1.Explain your project architecture?

2.Explain spring MVC flow?

3.List out all the annotations in spring?

4.what are all the annotations you used in your project on restful services?

5.What is the use of cascade and inverse in Hibernate?

6.what is first level cache and second level cache in hibernate?

7.what is diff b/w ArrayList and LinkedList?

8.can you explain the internal flow of HashMap?

9.what is the diff b/w HashMap and Hashtable?

10.Diff b/w Array and ArrayList?

11.Diff b/w ArrayList and Vector?

12.In your project where you used cuncurrent hashmap?

13.What is java annoying?

14.Diff b/w callable interface and future interface in concurrent package?

15.What is class loaders?

16.How can you take List into Map?

17.How can you take Map into List?

18.when you will get ClassNotFoundException and NoClassDefFoundError?

19.How you implement exception handling in your project?

20.where you implement multi-threading in your project?

21.what are all the design patterens you observed in spring?

22.which design patters you used in your project?

23.what are all the critial situations you come across in your project?

24.why wait() placed in object class ? why not it is placed in Thread class?

25.what is use of intern() in spring?

26.what is diff b/w String str="xyz"; and String str2= new String("xyz");

27.Expalin about java architectue?

28.Explain about jvm architecture?

29.Data base queries?

30.I have a company table in remote database. by using rest i need to get the table data and print into a file?

31.how to read book pages on online library by using bookid or author id(by using restful services)?

32.I have a table in remote database, how to update the data in that table using rest?

33.Diff b/w rest and web(soap)?

34.Agile methodolgy?

35.How to create web-services project and spring project using mavan?

36.what is diff b/w throw and throws?

37.can you tell me java8 features?

38.what are all the contents in wsdl?

39.Refer regular expressions?

40.can i add elements to list , if it is defined as final

ex:final List<String> list= new ArrayList<>();?

41.if you pass duplicate key to map what will happen?

42.Diff b/w abstract class and interface?

43.Diff b/w comparator and comparable?

44.How to compare two database tables(clue: comparator, compare(),

you have to compare database objects.)?

45.How to set timeout for the browser?(clue: restful client api.)?

46.what workflow you used in your project?

47.why java? why not c & c++?

48.In written test they are asking sorting programs(bubblesort,quicksort,...)?

49.what is time complexity? if you are going to implement sorting by your own which sorting you prefer? and why?

50.what is the use of volatile and synchronized?

51.what is serialization? have you implement serialization in your project?

52.programs on io streams?

53.why we are using @qualifier?

54.Diff b/w BeanFactory and ApplicationContext?

55.Explain about IOC container?

56.programs on string manipulations?(they are expecting solve by using regular expressions).

57.How you are implemented polymorphism in your project?

58.How you iterate map having key and list<values>?

59.Diff b/w Iterator and ListIterator and Enumarator?

60.what are all the collections are supporting ListIterator?

61.How to make non-synchronized map and list as synchronized(by using collection method)?

62.what is diff b/w collection and collections?

63.write the junit test case for the below senario..

-->read array of elements into list<>.

64.what are all the modifiers we can use inside method?(ans: only final)

65.what is diff b/w spring-jdbc and hibernate?

66.what are all the drawbacks of jdbc over hibernate?

67.what are all the problems with inheritance?

68.what is the use of hibernate session?

69.They given one query in sql and they are asking corresponding criteria api query?

70.why we are using @transient in hibernate?

71.what are all the inputs we are giving to SessionFactory?

72.what we are writing in hibernate-mapping file?

73.what we are writing in hibernate-configuration file?

74.senario: in jsp page with 2 buttons, one for addbook and another is for showListOfBooks(by using spring and hibernate)?

75.what is use of @ComponentScan?

76.what is use of dispatcher servlet?

77.what are all the pre-processings tasks done by DispatcherServlet?

78.How to render excel and pdf view to the enduser(using poi and itext api's)?

79.How to validate valid username and password in spring?for validating

can i directly interact with dao without service?

80.By defalut servlet container will handle multi-threaded applications ,

then why you are implementing multi-threading in your application

-----------------------------------------------------------------------------------------------------------------------------------

1) Immutable Object is thread safe ?

2) I have two thread one thread need to read data from file and another thread need to write that data in another file?

3) By using wait and notify methods need to print even and odd number?

4) I have 9 users and i have 3 passwords so here how to share these 3 passwords with those 9 users ?

5) I have one integer[] ar={10,20,30,40,10,5,46,30}with some elements so how to remove duplicate elements with this array without using collection?

6) I have 2 Strings like String s1="hello world &&@" and String s2="Hai Ram #@#" so here with these two Strings i want to remove common elements without using collections?

7) I have one String like String s="i am a java developer" for this String i want to count how many times each word is repeated without using regular expressions ?

8) I have one hash Map object with this object i want to display only repeated values means suppose if you have one hash map it contains 1 "a",2 "b", 3 "c" 4 "d" 5 "a", 6 "b", 7 "a" with this hash map i want to display only repeated key and value pair ?

9) I have one array list object it contains some elements so here i want to display only duplicates please write the logic?

10) In my application end user as part of search he is entered String like "amaz#$#%n" so here how you will replace with those special characters with some space.

1. Difference between annotation and interface?

2. Difference between String, StringBuffer, StringBuilder?

3. Internal data structure used in AL?

4. Find duplicate numbers from an Array?

5. String anagram?

6. Occurrence of each word in String?

7. Java is what pass by value or reference, justify?

8.Oops concept?

9.How to create custom exception?

10.How to create custom error?

11. Servlet life cycle?

12.Jsp implicit objects?

13. What is IOC?

14.Different types of Injection?

15. How to make employee object as key in hashmap?

16.What is PrepareStatement?

17. What are the methods available in resultset?

18. How to traverse a resultset in reverse order?

19. How to update data in resultset?

20.What is Agile? How you are using agile in your project?

21.Difference between Agile and Waterfall?

22. MVC architecture?

23.How to integrate WS with Spring?

24.How to integrate Restful with Spring?

25.Types of IOC?

26.Difference between ApplicationContext and BeanFactory?

27.How to manage transaction if there are multiple transactional resources(database, jms)?

28.What is spring transaction?

29.How to execute store procedure in java?

30.Difference between HashMap and HashTable?

31.What happen when we are using annotation?

32.Http status codes?

33.Difference between put and post?

34. What are the different http methods?

35.What is Hibernate Dialact?

36.What is HibernateTemplate and what are the advantages of using HibernateTemplate?

37. Define your project architecture?

38.What is CI? How to use Hudson?

39.What is CDN server? Why it is used?

40.How website is hosted?

41. What is scrollable?

42.How to redirect a .htm page to a .jsp page in spring?

43.If there is a css file for a table which is included into a jsp page and if you are overwriting the stylesheet in the jsp file for the table then what will happen?Will it reflect the jsp styles or css file style? 44.How to make a class singleton? What happen if I have a custom classloader , will the singleton class will work or not?

45.Define the classloader hierarchy? 46. Bean Life cycle?

47.Bean scopes?

48.Difference between Struts and Spring?

49. Difference between SOAP and Rest?

50.When destroy method will be called in a servlet?

51. How to create a session inside a servlet?

52. What are the exceptions thrown by a DAO class in spring?

53.If there is a complex query how do you execute it? Will you take the help of hibernate or native sql query?

54.How to normalize with the help of Hibernate?

55.What are the new features added in jdk1.8, explain?

56.Annotation support is there in Hibernate3 and Hibernate4?

57.What is the percentage of code coverage in your project?

58.What are the different ways to crate a servlet?

59.Which method will be called in a servlet when a request is coming?

60.If multiple requests are coming how many objects will be crated for the servlet?

61.What is a marker interface?

62.What do you mean by persistency?

63.What is DriverManager?

64.Difference Aggregation and Composition?

65.What is caching in Hibernate? What are different levels of caching available?

66.What are the other ORM frameworks available in market? What is the difference between them and Hibernate?

67.What is Mocking? What are the libraries available in market? Which one you are using?

68.What is PrepareStatementCreator?

69.If you are using ApacheCXF and if you want to expose them as rest resources, then how to do it?

70.How did you get the offer letter from your previous company?

71.Why are trying to change the company?

72.Why do you want to join a startup company(Asked in a startup product based company)?if you got a offer letter from Congnizant or any big company and if you got another offer letter from a startup company then which one will you join( Both are offering the same salary package)?

73.Why did you came to Bangalore?

74.How much salary you are expecting ? Justify ?

75.What do you mean by Techno function?(If you are mentioning in your resume that you are Techno function then they are asking , How do you became a Techno function within 3year)?

=====================================================================================

**Some Commonly asked interview questions.....**

Q)What is the output?

public class StringOperation {

public static void main(String[] args) {

String s1="abc";

String s2="def";

String s3=s2;

s2="ghi";

System.out.println(s1+s2+s3);

}

}

Ans-abcghidef

Q)What is the output?

String a="newspaper";

a=a.substring(5, 7);

char b=a.charAt(1);

a=a+b;

System.out.println(a);

Ans-app

Q)What is the output?

int i=1;int j=10;

do

{

if(i++ > --j){

continue;

}

}while(i<5);

System.out.println("i="+i+" and j="+j);

Ans-i=5 and j=6

Q)Wap to find sum of first 100 prime numbers?

Q)Wap to print prime nos from 1 to 100?

Q)Wap to find sum of digits in the given nos?

Q)Wap to find top two maximum nos in an array?

Q)Wap to sort a map by value?

Q)Wap to get a line with max word count from a given file?

Q)Suppose an ArrayList contains a list of employee objects with empid,name,dept as its attribute.Wap to sort this list by empid?

Q)Wap to craete deadlock between two threads?

Q)Wap to find maximum repeated words from a file?

Q)Wap to find out duplicate characters in a string?

Q)Wap to find max no from an array?

Q)Wap to add objects and then list the content from an ArrayList<String> ?

Q)Wap to print 1-50 using onr thread and 51-100 from another one simultaneously?

Q)Wap to get distinct elements from an array?

Q)Wap to convert decimals to binary format?

Q)Wap to print Fibonacci series?

Q)Wap to remove duplicates from sorted array?

Q)Wap to find given no is Armstrong or not?

**interview questions as part of my interviews till now**

----------------------------------------------------------------------------

Have you implemented Security in your application?

What kind of security you have implemented, tell me about it!

How you can get list of unique words from a sentence without using iterator 2nd time.

Is Hibernate is completely independent of database?

When we use @GeneratedValue(Identity) in Hibernate, is it possible to work for all DB like Oracle/MySQL/postgre sql …

When we go for comparable and comparator?

Where you have implemented comparable and comparator in your project give one scenario.

Can we sort a Map based on its value? If yes how?

Show the Collection hierarchy!

What's difference between Collection and Collections?

Where you have used multi-threading in your application?

Where you have used Linked List in your application?

What's different between HashSet and TreeSet?

Where you have used logging and auditing? How you implemented this? (I mentioned this as part of my resume)

Do you know string pooling

Do you know connection pooling design pattern

If multiple browsers send multiple requests to server, how does server identifies for which request it should response differently? Is the connection live after making a request to the server from a browser? If yes for how long time? It no how does server the request has been timed out/connection lost! How it sends the response back to the browser?

Do you know connection pooling? Do you know connection pooling design pattern?

I have a file to read or writing. Provide me a connection pool design pattern, come up with your own classes and components by which I'll able to call your classes whene'er I'll need reader and writer.

I've a requirement, I've a application (like ICICI ) whene'er user want to login , he'll try to get the pre-login page. And asking with the pre-login page I want to show the stock Price graph of one suggested stock. The stock should be keep showing without user interaction. The stock details should be fetched from NSE rest api. After getting the stock details at a point of time, how you'll maintain the previous data and recent data? How you'll send all these data to login page continuously? If you think Ajax, then what is Ajax? Come up with complete flow and design when I'll able to understand and I'll think yes this way it can be implantable.

What's collection?

How do you put a list of strings into set? Is there any utility method?

Do you use Rest in your project? Tell me about it?

What are components, methods of rest?

You know SQL query?

How you'll fetch multiple column values from 2 tables based on one column sorted in descending.

How you'll iterate a Map?

What's your difficulty task in your project?

How do you work on daily basis?

What are all the responsibilities of yours in your project?

How you made yourself learn all these technologies?

How do you communicate with others in your team?

Do you follow agile? What's agile?

Do you have any experience in deployment?

Tell me the project flow with code diagram?

Do you know spring elastic?

Do you know ELK?

Why don't you update to latest things like spring elastic?

Tell me some of your classes names based on a module?

How do you transfer data between the project's?

What was your responsibility?

What's Jira?

Do you have implemented any security?

How you made performance optimization in your application?

=====================================================================================

[11:24 PM, 3/20/2018] +91 89829 43755: MANDATORY

Support a current transaction, throw an exception if none exists.

NESTED

Execute within a nested transaction if a current transaction exists, behave like PROPAGATION\_REQUIRED else.

NEVER

Execute non-transactionally, throw an exception if a transaction exists.

NOT\_SUPPORTED

Execute non-transactionally, suspend the current transaction if one exists.

REQUIRED

Support a current transaction, create a new one if none exists.

REQUIRES\_NEW

Create a new transaction, and suspend the current transaction if one exists.

SUPPORTS

Support a current transaction, execute non-transactionally if none exists.

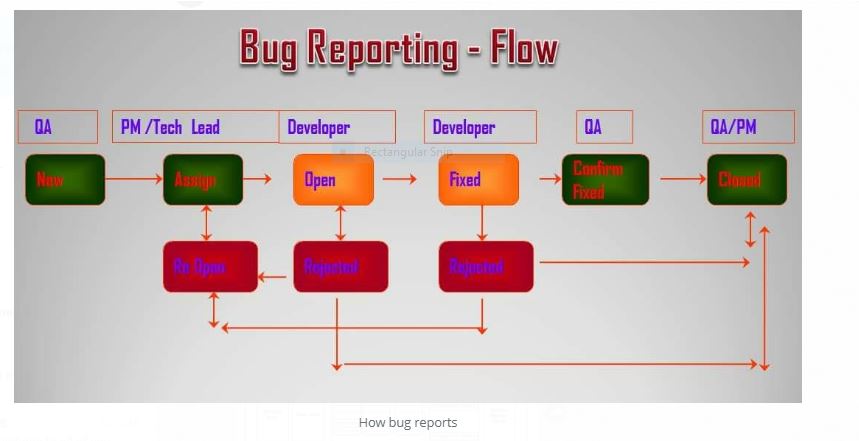
[11:24 PM, 3/20/2018] +91 89829 43755: Why we should place view component under WEB-INF?

To abstract the presentation tier technology to the end user, if in future my presentation tier technology has been changed then end user has to know for sending the request to my application.

-security

-incorrect view might access by the end user which will give a ugly error page to end user

**Bug Reporting Flow:**

****

In this blog am going to explain the bug reporting flow:

In software testing process a defect will undergo different status like

New Open Assigned Reject Fixed Reopen Retest closed deferred etc.

There are three different possible situations in any automated and manual testing:

· New-closed

· New-reject

· New-deferred

The time taken for defect to be closed or rejected or deferred is called BUG LIFE CYCLE.

Status: description:

New: a defect/bug detected by tester.

Open: a defect/bug that has been reviewed and verified.

Rejected: a defect/bug that has been found not to be a real defect.

Assigned: a defect/bug that has been verified and now assigned to the developer for fixing.

Fixed: a defect that has been fixed by developer and it will ready to retest that defect.

Retest: a defect that has been fixed by developer and it will ready to retest that defect for to make it close.

Reopen: a defect that has been failed during retesting and it need to reopen that defect.

Closing: a defect that has been successfully retest and we need to mark it as close.

Deferred: a defect that has been move to the next release after the mutual agreement of development lead and test lead.

How to report a track defect?

Here am just take a dummy example like duplicate form submission.

After the development of an application it will moves from development environment to QA environment.

And then test engineer will start the testing by sending request from the login page of the web application. So while testing login page tester found that it will supports duplicate form submission. So immediately tester logs to the test management software and register a new defect.

What is the roll and responsibility of tester, test lead and developer and development lead and developer:

Tester finds the defect and report to the testing lead. Test lead verify that defect and if it is valid defect create a NEW status for that defect and this defect will forward to the development team. Again development lead verify the defect and if it is a valid defect then mark or maintain the status ASSIGNED and assigned to the developer.

Developer works on that issue and after fixing it he will mark it as FIXED.

And this is reported to testing lead. Test lead change the status RETEST and assign to the tester. The test engineer retest it and if everything is close then he will mark the status as CLOSED.

**What do following junit test annotations mean?**

Ans : Following is a list of frequently used JUnit 4 annotations:

@Test (@Test identifies a test method)

@Before (Ans: @Before method will execute before every JUnit4 test)

@After (Ans: @After method will execute after every JUnit4 test)

@BeforeClass (Ans: @BeforeClass method will be executed before JUnit test for a Class starts)

@AfterClass (Ans: @AfterClass method will be executed after JUnit test for a Class is completed)

@Ignore (@Ignore method will not be executed)

[11:25 PM, 3/20/2018] +91 89829 43755: How do one do exception handling unit tests using @Test annotation?

Ans : @Test(expected={exception class}. For example: @Test(expected=IllegalArgumentException.class)

[11:25 PM, 3/20/2018] +91 89829 43755: 1) thread - This is called current session context class. Session object we created using the SessionFactory will be scoped to the Current Thread of execution and once the thread finished execution the session will be closed automatically. In short for session management in application we use this.

2) NoCacheProvider - In hibernate for Performance optimization and Better memory management we use Cache at various different levels like First-Level and Second-Level cache. While using the second-level cache we need to configure the cache provider. The above configuration indicates we are not using second-level cache.

3) In hibernate application we need to create only one SessionFactory object for one database in an application. This can be better done in web application using ServletContextListener implementation class as it would be called only two times at the start of application and end of application.

[11:25 PM, 3/20/2018] +91 89829 43755: Differences between the Query, Path and Matrix Params and when to use them:

Irrespective of parans we can send the data as part of path of the URI.

1. Query and Matrix params are optional but Template params are mandatory.

2. Query params always can appear at end of the req URI. Template params can be appear in anywhere of the URI and these params are mandatory. Matrix params can be appear in where in the req of part of path of URI and before the Query params.

3. The Template params involves/influences the way of resolving the req bcz these are mandatory but not by Query or Matrix params bcz they are optional.

4. The scope of the Matrix params is part/segment scope but not entire URI but the Template params scope is of entire req URI level. The scope of Query param is also entire req scope but scope is not make sense for Query params bcz these are always can be appear at end of the req URI but scope make sense for Matrix and Template bcz these can be passed in anywhere of the req.

5. We can send multi-values for an single param in case of Query and Matrix (in case of multiple values it takes 1st values only) but we cannot send in case of Template param.

=====================================================================================

====================================================================================

Ans:------->

**Internal flow of HashMap ?**

Normally HashMap follows below 3 steps

1. hashCode() to find bucket

2. == operator to reference comparison

3. equals () to content comparison

When we are save object in map means map.put(Object,Object) then first hashMap internally use hashing technique to find the bucket so it find the hashCode of object then % by initial capacity then bucket number will be get and object will placed on that specific bucket no Ex: (key.hashCode()%11)

After placing object if in same bucket 2 objects are placed then there may be a chance key collision .to avoid this situation it internally uses == operator so here it will check whether these 2 objects have same hashCode or not? If hashCode is different then it will place the object in map else if both have same hashcode then there may be chance hashing collision.

If hashing collision occurs then to overcome this issue it internally use  equals method to content comparison  it check obj1.equals(obj2) if content same then replace the value else add that object into map.

**AOP INTERNAL: ----**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

public class LoggingAspect implements MethodInterceptor {

@Override public Object invoke(MethodInvocation methodInvocation) throws Throwable { methodName = methodInvocation.getMethod().getName();

args = methodInvocation.getArguments();

targetObject = methodInvocation.getThis(); // log before target class method has been executed System.out.println("entered in to " + methodName + "(" + args[0] +"," + args[1] + ")");

args[0] = (Integer) args[0] + 1;

args[1] = (Integer) args[1] + 1; // let the target class method executes Object ret = methodInvocation.proceed();

int modifiedRet=(Integer)ret+1; // log after target class method has been executed and before returning System.out.println("returning from " + methodName + " before ret with return value :" + ret);

return modifiedRet;

}

}

public class AroundAdviceTest {

public static void main(String[] args) { // Perform the Weaving to build the proxy ProxyFactory pf = new ProxyFactory();

pf.setTarget(new Calculator());

pf.addAdvice(new LoggingAspect());

Calculator proxy = (Calculator) pf.getProxy();

int sum = proxy.add(10, 20);

int sum = proxy.add(10, 20);

int sum = proxy.add(10, 20);

}

}

class Calculator {

private int delta; //setter }

Calculator cal=new Calculator();

c.setDelta(1);

class Cache { } public class CacheAspect implements MethodInterceptor {

public Object invoke(MethodInvocation methodInvocation) throws Throwable {

cache = Cache.getInstance();

if (cache.containsKey(key)) {

return cache.get(key);

} // Note: // If if condition is true then there only control given back // and below lines for execute the target class method and will // not happen bcz after return key word there no execution // bcz control given back already // let the target class method executes if if-block is fails ret = methodInvocation.proceed();

cache.put(key, ret);

return ret;

}

}

class Calculator$Proxy extend Calculator {

private List<Object> advices;

private Object target; public Calculator$Proxy(List<Object> advices) {}

public int add(int a,int b) {

MethodInvocation methodInvocation=new MethodInvocation();

methodInvocation.setMethod(//stack trace);

methodInvocation.setArguments(//stack trace);

methodInvocation.setThis(this);

return advice.invoke(methodInvocation);

}

}

class MethodInvocation {

private Method method;

private Object[] args;

private Object target;

public Object proceed() {

return method.invoke(target,args);

}

}

**Spring Annotation: ---**

@Component---------------@Component ,@Service, @Controller, @Repository all these are stereo type annotations supplied by spring. @Component annotation is used to configure java class as spring bean.

@Service---------------------it is used to make spring bean as service layer component

@Controller-----------------it is specialization of @Component annotation used to configure as controller layer component to received request and process the requests received from clients.

@Repository----------------it is also a specialization of @Component annotation which is used to make spring bean as DAO layer component. It will be able to convert the one form of exceptions into another form of exceptions.

@Required------------it is used to specify that DI is mandatory on some properties but it is removed from spring 2.5

@Autowired----------it is used to enable byname, byType, constructor mode of auto wiring. It is spring supplied annotation. It can be applied at field level. constructor level, setter method level, orbitrary method level. it can be applied on only one parameterized constructor but there is no limit for methods and fields.

@Value-------------it is used to perform DI on simple type properties

@Qualifier---------it is spring supplied annotation. When we enable byType mode of auto wiring there is a chance to get ambiguity problem when multiple properties are found to inject to the target. so to overcome this problem we can use @Qualifier annotation so that IOC container performs byname mode of autowiring.

@Configuration----------it is used to make java class as Configuration class .it is alternate for spring bean xml configuration file. configuration class will have methods that each method will create and return the object of spring beans.to make the returned objects as spring beans we have to use @Bean annotation.

@Bean------------------------it used to make the return type of methods of configuration class as spring beans.

@ComponentScan---------it is alternate for < context:component-scan >tag used in xml file. using this we can specify the list of packages where we used stereo type annotations annotations like constroller,service….so that IOC container will search the packages and apply the functionality at runtime.it is mainly used to reduce the scan time.

@Dependson----------------in order to create one bean class object after creating one bean class object only.then we can use this annotation.it will take bean id of the depedent bean.

@Lazy---------------as we know that all the singleton scpe beans will be preinstantiates at the time of activating the container itself. But if you want to delay the instantiation of singleton beans you can use @Lazy annotation .it will take Boolean value true or false.if true it performs lazy loading.

@Scope----------used to specify the scope of spring bean.values can be singleton,prototype,session,request.

@RequestParm------------to specify the request parameters

@PathVariable-----------------it is used to specify the path variable which will be appended to the url as wury param.it will be appended to url with “/”.multiple parameters will be separated with “/”.

@ModelAttribute------------it is multi purpose annotation. it is used to make the return type or parameters of the handler methods or command class object as model data so that can be passed to the webview.

@RequestMapping----------it is spring MVC specific annotation. It is used to map the controller/handler class methods with the incoming request urls based on the virtual path of te uri.

@InitBinder -----------it is alternate for initBinder() method of SimpleFormController.it is used to convert the form page supplied properties into command class supported properties we need to override initBinder() method.we can bring the same effect using @InitBinder annotation.

**Semaphore**

A semaphore controls access to a shared resource through the use of a counter. If the counter is greater than zero, then access is allowed. If it is zero, then access is denied. What the counter is counting are permits that allow access to the shared resource. Thus, to access the resource, a thread must be granted a permit from the semaphore.

Working of semaphore

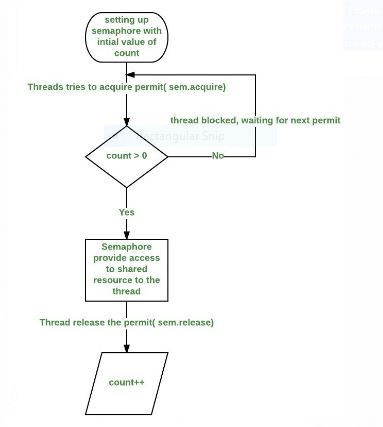
In general, to use a semaphore, the thread that wants access to the shared resource tries to acquire a permit.

If the semaphore’s count is greater than zero, then the thread acquires a permit, which causes the semaphore’s count to be decremented.

Otherwise, the thread will be blocked until a permit can be acquired.

When the thread no longer needs an access to the shared resource, it releases the permit, which causes the semaphore’s count to be incremented.

If there is another thread waiting for a permit, then that thread will acquire a permit at that time.



+91 89829 43755:

**What is a business delegates? Why do we need business delegate?**

**Action class directly call the DAO, then why the action class calling the bussiness delegates and business delegates will call dao?**

Ans:- If the action class is directly calling the dao, then my presentation tier component will be exposed to the persistency tier. So that any changes happens in the presentation tier will impact to the presentation tier (eg: changes in table schema, or change in technology), if an exception has been reported by the dao it will directly reported to presentation tier component.

In case of struts based project architecture, jsp will act as view component. When we submit the request from the jsp page, the request will comes to Action class, action class has to read the data in term of action-form. Bussiness delegates and dao is responsible for performing the business and persistency operation. Upon receiving the request from the jsp page Action class will call the business delegate and dao. While performing the persistency operation, dao runs into exception, then dao has to throw the exception to the delegate. And business delegate shouldn't re-throw the same exception back to the action class because if delegate do so, then my presentation tier component will becomes tightly coupled to the persistency tier and its technology.

**So what does the bussiness delegate will do? Why we need business delegate?**

Business delegate should throw the exception to the other class by ensuring that it shouldn't propagate to other part of the application, by translating technology specific exception to application specific exception, and throw the application specific exception to the action class.

**What is application specific exception? what is the purpose of application specific exception**?

Application specific exception are user defined exception or define by the programmer which are technology neutral.

**How do you create application specific exception? What is the granularity of it? How many application specific exception are there in your application?**

Design the application specific exception with different levels of hierarchy, for all the exception dealing with in my application we define, there is one exception class class which is acting as a base class for my whole application, and from there several exception classes for one-one module,(when the exception is nothing matching any of the defined exception then, we have a generic exception acting aa module level exception) and based on the business process flow there could be lots of exception

**Magic method in serialization?**

This method not realted to any package or any class, this is a contractual method in serialization api.

defaultWriteObject()

defaultReadObject()

writeObject()

void readObject()

Object writeReplace()

Object readResolve()

==================================================================================

**Hey boss it’s not that which one we need to use depends on various factors which we need to choose in our project**.

If the no.of elements within the xml is more it is not recommended to use DOM bcz DOM is memory intensive parser if we use DOM then we may encounter with out of memory exception.

If the no.of elements within the xml is less, static and fixed then we can use DOM but is always not recommended. But it is easy to parse/read the contents of an xml if size is small.

If we can impose size restriction on the xml while parsing then if we want we can use DOM even.

If we want to read the contents of xml only sequentially then we can use SAX. If are parsing the contents of an xml sometimes sequentially randomly then we need to DOM instead of SAX.

If we want to read only without modifying the contents of an xml then we need to go for SAX. If Sometimes we wanted to read and sometimes we wanted to modify then the contents of an xml then we need to use DOM so it is completely depends on then various factors our project requirement.

JPA is just a specification which needs concrete implementation. The default implementation provided by oracle is "Eclipselink" now. Toplink is donated by Oracle to Eclipse foundation to merge with eclipselink.

Using Eclipselink, one can be sure that the code is portable to any implementation if need arises. Hibernate is also a full JPA implementation + MORE. Hibernate is super set of JPA with some extra Hibernate specific functionality. So application developed in Hibernate may not be compatible when switched to other implementation. Still hibernate is choice of majority of developers as JPA implementation and widely used.

Another JPA implementation is OpenJPA, which is an extension of Kodo implementation.

[The key difference between Anonymous class and Lambda expression is the usage of 'this' keyword. In the anonymous classes, ‘this’ keyword resolves to anonymous class itself, whereas for lambda expression ‘this’ keyword resolves to enclosing class where lambda expression is written.

Another difference between lambda expression and anonymous class is in the way these two are compiled. Java compiler compiles lambda expressions and convert them into private method of the class. It uses invokedynamic instruction that was added in Java 7 to bind this method dynamically.

**Transition State:**

when an object is newly created with new operator then, we can say that the object at the transaition state, means this object is not attach with any session scope, if any changes are done then those changes will not get reflected in the database automatically then that object is in transition state.

**Persistence State:**

When an object is attached to the session object, and any change in the object will be reflect to the database directly thne those object is said to in the persistence state.

We can move the object in the persistence state in two ways:-

1) By directly loading the data from the database using get or load() methods.

2) By calling save(), persist(), saveOrUpdate() methods.

**Detached State:**

When there is the the record in the database and there is no corresponding data in the session, then it is in the detached state.

close(), evict(), clear()

**Race Condition**

Two or more thread executing in a system with an illusion of concurrency and accessing shared data, may try to change the shared data at the same time. Since the thread scheduling can swap between threads at any time, we won't actually know he order in which the thread will access tha data, which leads to data inconsistency issue.

**CopyOnWriteArrayList** creates a Cloned copy of underlying ArrayList, for every update operation at certain point both will synchronized automatically which is takes care by JVM. Therefore there is no effect for threads which are performing read operation. Therefore thread-safety is not there in ArrayList whereas CopyOnWriteArrayList is thread-safe.

While Iterating ArrayList object by one thread if other thread try to do modification then we will get Runt-time exception saying ConcurrentModificationException. Where as We won’t get any Exception in the case of CopyOnWriteArrayList.

ArrayList is introduced in JDK 1.2 whereas CopyOnWriteArrayList is introduced by SUN people in JDK 1.5.

Iterator of ArrayList can perform remove operation while iteration. But Iterator of CopyOnWriteArrayList cant perform remove operation while iteration, otherwise it will throw run-time exception UnsupportedOperationException.

**CountDownLatch**

CountDownLatch is used to make sure that a task waits for other threads before it starts. To understand its application, let us consider a server where the main task can only start when all the required services have started.

**Working of CountDownLatch**:

When we create an object of CountDownLatch, we specify the number if threads it should wait for, all such thread are required to do count down by calling CountDownLatch.countDown() once they are completed or ready to the job. As soon as count reaches zero, the waiting task starts running.

**Facts about CountDownLatch:**

Creating an object of CountDownLatch by passing an int to its constructor (the count), is actually number of invited parties (threads) for an event.

The thread, which is dependent on other threads to start processing, waits on until every other thread has called count down. All threads, which are waiting on await() proceed together once count down reaches to zero.

countDown() method decrements the count and await() method blocks until count == 0

**\*CyclicBarrier \***

CyclicBarrier is used to make the thread wait for each other. It is used when the diffrent threads process a part of the computation and when all threads have completed the execution, the result needs to be combined in the parent thread. In the other words, a CyclicBarrier is used when multiple thread carry out different sub tasks and the output of these sub tasks need to be combined to form the final output. After the completing its execution, threads call await() method and wait for the other threads to reach the barrier. Once all the threads have reached, the barriers then give the way for threads to proceed.

CyclicBarrier newBarrier = new CyclicBarrier(numberOfThreads);

**The need for Callable**

There are two ways of creating threads – one by extending the Thread class and other by creating a thread with a Runnable. However, one feature lacking in Runnable is that we cannot make a thread return result when it terminates, i.e. when run() completes. For supporting this feature, the Callable interface is present in Java.

**Callable vs Runnable**

For implementing Runnable, the run() method needs to be implemented which does not return anything, while for a Callable, the call() method needs to be implemented which returns a result on completion. Note that a thread can’t be created with a Callable, it can only be created with a Runnable.

Another difference is that the call() method can throw an exception whereas run() cannot.

public Object call() throws Exception;

What is Thread Pool in Java?

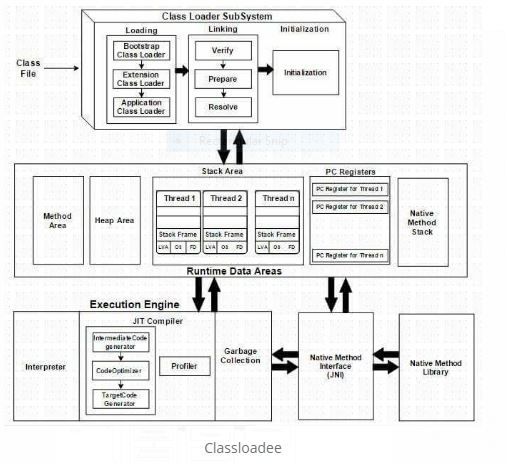
A thread pool reuses previously created threads to execute current tasks and offers a solution to the problem of thread cycle overhead and resource thrashing. Since the thread is already existing when the request arrives, the delay introduced by thread creation is eliminated, making the application more responsive.

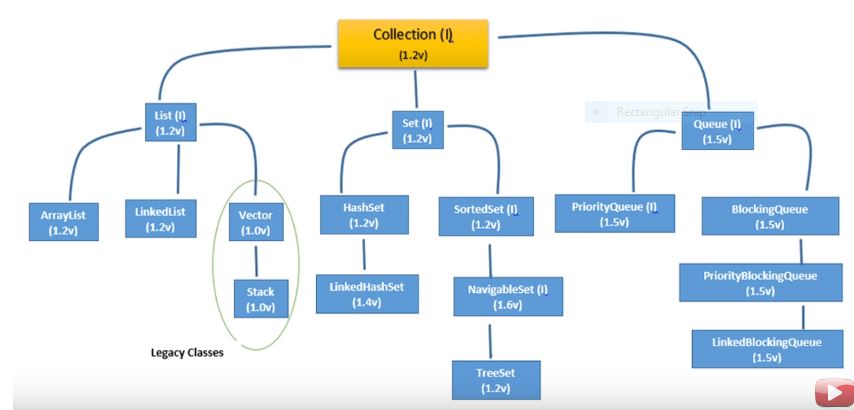
Java provides the Executor framework which is centered around the Executor interface, its sub-interface –ExecutorService and the class-ThreadPoolExecutor, which implements both of these interfaces. By using the executor, one only has to implement the Runnable objects and send them to the executor to execute.

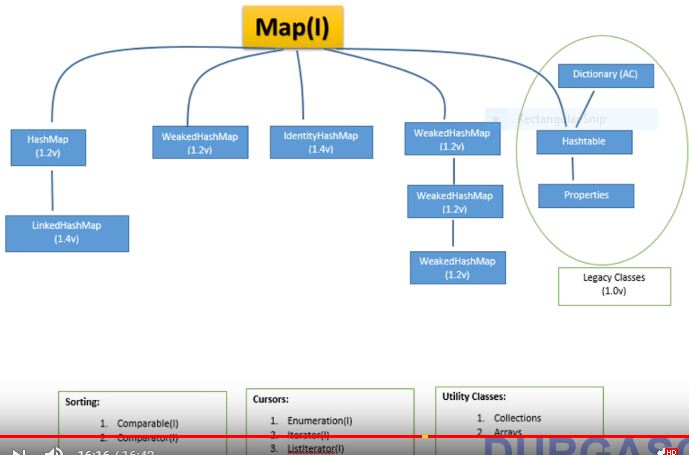
They allow you to take advantage of threading, but focus on the tasks that you want the thread to perform, instead of thread mechanics.

To use thread pools, we first create a object of ExecutorService and pass a set of tasks to it. ThreadPoolExecutor class allows to set the core and maximum pool size.The runnables that are run by a particular thread are executed sequentially.

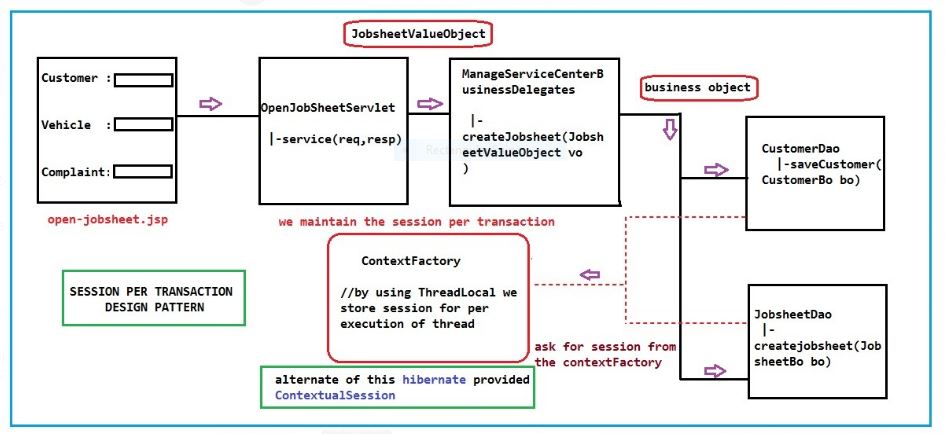
===================================================================================

****

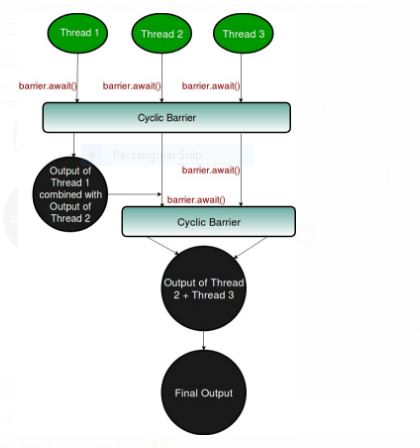
****

****

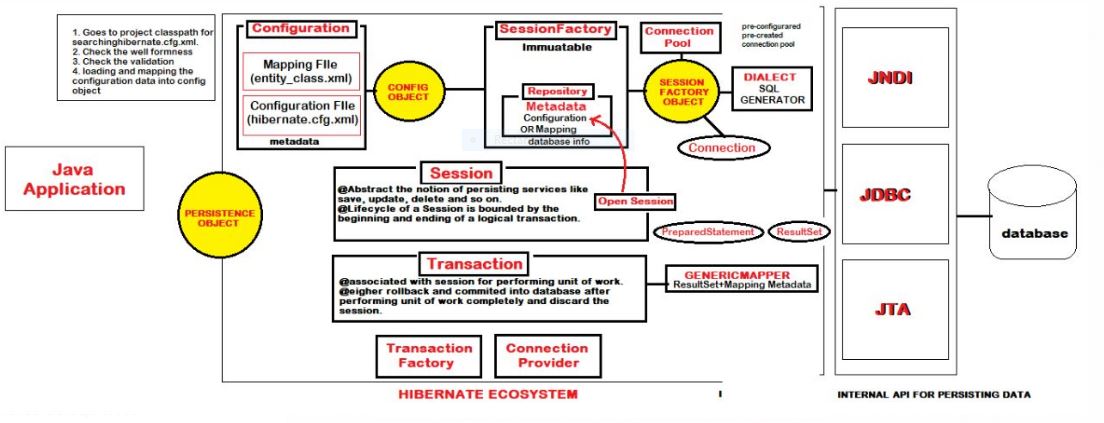
**Contextual session**

****

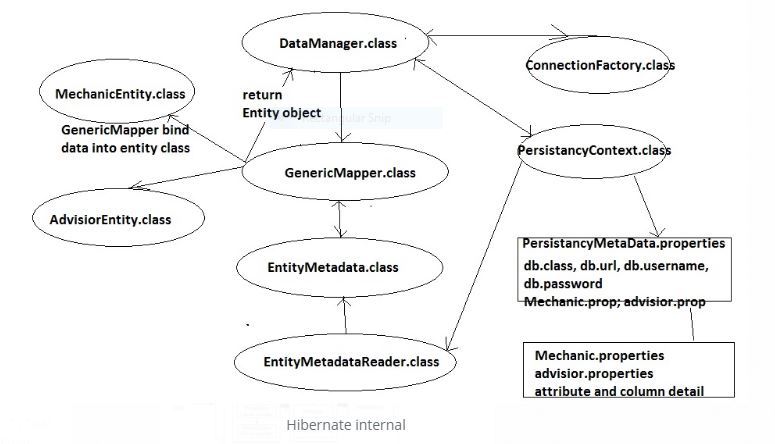
**Cyclic barrier**

****

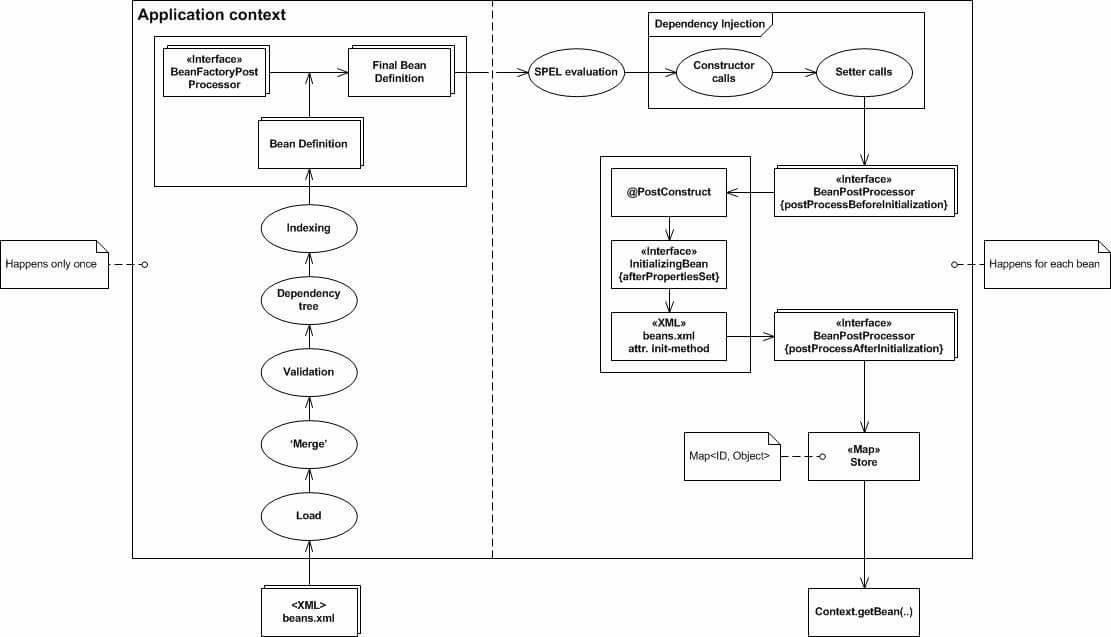
**Hibernate eco system**

****

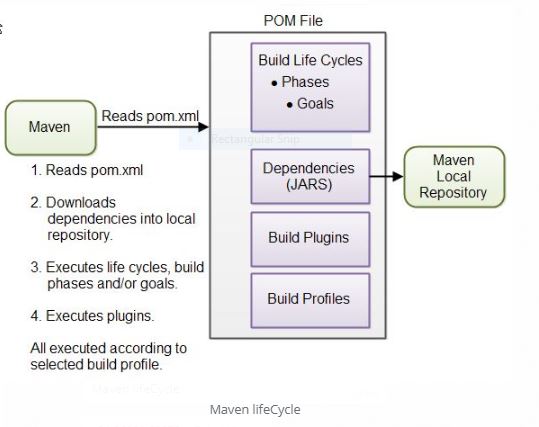
**Hibernate internal**

****

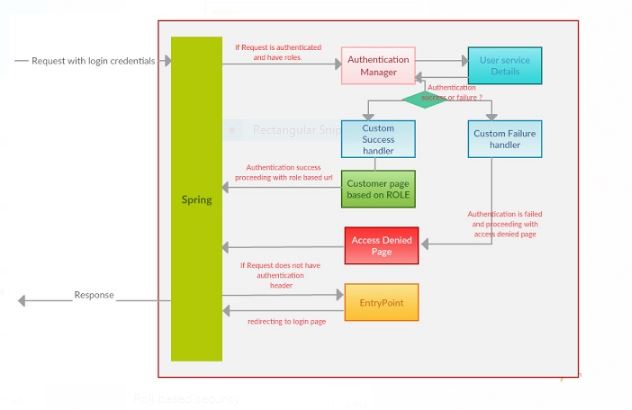
**Ipc Container**

****

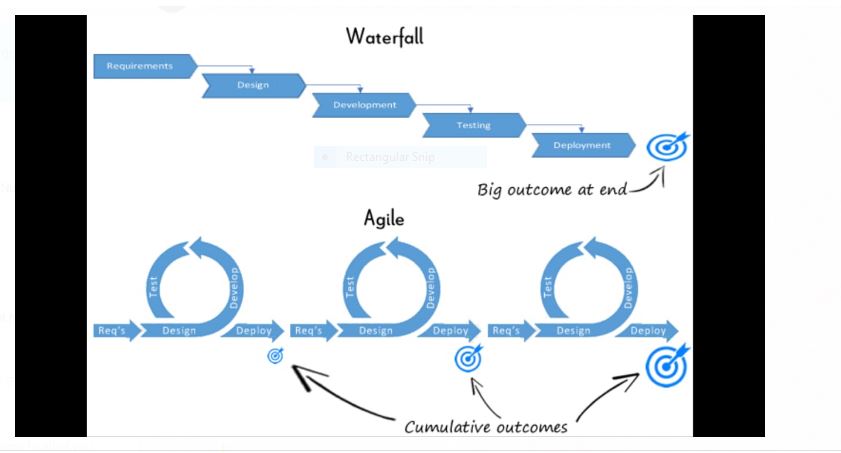
**Maven life cycle**

****

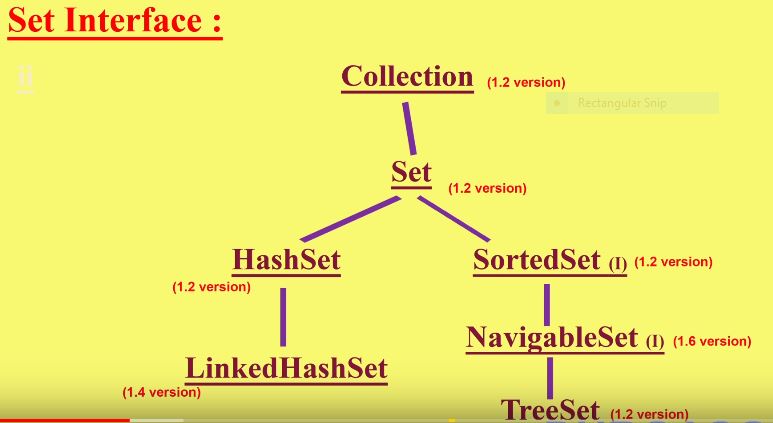
**Role based security**

****

**Waterfall vs agile**

****

**Set Interface**

****

**================================================================================**

**Key features in JDK 10 include:**

Local variable type inference, to enhance the Java language to extend type inference to declarations of local variables with initializers.

Parallel full garbage collection for the G1 garbage collector, to improve worst-case latencies.

Application class-data sharing to optimize startup time and footprint. The existing Class-Data Sharing feature is extended so application classes can be placed in the shared archive.

An experimental just-in-time compiler, Graal, can be used on the Linux/x64 platform.

Docker awareness. When running on Linux systems, the Java Virtual Machine (JVM) will know if it is running in a Docker container. Container-specific information—the number of CPUs and total memory allocated to the container—will be extracted by the JVM instead of it querying the operating system. (The number of CPUs available to the Java process is calculated from specified sets, shares or quotas of processors.)

Three new JVM options, to give Docker container users greater control over system memory.

A bug fix to correct the attach mechanism when trying to attach from a host process to a Java process that is in a Docker container.

Shorter startup times for the jShell REPL tool, particularly when a start file with many snippets is in use.

New APIs to better enable creation of unmodifiable collections. The

copyOf,Set.copyOf , and

Map.copyOf methods create new collection instances from existing instances. New methods toUnmodifiableList ,

toUnmodifiableSet , and

toUnmodifiableMap were added to the

Collectors class in the Stream package, allowing the elements of a Stream to be collected into an unmodifiable collection.

A local-variable type inference, to enhance the language to extend type inference to local variables. The intent is to reduce the “ceremony” associated with coding while maintaining a commitment to static type safety.

A clean garbage collector interface to improve source-code isolation of different garbage collectors. The goals for this effort include better modularity for internal garbage collection code in the HotSpot virtual machine and making it easier to add a new garbage collector to HotSpot.

Parallel full garbage collection for the G1 garbage collector. The intent is to improve worst-case latencies by implementing parallelism.

Enabling HotSpot to allocate the object heap on an alternative memory device, such as an NVDIMM memory module, specified by the user. This feature envisions that future systems may have heterogeneous memory architectures.

Enabling the Grall Java-based just-in-time compiler to be used in an experimental fashion on the Linux/x64 platform.

Consolidation of the repositories of the JDK forest into a single repository, to streamline development. The code base until now has been broken into multiple repos, which can cause problems with source-code management.

Application class-data sharing, to reduce the footprint by sharing common class metadata across processes. Startup time is improved as well.

Thread-local handshakes, for executing a callback on threads without performing a global VM safepoint. Individual threads could be stopped instead of either all threads or no threads.

Provision of a default set of root certificate authority certificates in the JDK. The goal is to open-source root certificates in Oracle’s Java SE Root CA program to make OpenJDK builds more enticing to developers.