в случае, если объект класса-потомка находится в переменной класса предка, и при этом мы вызываем у этой переменной метод, объявленный в предке и переопределённый в потомке, С++ вызывает всё-таки метод предка

Отличие поведения виртуального метода от не виртуального проявляется только в одном случае: когда объект класса-потомка лежит в переменной класса-предка и у предка вызывается виртуальный метод, перекрытый в потомке. Так, например, если объект Cat находится в переменной Cat, то никакой разницы между виртуальными и не виртуальными методами не будет.

Виртуальные методы нужны в тех случаях, когда объекты-потомки могут/будут лежать в переменных классов-предков И мы хотим со всеми ними работать единообразно, не зная, какому конкретно классу-потомку они принадлежат, а просто вызывая при это общие для всех них методы, определённые в классе-предке. При этом мы хотим, разумеется, чтобы при вызове этого общего для всех них метода, на самом деле для каждого объекта вызывался метод свой специфический

Так что – этот метод нужен только для того, чтобы просто быть. Он должен быть пустым в базовом классе для того, чтобы классам-потомкам было, что перекрывать. Он может так и остаться пустым – тогда какие-то из потомков, которые не перекроют его, просто унаследуют пустой метод, который ничего не делает. Но есть и другая альтернатива: этот метод можно сделать больше, чем пустым, можно удалить у него реализацию вообще.

Метод без реализации называется абстрактным (или ещё говорят, что это «чистый виртуальный метод»), а класс, содержащий хотя бы один абстрактный метод, называется абстрактным классом.