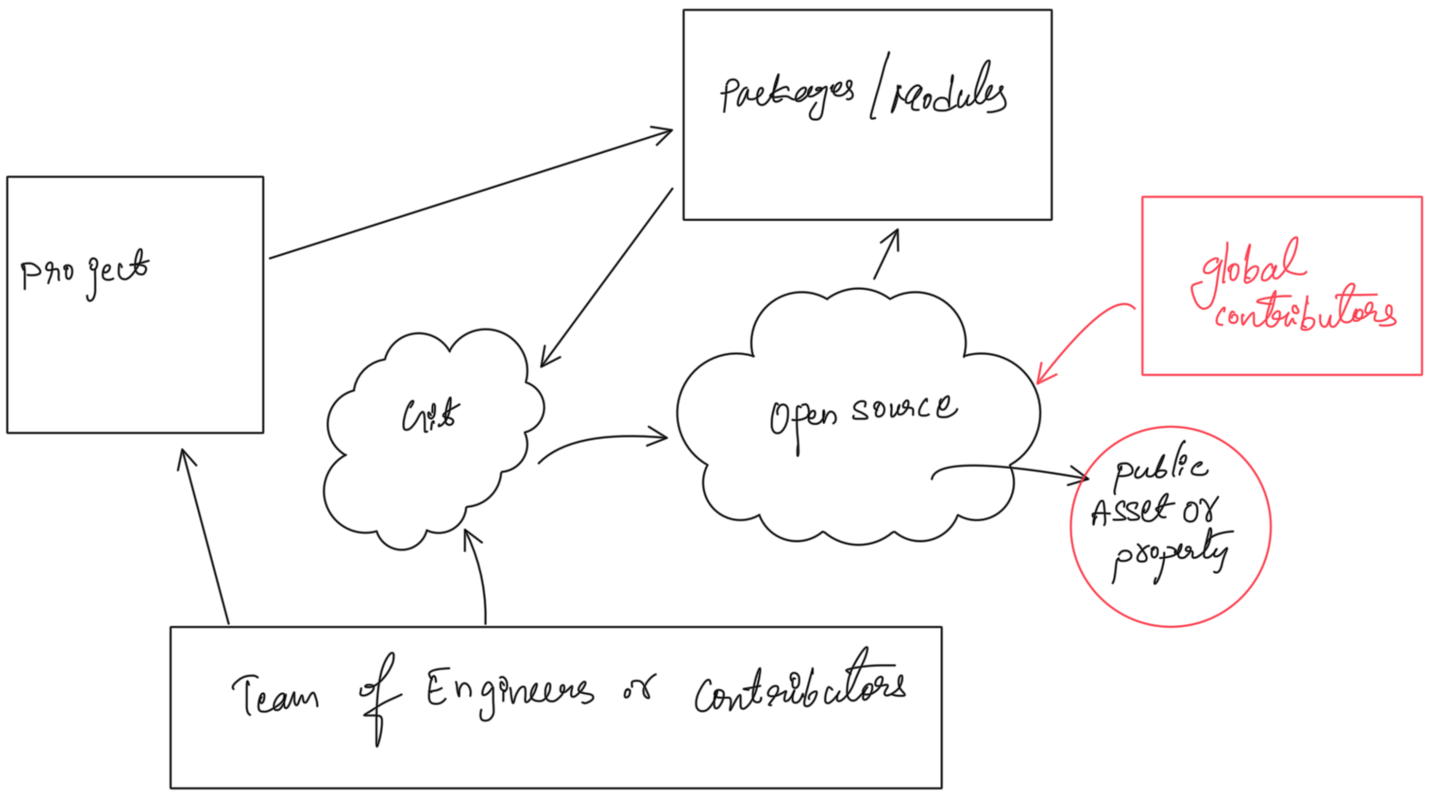
INTRODUCTION To Working With Packages And Modules.

Day 30
09/08/2025

Any Realworld software containes Millions of Lines of code under which Thousands of Engineers working together

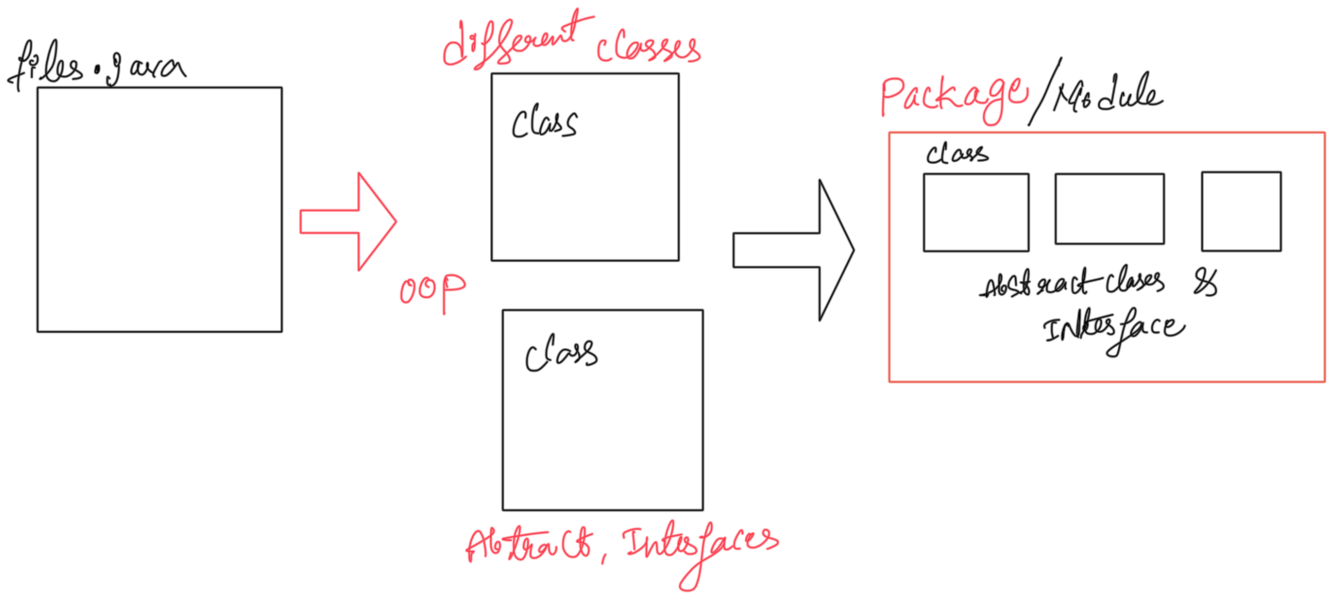
So We Have to organize and maintain the code In a Systamatic Way.

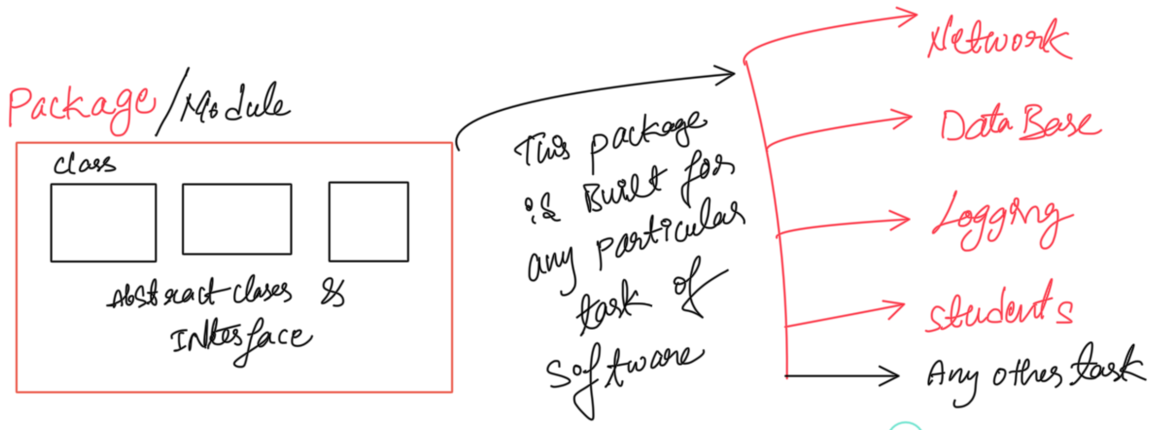


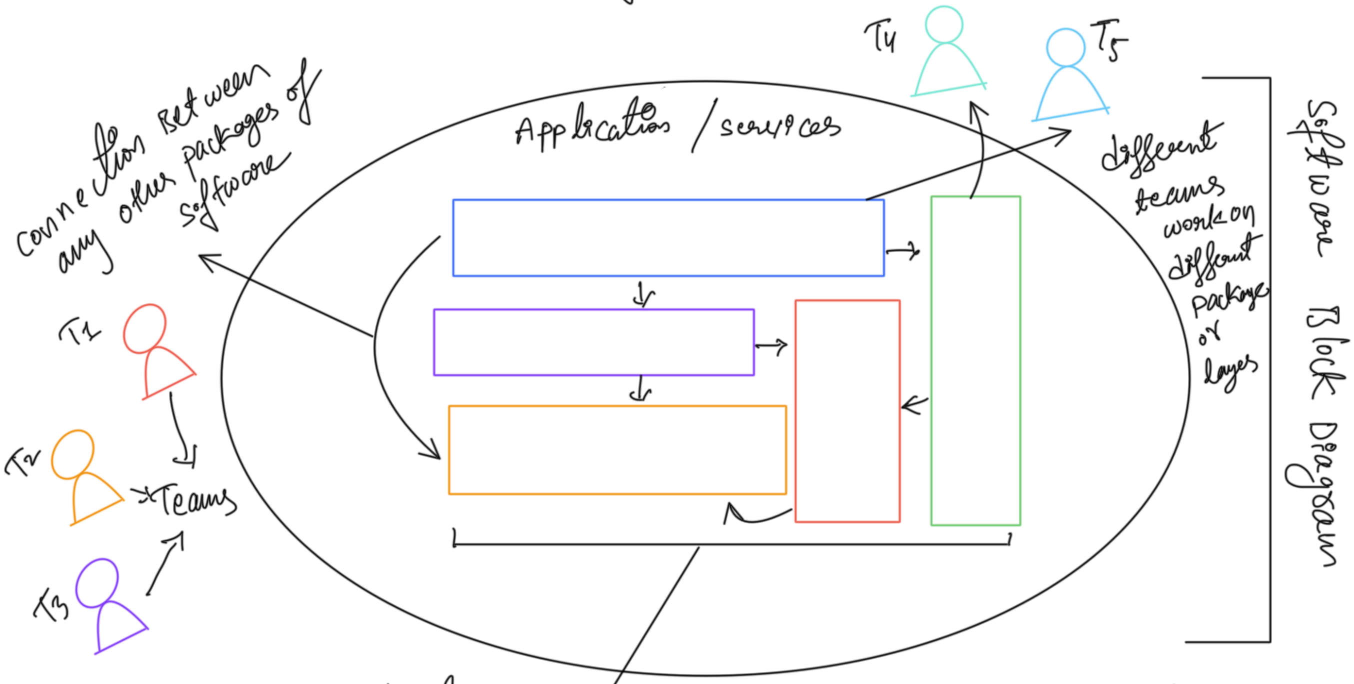
Consider Any Project It has to be maintained and well organized.

When it comes to opensource Most of the Common code which solves the common problems or provides Services to different types of Bussiness Have Same Service Need.

So the code made open Source So that any global contributors can contribute to the Software and any one can make use of that open Source Software.





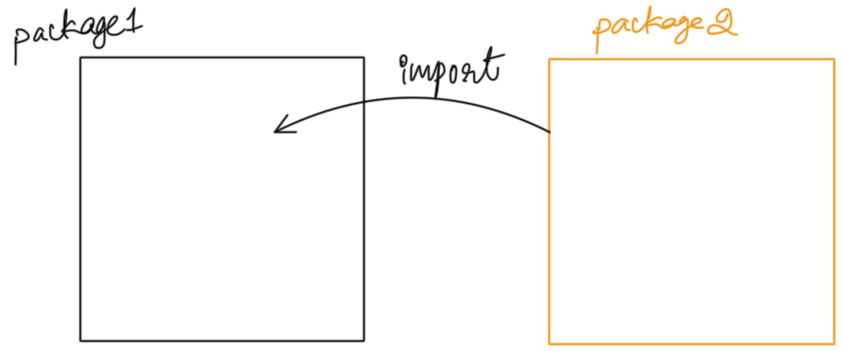

connection Between any other packages of software
Application / services
T4
T5
different teams work on different package or layers
Software Block Diagram
T1
T2
T3
Teams

An Application or a software Service contains different packages, components, layers, Relations

If we organize and maintain the Software in a systamatic way so other Software developers can work collaboratly.

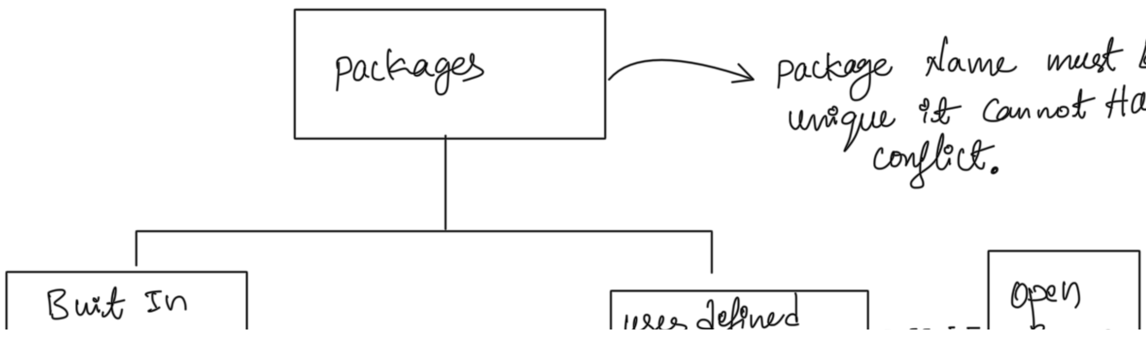
different Teams work on different package or Module of Software.

Import (getting reference from another package to a package)



Advantages of using Packages in JAVA

- Modularity :-> Packages Break large projects into smaller, Managable, development Easy
- Reusability :-> classes inside a package can be reused across different projects or Part of Same project. helps to avoid Redundancy.
- NameSpace Management :-> Prevents class Name conflicts
- Organization :-> Package organize classes and interface into meaningfull related groups improving code Readability.
- Access Control :-> Packages Provide different levels of access protection (default, public, protected) to Encapsulate classes and Interfaces.

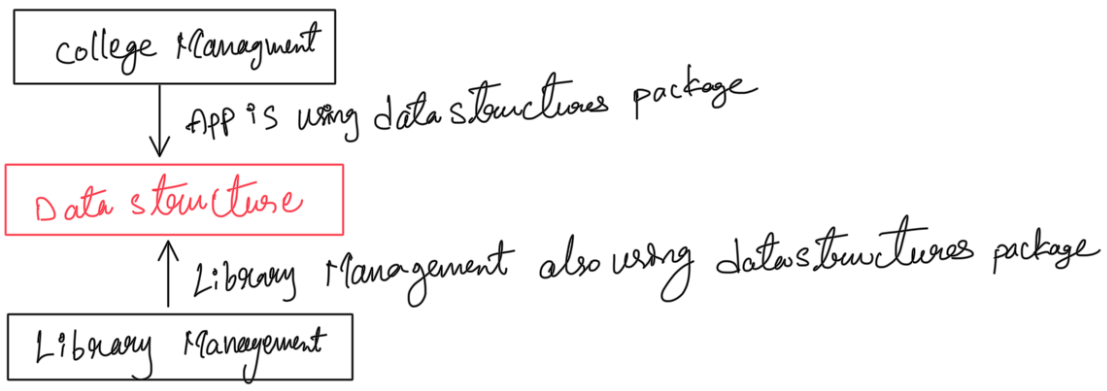


package Name must be unique it Cannot Have conflict.

Multiple packages in a Software But only the required things are used in a file to perform Some operation.

import describe which package is connecting with which package. only that connected package is used in a particular package.

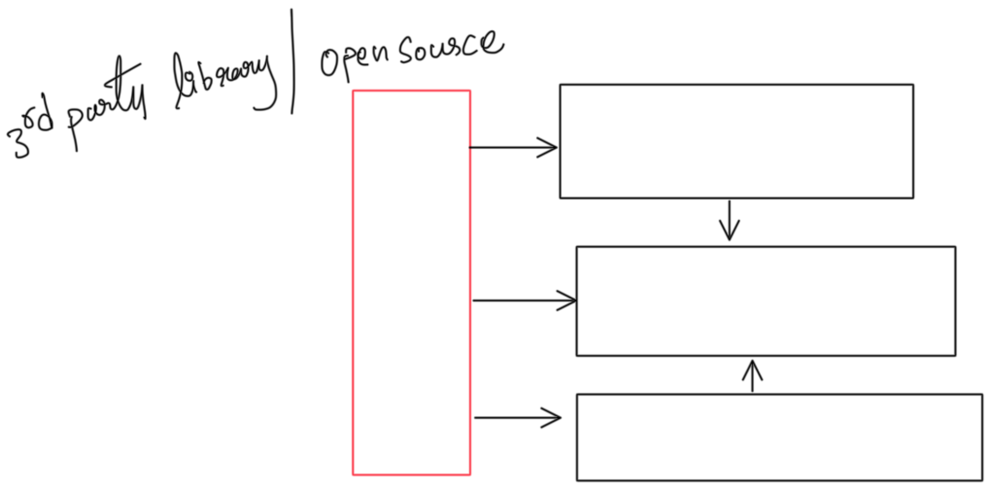
Lets understand with Simple Example.


college Managment
App is using data structures package
Data structure
Library Management also using datastructures package
Library Management

We can observe in the above figure that the college Management Not using the Library Mangament & Vice Versa.

Block diagram :- Understanding Which package is used By which file and shows the connectivity with Neat picture.

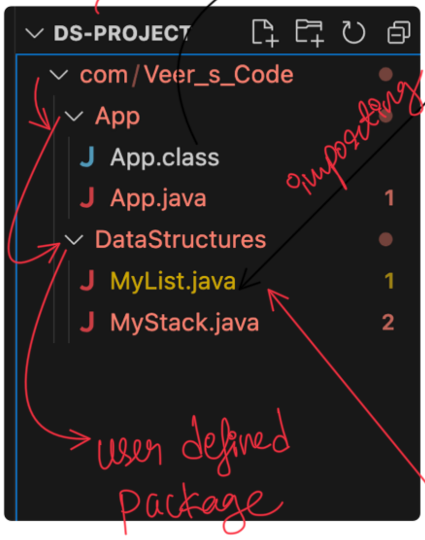
With the help of Block diagram We can Have the Clear picture of the Software its working packages & Connectivity.

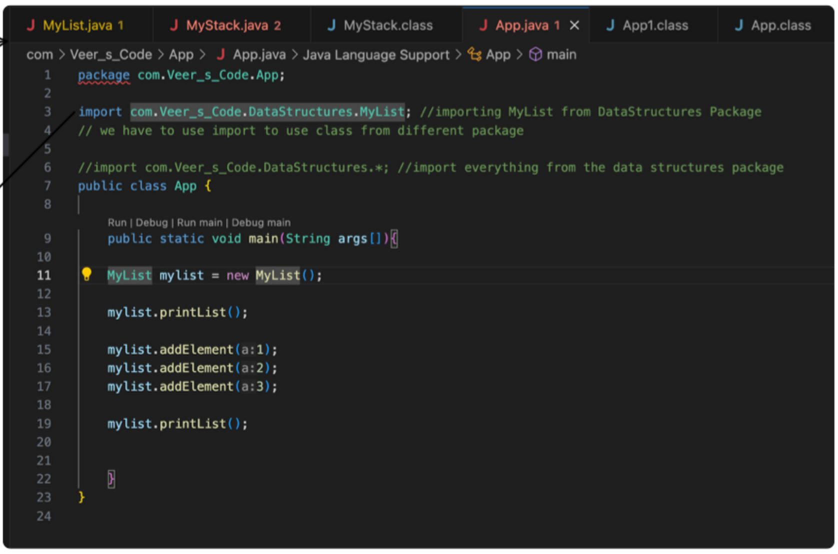


Block diagram.

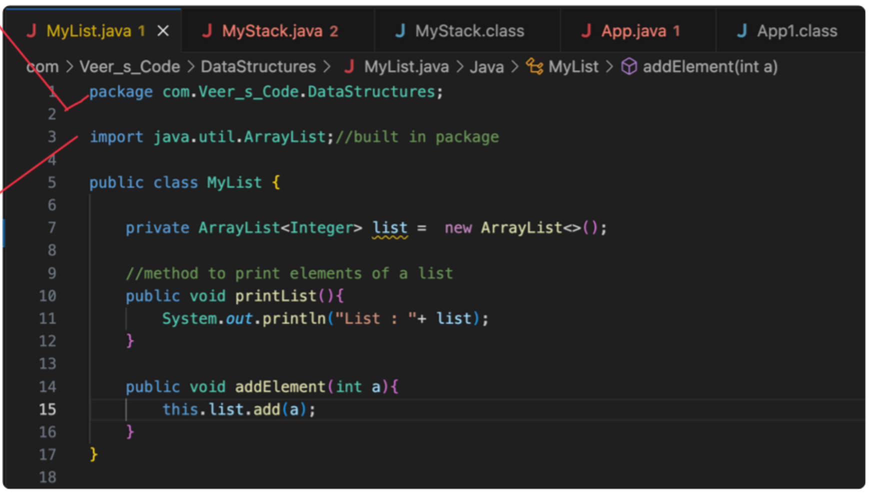
let us understand with Example

Project folder




MyList.java 1
MyStack.java 2
MyStack.class
App.java 1
App1.class
App.class
com > Veer_s_Code > App > App.java > Java Language Support > App > main
Run | Debug | Run main | Debug main

```java
package com.Veer_s_Code.App;

import com.Veer_s_Code.DataStructures.MyList; //importing MyList from DataStructures Package
// we have to use import to use class from different package

//import com.Veer_s_Code.DataStructures.*; //import everything from the data structures package
public class App {

    public static void main(String args[]){

    MyList mylist = new MyList();

    mylist.printList();

    mylist.addElement(a:1);
    mylist.addElement(a:2);
    mylist.addElement(a:3);

    mylist.printList();


    }
}
```


MyList.java 1
MyStack.java 2
MyStack.class
App.java 1
App1.class
com > Veer_s_Code > DataStructures > MyList.java > Java > MyList > addElement(int a)

```java
package com.Veer_s_Code.DataStructures;

import java.util.ArrayList;//built in package

public class MyList {

    private ArrayList<Integer> list =  new ArrayList<>();

    //method to print elements of a list
    public void printList(){
        System.out.println("List : "+ list);
    }

    public void addElement(int a){
        this.list.add(a);
    }
}
```

Importing Built in package