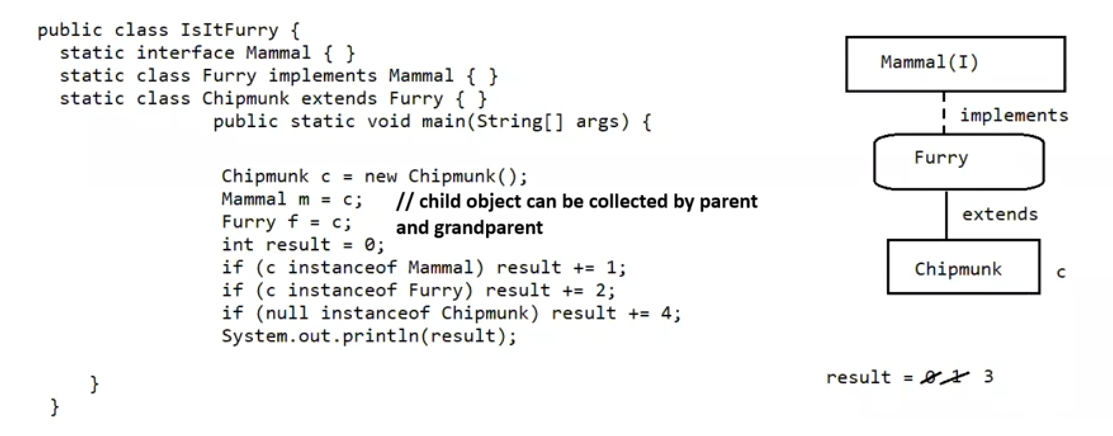


If the class implements interface and it doesn’t give the implementation for interface methods it must be marked as abstract. Otherwise it leads to compile time error.

To give the implementation to the interface method , method signature and return type should be same otherwise it leads to compile time error.



Eg: Interface\_Eg3

//from jdk 1.8 we can have concrete methods in interface

Eg: Interface\_Eg4

// go through the code

Note: static methods are inherited in Java, but they are not overridden. Subclasses have access to static methods from their superclass, but redeclaring a static method with the same signature in the subclass creates a new method that hides the superclass method. ( class-level)

Eg: Interface\_Eg5

// go through the code

Eg: Interface\_Eg6

// go through the code

public interface InterfaceX

{

public int geek();

}

public interface InterfaceY

{

public String geek();

}

Now, Suppose we have a class that implements both those interfaces:

public class Testing implements InterfaceX, InterfaceY

{

public String geek()

{

return "hello";

}

}

The question is: Can a java class implement Two interfaces with same methods having the same signature but different return types??

No, its an error

If two interfaces contain a method with the same signature but different return types, then it is impossible to implement both the interface simultaneously.

What if the methods have same signature and return type ?

No problem absolutely works fine

Eg: Interface\_Methods\_With\_Same\_Method\_Signature\_Return\_Type

// go through the code

