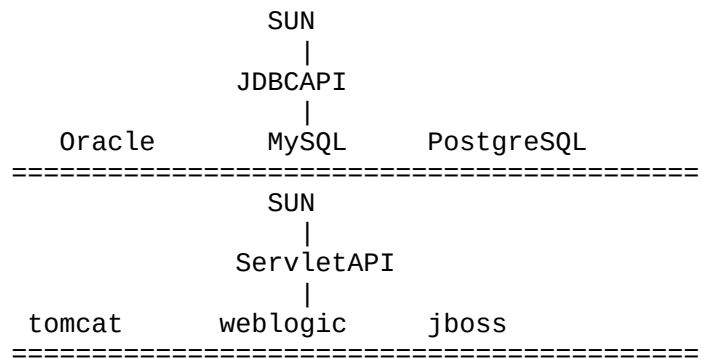
Interface
=======
  Def1::Any service requirement specification(SRS) is called interface.

eg:: SUN people responsible to define JDBC API and database vendor will provide implementation for that



Def2::From client point of view an interface define the set of services what is expecting.
         From service provider point of view an interface define the set of services what is "offering".
           So interface acts a contract b/w client and serviceprovider.

   eg:: GUI screen of ATM defines the set of services what the customer is expecting,
                Bank people offered the same set of services what the customer is expecting.
            Cusomter => GUI => Bank

Def3:: Inside interface every method is always abstract whether we are delcaring or not hence interface is considered as
          100% pure abstract class.

eg:
```
interface  Account
{
      //It is 100% abstract class

      //The methods are by default "abstract and public"
       void withDraw();
       void deposit();
       void checkBalance();
}
```

Summary::
Interface corresponds to Service Requirement Specification(SRS) or contract b/w client and service provider or 100% pure abstract class.

Declaration and implementation of Interface
==========
a. Whenever we are implementing an interface compulsorily for every method of that interface
     we should provide implentation otherwise we have to declare class as abstract class in that case child class is responsible to provide
     implementation for remaining methods.

b. Whenever we are implementing an interface method compulsorily it should be declared as public otherwise it would result in
CompileTime Error.

```
eg:: interface ISample{
            void methodOne();
            void methodTwo();
       }

     abstract class Sample implements ISample{
       public void methodOne(){...}
     }

     class SubServiceProvider extends Sample{
        @Override
      public void methodTwo(){...}
     }
```