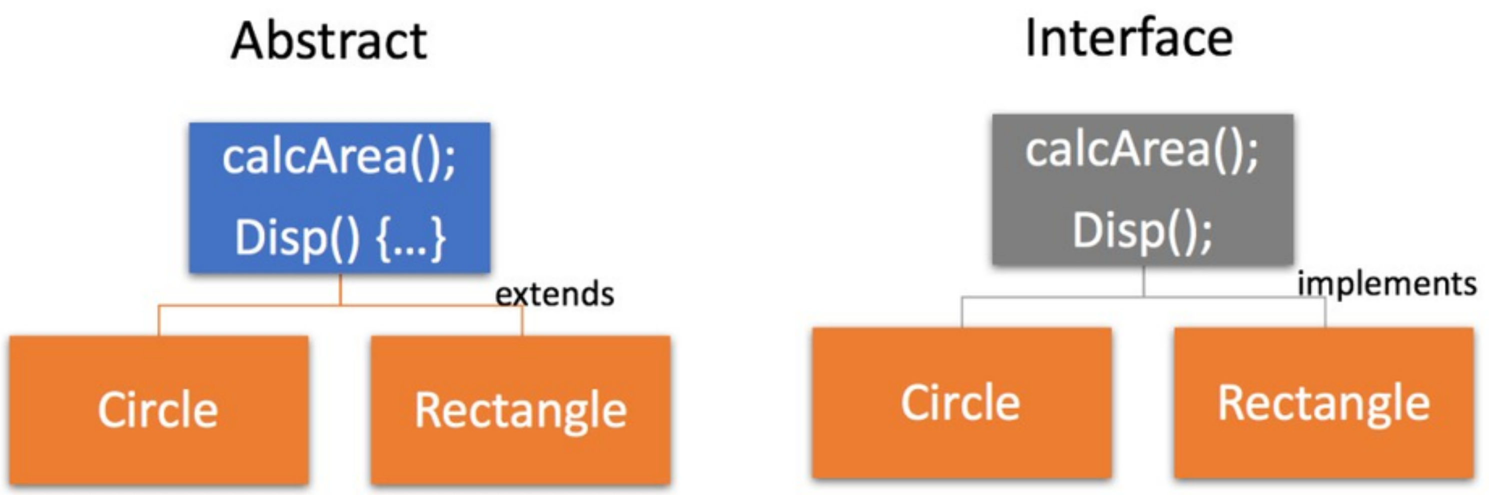
AlgoTutor

ABSTRACT CLASS VS INTERFACE IN JAVA



+91-7260058093
www.algotutor.io

Introduction:

After Java 8, the interfaces can now have default methods with implementations.

The interface can now have static methods with implementations.

But why It's required make changes in Interface?

The Reason:

To support lambda functions better,the default method is introduced without breaking the existing implementations.

To provide backward compatibility to the Collection Framework with the new lambda functions.

And, static methods are introduced to directly add the utils method in the interface without having to create a new class.

The Real confusion

Previously, Abstract classes can have implementations and after Java 8, the interfaces can also have it now.

So ain't they same now?

Let's find out, how they're still different.

State:

An abstract class can have a state, its methods can access a state of any of its instances.

But the interface can't have a state.

Constructor:

Abstract classes can have constructors.

Interfaces cannot have any constructor hence we cannot write any logic on object creation.

Access Modifiers:

An abstract class can declare instance variables, with all possible access, and modifiers they can be accessed in child classes.

An interface can only have public, static, and final variables and can't have any instance variables.

Methods and Blocks

An abstract class can declare instances and static blocks, which is not possible. with interfaces.

The interface can have a single abstract method that can refer to a lambda expression. But abstract class can't.

Conclusion

Always try to prefer an Interface.

Because, you can use multiple Interfaces in concrete classes and also extend one parent class.

Use Abstract class only when required because then you can't extend other classes so it's a bit restrictive.