Advanced Java

Monday, December 20, 2021 10:45 AM

Distributed Application ---- 2 or more than 2 processes communicate with each other in one application.
JEE = Java Enterprise Edition

Enterprise= distributed business application

JEE = Supports the development of Enterprise application
It provides STANDARDS for developing Enterprise COMPONENTS.

For example --
Laptop is an **integration** of different **components** --

Components are developed by different vendors
and integrated by yet another vendor
All vendors MUST follow the standard ---- then only they can be integrated .

We want to integrate software components …
What could be the standards to integrate software components ????

Signature = prototype= declaration = argument list (number ,type and sequence of parameters) , return type, name, exceptions throws, scope)

```
Void component1(int x, float y, char * t)
{
        …
        ….
        …
}
//component2
Void main()
{
        Char * p = …..
     //Integrate ----Use the component1
       compoent1(12,34.56, p )----call must follow signature
}
```

STANDARDS in software are API signatures !!!
Standards should be followed by BOTH components to be integrated------
One components should give the Implementation of the signature and the other component should give the CALL to that signature

JEE -- provides the standards --- it provides a huge set of INTERFACES !!!!

ADVANCED JAVA based on JEE
ADVANCED JAVA is based on COMPONENT ARCHITECTURE !!!
JEE provides support by giving standards to write components such that they can INTEGRATE !!!

JEE standards FOR
1. Presentation layer ---- Servlets and JSP
2. Business Logic Layers ----- EJB, Servlets, Java Beans , JTA , JPA , JAX-RPC …..

3. Database Layer ---- JDBC !!!!

```
//STANDARD -JEE
 interface  Servlet
{
        Init(…);
        Service(…);

}

//Component1
Class MyServlet  implements Servlet
{
        Init(…)
        {….
        ….
        }

        Service(…)
        {
        …
        }

}

//Component 2
Class WebServerContainer
{
        someJob(Servlet  obj)
        {
                obj.Service(……);
        }
}
```

How to use a component in our program ?

1. Somebody should develop a component -- with great algo and efficiency
2. Somebody should pack all the classes related to the component in a JAR file

JAR = **J**ava **Ar**chive = archive file means a file containing folders and files --- USED for PACKING

To create a JAR
Through eclipse we can create a JAR file --- using project -right click ---export ---Java ---jar ---give the destination

3. Someone else should use the component !!!
    a. Acquire the JAR file somehow and keep it in some folder
    b. Add the Path of that JAR file in your project's build path from eclipse (project - right click --build path, libraries tab--- add external jar --- select the jar file in the folder --- apply and close .
    c. Simply use those classes and CALL the methods

HW ------

```
In Workspace1  --------
        Create Component1 java project
                Add interface study.FeaturesList having 2 methods
                        public double add(double a, double b)
                        public int integerDivision( int a , int b )

                Implement the above interface in study.impl.HiFiCalc  class

                Create a JAR file in some folder  ----  mysupercalc.jar
```

```
Close the above workspace and open Workspace2 -----------
        Create  Component2  java project
                add a class User
                        Add main
                        {
                         doSomeCalculations( new HiFiCalc() );
                        }
                        Add a method
                                Public static void doSomeCalculations(FeaturesList  obj)
                                {
                                        Sysout ( obj.add(…, …)  )
                                }
```

Firstly u get errors FeaturesList and HiFiCalc cannot be resolved
You add the mysupercalc.jar in the build path , you find that the errors are resolved
You run the code!!!

JDBC = Java Database Connectivity

JDBC ---interfaces ---standards given by JEE
Component 1 ---implement the interfaces ---- download this JAR ---- mysqlconnector.jar
Component 2 ---- calls the methods -------- WE will write !!!