Q1) Why class cannot be declared as protected in Java??

Q2) Why class cannot be declared as private in Java??

Q3) Why abstract methods cannot be declared as protected in Java??

Q4) Why abstract methods cannot be declared as private in Java??

Q5) What is the difference between abstract and interface in Java??

Chapter - 8

For a top-level interface, access is either public or not used. When no access modifier is included, then default access results, and the interface is available only to other members of its package. When it is declared as public, the interface can be used by any other code. (When an interface is declared public, it must be in a file of the same name.) name is the name of the interface and can be any valid identifier, except for var (which is a reserved type name added by JDK 10).

In the traditional form of an interface, methods are declared using only their return type and signature. They are, essentially, abstract methods. Thus, interface are like abstract declaration where you can declare the whole class as an interface.

Once an interface has been defined, one or more classes can implement that interface. To implement an interface, include the implements clause in a class definition and then create the methods required by the interface.

The methods that implement an interface must be declared **public**. Also, the type signature of the implementing method must match exactly the type signature specified in the **interface** definition.

It is both permissible and common for classes that implement interfaces to define additional members of their own.

One more point: If a class includes an interface but does not fully implement the methods defined by that interface, then that class must be declared abstract.

You might be somewhat surprised to learn that you can declare a reference variable of an interface type. In other words, you can create an interface reference variable. Such a variable can refer to any object that implements its interface.