**Core Java Interview Questions :**

1. **What are the principle concepts of OOPS?**

There are four principle concepts upon which object oriented design and programming rest. They are:

* Abstraction
* Polymorphism
* Inheritance
* Encapsulation

1. **What is Abstraction?**

Abstraction refers to the act of representing essential features without including the background details or explanations. It focus an outside view of an object. It solves the problems in the design side. Abstraction is achieved by abstract class and interface.

1. **What is Encapsulation?**

Encapsulation is a technique used for hiding the properties and behaviors of an object and allowing outside access only as appropriate. It prevents other objects from directly altering or accessing the properties or methods of the encapsulated object. The encapsulation can be achieved by defining the properties as private and functions as public.

1. **What is the difference between abstraction and encapsulation?**

* **Abstraction** focuses on the outside view of an object (i.e. the interface) **Encapsulation** (information hiding) prevents clients from seeing it’s inside view, where the behavior of the abstraction is implemented.
* **Abstraction** solves the problem in the design side while **Encapsulation** is the Implementation.
* **Encapsulation** is the deliverables of Abstraction. Encapsulation barely talks about grouping up your abstraction to suit the developer needs.

**5) What is Inheritance?**

* Inheritance is the process by which objects of one class acquire the properties of objects of another class.
* A class that is inherited is called a superclass.
* The class that does the inheriting is called a subclass.
* Inheritance is done by using the keyword extends.
* The two most common reasons to use inheritance are:
  + To promote code reuse
  + To use polymorphism

6) What is polymorphism ?

Polymorphism means which can take many forms. It is a concept force to use common interface instead of concrete implementation while writing the code.

Polymorphism is achieved by method overloading and method overriding.

6) What are the memory areas available in JVM ?

1) Class (Method) area

2) Heap

3) Stack

4) Program Counter Register

5) Native Method Library

7) What is Classloader ?

It is the subsystem of JVM that is used to load class

8) What is an Arrays?

-it is an class and it is an data structure used to store the similar type of data elements. It has fixed size cannot grow in size at runtime.

8) How to sort an elements in an array?

Int a[]={1,7,6,8}

Arrays.sort(a)

System.out.println(Arrays.toString(a));

9) How to search an element from an array?

- using the Arrays.binarySearch(Object[] a, Object key)

Static int binarySearch(Object[] a, Object key)

It searches specified value from an array using the binary search algorithm and returns an index position.

10) How to copy an array to an another array ?

- We can use the arrayCopy method from the System class.

System.arrayCopy(copyFrom,startingIndexPos,copyTo,startingPos,endingPos);

11) How to compare two arrays in java?

Public static Boolean equals(object[] a, Object b[])

It compares the given two arrays with size , value and datatype. It returns true if two arrays are equal in size and the pair of elements are same.

12) How autoboxing and unboxing done internally?

Autoboxing – converts primitive to Object. (Integer.valueOf(int value))

Unboxing – converts Object to primitive. (Integer.intValue(object a))

13) How strictfp used in java?

- strictfp used to provide control over the floating point arithmetic operation.

- floating point arithmetic operation differs in platform to platform on its precision value,

- to ensure the same value in all platforms , the strictfp keyword is used.

- it can be applied to classes, intefaces and methods but not to variables , constructors and abstract methods.

14) What is Javadoc toll?

- it is an tool used to create document API about classes ,methods and constructors.

- document comment /\*\*….\*/ should be used in the java program and run it by Javadoc m.java

- html files are generated and refer the index.html files.

**15) what is collection framework? Where it is used?**

16) What is map and how to retrieve an value from map?

- Map is an collection of key value pairs.

- To retrieve an value from Map , we have to convert the map into set and then take the iterator from the set to iterate the elements.

- to declare an map,

Map<Integer,String> map=new HashMap<Integer,String>();

Add an key value into map,

Map.put(1,”aaaa”);

Map.put(2,”bbbb”);

Retrieve an element from map we have to use the entrySet() method which returns the set

Set<map.Entry<Integer,String>> set=Map.entrySet();

Take the iterator from the set,

Iterator<Map.Entry<Integer,String>> it=set.iterator();

While(it.hasNext()){

Map.Entry e=it.next();

System.out.println(e.getKey());

System.out.println(e.getValue());

}

17) How exceptions are handled in spring-hibernate application?

18) How logging is handled in yr application?

19) explain entrySet?

20) Explain DAO layer?

DAO is an Data Access Object used to interact with the database via model object. It provides the abstract interface to data base or other persistence mechanism . The persistence mechanism may be the database, file system or web service.

Instead of the domain logic directly communicates with the persistence mechanism , it talks with the DAO layer then the DAO communicates with the data base or persistence service or system.

Instead of exposing the persistence logic and data base details used in the application to other layers , it simply provides an interface with applicable methods to be used by other layers to achieve the persistence mechanism.

Advantage:

When there is change in persistence mechanism , we can change the logic only in DAO layer , not in service or view layer.

Components:

There are three components involved in the DAO layer.

1)DAO Interface : it defines the standard operations to be performed on model object.

2) DAO Concrete class : it provides the implementation on the operations defined in the interface

3) Model object or Value Object : it is an simple pojo with getters and setters method

Handling the Exceptions:

Most the of the time , DAO layer it throws an SQLException.

We may create own independent exceptions for each layer. Throw an ApplicationException should wrap the SQLException.getMessgae() from the DAO layer to Service layer to handle the exception.

DAO layer throws an exception to service layer and it can rethrow an exception to the UI layer.

DAO layer may throw an SQL Exception when it interact with database , and throw an Remote exception when it interact with Webservice and it throw an IOException when it interact with the file. So create some generic DAOException to wrap all three type of exceptions. Later , if the DAO throws some new exceptions , it can be wrapped up with DAO Exception and the service layer still unaffected.

DAO layer throw an more generic application specific DAOException to service layer. Once the exception is rethrown from the service layer to UI layer, it converted to a user friendly message and log them in console or file them displayed to the user.

UI layer catch this exception in UI Specific classes.

21) how transactions handled in Hibernate ? web application?

Transaction is an unit of work . it must be rolled back , if somethings goes wrong with in the unit.

So we can say service layer is the place doing the unit of work and is the place to handle the transactions.

Transactions are useful to maintain the data integrity and consistency.

Transactions are handled using the Transaction interface in hibernate.

EX:

Configuration cfg=new Configuration().configure(“hibernate.cfg.xml”);

Sessionfactory sessionFactory=cfg.buildSessionFactory();

Session sesfac=sessionFactory.openSession();

Transaction tx=session.beginTransaction();

Tx.commit() ; Or

Tx.rollback();

22) What is generics in java? Where its used?

- Generics are introduced in java 1.5. Generic classes are class which can accept any type of objects.

- Generics are used to deal with type safe objects.

- before generics, collection can accept any type of objects but by using generics collection can be restricted with specific type of objects.

Advantage:

* + - Type casting is not required becoz while declaring the collection itself we are restricting the type of object stored
    - It can provide the type safety that is it accepts specific type pf objects
    - Compile time checking , so it reduces error during runtime , it is a good programming practice

Ex: List <Integer> list=new ArrayList<Integer>

Iterator<Integer> it=list.iterator();

While(it.hasNext()){

System.out.println(it.next());

}

23) how to use advanced for loop?

It introduced in java1.5.

Advanced for loop or enhanced for loop or for each loop is used to iterate over the array or collection elements.

Syn:

For(datatype var:array(or ) Collection)

Ex:

For(int i:a[])

Advantage :

* 1. Makes code more readable
  2. It eliminates the programming errors

24) What are the annotations used in java?

Annotations are the tag which gives extra information(metadata) about the class , constructor and methods to the JVM. It can be used instead of the marker interface.

Built in annotations are @Deprecated @SuppressWarnings @Override

25) How to handle the exceptions in Hibernate ?

Catch the **exception** where you can do something with the **exception**. If you catch it in DAO layer, then the information specific to dao layer needs to be extracted. e.g. if it is can not insert null value, then log the field details where it is failing.

Then rethrow the exception to the service layer, exception handler is set up on the view or controller to handle the exception

All the exceptions in hibernate are runtime exceptions. So it is better to caught and handle the exceptions in dao layer itself.

Exception.getMessage() method used to get the exception message from stack trace

26) how triggers works in database?

Data base trigger is an stored program associated with the table or view. Oracle executes trigger automatically , whenever DML operations applied to the table or view.

Triggers are set of Pl/SQL statements associated with the table and gets executed automatically whenever insert , update and delete operation happens on that table. Triggers are executed automatically by oracle engine.

Triggers are stored programs , gets invoked automatically when some event occurs.

Triggers can be created for the table , view, schema or database.

Advantages :

* + 1. It generates derived column automatically
    2. Enforces data integrity
    3. Synchronization of data on tables

Types:

* + 1. Before – it gets executed before DML(insert , update and Delete) operation is performed
    2. After – it gets executed after the DML operation is performed
    3. Drop – it is used to drop the trigger by using the DROP Trigger trigger\_name statement
    4. Disable – ALTER Trigger statement can be used to disable the trigger. ALTER TRIGGGER trig\_name DISABLE
    5. Enable – ALTER Trigger statement can be used to enable the trigger. ALTER TRIGGGER trig\_name ENABLE.

27) What is the replacement for try with finally?

In JDK1.7 , try-with resource statement replaces try ..finally.

Try with resources is an try statement that declares one or more resources. Resource is an object must be closed after program used it.

Before java1.7 we use finally to close the resources, but now if we use try with resources statement it will be closed automatically. Resources can be of type Closable or AutoClosable.

Resources are BufferedReader or FileReader.

Ex: **try (BufferedReader br =**

**new BufferedReader(new FileReader(path)))** {

return br.readLine();

}

28) multi—catch statement?

Java1.7 supports , the catch clause catches multiple exceptions.

Ex: try{

}catch(SQLException | IOException Ex){

}

29) Multithreading in java?

Threads shares the same common memory area. Since it doesn’t allocate separate memory area, it saves memory.

30) How is the security achieved in the Web service?

- web service security is achieved via SSL Certificate.

- SSL is an Secure Socket Layer which is supported by HTTP .

- SSL Digital Certificate is a small data files that digitally bind the cryptographic key to an organizatiosn details. When installed on the web server , it enables the pad lock and https protocol which allows secure connection from the web server to browser.

- SSL is used to secure the login information, credit card information and data transformation

- SSL Certificates bind together,

1) A Domain name , server name or host name

2) An organizations identity and location

- An organization needs to install the SSL certificate on its web server to initiate the secure connections with the browser.

- Once secure connection is established , all web traffic between web server and web browser will be secure.

31) How does an SSL works?

SSL uses public key cryptography concept. Which uses two keys of long strings which are randomly generated numbers. Public key is known to the webserver and is available to the public domain.

It can be used to encrypt any message.

SSL needs to be issued from the trusted certificate authority.

Browser : Establish the secured connection with the web server(https) and request the webserver to identify itself

Web server: web server send its copy of its digital certificate with servers public key

Browser: browser validates the web servers certificate with list of all trusted certificates . if browser trust the certificate it creates, encrypt and send the session key encrypted with public key

Web server: web server decrypt the session key with the private key and send the acknowledgement encrypted with the session key

Browser and the web server : establishes the encrypted connection with session key. All datas are encrypted.

32) What is the default scope of the spring bean?

Singletone

Scopes:

Singleton: only one bean instances per spring IOC container

Prototype: multiple bean instances created. For every spring request In spring IOC container

Request : bean instances created for every http request. Applicable only in spring aware web application context

Session: bean instances created for every http session. Applicable only in spring aware web application context

Global Session : bean instances created for global http session. Applicable only in spring aware web application context

33) How to test the DAO Layer ? how to conclude that the test results?

By using the Assert Statement. Assert statement is statement which is used to test the assumptions about your program. It is mainly used for testing purpose.

Always the assertion should be true. If it fails , it will throw an assertion error.

Advantage:

It is an effective way of detect and correcting programming errors.

34) How to verify the equality of the two objects ?

By overriding the hashcode() and equals() method from objects class. HashMap internally use hashcode to search an value from the map .

35) How to search the User defined objects used as key in HashMap?

We can achieve this by implementing the equals() and hashcode() method in the user defined object.

Ex:

class Price{

    private String item;

    private int price;

    public Price(String itm, int pr){

        this.item = itm;

        this.price = pr;

    }

    public int hashCode(){

        System.out.println("In hashcode");

        int hashcode = 0;

        hashcode = price\*20;

        hashcode += item.hashCode();

        return hashcode;

    }

    public boolean equals(Object obj){

        System.out.println("In equals");

        if (obj instanceof Price) {

            Price pp = (Price) obj;

            return (pp.item.equals(this.item) && pp.price == this.price);

        } else {

            return false;

        }

    }

    public String getItem() {

        return item;

    }

    public void setItem(String item) {

        this.item = item;

    }

    public int getPrice() {

        return price;

    }

    public void setPrice(int price) {

        this.price = price;

    }

    public String toString(){

        return "item: "+item+"  price: "+price;

    }

}

When we use this object as key in HashMap and do search

package com.java2novice.hashmap;

import java.util.HashMap;

import java.util.Set;

public class MyObjectKeySearch {

public static void main(String a[]){

        HashMap<Price, String> hm = new HashMap<Price, String>();

        hm.put(new Price("Banana", 20), "Banana");

        hm.put(new Price("Apple", 40), "Apple");

        hm.put(new Price("Orange", 30), "Orange");

        printMap(hm);

        Price key = new Price("Banana", 20);

        System.out.println("Does key available? "+hm.containsKey(key));

    }

    public static void printMap(HashMap<Price, String> map){

        Set<Price> keys = map.keySet();

        for(Price p:keys){

            System.out.println(p+"==>"+map.get(p));

        }

    }

}

36) Factory pattern ?

Define an abstract class or interface for creating the object and let the sub class decide which class to instantiate.

37) How to create singleton in java ?

1) Using the Singleton design pattern and using the spring framework.

2) we can use the single element enum class used to achieve the singleton

Public enum enumEx{

INSTANCE;

}

Internally it has private constructor. And no need to write the public getInstance method to get the instance , instead we can use like this as,

enumEx.INSTANCE; = it will return the instance of the enum class.

38) Different approach to sort the elements in an ArrayList?

1) String objects and wrapper class object are sorted using the Collections.sort(list) method. It will sort the elements in the natural ordering.

Since these objects are implements Comparable interface by default. So if ArrayList objects needs to be compared , it should implement the comparable interface

2) User defined objects are sorted using the Comparable or Comparator interface

Java.lang.Comparable : it has one method to be implemented compareTo(Object O1) and used to compare object based on the only one field.

Java.util.Comparator: it has two methods to be implemented Compare(Object O1,Object O2) and equals and used to compare the objects using multiple fields.

39) What is the datastructure that arraylist uses to store an element ?

DynamicArray

40) how to create bean in spring ?

41) How to find the occurrence of the each character in string?

String s=”welcome”;

HashMap<Character,Integer> map=new HashMap();

For(char c:s.toCharArray()){

If(map.containsKey(ch)){

Int val= Map.get(ch);

Map.put(ch,val+1);

}else{

Map.put(ch,1);

}

system.out.println(map);

}

43) how to fetch the two different columns from two different tables using HQL?

Select p.name ,a.street from Person p, Address a where p.id=s.id))

44) What are the factors needs to be considered while reengineering the j2ee applications.

45) Roles and responsibilities of java architect?

46) What is scalability and interoperability of the j2ee application?

48)What are the java1.8 features ?

1) default and static methods in Interface:

In java 1.8 , we can have the default and static methods inside the interface. Default method is defined using the default keyword and static method is defined using static keyword . advantage of having the default method is to implement the common functionality in interface. Default methods can be overridden. Static method cannot be overridden. After having the default method in interface the only difference between abstract class and the interface is we can create the constructor in abstract class.

2) Functional Interface:

Interface that have only one abstract method is known as functional interface. It can have any number of default and static methods but should have only one abstract method. Functional interface can have methods of object class (toString(),hashCode(),equals()) . Functional interface can extend another interface only it doesn’t have any abstract methods. Java 8 has predefined functional interfaces available in java.util.functional. @FunctionalInterface is used to define the functional interface.

1. EX: @FunctionalInterface  //It is optional
2. **interface** Drawable{
3. **public** **void** draw();
4. }
6. **public** **class** LambdaExpressionExample {
7. **public** **static** **void** main(String[] args) {
8. **int** width=10;
10. //with lambda
11. Drawable d2=()->{
12. System.out.println("Drawing "+width);
13. };
14. d2.draw();
15. }
16. }

3) Lambda Expressions :

It provides the implementation for the functional interface.It helps to iterate , filter and extract data from collection. Advantage is less coding and providing the implementation for the functional interface. It is an replacement of anonymous inner class.

Syntax:

argument-list -> body

it consists of three components.

Argument-list :- which are kind of variable names

Arrow-token :- which gives association between argument-list and body of the expression

Body :- which contains the statements and expressions for the lambda expression.

Lambdaexpressions can have zero or more arguments in the argument list. Lambda expressions used in the forEach loop of the collections to iterate the elements.

Run() method from Runnable interface can be implemented by using the lambda expression.

Collections are filtered using the filter method in Stream class(java.util.stream.Stream)

// create the arraylist and populate the user defined objects.

Filter like this

Stream<Product> stream=list.stream().filter(p->p.price>2000)

Iterate like this

Stream.forEach(product->sysout(product.name));

Lambda expressions also used to implement the compareTo() method of Comparable interface.

Ex:

1. Collections.sort(list,(p1,p2)->{
2. **return** p1.name.compareTo(p2.name);
3. });
4. **for**(Product p:list){
5. System.out.println(p.id+" "+p.name+" "+p.price);
6. }
7. 1. Optional Class:

It is used to deal with nullpointerexception in java application. It is a public final class. It is used to check the presence of the value for Particular variable.

It avoids the abnormal termination of the program.

It has the following methods to deal with the NullPointerException.

Optional opt=Optional.ofNullable(variable) – returns an optional instance for the specified with the specified variable value. If the variable has a null value , it returns an empty Optional instance

Opt.isPresent() – returns true if the value is present in the optional instance else returns false.

Opt.get() – returns the value of the optional.

* 1. Stream:

Jdk1.8 introduces a new package called stream , which has classes, interfaces and enum to perform the functional operations on the element. It is used to filter,collect, print and convert one datastrucure to another etc. stream provides fast execution. Steram is lazy and evaluates code only when its required. It doesn’t store elements , it just supports the functional operations on the source.

List.stream().Filter(p->p.price>3000) //filter data from list

.map(p->p.price) //fetch data from filter result

.collect(Collectors.toList()); // collect the data into list

Stream is used to iterate the elements at any number of times. The iteration limit is specified by limit() method.

Ex:

Stream.iterate(1,element->element+1)

.limit(3)

.forEach(System.out::println);

Stream collect method is used convert the data structure

List to Set:

.collect(Collectors.toSet())

List to Map:

.collect(Collectors.toMap(p->p.id,p->p.name)

* 1. Method References:

It is a feature related to Lambda expressions. Using the method reference , we can refer the static method of class , instance method of instance and constructor

* 1. forEach() mthod in iterable interface:

it is used to loop through the collection elements . it takes either lamda expressions or method reference as an argument.

* 1. Concurrency improvement:

Running several parts of the program in parallel is known as concurrency. Thread is used to achieve the concurrency in java. In multithreaded application , multiple hreads try to access the shared resources , then concurrency issues happen.

49) Difference between save and persist ?

Persist returns void but save returns the serializable object. Persist is suitable for long running transactions but the save is not suitable for the long running transactions.

Insert happens immediately in save so return an identifier. But persist() insert the recode at flush time.

|  |  |
| --- | --- |
| Save | Persist |
| It inserts the record immediately | The insert may happens at flus time |
| It returns identifier as serializable object | It returns void |
| It is not suitable for long running conversations | It is suitable for long running conversations with extended session / PersistenceContext |

What is long running conversations in hibernate ?

This is session per conversation pattern. The session is disconnected from transaction once the transaction is committed. It can be reconnected for the next client request. In this case session needsd to be flushed at manually not automatically.

50) Difference between merge and update?

Both are used to update the object into the database. Both are move the detached object to persistent state.

Update() :- if we are sure that the record is not already in persistent state(associated with session) with same identifier , then use update. It just update the record. We cannot use update if the same object already in session with same identifier. If we try to update the object if it is already in session(persistence state) , it throws nonuniqueobject exception.

Merge() :- if we want to save our modifications at anytime without considering about the state of the session , then use merge. We can use merge to merge the changes even is the object is already in the session. The latest change is stored in the database.

**51) how do you handle the production issues?**

52) Explain JDBC template.

It is an mechanism to connect the database and execute the simple sql queries. It reduce the lots of code for making the database connection and creating the statement . It replaces the disadvantages from JDBC API. Whne we use jdbc API , we have to write code for creating connections , handling transactions , handling the exceptions. JDBC template takes care of all these and providing methods to execute the simple query.

53) What is autowiring in spring?

It is useful to inject the object dependency implicitly via setter injection or constructor injection. Autowiring cannot inject the primitives and strings

Advantage :

Less code : no need to write the code for injecting the dependent objects.

Modes:

Autowiring can be done in different modes.

byName – based on the preperty name

byType – based on the property type

constructor – based on constructor arguments

by default no autowiring enabled.

54) Why multiple inheritance is not supported in java ? and What is diamond problem?

When a class inherits multiple classes is known as multiple inheritance. One child cannot have two parent. Multiple inheritance creates an ambiguity. To reduce the complexity and simplify the language , java doesn’t support multiple inheritance. If the class extends multiple classes then child class causes an ambiguity to decide which version of (from parent1 or parent2) method should be orverrided.

55) Explain JVM architecture?

JRE is the implementation of the JVM. JVM is platform dependent

56) How many containers available in spring?

Spring has two IOC containers. BeanFactory and ApplicationContext.

57) when the spring bean is created ?

If we use BeanFactory , beans are created when we invoke getBean() method .it supports lazy initialization.

If we use ApplicationContext , beans are created at the time of spring container starts.

58) Access specifier rule in Overriding?

When we override the method, access specifier in overridden method (in subclass) should not be more restrictive.

59) How to achieve encapsulation ?

Making the data member as private and use getters and setters method to get the data in it.

60) Explain:

String s =new String (“servion”); creates the string literal “servion” in string constant pool. New string object in heap . the reference variable ‘s’ in stack.

s=s+”Global” //the literal “Servion Global” will be created in the string pool and refered by the reference varable “s”.

system.out.println(s) // it prints “Servion Global” and the object created in heap in line 1 available for garbage collection.

61) Difference between NamedQuery and SQL query in hibernate ?

Named queries are reusable.

62) How to compare the two different versions of file in tortoise SVN?

63) How to compare the tortoiseSVN file with local file ?

64) How to revoke the versions in svn?

65) Explain SessionFacade and BusinessDelegate design patterns.

66) Can we extend the class which has private constructor?

No. We cannot extend the class which has private constructor.

67) what is the advantage of Annotations over the xml in java?

1) configuration and coding goes into single file . No need to refer two files . So I can code much faster

2) it gives good readablility

3) less coding – Annotations need less coding when compared to xml configuration

4) it supports static type checking : Static type checking is the process of verifying the type safety during compile time. Dynamic type checking is the process of verifying the type safety during runtime. Static type checking is helpful to analyses the code and find the errors at compile time.EX: Child class variable doesn’t hold the parent class object , it throws complie time error. It has to be corrected by casting the parent class object to type of child class name.

5) it is the standard way to define the metadata. Before annotations they were used marker interfaces and comments.

6) Annotations can be applied to classes , methods , interfaces and field

68) what are the differences between annotations and xml ?

|  |  |
| --- | --- |
| Annotations | XML |
| It gives good readability by referring the single file itself | Always we need to verify more than two files xml and POJO |
| If any changes made , it needs to be recompile | Recompilation is not required |
| It gives tight coupling of the code and configuration.So if any changes , can do it in signal file itself | It supports loose coupling. So if any changes it needs be done in more than two files |
| Maintenance is easy | Maintenance if little difficult |

69) Explain the design phase of software application.

Design is the order of the system that connects individual components. There are 2 phases in the design phase.

* 1. Architectural or High level design
  2. Detailed or Low level design

70) what is low level design document ?

- it a component level design process.

- it covers data structures , required software architecture and performance related algorithms.

- it referred by programmers

- it gives internal logical design of the program code

- it describes the class diagrams with methods and relations with the classes.

- it describes the modules , so that programmer directly refer this document and code.

- it makes the programer easy to develop when proper analysis is utilized

71) What is the High level design document ?

- it explains the architecture that are going to use to develop the application in form of architecture diagram. It describes the components to be developed for the system and their interfaces..

- it is an overview of structure is the system such as database architecture, application architecture(layers) , application flow (navigation) , security architecture and technology architecture.

- it s complete view of the entire system. It describes the suitable model for the coding

- it is represented to administrators to get the understanding on the system

- it is created based on the low level design

72) what is webservice ?

It is software or application component used for communication between two systems via some message format. It supports client server communication. It supports interoperable machine to machine communication. It is the method of communication between two applications over the web.Web services are part of service oriented architecture.

73) what are the types of web service and when to choose what?

Two types of web services available .

SOAP :- Simple Object Access Protocol .

RESTFULL :- Representational State Transfer .

There are two factors needs to be considered for choosing this web service.

1. Type of Client : if it is an light weight client (mobile phones) go for REST
2. Flexibility & standardization used by the company culture.

74) how the legacy systems are exposed as web service ?

We can use web service concept with SOA architecture to expose the legacy system as web service. Refer the following url : <https://www.ibm.com/developerworks/library/ws-legacy/> . we should need some interface(web service adapter) which connects to the legacy application and hide the complexity of the legacy application. One adapter may call one backend application. Sometime it may build complex web service request to access multiple backend system and form a web service response which is passed back to service requester. The web service adapter hides the complexity of the backend systems and requires only necessary parameters to set to execute the service.

Each backend system has its own transaction manager to handle the transaction. When we use the EJB session bean for the adapter implementation , EJB session bean transaction coordination capabilities helpfule to coordinate the transaction from different backend systems. When the complex web service request starts from adapter, it send a middleware message to start the transaction on the backend systems. These messages send to one or more backend systems. If any response from the backend systems end with an error then the EJB roll back all the transactions from all the backend systems. This ensures the complete execution of the web service request. This avoids the any inconsistent state on the data base on different backend systems.

75) How to persist the state of the object?

Using serialization, we can persist the state of the object. The object must implement the Serializable interface for that.

76) what is Business Delegate design pattern?

It is used to decouple the business tier and presentation tier. We have following entities involved in this pattern.

1. Business Service : It is an business service interface and needs to be implemented by concrete classes to provide the actual implementation logic
2. Lookup Service : it provides the business object access to business delegate object. It returns the business object based on the types passed.
3. Business Delegate : Single entry point for client to get access the business methods. It use lookup service to get the business object and call the business methods using the object returned by lookup service.
4. Client: Presentation tier code such as JSP , servlet or UI code

77) What is prototype pattern ?

It is used to creating duplicate objects . This is creational pattern. This is one of the best way to create an object.

**78) What is micro services?**

79) How to handle exceptions in Web service ?

Exception is an error condition or unexpected behavior occurred during the program execution. Web services exception needs to conveyed to consumer in the form of SOAP supported format(Platform – independent manner).

SOAP supports standard and platform independent way of conveying the exception to consumer within the SOAP message using the SOAP fault. SOAPBinding maps the exceptions to SOAP fault. The will be only one SOAP fault elements in the message. There are 2 elements in the SOAP fault , error and status.

There is an nother way of handling the exception in webs ervice. The code which cause an exception should be placed in the try catch. And the web service exception should be handles by using the Customer Exception class.

All the SOAP Exceptions are handles with SOAP-Fault.

80) What is Adapter Pattern ?

Adapter works as bridge between two incompatible interfaces. This is structural pattern. This combines the capability of two interfaces.

81) what are inheritance mapping in hibernate ?

There are 3 inheritance mapping available in hibernate

1. Table per hierarchy :- There will be only one table for the entire hierarchy. There will be an extra column (discriminator column) used to identify the class or record

Child classes are mapped with using the subclass element . Class and subclass have the discriminator-value property which stores the value to identify of the record type.

1. <**class** name="com.javatpoint.mypackage.Employee" table="emp121" discriminator-value="emp">
2. <id name="id">
3. <generator **class**="increment"></generator>
4. </id>
6. <discriminator column="type" type="string"></discriminator>
7. <property name="name"></property>
9. <subclass name="com.javatpoint.mypackage.Regular\_Employee" discriminator-value="reg\_emp">
10. <property name="salary"></property>
11. <property name="bonus"></property>
12. </subclass>
14. <subclass name="com.javatpoint.mypackage.Contract\_Employee" discriminator-value="con\_emp">
15. <property name="pay\_per\_hour"></property>
16. <property name="contract\_duration"></property>
17. </subclass>
19. </**class**>

1. Table per concrete class :-

Each class is stored in separate table and the mapping between parent and child class is done by union-subclass element. The child class tables stores the parent class table column also.

1. <**class** name="com.javatpoint.mypackage.Employee" table="emp122">
2. <id name="id">
3. <generator **class**="increment"></generator>
4. </id>
6. <property name="name"></property>
8. <union-subclass name="com.javatpoint.mypackage.Regular\_Employee" table="regemp122">
9. <property name="salary"></property>
10. <property name="bonus"></property>
11. </union-subclass>
13. <union-subclass name="com.javatpoint.mypackage.Contract\_Employee" table="contemp122">
14. <property name="pay\_per\_hour"></property>
15. <property name="contract\_duration"></property>
16. </union-subclass>
17. Table per sub class:

Each of the class has separate tables both parent and children’s. But they are linked with primary key and foreign key. The sub classes are linked with parent by using the joined-subclass element in hbm file.

Here the parent table primary id is stored as foreign key in child tables.

1. <**class** name="com.javatpoint.mypackage.Employee" table="emp123">
2. <id name="id">
3. <generator **class**="increment"></generator>
4. </id>
6. <property name="name"></property>
8. <joined-subclass name="com.javatpoint.mypackage.Regular\_Employee" table="regemp123">
9. <key column="eid"></key>
10. <property name="salary"></property>
11. <property name="bonus"></property>
12. </joined-subclass>
14. <joined-subclass name="com.javatpoint.mypackage.Contract\_Employee" table="contemp123">
15. <key column="eid"></key>
16. <property name="pay\_per\_hour"></property>
17. <property name="contract\_duration"></property>
18. </joined-subclass>

82) What are collection mappings in hibernate?

1) List

2) Set

3) SortedSet

4) Map

5) SortedMap

6) Bag

7) implement org.hibernate.usertype.UserCollectionType

List and Map are indexed collections and it needs extra <index> element in hbm file and it will create extra index column in the table . Bag and set are non indexed collections.

List : index acts as array index position .it use arraylist to store data

Map: index column acts as key.

List collection mapping for String:

<list name=”answers” table=”ans”>

<key id=”qid” />

<index column=”type”/>

<element column=”answer” type=”string” />

</list>

List collection mapping for entity references:

To have the entity references as collections , we should use any of the associations one-to-many, many-to-many

<list name=”answers” table="ans102">

<key id=”qid” />

<index column=”type”/>

<one-to-many class=”com.Answers” />

</list>

<key > :- defines the foreign key associated with parent table. By default it is nullable , to make non-nullable we should use not-null=”true”

<index> :- is used to identify the type

<element> :- it defines the element of the collection.

Hibernate pojo has List element it can also be mapped as bag in hbm.xml file.

Bag collection mapping for string:

1. <bag name="answers" table="ans101">
2. <key column="qid"></key>
3. <element column="answer" type="string"></element>
4. </bag>

Bag Collection mapping for entity references:

1. <bag name="answers" cascade="all">
2. <key column="qid"></key>
3. <one-to-many **class**="com.javatpoint.Answer"/>
4. </list>

Both List and Bag use the List mapping in the persistent class as like,

1. **public** **class** Question {
2. **private** **int** id;
3. **private** String qname;
4. **private List<Answer> answers;**
6. //getters and setters
8. }

Set mapping for String:

1. <set name="answers" table="ans102">
2. <key column="qid"></key>
3. <element column="answer" type="string"></element>
4. </set

Set collection mapping for entity references:

1. <set name="answers" cascade="all">
2. <key column="qid"></key>
3. <one-to-many **class**="com.javatpoint.Answer"/>
4. </set>

Map collection mapping for Strings:

1. <map name="answers" table="answer736" cascade="all">
2. <key column="questionid"></key>
3. <index column="answer" type="string"></index>
4. <element column="username" type="string"></element>
5. </map>

Map Collection mapping for entity references:

1. <map name="answers" table="answer738" cascade="all">
2. <key column="questionid"></key>
3. <index column="answer" type="string"></index>
4. <many-to-many **class**="com.javatpoint.User" column="userid"></many-to-many>
5. </map>

83) How to disable first level cache in hibernate ?

First level cache is associated with the session object. By default it is enabled , there is no way to disable it. But we have some methods to clear the cache or remove the object from cache.

Methods used to clear the first level cache:

1. Evict() : remove the particular object from session
2. Clear (): remove all the object from cache
3. Contains() : check the presence of the object and returns true if it is present

Object cached in one session will not be visible to other sessions. When the session is closed , all objects associated with the session will be lost.

84) how to handle the network threat in soap web services?

we can use firewall and filter to filter the values in SOAP Header and message content type from HTTP POST.

We can build the application level firewall to prevent this kind of network threat. This firewall validate the credentials against the source (LDAP)(Lightweight Directory Access Protocol). Once authenticated , the webservice , operation are validated . All kind of XML encryption and decryption done here and XML verification against schema are done here. Once all the checks are gone thru it reach the web service.

Firewall software are installed on our computer to monitor the incoming and outgoing packets and block that if something is coming from untrusted source.

This XML Application firewall is completely different from IP-level firewall.

85) What is index ? how to run it ?

Index is an datastructure that speed up the retrieval process.

86) difference between primary key and unique key ?

|  |  |
| --- | --- |
| Primary key | Unique Key |
| It is used to identify the record | It used to prevent the duplicate values in the column |
| There should be only one primary key in the table | There can be multiply unique key in the table |
| It cannot have null values | It can have null values |
| We cannot delete or change primary key value | We can change unikey vakue |
| It use clusterd index | It use non clusterd index |

87) difference between varchar and varchar2 ?

|  |  |
| --- | --- |
| Varchar | Varchar2 |
| It support distinction between NULL and empty string | It doesn’t support distinction between NULL and empty string |
|  |  |

If we want to support both NULL and empty string is same , then we can use varchar2.

88) difference between instr and substr?

|  |  |
| --- | --- |
| Substr | Instr |
| It returns the specified portion of the string specified by the index position | It provides the character position of the pattern in the given string |
| Substr(‘BCDEF’,2,3) =CD | Instr(‘ABCBEF’,’B’,2) = 4 (2nd occurrence of the character B. |
| It returns character | It returns number |

89) How to integrate spring with hibernate?

In applicationContext.xml , define the datasource , configure the hbm files . Use springframework sessionfatorybean class to get the sessionfactory by passing the datasource,hibernate properties and hibernate properties file as properties.

90) What is thread local?

ThreadLocal is a java classes which can create the threadlocal variable and can be read and write only by same thread. If two thread access the same code if it has thread local reference then two threads cannot see eachothers thread local variables.

Usage:

private ThreadLocal myThreadLocal = new ThreadLocal();

myThreadLocal.set("A thread local value");

String threadLocalValue = (String) myThreadLocal.get();

**91) How to increment the key value in hibernate without using code and database logic?**

92) Why character array is better than string for storing the password in java?

Since strings are immutable in java , it will be in memory(String pool) for long time until garbage collector runs and clear it . During this time , if any one has access to the memory dump they can easily get the value of the password. We cannot change the value of the string contents also , if we change it will produce new string object in memory. String has toString() method which prints contents of the string. It causes security risk.

But in char array we can change the contents easily. It prevents the security risk. Once the password is processed , we can easily wipe the data in char []. When we print char[] it wont print the contents.

Java itself supports the getPassword() method from TextFiled which returns char[].

93) What are the security issues arise in web service ?

1) Confidentiality :- client send an XML request to server as SOAP which runs on top of HTTP. HTTP supports SSL. SSL is used to encrypt the communications messages. SSL is widely used technology for security. SSL gives point to point security not end to end security.

When the web service communicate with chain of applications , data needs to be encrypted in all of these applications . We can use W3C XML encryption standards. This is the standared framework for encrypting and decrypting the xml messages.

2) Authentication

It can be done by using the SSL digital certificate. Users identity is verified based on the credentials given by the user.

3) Network security

It can be achieved by user authentication and data encryption and firewalls. We can also filter the SOAP Headers and validate

1. Integrity: it can be achieved by digitally sign the message.
2. Authorization : it can be done based on the users entitlements.

94) Difference between Beanfactory and ApplicationContext .

|  |  |
| --- | --- |
| BeanFactory | ApplicationContext |
| It gives basic features like Dependency Injection | It gives advanced features like spring AOP support , handling message resource (i18) , event propagation and web application support. |
| It instantiates the bean when the getBean() method to be called | It instaniates the bean when the container is started |
|  | It propagates event to bean that are registered as listeners |
|  | It support enterprise services like JNDI access , EJB integration |

95) How HashTable internally works in java ?

HashTable is an array of List . Each list is known as buckets. The bucket position is identified by keys.hashCode() method.

While putting items :

* 1. Calculate the hashcode of key by key.hashCode()
  2. If bucket with that hashcode already exists , then use the equals() method on that key to search the key to determine whether that key already exist or not. If already exist , replace the value , if not add the value.
  3. If second keys also returns the same hashcode , even if they are not equal . it will be added into the list(bucket)
  4. If the bucket with particular hashcode not exist, it will create an bucket and add an item into the list.

While retrieving the items :

* 1. Calculate the keys hashcode.
  2. Go to the bucket allocated for that particular basket.
  3. Use keys.equals() method to get the same key , and this will return the value associated with that key.

96) Why map is not part of Collections?

Collections :- it support the list of single elements

Map :- it support key value pairs.

97) How to override the parent method which throw NullPointerException.

If the super class method throws NullPointerException , then the subclass overridden method can thro same exception or no exception or Runtime Exception but never throw object of Exception class.

98) What is the overriding rules when the parent class throws unchecked exceptions ?

If the parent method throws unchecked exception , then the overridden method in child classes should not throw any checked exceptions and it can throw parent exception of unchecked exception .

99) What is concurrency in java ? what are the concurrency issues occur?

Concurrency is the ability to run several parts of the program parallel. Thread is a lightweight process. Threads has its own call stack but it can access the shared data. There are two issues arise in concurrency

* 1. Visibility : if thread A read the value and later it is modified by thread B ,and Thread A doesn’t aware of this change.
  2. Access: if several threads access and change the same data at the same time.

100) How to deal with concurrency ?

Java provides the java.util.concurrent package has set of claases and interfaces to deal the concurrency issues.

101) how to achieve thread safety in java ?

By using the synchronization mechanism.

102)What is the preliminary condition on objects equality check ?

1) when the two objects are equal by equals() method , they must return same hashcode

2) Two unequal object can have the same hashcode

103) What is the preliminary condition for using the object as key in the hashing collections like HashMap, HashTable and ConcurrentHashMap ?

The object must implement the equals() and hashcode() methods.

103) What happens if two keys has the same hashcode ?

During put operation in the Map , it will form an linked list data structure.

104) What is the preliminary condition on overriding the equals() method?

The hashcode() method should also be overridden, when we override the equals() method.

They both should use the same set of fields to check the equality.

105) How to override the hashCode and equals() method ?

@Override

public boolean equals(Object object) {

boolean result = false;

if (object == null || object.getClass() != getClass()) {

result = false;

} else {

Tiger tiger = (Tiger) object;

if (this.color == tiger.getColor()

&& this.stripePattern == tiger.getStripePattern()) {

result = true;

}

}

return result;

}

// just omitted null checks

@Override

public int hashCode() {

int hash = 3;

hash = 7 \* hash + this.color.hashCode();

hash = 7 \* hash + this.stripePattern.hashCode();

return hash;

}

106) what is hibernate dialect ?

Which is used to connect the application to database. Which also specify the type data base used to generate the database specific queries.

107) If a method throws NullPointerException in the superclass, can we override it with a method which throws RuntimeException?  
  
The answer is you can because both r runtime exception. But you can not do same if its checked Exception  
  
107) When the instance initializer block gets executed ?

At the time of object creation. Compiler copies the code from instance block to constructor after the super() statement. So the instance block statements are executed first then the constructor statement are executed.

108) How to convert arraylist to Array ?

- create the string array with size of arraylist

- call the toArray() of arraylist by passing the created arrayobject.

EX:

List al = **new** ArrayList<String>();

al.add("One");

al.add("Two");

al.add("Three");

al.add("Four");

al.add("Five");

String[] stringArrayObject = **new** String[al.size()];

al.toArray(stringArrayObject);

109) How to remove unwanted characters from string ?

We can use replaceAll() method from String to replace the unwanted characters.

String phoneNo="(952)-594-5676";

phoneNo=phoneNo**.replaceAll("[^0-9]", "");**

System.***out***.println(" New Phone number :::"+phoneNo);

ReplaceAll() takes 2 arguments, first one is the expression tells what needs to be replaced. Second one tellsreplaced to what ?

^ - implies except

[]- implies expression

In the above example , it will replace all characters and symbols except 0-9. So the output is 9525945676.

110) How to remove duplicate values from an arraylist ?

Create another arraylist like this ,

List<String> list=**new** ArrayList<String>();

list.add("two");

list.add("one");

list.add("three");

list.add("one");

list.add("three");

list.add("four");

System.***out***.println("list values .........."+list);

Set<String> set=**new** TreeSet<String>(list);

System.***out***.println("set values .........."+set);

**List list2=new ArrayList<String>(new TreeSet<String>(list));**

System.***out***.println("list2 values .........."+list2);

Set doesn’t allow duplicates , conver the list to set removes duplicates.

111) How to sort array in java ?

We can use the sort method from Arrays Class.

String strArray[]={"Vasantha","Ian","H"};

**for**(String s:strArray)

System.***out***.println(" before sort ..............."+s);

Arrays.*sort*(strArray);

**for**(String s1:strArray)

System.***out***.println(" Soreted array ........."+s1);

112) How to sort arraylist in java ?

Using the Collections.sort(list) , we can sort the list.

List list=**new** ArrayList<String>();

list.add("Yak");

list.add("Hi");

list.add("Cat");

list.add("Apple");

System.***out***.println(" list before sort ............."+list);

Collections.*sort*(list);

System.***out***.println(" list after sort ............."+list);

113) How to find the first non occurance of character in string ?

If it s a string , we can do like this,

String testString= "ACBA";

HashMap charMap=**new** HashMap();

**for** (**char** ch:testString.toCharArray()){

System.***out***.println(ch);

**if**(charMap.containsKey(ch)){

**int** val=(Integer)charMap.get(ch);

charMap.put(ch, val+1);

}**else**{

charMap.put(ch,1);

}

}

**for** (**char** c:testString.toCharArray()){

**if**(((Integer)charMap.get(c))==1){

System.***out***.println("non occurance characters..."+c);

}

}

System.***out***.println(charMap);

There is an simple alternate way is,

String input ="AACBC";

**for**(**char** i :input.toCharArray()){

**if**(input.indexOf(i)==input.lastIndexOf(i)){

System.***out***.println(i);

**break**;

}

}

114) How to find the occurance from the list ?

We have the Collections.frequency() method to find the occurrence of element from list.

List<String> list=**new** ArrayList<String>();

list.add("C");

list.add("A");

list.add("B");

list.add("T");

list.add("C");

**for**(String str:list){

**int** nooftimes=Collections.*frequency*(list,str );

System.***out***.println( str + " "+nooftimes);

**if**(1==Collections.*frequency*(list,str )){

System.***out***.println(" first non occurance is " + str );

}

}

**int** no=Collections.*frequency*(list,"B" );

System.***out***.println(" Occurance is "+no);

115) There is an abstract class called “Bird” has fly() method. There are two concrete classes “Parrot” and “Pigeon” which are implementing the fly() method. This is existing design. There is a new requirement comes in that “Penguin” is a class which is also “Bird” but it cannot fly .The penguin object cannot call the fly(). How to solve this problem ?

Since this is the existing design , we have to handle this in new class “Penguin” . We can create some custom exception(“I canoot fly”) , and throw this exception from Penguin’s fly(). This way can convey to user penguin doesn’t support fly behavior.

116) What are the fail-fast and fail-safe iteratiors?

Both are iterators that define how the system behave during the failure condition.

Fail-fast iterator throws ConcurrentModificationException when the collection is modified during the iteration process. Ex: Iterator of ArrayList.

Fail-safe iterator doesn’t throw ConcurrentModificationException when the collection gets modified during the iteration process.sinces it works on clone of the collection. Ex: Iterator of ConcurrentHashMap.

117)What are the ConcurrentCollections available ?

We make the collections synchronized by calling the synchronizedMap(),synchronizedList() and synchronizedSet().

But Java 1.5 & 1.6 introduces the following collections to deal with concurrency in java.

ConcurrentHashMap- Alternative for synchronized Map or HashTable

CopyOnWriteArrayList – Alternative for synchronized List.

CopyOnWriteArraySet – Replacement for synchronized Set.

ConcurrentSkipListMap – alternative for synchronized TreeMap.

ConcurrentSkipListSet – alternative for synchronized TreeSet.

BlockingQueue –

118) How to filter java collection ?

We can use java 8 to filter the collection.

List<Person> personList=students.stream().filter(p->p.getMark()>80).collect(Collectors.toList())

119) How to convert List to Set ?

Before java 8 ,

Set set=new HashSet(list);

After java 8, there is an another option,

Set set=list.stream().collect(Collectors.toSet());

120) When to use ConcurrentHashMap ?

When there is more number of updates and less number of retrievals the we can use concurrentHashMap.It doesn’t lock the entire table like HashTable.

121) How does ConcurrentHashMap works internally ?

It allows concurrent read and thread-safe update operation. It allows multiple threads to access the part of the map. It don’t lock the entire table . it is dived as segments. By default it can be accessed by 32 threads at highlevel.it works same as HashMap. It doesn’t require explicit synchronization. Read Operation doesn’t block the map . Only update can do. The retrieval operation may overlap with the updation operation on the hashmap. So the retrieval process and the iteration process may not return the latest update.

Default concurrent level is 16 . so 16 threads can access the map concurrently. We can change the concurrency level in the constructor.

122) Can two threads update the ConcurrentHashMap simultaneously?

Yes. But two thread lie on the same segment than its not possible.

123) How serialization works?

Serialization serialize the state of the object not the class file or methods.

124) I have a class which has three variables with some initial values. It has a method which is changing the variables values. While serializing this object an send it over the network , I wanted to get only the default initial values not the values supplied by a method. How to achieve it?

Serialization support serializing the state not the methods . So when we serialize this object only the variables with default values will be serialized , the method wont be serialized. Static and transient variables cannot be serialized.

**125) What is the return type of the Dispatcher Servlet?**

**126) If I have spring listener , if the request comes in which will execute first DispatcherServlet or Listner ?**

127) How to read lines from notepad file and find the occurrence of each word ?

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**try**{

FileInputStream fi=**new** FileInputStream("V:\\TODO.txt");

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(fi));

String line=**null**;

String[] word=**null**;

**int** lineNo=0;

HashMap map=**new** HashMap();

**while** ((line=br.readLine())!=**null**){

**int** occur=0;

lineNo++;

word=line.split(" ");

**for**(String wr:word){

System.***out***.println(" str array " +wr);

**if**(wr.matches("done")){

occur++;

}

}

map.put(lineNo,occur);

}

System.***out***.println(map);

}**catch**(FileNotFoundException fe){

System.***out***.println("File not found");

}**catch**(IOException ie){

System.***out***.println(" IO exception");

}

}

Output :

{1=1, 2=1, 3=1, 4=0, 5=1, 6=1, 7=0, 8=0, 9=1, 10=1, 11=2, 12=0, 13=0, 14=0, 15=0, 16=0, 17=0, 18=0, 19=0, 20=0, 21=0, 22=0}

128) what is the return type of the Spring Controller methods ?

It returns ModelAndView Object.

129) what is serialization ?How serialization happens for the different types of variables ?

Serialization is the process which converts the object into binary format and saved into disk or send over the network or send to other JVM. The reverse process is known as deserialization. The API’s involved are Serilizable,Externalizable , ObjectInputStream and ObjectOutputStream. This process serilize class name and fields.

1) instance variables:

2) static variables :

3) transient variables:

4) super variables :

130) What is the difference between embedded id and primary id in hibernate ?

When more than one column makes the primary key is known as composite primary key. In hibernate the composite primary key fields are defined as separate entity with annotated as @Embeddable. This will be used as composite primary key of the entity using @EmbeddedId.

Ex:

@Embeddable

Class ProjectId{

Int deptid;

Long projectId;

}

@Entity

Public class Project{

@EmbeddedId ProjectId id;

}

|  |  |
| --- | --- |
| @Id | @Embeddedid |
| Represent the Simple Primarykey | Represent the composite primary key |

The entity class always has the single primary key. Here more than one fields forming the primary key. So we defining as embeddable class , and using this as primary key in the entity class.

131) what is the difference between Aggregation and Composition ?

Composition: - which implies the has-a relationship between two objects. One objects has the reference of other objects. Ex: HashMap has an Entry class .If map is deleted all entries in it should also be deleted. The object which holds an reference of other object is called container and is responsible for the lifetime of the object it holds. When one object owns other object the relationship defined as composition. Without parent , the child cannot exist meaningfully. Composition gives strong relationship.

Aggregation :- which is also implies the has-a relationship between two objects. One objects has the reference of other objects. The object which has reference to other object cannot control the life time of the reference object which it holds. When one object uses other object the relationship between these objects is called as Aggregation. Without parent , the child can exist. EX: Organization and Employee.

132) How to make lazy initialization on singleton?

Instead of creating object at the time of variable declaration , we can create instance in the getInstance() method. But this is good in single threaded application and not suitable for multithread application. When multiple thread accec the getInstance() at time , it will create multiple instances.

132.1) Does this type of singleton support thread safety?

No. when multiple threads access it, it may create more than one instances. To achieve thread safety , we can use early initialization on instance or we can go for static inner class or use synchronization.

133) How to make the singleton as multithreaded ? Or How singleton works in multithreaded environment?

We can have the private static inner helper class which creates an object. Which is suitable for multithreaded application also supports lazy initialization.

Lazy initialization supports : Since static inner class is taking care creating the instance and it will not be loaded at the time of main class loading. This inner helper class is loaded when the getInstance() called at the first time . This way it supports lazy initialization.

Multithreaded Support: The class loader has its own synchronization which gives thread safety. We can also use single element Enum to achive the singleton in multithreaded environment.

public class BillPughSingleton {

private BillPughSingleton(){}

private static class SingletonHelper{

private static final BillPughSingleton INSTANCE = new BillPughSingleton();

}

public static BillPughSingleton getInstance(){

return SingletonHelper.INSTANCE;

}

}

134) How do we synchronize the changes done in the cache from one server to other in a distributed environment ?

We can use JavaMessaging API to send updated messages across containers . Each containers has listener points to the topic and the message is published , it cab ne accessed by other containers and can update their caches.

Or we can use any cluster aware cahce like SwarmCache or JBossTreeCahce.

135) How singleton works in distributed environment ?

Singleton instance is created per container . So in distributed environment each container has its own singleton instance. To avoid this or to make it synch we can use vendor specific singleton service available in the application servers like, Websphere , JBoss and WebLogic.

The best approach is to use vendor specific singleton support.

Singlton in Websphere :

Websphere supports objectgrid concept to make use of the singleton service.

136) Difference between DOM and SAX Parser.

|  |  |
| --- | --- |
| **DOM** | **SAX** |
| It works based on document object model | It works based on event |
| It occupies more memory since it loads the whole XML as DOM tree in memory | It doesn’t require more memory since it parse the XML step by step.It fires an event whenever it encounters and element tag , element and attributes. |
| It is suitable for small xml files. If it is large xml files then it maynot be loaded into memory | It is suitable for large XML files. |
| It doesn’t require more code to pare the XML. | It needs more code to read the XML. |
| DOM-Document Object Model | SAX-Simple API for XML |

Java supports both xml parser via JAXP.

137) What is circular dependency in spring ?

The Spring container usually creates the bean in the order specified.

When bean A has reference of Bean B and Bean B has a reference of Bean A , the container gets confused which one should be created first. This is called circular dependency. This happens when the constructor injection is used.

How to overcome this issue?

To overcome this ,

* 1. Use setter or field injection :

This way , spring creates the bean but the dependencies are created when they are first used.

**@Component**

**public class CircularDependencyA {**

**private CircularDependencyB circB;**

**@Autowired**

**public void setCircB(CircularDependencyB circB) {**

**this.circB = circB;**

**}**

**public CircularDependencyB getCircB() {**

**return circB;**

**}**

**}**

* 1. Use @Lazy annotation in the constructor parameter(any one of the bean) :

**public class CircularDependencyA {**

**private CircularDependencyB circB;**

**@Autowired**

**public CircularDependencyA(@Lazy CircularDependencyB circB) {**

**this.circB = circB;**

**}**

**}**

It will create the proxy and inject. The fully initialized bean is created , when its first needed.

* 1. Use @ PostConstruct :

Use PostConstruct method to give its dependency to other bean. Use @Autowired to inject the bean.

@Component

**public class CircularDependencyA {**

**@Autowired**

**private CircularDependencyB circB;**

**@PostConstruct**

**public void init() {**

**circB.setCircA(this);**

**}**

**public CircularDependencyB getCircB() {**

**return circB;**

**}**

**}**

**@Component**

**public class CircularDependencyB {**

**private CircularDependencyA circA;**

**private String message = "Hi!";**

**public void setCircA(CircularDependencyA circA) {**

**this.circA = circA;**

**}**

**public String getMessage() {**

**return message;**

**}**

**}**

138) What is the purpose of database index?

Database index are data structures which stores the values of particular column. Index is created for particular column which improves the performance of the retrieval process.

139) What is the difference between DELETE and TRUNCATE ? How the DELETE and TRUNCATE behave on ROLLBACK ?

|  |  |
| --- | --- |
| TRUNCATE | DELETE |
| It will delete all rows from table | It will delete only particular row based on the condition |
| It is an DDL statement | It is an DML statement |
| In case of rollback, it wont rollback the deleted records | We can rollback or commit the deleted records |
| It wont fire any triggers | It fire the delete triggers |

140)

141) What are the DDL statements ?

DDL Statements are ,

* 1. Create , Alter and Drop schema objects
  2. Grant , Revoke , Analyze , Audit and Comment :- Grant and revoke privileges and roles

**142) What is lock object in java?**

This is same as implicit lock used by the

**143) How to handle the SQLException in hibernate?**

144) Diff between sessionfactory and session ?

|  |  |
| --- | --- |
| SessionFactory | Session |
| It is an thread safe object. It is intended to use by all threads in an application | It is an non thread safe object. |
| This object is one per application | This object is one per client |

145 )How SessionFactory is thread safe ?

The internal state of the sessionfactory is immutable. Once it is set , it cannot be changed and it includes all meta data about ORM mapping.So none of the thread can change the data until rebuild the sesson factory object. All threads can access it , that’s why it is thread safe.

145) Can a prototype bean injected into the singleton bean in spring ?

We cannot inject the prototype bean into the singleton bean.The container creates the only one singletone bean and only one prototype bean at the time of container instaniation . But the injected prototype bean needs to be created for every request. If both beans are in same scope then no issues .But here there is an issue creating the prototype been for every request.We can use the look-up method injection to achieve this. Method injection is only applicable via configuration not by annotation.

To make use of method injection , the singletone bean should be abstract abd it has abstract method for creating the prototype bean. There will not be direct reference of prototype bean in singleton bean.

Ex:

public abstract class Singleton {

protected abstract Prototype createPrototype();

public void doSomething() {

createPrototype().foo();

}

public void doSomethingElse() {

createPrototype().bar();

}

}

Bean Configuration :

<bean id="prototype" class="ch.frankel.blog.Prototype" scope="prototype" />

<bean id="singleton" class="sample.MySingleton">

**<lookup-method name="createPrototype" bean="prototype" />**

</bean>

<lookup-method> is used for the method injection. createPrototype() is the abstract method available in singleton bean and its implementation is given by spring.

**Can a Singletone bean inject the Prototype bean ?**

146) How do the method return list which has return type void ?

**public** **class** VoidReturnValue {

**public** **static** **void** asList(List list){

list.add("1");

list.add("2");

system.out.println(" list value is ....."+list);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

List list=**new** ArrayList<>();

*asList*(list);

System.***out***.println(" list value is ....."+list);

}

}

147) Difference between JAXP and JAXB ?

|  |  |
| --- | --- |
| JAXP | JAXB |
| Java API for XML Processing | Java Architecture for XML Binding |
| It is an low level API which use DOM , SAX parsers to parse the XML | It is an High level API which uses XSD binding to process the xml |
| Which requires content handlers and call back methods to be needed to parse the xml | Which doesn’t require the content handlers and call back methods and it just bind the xml schema with xml and do the marshalling and unmarshalling. |
| It needs more code | It needs less code |

148) How to validate the XML in java ?

149) What is the purpose of having the service layer and DAO layer ?

DAO Layer :- it should be limited to only add / update / insert / select entity objects to and from database.

Service Layer :- All other business logic and transaction handling codes should be there in service layer.

150) which design pattern used between service layer and DAO layer?

151) Difference between session.commit() and session.flush() in hibernate?

|  |  |
| --- | --- |
| Commit | Flush |
| Commit do both the flush and commit the unit of work | It synchronize the state of the object available in the session into database. It just executes the SQL query on the database but does not committed until explicitly call the commit on the Transaction |
| No way to rollback the changes once committed | It throws exception if any constraint violation occurs when insert an record |

152) Does the session.clear() make an database operations in hibernate?

No. It will remove all the objects from session.(first level cache)

IT moves the objects from persistent to detached.

153) Difference between session.clear() and session.evict() ?

|  |  |
| --- | --- |
| Clear | Evict |
| It removes all the objects from session | It remove only particular object from session |
|  |  |

154) Difference between session.close() and session.clear() in hibernate .

|  |  |
| --- | --- |
| Close | Clear |
| It will close the session and will be called after the transaction is ended | It will clear the session and will be called any where with in the transaction boundary |
| It will remove the object from session and close the database connection and do cleanup | It will only remove the objects from session |

155) Explain spring bean lifecycle ?

156) SQL Query - Count of all employees in each department ?

Select dept\_id,count(\*) from employee group by dept\_id;

157)Explain - Spring + hibernate transaction management ?

There are two ways to create the transaction in spring hibernate application.

* 1. Programmetic transation
  2. Declarative transaction : Declarative transaction can be achieved by using the spring AOP.

158)Explain spring bean life cycle ?

159) Autowiring concept: by type. The bean definition has autowiring by type is enabled , and it has two referenced bean with the same type , What will be the result ?

It throws exception when container loads.

160) How logging framework integrated with Spring AOP?

161) How to handle the transactions in spring hibernate application?

There are two ways to create the transaction in spring hibernate application.

* 1. Programmetic transation
  2. Declarative transaction : Declarative transaction can be achieved by using the spring AOP.

**Kumaran systems interview:**

162) what is the purpose of datasource ?

It increase the application performance as connections are not created and closed within the class. They are managed by the application server and can be fetched while at runtime.

It supports the facility to create connection pool.(it supports connection pool facility)

Getting database connection using the datasource as like,

DataSource dataSource =(DataSource) new InitialContect.lookUp(datasourceName);

Connection connection=dataSource.getConnection()

165) Difference between component scan and package scan?

167) There is a component in UI screen and it needs to populate the value from bean with sorting? How to do sorting at bean level ?

The bean level sorting can be done by implementing the comparator or comparable interface. The have the following methods used to compare and sort the list of objects.

Comparable :- compareTo(object o1)

Comparator :- compare(object o1,object o2)

163) When to use MDB ?

To receive messages asynchronously and process it , use message driven bean. It has business logic and doesn’t maintain the state like stateless session bean.

164) There is an topic which has 5 subscribers, 3 are online and 2 are offline . when the first message is posted in topic ,3 subcribers received the message and 2 subs are still offline. When the second message posted to topic , how many message are there in topic and the 2 offline subscribers comes to online which message has been subscribed?

If the subscriber uses durable subscription, the topic has 2 messages are available for the offline subscribers and when they come online those 2 messages are delivered.

**165) What is hibernate interceptor ?**

Design principles :

Logging framework

Auditing framework

Performance

DAO