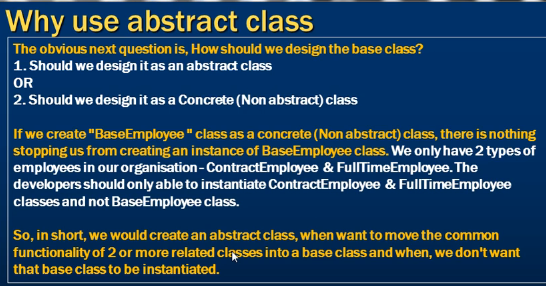
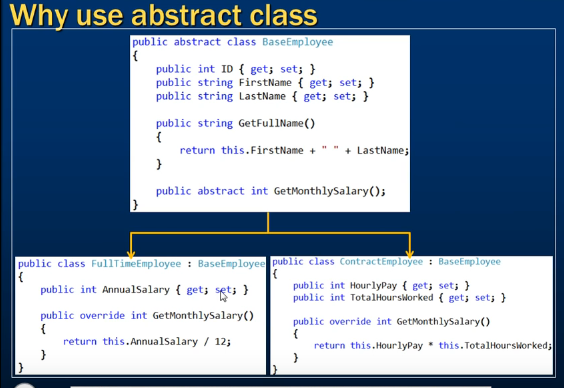
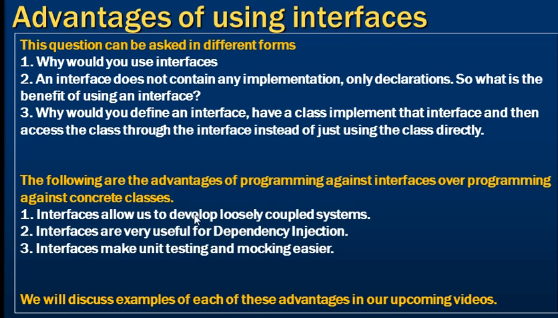
# Abstract class

Common code are duplicated in two classes ContractEmployee and FullTimeEmployee





# Interfaces



# Abstract classes vs Interfaces

Короткое различие.

Абстрактный класс — это класс, у которого не реализован один или больше методов (некоторые языки требуют такие методы помечать специальными ключевыми словами).

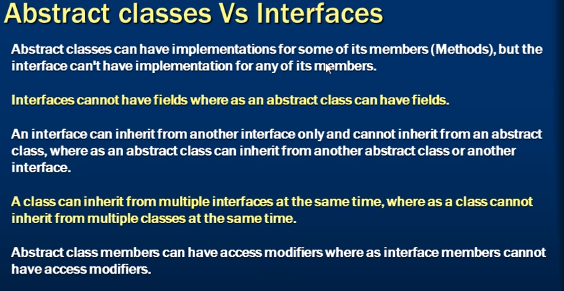
Интерфейс — это абстрактный класс, у которого **ни один** метод не реализован, все они публичные и нет переменных класса.

Интерфейс нужен обычно когда описывается только интерфейс (тавтология). Например, один класс хочет дать другому возможность доступа к некоторым своим методам, но не хочет себя «раскрывать». Поэтому он просто реализует интерфейс.

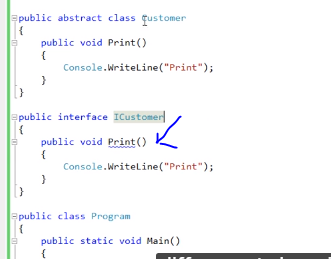
Абстрактный класс нужен, когда нужно семейство классов, у которых есть много общего. Конечно, можно применить и интерфейс, но тогда нужно будет писать много идентичного кода.

В некоторых языках (С++) специального ключевого слова для обозначения интерфейсов нет.

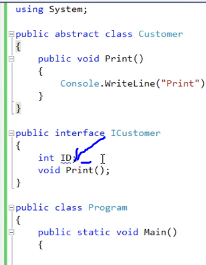
Можно считать, что любой интерфейс — это уже абстрактный класс, но не наоборот.



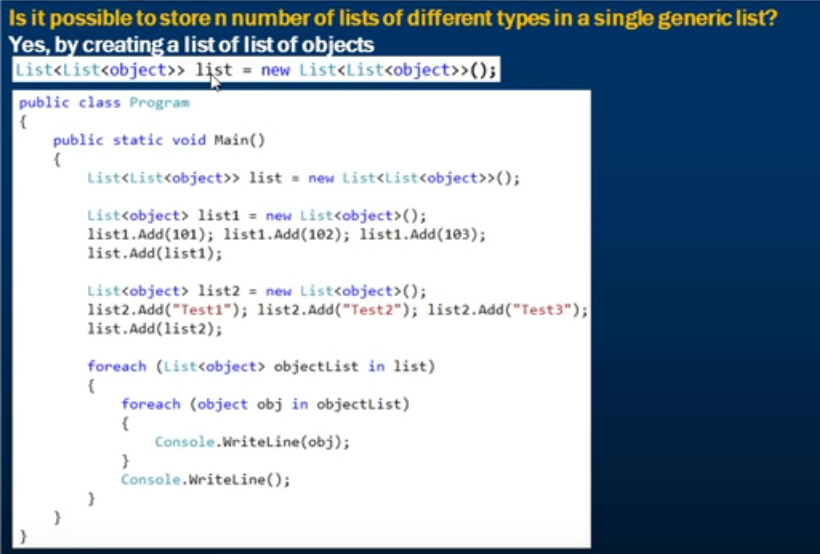
Interfaces cannot have implementation



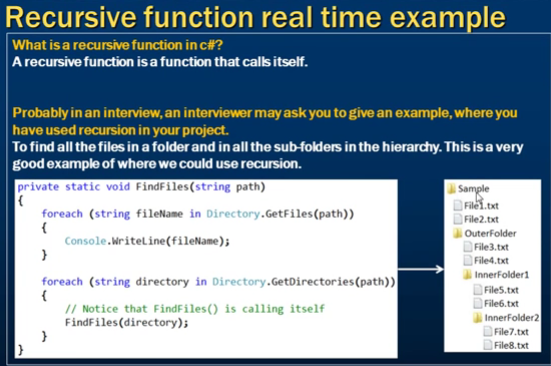
Interfaces cannot have field



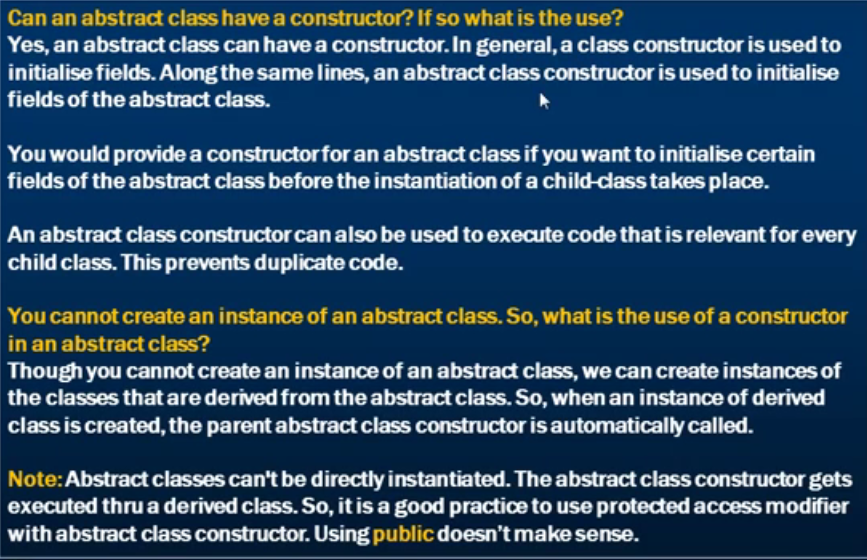
# Storing different types lists in list

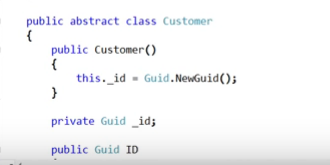


# Recursive

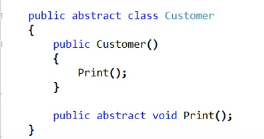


# Abstract class constructor

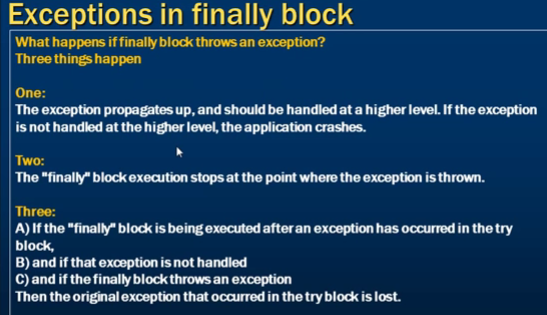




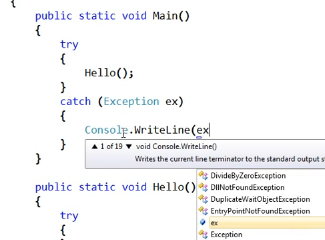
Calling an abstract method from an abstract class



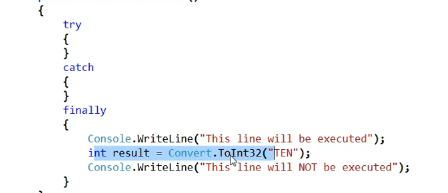
# Finally block exception



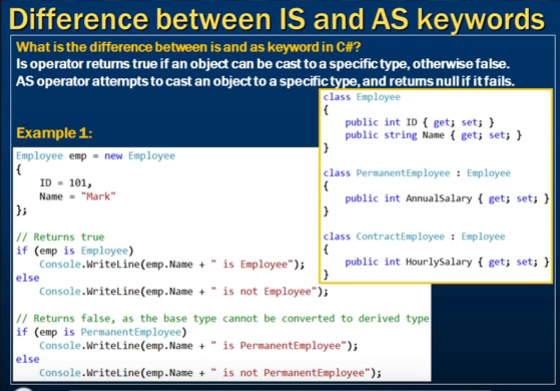
Higher level catching

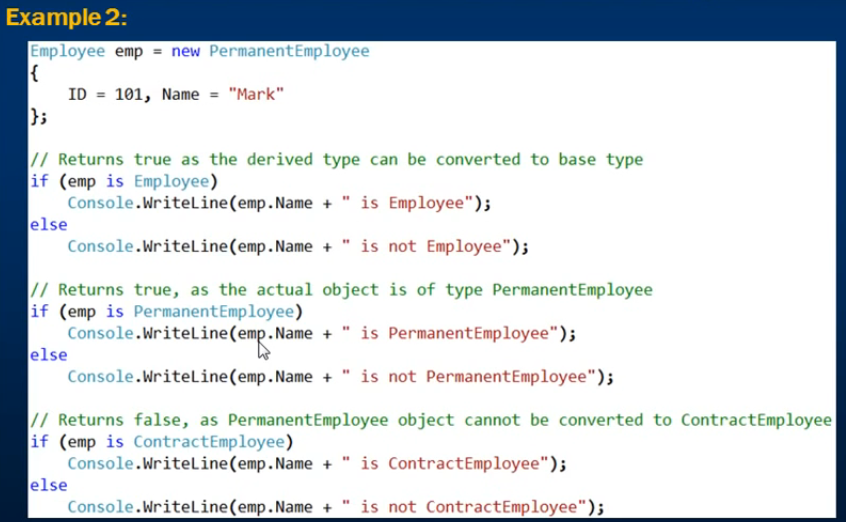


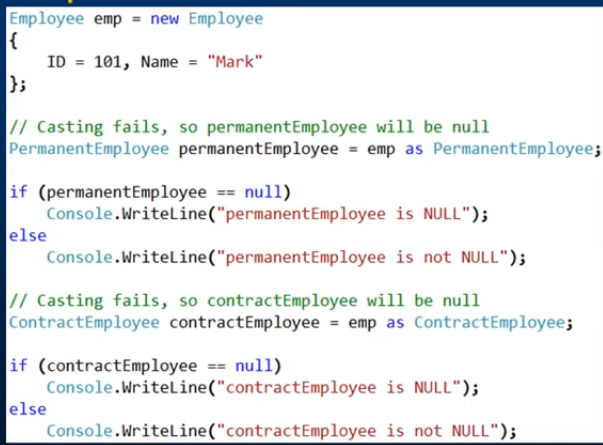
Stop at the point where the exception throw

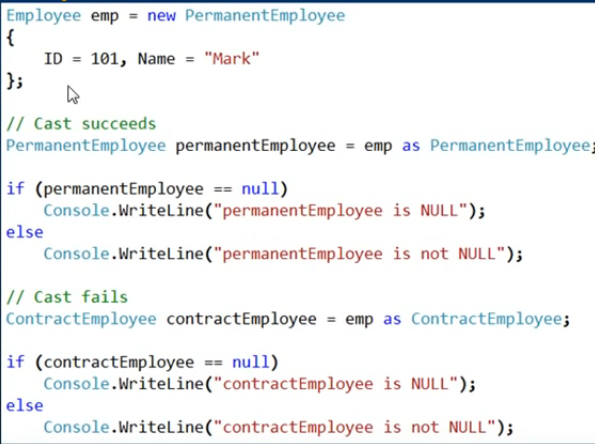


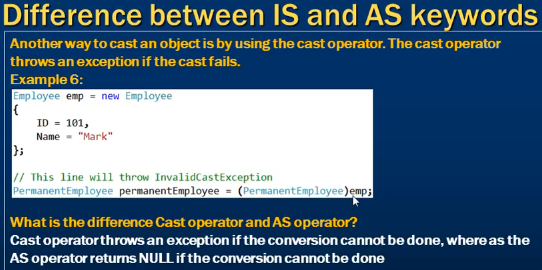
# Is and As differences



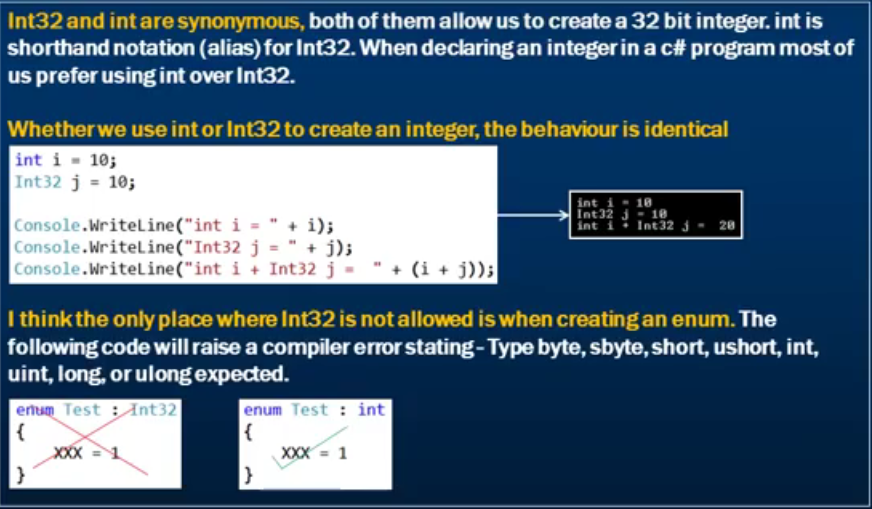








# Differences between Int32 and int



# Reverse each word in string

