

Advance Java .

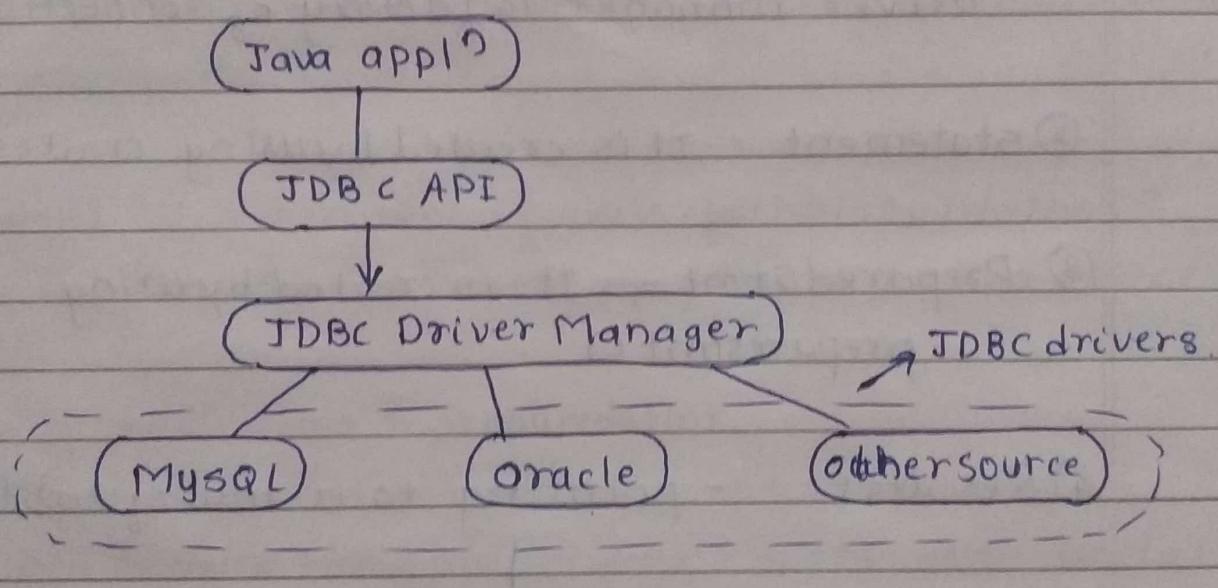
JDBC .

- It is java database connectivity
- it is an API that allows java programs to access database management system.

-

Architecture -

specifying the relational databases.



Steps of JDBC

- 1> import JDBC packages.
- 2> Load the driver. (`ClassName("oracle.jdbc.driver.OracleDriver")`)
- 3> open the connection. (`P.S. Conn getConn(string)`) throws `SQLException`
- 4> Create a statement. (`P.S. createStatement()`) throws `SQLException`
- 5> execute the statement. (`P. resultset executeQuery()`)
- 6> process the result.
- 7> close the result. (6)
- 8> close the connection. (`P. V. close()`) throws `SQLException`

driver class -

is a utility class that are used to carry out some task..

* what is JDBC API components

interfaces -

① Connection - It is created by using getconnection() method.

Driver manager is factory of connection.

② Statement - It is created by using createstmt()

③ PreparedStatement - It is created by using preparestmt()

④ ResultSet - pointing to a row of table

classes -

DriverManager - acts as an interface bet' user & drivers.

SQLException - provides information on database access errors.

* what are the JDBC stmt ?

① preparedstmt -

it is used to access the database & execute static SQL query runtime.

when we need to provide input parameters to the query at runtime

② callable stmt -

when we need to access the database stored procedures.

* what is return type of class.forName() method ?

It returns the object of java.lang.class object.

* diff betⁿ stmt & preparedstmt.

stmt ① it execute the queries with the database.

pstmt ② it execute the parameterised query with database.

stmt → p. stmt

③ Query is compiled at each of the time.

Query is compiled at once.

* diff. betⁿ execute, executequery, executeupdate.

Execute

it is used for both.

Select & updatestmt.

ExecuteQuery

it is used only select

stmt.

executeupdate.

it is used to update,

delete, insert operation.

it return boolean value.

it returns object type value.

it returns integer type value.

diff betⁿ procedure & function.

Procedure

It is used to perform business logic

does not have return type

may return 0 or more values

support input & output parameters

* diff. betⁿ.

java.util.Date

it represents only Date

java.sql.Date

it represents both date and time.

functions

it is used to perform calculation.

must have return type

may returns only one values

supports only input parameters

Hibernate

* what is hibernate ?

it is java-based persistence framework

it is an ORM (Object relational mapping) of java

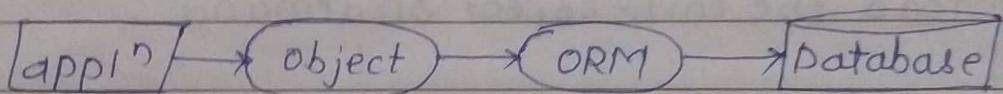
languages to map the java classes to database & also map the datatypes to sql database.

* what is ORM ?

ORM is object relational mapping.

it simplifies the data manipulation, data creation & data retrieve

It is object oriented technique to map the object and stored into database.



* Advantages of hibernate over JDBC.

① clean readable code

② HQL - allows you to write the object oriented queries for advance operation

③ Transaction - enable to achieve the correspondance

④ Exception handling - rollback in case it does something wrong.

* Interfaces of Hibernate.

① Configuration -

extending the properties of the session factory .

② Session factory -

collect the info for any listed files to initiate the dependency appropriately.

* ^③ what is Session.

It is an combination of app1n & database which initiate the session factory.

^④ criteria

^⑤ Query

^⑥ Transaction - enable to achieve the data consistency & rollback in case does have unexpectedly something wrong.

* what is lazy loading ?

to improving the app1n performance by helping to load the child object on demand.

* diff. betⁿ first cache level & Second cache level.

first level

shared only local
session object

Second level

shared multiple session
object.

It is enabled by default
it no way to disable it

it is disabled but we
enable it through configurtⁿ

it is available only when
Session is Open until
it's destroyed.

it's available through appⁿ
life cycle, only destroyed &
retrieve whe app1n is
restarted.

* what is configuration file ?

mapping the appⁿ in to the database

say : it is hibernate.cfg.xml.

- set the ~~the~~ driver class, URL, username & password

* How to create immutable class.

using annotations,

@immutable annotations

* what is hibernate mapping.

used to define the bean fields & corresponding database columns.

* Annotations -

① javax.persistence.Entity -

it tells that classes are entity beans

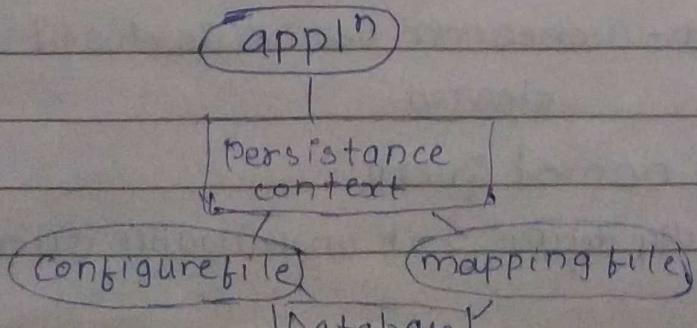
② javax.persistence.Table -

class are map to the tablename in database

③ javax.persistence.Id -

indicate the primary key value in bean.

* Hibernate structure -



diff b/w get() & load()

get()

load()

method get the data

load the data when it's required

everytime method is called

nearly time method

* merge() method

Used to ~~remove~~ update the existing value.

* hibernate support native sql queries

Yes, it provides the createSQLquery()

* can we declare entity class as final?

→ No, because hibernate use proxy classes & objects

* what is persistent & state

persistent - whenever the class is persisted.

state - represents only one row in database & retrieve the identifier value.

① Transient - it is an initial state.

Object exist in heap memory

② persistent - object can associate with session.

③ Detached - whenever session is closed the cache is cleared.

* benefit of named query -

it is statically define & it unchangeable query string.

Hibernate

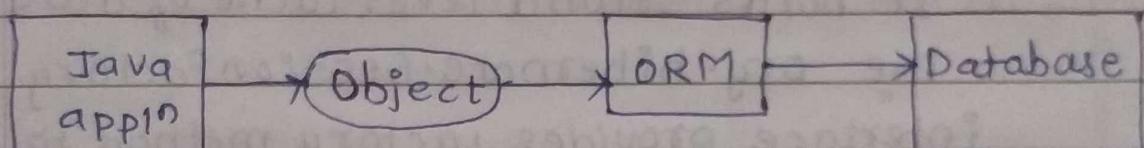
* Intro -

It is a java framework that simplify that development of java application to interact with the database.

It is an open source, light weight ORM tool.

* ORM tool :- (Object Relational Mapping)

It is a programming technique that maps the object to the data stored in database.



* Advantages of Hibernate -

- 1> opensource & lightweight
- 2> fast performance
- 3> Database independent Query
- 4> Automatic Table creation.
- 5> simplify complex join

* Hibernate architecture -

It include many objects such as persistent object, session factory, transaction, connection factory, session, transaction etc.

* Hibernate architecture layer.

1) Java application layer.

2) Hibernate framework layer

3) Backend API layer

4) Database layer.

* Elements of Hibernate.

1) Session factory -

- It is a factory of Session & client of connection provider.

- It holds second level cache of data.

- The org.hibernate.SessionFactory interface provides factory method to get the object of Session.

2) Session -

- Session object provide connection with relational database.

- session object is created using SessionFactory

3) Transaction -

- The transaction object specifies atomic unit of work.

- It is optional.

- The org.hibernate.Transaction interface provides method of transaction.

4) configuration interface -

- It is used to configure hibernate.
- It is used to load the configuration.
- files & class mapping files.

Ex - new Configuration().configure(configfile)

Hibernate lifecycle -

~~EE~~

1) Transient state -

objects that are currently mapped with a session & has corresponding view in database.

2) Transient -

objects which are never mapping with session.

3) Detached - object that are mapped with session but session is closed later.

* Hibernate JPA Annotations

1) Entity - import javax.persistence.Entity.

2) Table -

3) Column -

4) Id -

5) Generated value -

6) Version -

7) Transient -

8) Lob -

Page No.	
Date	

- * Hibernate Association mapping Annotation
 - 1> one to one
 - 2> many to one
 - 3> one to many
 - 4> many to many
 - 5> primary key join column.
 - 6> join column

Spring

- * what is spring framework?
 - It is Java framework, it's powerful open source, loosely coupled, lightweight.

- * spring modules ?

- ① Core container.

- ② Data access.

- ③ Web

- ④ AOP

- ⑤ Test

- * Features of spring framework -

- ① layered architecture

- ② separation of appl'n business logic

- ③ highly configurable.

- ④ creation & management.

- * what is spring IOC container.

- it is predefined program.

- component of Spring fw.

- create the object and hold it in memory and inject to another object as required.

- * what is dependency injection?

The components & services need not be connected by us we need to describe the services in the configuration file

* diff. bet" setter & constructor injection

Setter

partially allowed setter
injection

constructor

Partially not allowed
constructor injection

it does not override
setter injection
property

It is override the
constructor property

works many properties

works few properties

* what is spring Beans ?

It created the configuration metadata to user supply
to the container.

* Scopes in spring.

- ① singleton - single instance IOC container
- ② prototype - any no. of object instance
- ③ Request - definition is an HTTP request.
- ④ session - def'n of HTTP-session.

* what is autowiring ?

It is auto wired spring from beans for a
classes of constructors & variables where as inject
in a dependency.

- ① no autowiring.
- ② byname
- ③ bytype

* limitations of autowiring.

① Overriding possibility -

used to <constructor-arg> & <property> to override

② Datatype restriction -

primitive datatypes, string & classes can't be autowired.

* diff b/w

Spring

They have multiple features like, dependency injection, data binding, AOP.

Spring Boot

it simplifies commonly used dependency

It is loosely coupled.

applicable for straight command line.

* Features of spring boot.

① Starter dependency -

improves productivity & reduce burden.

② Spring initializer -

helps to developer in creating internal project structure.

③ Auto-configuration -

helps to loading the default configuration

④ Spring actuator -

provide the management end points.

* diff betn ↗

request param

used for accessing the query parameter values from the request

path variable

it identifies the pattern that used in URI incoming request

* diff betn ↗

@ Autowired

It has required attribute

@ inject

not required attribute

Singleton is default scope

prototype is default scope

In case of ambiguity

@ Qualifier annotation used.

In case of ambiguity

@ Named Qualifier is used

* what is data jpa -

specification for persisting object in the enterprise appn.

Web Services.

* what is web services -

specially designed to communication b/w
client & server app's

* Types of web services.

- ① SOAP (Simple object access protocol)
- ② RESTful (Representational state transfer)

① SOAP -

transport-independent messaging protocol to
transfer a message & based on XML protocol.

② RESTful - to make the web service more efficient.

* Features of web services?

XML messaging system.

support loosely coupled

can be synchronous & asynchronous

* Components of web services.

SOAP (Simple object access control)

UDDI - Universal Description, Discovery integration

RDF - (Resource description framework)

XML (extensible markup language)

* diff bet

API

it can be online or
offline

web services

must be web network

lightweight architecture. not lightweight
becoz send & receive data

It is open source

It's not open source

* advantages of Restful Service

platform independent

easy to implement.

lightweight

faster & provide better performance

* Which protocol is used in Restful services?
HTTP.

* UDDI?

Discover the web services.

Diff betⁿ applⁿ context & bean factory.

applⁿ context

bean factory.

- * what is AOP? (Aspect-oriented programming)
→ add the enterprise ~~app~~ functionality to the applⁿ.

what is IOC? (Inversion of control).

- configure the object
- ~~access~~ assemble the dependencies betⁿ the objects

Types.

- Bean Factory
- applⁿ context

Bean Factory

applⁿ context

It uses lazy initialization

It uses aggressive initialization

It doesn't support annotation

It supports annotation based

based dependency

dependency.

It explicitly provides resource
object using syntax.

It creates a resource obj by its
own.

Diff. betⁿ

Spring Boot

It is module spring based
applⁿ

It provides spring powered
framework.

Spring MVC

It is model view controller based
applⁿ.

It provides ready to use
features.

not required deployment
descriptor

It is required deployment
descriptor.

* spring Boot annotations

@Required
@autowired
@configuration
@bean
@service
@Repository

* Spring MVC Rest Annotations

@Requestmapping
Get mapping
post mapping
Delete mapping

Data JPA

easy to access the different kinds of persistance
stores both relational database & No SQL data
stores.

Java 8

Page No.	
Date	

1 lambda expression

- it is anonymous functions
- no need to provide a name for implementation
- doesn't have any name or return type.

* what are new features introduced in java 8

lambda expression.

functional interface.

static & default method in interface

stream API

Date / time API change.

method reference.

* Advantages of java 8

more compact code.

less boiler plate code.

more readable & reusable code.

parallel operations.

* what is functional interface & why it use?

It can have only one abstract method is called functional interface.

to achieve functional programming by passing method arguments.

ex. in general

void m1 (int a, int b)

{ }

? ~~execute~~

but using FI

void m1 (method c)

{ }

? .

ex - Comparable with compareTo() method.

Runnable with run() method.

* Method reference in java 8?

- - it used to refer method of functional interface
- replace lambda expression with functional references method.

Syntax - class:: methodname.

System.out:: println;

* what is optional?

to avoid null pointer exception.

* what are default methods in java 8.

- interface public abstract methods (Till java 8)
- in java 8.

default & static methods also allowed.

↳ allow to add new methods to interface without affect classes.

* static method in java 8
method hiding.

* what is streams API.

used to process the collections.

Properties	
Lists	

diff. bet' collection & stream.

Collection	Streams
It is a group of data	perform operations on data.
we can add or remove elements.	we can not add or remove elements.
Eagerly constructed ex - list, set, map	lazily constructed filter, map

* Intermediate operations in streams
filter, map, sort, sorted.

* Terminal operations

collect

foreach. - iterate

Find the employees whose salary is above 10000 using streams

P. class StreamFilterEx {

P. S. V. M (String a[]) {

List<Employee> emp = new ArrayList<>();
emp.add(new Employee("Varshu", "Account", 8000))

emp.stream().filter(em → em.getSalary() > 10000)
, forEach(s → System.out.println);

}