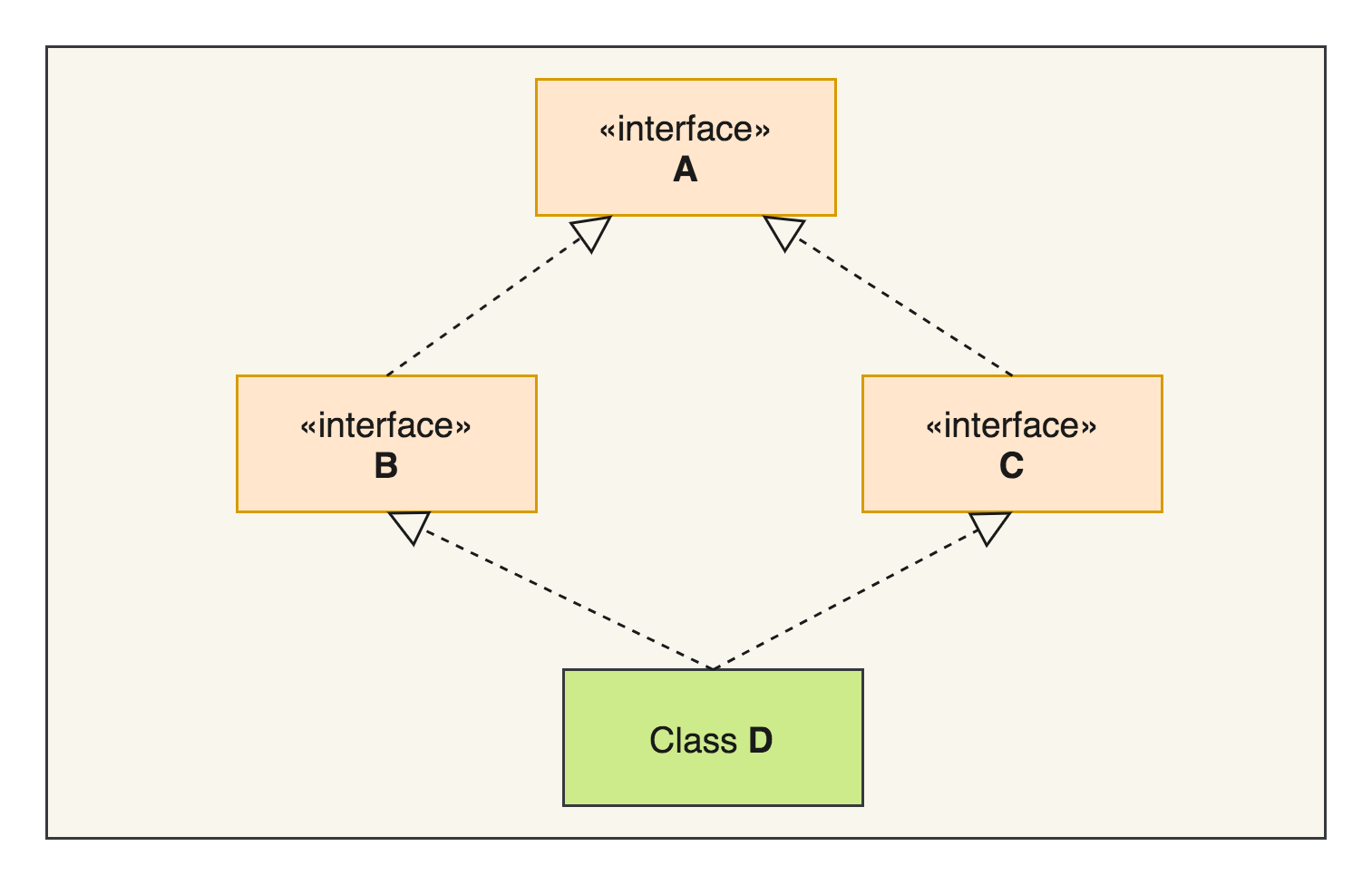
Diamond Problem of inheritance is an ambiguity that can arise as a consequence of allowing multiple inheritance in language like C++.

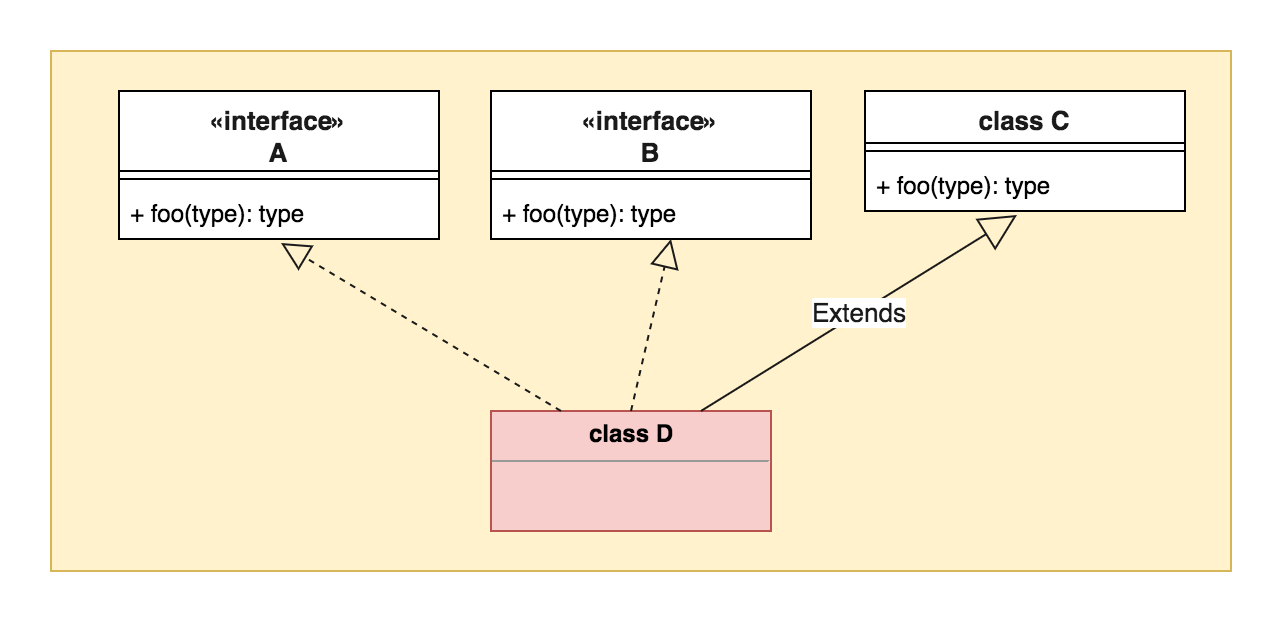


How does Java 8 tackles diamond problem of multiple inheritance?

Java 8 brought a major change where interfaces can provide default implementation for its methods. Java designers kept in mind the diamond problem of inheritance while making this big change. There are clearly defined conflict resolution rules while inheriting default methods from interfaces using Java 8.

Rule 1

Any method inherited from a class or a superclass is given higher priority over any default method inherited from an interface.

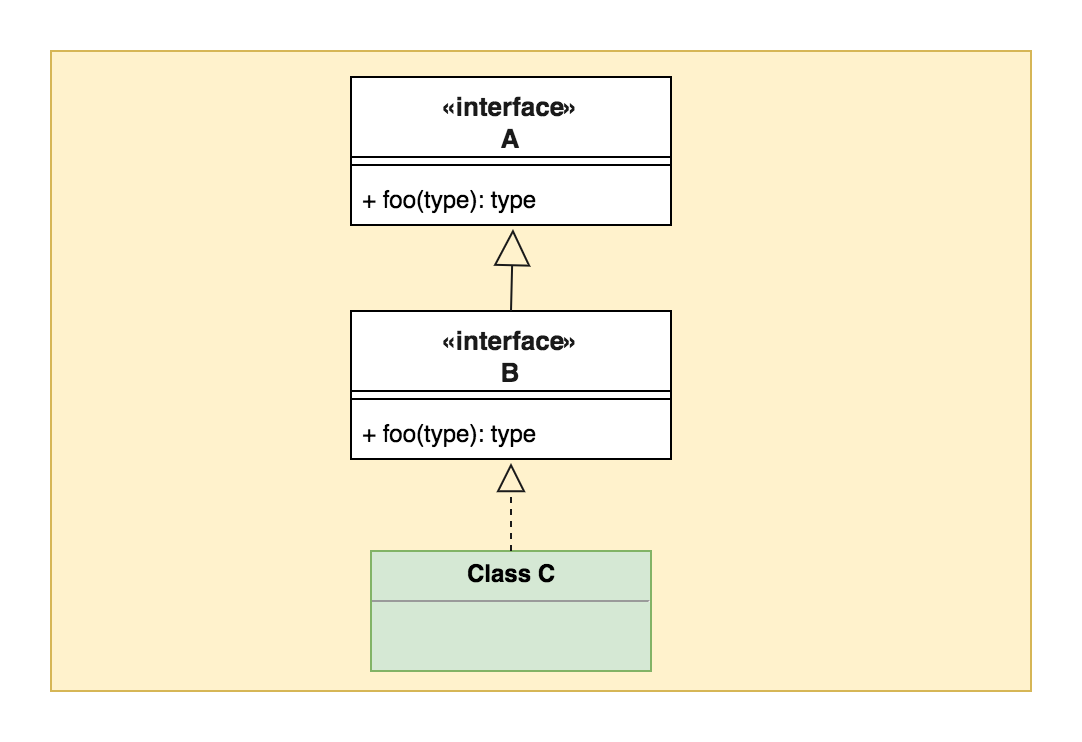


class has higher precedence than interface default methods.

In the diagram above, foo() method of class D will inherit from class C.

Rule 2

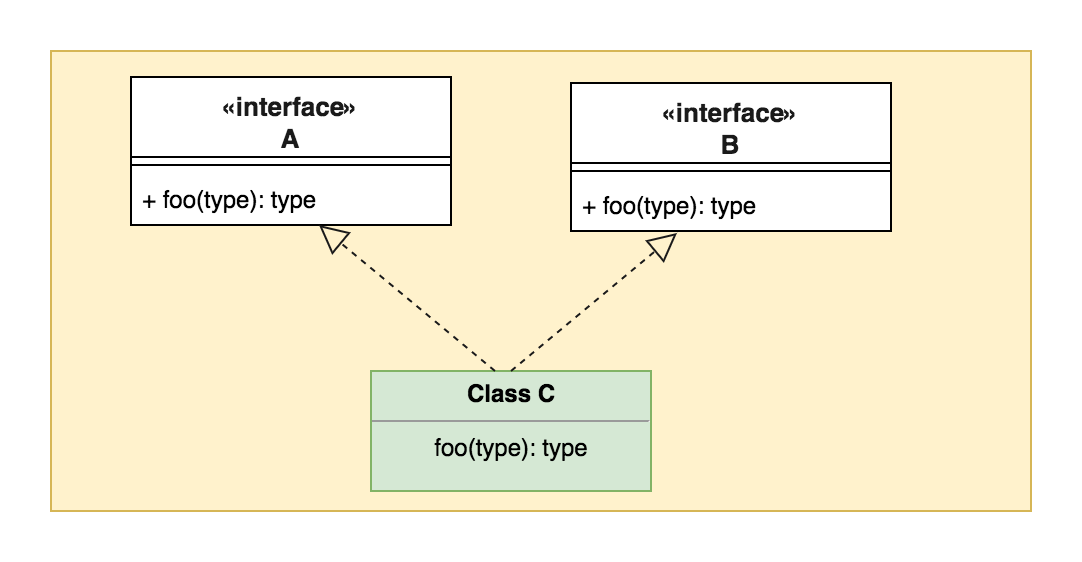
Derived interfaces or sub-interfaces take higher precedence than the interfaces higher-up in the inheritance hierarchy.



sub-interface has higher priority

In the above class diagram, foo() of class C will inherit from default method of interface B.

Rule 3

In case Rule 1 and Rule 2 are not able to resolve the conflict then the implementing class has to specifically override and provide a method with the same method definition.

class C must override foo method

In above class diagram, since interface A & B are at same level, to resolve conflict, class C must provide its own implementation by overriding method foo().

Rule 3

class C {

void foo() {

B.super.foo();

}

}

foo() method can refer to A or B’s default implementation using A.super.foo() or B.super.foo()