JAVA EXAM

1. Collection Framework
   a. What are the 9 interfaces in collection framework? Give Example for each interface?

ANS: 9 key interfaces of collection framework:

1.Collection Interface: If we want to represent a group of individual objects as a single entity then we should go for collection.

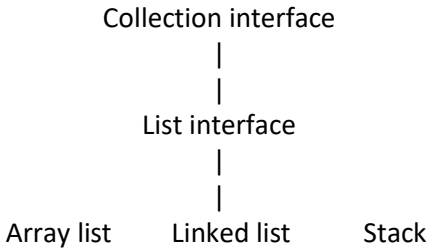
Collection is the root of all the interfaces and classes.

Collection interface defines the most common methods which are applicable for any collection object.

The interface is considered as the root interface of the collection framework.

There is no concrete class that implements collection interface directly.

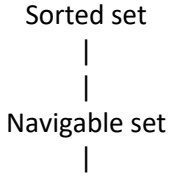
2.List Interface: It is child interface of collection. If we want to represent a group of individual objects as a single entity where duplicates are allowed and insertion order must be preserved then we should go for it.



3.Set Interface: It is the opposite of list. It is the child interface of collection. If we want to represent a group of individual objects as a single entity, where duplicates are not allowed and insertion order not required then we should go for set interface.

4.Sorted Set: It is the child interface of a set .If we want to represent a group of individual objects as a single entity where duplicates are not allowed and all objects should be inserted according to some sorting order then we should go for a sorted interface.

5.Navigable set: It is the child interface of the sorted set. It contains several methods for navigation purposes.



|
Tree set

6.Queue: It is the child interface of collection. If we want to represent a group of individual objects prior to processing, then we should go for queue. Usually, the queue follows first in first out order but based on our requirement we can implement our own priority order also.

Iterable
|
|
Collection
|
|
Queue
|
|
Deque
|
|
Abstract
|
|
Array deque

7.Map interface: It is not the child interface of collection. If we want to represent a group of objects as key value pairs then we should go for a map. Both key and values are objects, only duplicate keys are not allowed but values can be duplicated.

8.Sorted map: It is the child interface of map. If we want to represent a group of key value pairs according to some sorting order of keys. In a sorted map the sorting should be based on key but not on value.

9.Navigable map: It is the child interface of sorted maps it defines several methods for navigation purposes.

b. What is the difference between Array List and LinkedList? Give Examples?

Ans: Difference Between ArrayList and LinkedList

1. ArrayList internally uses a dynamic array to store the elements whereas LinkedList internally uses a doubly linked list to store the elements.

2. Manipulation with arraylist is slow because it internally uses an array .If any element is removed from the array all the other elements are shifted in memory whereas Manipulation with Linkedlist is faster than arraylist because it uses a doubly linked list so no bit shifting is required in memory.
3. An ArrayList class can act as a list only because it implements List only whereas LinkedList class can act as a list and queue both because it implements List and Dequ e interfaces.
4. Arraylist is better for sorting and accessing data whereas linked list is better fpr manipulating the data.
5. The memory location for the elements of an ArrayList is contiguous whereas The location for the elements of a linked list is not contagious.
6. Generally when an arraylist is initialized a default capacity of 10 is assigned to the array list whereas there is no case of default capacity in linked list.In linked list an empty list is created when the linked list is initialized.
7. To be precise, an ArrayList is a resizable array whereas LinkedList implements the doubly linked list of the list interface.

Example:

```
import java.util.*;
class TestArrayLinked{
 public static void main(String args[]){

 List<String> al=new ArrayList<String>();//creating arraylist
 al.add("Ravi");//adding object in arraylist
 al.add("Vijay");
 al.add("Ravi");
 al.add("Ajay");

 List<String> al2=new LinkedList<String>();//creating linkedlist
 al2.add("James");//adding object in linkedlist
 al2.add("Serena");
 al2.add("Swati");
 al2.add("Junaid");

 System.out.println("arraylist: "+al);
 System.out.println("linkedlist: "+al2);
 }
}
```

Expected output:
Arraylist : [Ravi, Vijay, Ravi, Ajay]
Linkedlist : [James, Sarena, Swati, Junaid]

2.Spring Boot

a. What is Spring boot? What are the advantages over Spring boot? What is The Spring Boot Architecture

ans: Spring Boot is a project that is built on the top of the spring framework. It provides an easier and faster way to set up, configure and run both simple and web based applications.

It is a spring module that provides a RAD (Rapid Application Development) feature to the spring framework. It is used to create stand alone spring based application that you can just run because it needs minimal spring configuration.

Spring framework + Embedded HTTP Servers -XML Configuration = Spring Boot

In short Spring Boot is the combination of both soring framework and Embedded Servers.

In spring boot there is no requirement for xml configuration. It uses convention over configuration software design paradigm that means it decreases the effort of the developer.

We can use spring initializer to develop spring java applications

Uses of spring boot: spring boot is used because

The dependency injection approach is used in spring boot.

It contains powerful database transaction management capabilities.

It simplifies integration with other java frameworks like LPA/Hibernate ORM, Struts etc.

It reduces the cost and development time of the application.

Advantages:

It creates stand alone spring applications that can be started using java jar.

It tests wen applications easily with the help of different embedded http servers such as tomcat, jetty, etc. We don’t need to deploy war files.

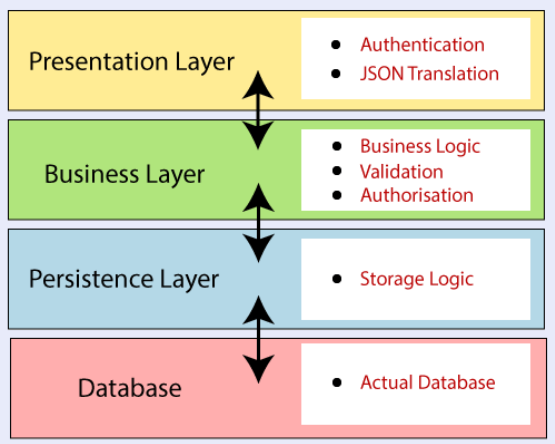
It offers the number of plugins.

It increases productivity and reduces development time.

There is no requirement for XML configurations.

It provides production ready features.

Spring Boot Architecture:



There are four layers in spring boot:

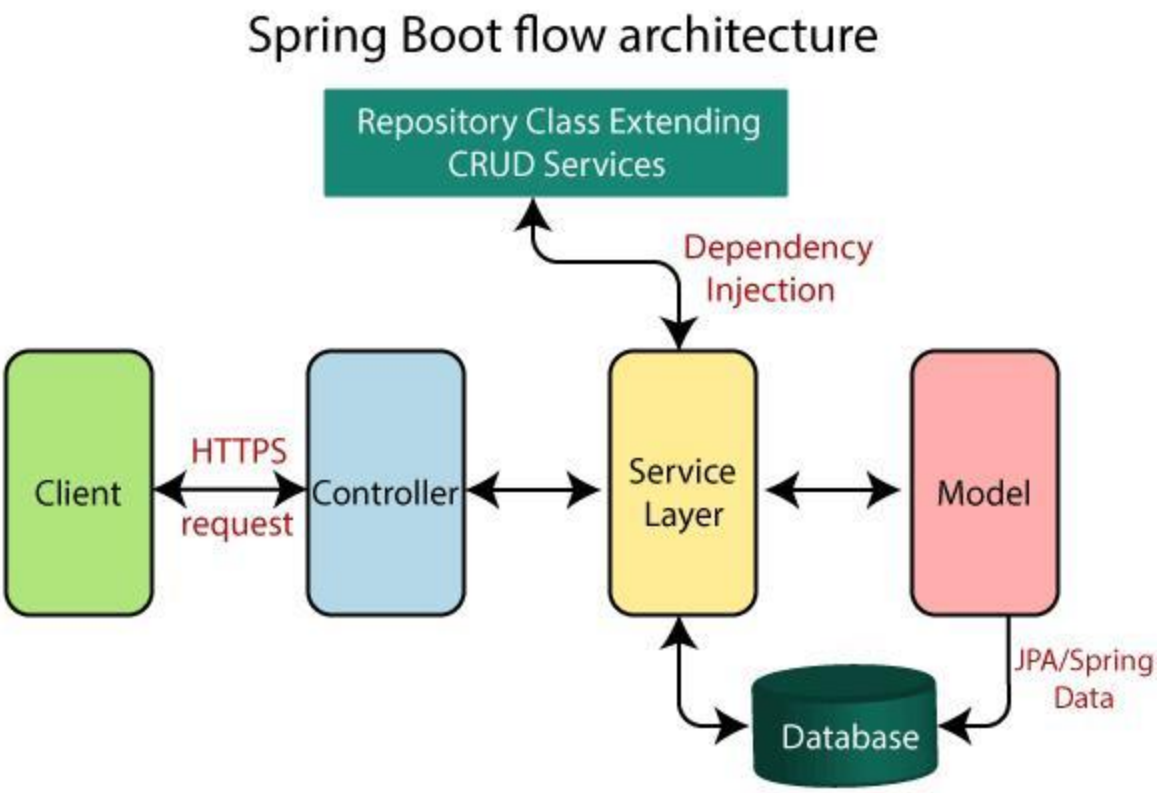
Presentation Layer: It handles the http requests, translates the json parameter to object and authenticates the request and transfer it to the business layer.

Business Layer: It handles the business logic. It consists of service classes and uses services provided by data access layers. It also performs authorization and validation.

Persistence Layer: It contains all the storage logic and translates business objects from and to database rows.

Database Layer: In this layer CRUD operations are performed.

Spring Boot flow architecture:



Spring Boot uses all the modules of spring like spring MVC, Spring data etc. The architecture of spring boot is the same as the architecture of spring MVC except one thing, there is no need for DAO classes in Spring boot.

Creates a data access layer and performs CRUD operation.

The client makes the HTTP requests (PUT or GET).

The request goes to the controller, and the controller maps that request and handles it. After that, it calls the service logic if required.

In the service layer, all the business logic performs. It performs the logic on the data that is mapped t o JPA with model classes.

A JSP page is returned to the user if no error occurred.