Java and AWS Training

Day 1 : 14-11-2022

**Program :** set of instruction to perform a specific task.

Structure programming language

OOP : Object oriented programming language

Functional programming language

AOP : Aspect oriented programming language

C: is a basic structured programming language.

#include<stdio.h>

Global declaration

Pre defined function or user defined function.

#include<stdio.h>

void main() {

printf(“Welcome to C language “);

}

Data types

Operators

If statement

Switch statement

Looping

Pointer

Function

Enum

Structure

struct Emp {

int id;

float salary;

char name[10];

};

void main() {

struct Emp e1;

e1.id, e1.salary, e1.name

}

Limitation of procedure language

Data security

Re-usability

void mno() {

}

void xyz() {

mno();

}

void abc() {

xyz();

}

void main() {

abc();

}

OOPs : Object Oriented Programing system

object : object is any real world entity.

Properties or state-🡪 have -🡪 variables / fields

Person

Behavior --🡪 do/does 🡪 functions / methods

Bank

Car

Animal

Customer

Employee

class : blue print of object or template of object or user defined data type which help to create object.

syntax of class

class ClassName {

variable or field declaration;

methods or functions;

}

class App {

public static void main(String args[]) {

System.out.println(“Welcome to Java..”);

}

}

We need to save the program using ClassName.java

Please download java 8 or 11 version

class App {

public static void main(String args[]) {

System.out.println("Welcome to Java...");

}

}

Save the program App.java

javac App.java : compile the program

java App : run the program

class App {

public static void main(String args[]) {

System.out.println("Welcome to Java...");

System.out.println("Welcome to Java...");

System.out.println("Welcome to Java...");

System.out.print("Welcome to Java...");

System.out.print("Welcome to Java...");

System.out.printf("Welcome to Java...");

}

}

Variable : variable is name which hold value and value can change during the execution of program.

Data types: data type is a type of data which tells what type of data it can hold.

It divided into two types.

1. primitive data types : this data types is use to store only value

8 types

1. byte 1byte
2. short 2 byte
3. int 4 byte
4. long 8 byte : without decimal
5. float 4 byte
6. double 8 byte : with decimal
7. char 2 byte : single character
8. boolean 1 bit : true or false.
9. non primitive data types : this data type is use to store value as well as reference of another data types.

4 types

array

class : can pre defined or user defined

interface can pre defined or user defined

enum can pre defined or user defined

type casting :

Converting one data type to another data type is known as type casting.

2 types

Implicit type casting:

Explicit type casting:

------------------🡪 implicit --------------🡪

byte short int long

🡨------------ explicit ------------------------

-----🡪 implicit----------🡪

int float

🡨------explicit ---------

Operator :

Arithmetic operator : +, -, \*, /, %

Conditional operator or relational : >, >=, <, <=, ==, !=

Logical operator : &&, ||, !

Assignment operator : =

Increment and decrement : ++, --

Bitwise : &, |, ^

instanceOf

if statement

1. simple if

if(condition) {

}

1. if else

if(condition) {

}else {

}

1. nested if

if(condition) {

if(condition){

}else {

}

}else {

}

1. if else if

if(condition) {

}else if(condition) {

}else if (condition) {

}else {

}

1. switch statement

in switch statement use can take the decision which block we want to execute

syntax

int choice=1;

switch(choice) { // variable type can be int or char or string

case 1:block1;

break;

case 2:block2

break;

case 3: block3

break;

default : default block

break;

}

Taking the value through keyboards in java

using Scanner class.

Scanner is a pre defined class part of util package. Package is a collection of classes and interfaces.

We need to create the Scanner class object

Scanner sc = new Scanner(System.in);

Once we created Scanner class object we will get the error because Scanner is part of util package.

import java.util.Scanner;

looping :it is use to execute the task again and again till the condition become false.

while loop

do while loop

for loop

initialization : start and end position

condition : if true it do the task.

Body of the loop

Increment or decrement

for each loop or enhanced loop : is use to retrieve the value from array or collection of classes.

Day 2 : 15-11-2022

array : array is a type of reference data type which is use to store more than one value of same types.

int a=10;

a=20;

syntax

int abc[];

int xyz[]={10,20,30,40,50};

we can retrieve the value from array index position start from 0.

xyz[0];

creating memory for the array

int data[]=new int[10];

**OOPs concept using Java**

object : object is any real world entity

class : blue print of object or template of object

class name must be follow pascal naming rules.

1. If class name one word. The first letter of class start with upper case.
2. If class contains more than one word then each word first letter.

Variable name and method name must be follow camel naming rules.

1. If variable or method name one word then we have to write in lower case.
2. If variable or method name contains more than one word then from second word onward each word letter must be upper case.

Types of variable or fields.

In Java variable are divided into 3 types.

1. Instance variable:
   1. The variable which declared inside a class but outside method is known as instance variable.
   2. Instance variable hold default value according to their data types.

int family ->0

float family 🡪0.0

boolean 🡪 false

char 🡪space

String 🡪 null;

* 1. Instance variable we can use inside all method directly but method must be part of same class and it must be non static method.

1. Local variable
   1. The variable which declared inside method including main method is known as local variable.
   2. Local variable doesn’t hold default value we have to initialize.
   3. Scope of the variable within that block where it declared.
2. Static variable

Constructor : it is a type of special method which help to create the memory (heap memory).

Pts

1. Constructor have same name as class itself.
2. Constructor doesn’t contains return type not even void also.
3. Constructor no need to call it will call automatically when we create the object.
4. If we not write any constructor by default JVM provide default constructor. Default constructor is always empty constructor.
5. But if we write explicitly empty or parametrized constructor then JVM doesn’t provide any default constructor.

In the life of the object if we want to perform any task only one time that type of task we need to write inside a constructor it can be empty or parameterized.

If we want to do the task more than one time that type of task we have to write inside a method.

Encapsulation : Binding or wrapping data (variables ) and code (methods) in a single unit is known as Encapsulation.

Class is good example for Encapsulation.

class Employee {

String name;

float salary;

void display() {

}

}

this keyword :this is a keyword which refer to current object.

when local variable and instance variable have same name then local variable hide the visibility of instance variable. So if we want to refer to instance variable then we have to use this keyword.

Inheritance : Inheritance is use to inherits or acquire the properties and behavior of old class to new class.

class OldClass { // super class, base class or parent class

properties

behavior

}

class NewClass extends OldClass{// sub class,derived class or child clas

properties

behavior

}

Types of inheritance

1. Single inheritance : one super class and one sub class

class A { }

class B extends A { }

1. Multilevel inheritance : one super class and n number of sub classes connected one by one

class A { }

class B extends A {}

class C extends B{}

class D extends C {}

1. Hierarchical inheritance : one super class and n number of sub classes connected directly to super class.

class A { }

class B extends A { }

class C extends A { }

1. Multiple inheritance : more than one super class and one sub class

class A { }

class B { }

class C extends A,B{ } wrong in Java

Java doesn’t support multiple inheritance using class. It can support using indirectly using interface but not with class.

OOPs relationship

1. is a
2. has a

Day 3: 16-11-2022

OOPs relationship

1. is a relationship
2. has a relationship

class Employee{

id,name,salary

}

class Manager extends Employee{

numberOfEmp

Address add = new Address();

}

class Programmer extends Employee{

projectName

}

class ProjectManager extends Manager{

clientId;

}

class Address {

cit,state etc

}

Manager is a Employee

Programmer is a Employee

ProjectManager is a Employee

Employee/Manager has a Address

Has a relationship : inside one class we are creating object of another class.

Has a relationship types

1. association
2. aggregation
3. composition

class A {

B obj1, obj2, obj3; 0 or 1 or many

}

class B {

A obj1, obj2, obj3; 0 or 1 or many

}

If we want to achieve has a relationship any one of the side we need to create at least one object then we can say has relationship.

class Manager {

Address ladd = new Address(); 0 or 1 or many

Address padd = new Address();

}

class Address {

city, state

}

This is also type of association but it is known as weak association. Weak association is known as aggregation.

class Student {

StudentHistory sh = new StudentHistory();

}

class StudentHistory {

}

This is also type of association but it is known as strong association. Strong association is known as composition.

Polymorphism : one name many forms.

2 types

1. compile time polymorphism or static binding or early binding

Example : Method overloading : method have same name but different parameter list ie type of parameter list or number of parameter list is known as method overloading.

class Operation {

read() {}

add() {}

abc() {}

display() {}

void add(int x, int y) {x+y}

void add(int x, int y, int z) {x+y+z}

void add(float x, float y){x+y}

void add(String x, String y){x+y}

}

1. run time polymorphism or dynamic binding or late binding

Example : Method Overriding : The method have same name and same method signature (number of parameter list, type of parameter list and return type must be same).

To achieve method overriding we need inheritance concept.

Annotation : annotation is known as meta-data. Data about data.

Java provided lot of pre defined annotation as well as we can create user defined annotation.

All annotation start with pre-fix @ followed by annotation name

If annotation we can use on class level or method level or property level.

@Override annotation we can use on those method only which are methods are override.

Java Non access specifiers

abstract, static and final

abstract : abstract is keyword we can use with method and class but not with variable.

1. abstract method : the method without body or incomplete method or without curly braces is known as abstract method

abstract void speed();

1. if class contains one or more abstract method then that class we need to declare as abstract class.

abstract class Bike {

}

1. Which ever class extends abstract class that class must be provide for all abstract method belong to that class.

That class can ignore if that class itself is an abstract class.

1. We can’t create abstract class object.
2. Abstract class can contains normal as well as abstract method.

It can contains zero or 1 or all abstract method.

1. class can be abstract but no abstract method.
2. abstract class can contains default as well as we can write parameterized constructor(this constructor is use to set the value for instance variable).

static

1. static keyword we can use with variable and method but not with class (we can use static keyword with class but class must be inner or nested class). Outer class we can’t use static keyword.
2. Static variable :if variable is static we can access or assign the value for that variable using class name.
3. Static method : if method is static we can call that method with help of class name object not required.
4. Even though we can assign the value for static variable with object also as well as we can call static method with the help of class name.
5. Inside non static method we can access static as well as non static variable directly but inside static method we can access only static variable directly of that class.

heap memory

static memory

every class we will get only one static memory

Employee

Id,name,salary - 🡪 instance variable

MgrId,ClientId,ProjectId; 🡪 static

Static is like a global to all object.



Day 4 : 17-11-2022

final keyword :

1. Final keyword we can use with variable, method and class
2. final variable : we can use final variable to declare constant in java.

final int A=10;

A=20; //Error

1. final method : if method is final we can’t override that method but we can use it or we can call it in sub class.
2. final class : if class if final we can’t inherits or extends that class.

interface : interface is a type of reference data types which is also known as 100% pure abstract class.

Syntax to create interface

interface interfaceName {

fields;

methods;

}

By default all fields in interface are public static and final

By default all methods in interface are public and abstract.

interface Abc {

public static final int A=10;

public abstract void dis1();

}

interface Abc {

int A=10;

void dis1();

}

interface Xyz {

int B=20;

void dis2();

}

interface Mno extends Abc,Xyz{ // multiple inheritance.

int C=30;

void dis3();

}

class Info implements Abc,Xyz {

need to provide body for dis1 and di2

}

Like a class one interface can extends another interface. But interface can extends more than one interface.

Class always implements interface. Class can implements more than one interface. Which ever class implements any interface one or more than interface must be provide the body for all abstract method belong to that interface.

Access specifiers while overriding interface method

Super class / interface Sub

public public

protected public

protected

default (nothing) public

protected

default (nothing)

private we can’t override method

difference between interface and abstract class.

1. Interface contains only final variable but abstract class not mandatory.
2. Interface contains only abstract method but abstract class not mandatory it can contains abstract as well as normal method.
3. Interfaces doesn’t contains default constructor as well as we can’t write parameterized constructor. But abstract class can contains.
4. We can implements more than one interface but we can extends only one abstract class.

Common points

1. We can create the object of interface as well as abstract class.
2. Whichever class extends abstract class or implements interface that must be provide the body for all abstract method belong to that interface or that abstract class mandatory.

Using abstract class we can achieve partial abstraction but using interface we can achieve 100% abstraction.

Abstraction :hiding the internal implementation without knowing background details.

Run time polymorphism using object creation.

this keyword, super keyword, this(), super()

this() and super() :these two use to do constructor code re-usability.

this() : it is use to achieve constructor chaining or calling the same class constructor. this() must be inside a constructor and it must be first parameter inside a constructor.

super(): it is use to achieve constructor chaining from sub class constructor to super class constructor. By default every sub class constructor contains super() parameter which always call super class empty constructor. And it must be first parameter inside a constructor.

Day 5 : 18-11-2022

EmployeeName\_EmployeeId\_Week1.zip

Access specifiers : java provided different types of access specifiers which help expose the visibility of classes, interfaces, variable and methods.

Four types of access specifiers

1. **private**

We can use private with : instance variable, static variable, non static method, static method, constructor but can’t use local variable and class(outer class).

**scope** : within a same class.

1. **default (nothing)**

We can use default with all.

**scope** : within same package other package we can’t access.

1. **protected**

We can use protected with : instance variable, static variable, non static method, static method, constructor but can’t use local variable and class(outer class).

**scope** : within a same package other package if class is sub class or inherits class.

1. **public:**

We can use with call but not with local variable.

We can use with class but in one editor or file if we write more than one class we can use for only one class as public other class must be default.

**scope** : same package as well as other package.

package : package is known as collection of classes and interfaces.

When two classes or interfaces which have same name but different purpose use then using package concept we can avoid the conflict.

Package is like directory or folder.

Package are divided into two types.

1. User defined package or custom package
2. Pre defined package or built packages.

education

school college

Attendance.java Attendance.java

Syntax to create the package

package packagename;

import packagename.subpacakge.\*;

import packagename.subpackage.ClassName;

pre defined package

pre-defined package

java -----🡪 root package javax -🡪root package

lang swing

io servlet

util ejb

sql sql

net net

etc

by default every java program imported lang package.

import java.util.\*;

import java.util.Scanner;

import java.\*; wrong

Scanne sc = new Scanner(System.in);

Lang package classes and interfaces

Exception, Error and more types of exception classes.

Thread , Runnable

String

StringBuffer

StringBuilder

Object -🡪 by default every java program it may be pre defined or user defined extends Object.

Class : name of class itself is Class.

Wrapper classes

Cloneable interface. : this interface is use to create Clone or duplicate object.

System

Day 6 : 21-11-2022

**Exception handling :**

Exception is a pre defined object or memory which occurs when unexpected or abnormal conditions occurs during the execution of a program. Using some technique we need to handle exception that is known as exception handling.

Java

Compile program run the program

javac : compiler java : interpreter

compile time error run time error

syntax error

or typo error error exception

run time error

Object

Throwable

Error Exception

Both are pre defined classes part of lang package. Error and Exception extends Throwable class. Throwable class internally extends Object.

Error : The error which generate at the run time which we can’t handle it. Ex: JVM crash, software or hardware etc.

Exception : exception is type of run time error which we can handle it. Ex: divided by zero.

Exception

CheckedException UncheckedException

IOException RuntimeException

SQLException

ArithmecticException

ArrayIndexOutOfBoundsException

All unchecked exception are sub class of RuntimeException.

To handle both the type of exception java provided 5 keyword.

1. try
2. catch
3. finally
4. throw
5. throws

Unchecked exception

try catch block (try with single catch block)

try {

}catch(Exception e) {

}

Try with single catch block

If we want do common task if any exception generate then we can use try with single catch.

If we don’t know which code generate what type of exception.

Try with multiple catch block :

Base upon exception we want to execute set of code and we know which code generate type of exception then we need use try with multiple catch block.

finally block

This block execute 100% sure if any exception generate or not. It will execute doesn’t matter any exception generate or not.

Catch block execute only if any exception generate. It is known as exception handler.

In try block we need to write the code which may generate exception.

It is use to close the resources like file handling, jdbc connectivity etc.

try

catch catch catch catch finally

catch finally catch

catch finally

try{

open file

read and write operation

}catch(Exception e) {

}finally {

close the file

}

throw : This keyword is use to throw checked or unchecked user defined or custom as well as pre defined exception base upon our conditions. With help of throw keyword we can raise or generate any exception

syntax

throw new Exception();

or

throw new ExceptionSubClass()

throws : this keyword is use to throw checked or unchecked exception to caller method.

This keyword we use with method signature.

void display() throws Exception, ExceptionSubclass {

}

Checked exception check twice one at compile time and another run time.

Unchecked exception we can avoid some extends but checked exception we can’t avoid because we can’t compile the program.

We need to handle the checked exception mandatory using try – catch or throws.

Introduction to Multi threading

**synchronization**

program : set of instruction to perform a specific task.

process : time taken to execute the code.

processor : processor is responsible to execute the code.

thread : thread is a small execution of a code within a process.

By default java is thread base programming language.

Thread t = Thread.currentThread();

System.out.println(t); Thread[main,5,main];

Main 🡪 name of the thread

5 🡪 priority of the thread

Main 🡪 group of thread

Inside a main method by default one thread always execute to that thread details we can take the help of currentThread.

Multi tasking using two ways

1. using process base
2. using thread base

In Java we can create user defined thread using two ways

1. Extends Thread class
2. Implements Runnable interface

Synchronization : it is a concept which help to allow to access all resource or particular resources for only one thread at time in particular period of time or it is use to block or lock the thread.

To achieve synchronization we need to use synchronized keyword.

This keyword we can use with method or inside a method we can use more than one synchronized block.

When method is synchronized work is thread safe but slow in performance.

Day 7 : 22-11-2022

String class : String is one character or more than one character enclose in double quote. In Java String is pre defined class or also known as reference. In Java String doesn’t end with null character.

Syntax to create the String class object.

String str1 = “Welcome to Java”; literal style object creation

String str2 = new String(“Welcome to Java”); using new keyword.

str1.

StringBuffer

StringBuilder

In String

== : it check value as well as reference code or memory code or hashcode

equals() : it check only value not reference code.



String is known as immutable class. We can’t change.

StringBuffer

StringBuilder

These two classes is known as mutable string class.

StringBuffer method are synchronized means they are thread safe but slow in performance.

StrinigBuilder method are not synchronized so they are not thread safe but performance wise fast.

Wrapper classes : Java provided totally 8 types of wrapper classes which help to convert primitive to object and vice-versa.

Primitive data types Wrapper classes

byte Byte

short Short

int Integer

long Long

float Float

double Double

char Character

boolean Boolean

byte a=10;

int b=20;

Integer c = new Integer(b); : converting primitive to object.

int d = c.intValue(); : convert object to primive

float e = c.floatValue();

JavaBean class :

According to Java bean class

1. Class must be public
2. All variable must be private and for each variable we need to provide setter and getter method.
3. Setter method name start with pre-fix set followed by variableName like setId
4. Getter method name start with prefix get followed by variable name like getId
5. Setter method is use to set the value with condition base upon requirement.
6. Getter method is use to get the value.

This class is known as pure encapsulation class.

Whenever we display any user-defined class reference using println internally it will to toString() of object class and that method return as a string ie [packageName.className@code](mailto:packageName.className@code).

But if we need proper output then we need to override toString method belong to object class.

equals() method is part of Object class. String class as well as all wrapper class override equals() method. They provided the logic how to compare to String as well as wrapper class reference.

When we check user defined class reference with equals method it will call equals() method of object. that equals method return false by default.

Now we need override equals method and inside that method we need to provide logic for equal when two object are equal base up property like id or name or salary or all

Clone : clone is use to create the duplicate object.

To create the clone java provided clone() method this method is part of Object class.

Which class object we want to create clone that class must be implements Cloneable interface. This interface is part lang package. This interface is known as marker interface. Marker interface means this interface doesn’t contains any method or zero methods.

Day 7 : 23-11-2022

IO : Input and Output : In Java We can do IO operation using stream.

Stream means flow of data or it is abstraction between source and destination.

Stream

byte char

Input Output input output

InputStream OutputStream Reader Writer

These are abstract classes part of io package.

DataInputStream DataOutputStream InputStreamReader OutputStreamWriter

FileInputStream FileOutputStream FileReader FileWriter

ObjectInputStream ObjectOutputStream BufferedReader BufferedWriter

PrintStream PrintWriter

Scanner sc = new Scanner(System.in);

System.in()

System.out.println(“”);

System is a class which contains three reference ie in, out and err

In is reference of InputStream

Out and err is reference of PrintStream

These 3 are property of system.

InputStream is = System.in; InputStream always refer to standard input device ie keyword.

PrintStream ps = System.out; PrintStream always refer to standard output device ie console

Byte wise

Source : keyword

Destination : console

But if we want to store primitive data types like id,name,salary,desg etc.

FileInputStream, DataInputStream, DataOutputStream,FileOutputStream

So if want to store the object.

Object Serialization : object contains three things

Identity ie reference of that object

Property : variable of that object

Behavior : functionality of that object

We can store only property not identity as well as functionality.

Storing the object itself or converting object into byte format is known as object serialization.

Object De-Serialization : converting byte format object back object format is known as object De-serialization.

Which class object we want to do serialization that class must be implements Serializable interface and it is a type of marker interface.

Day 9 : 24-11-2022

Collection Framework (Data Structure)

Variable

int a=10;

a=20;

array

int abc[]={10,20,30,40};

structure like In C or C++

Java doesn’t support structure

class Employee {

int id;

String name;

float salary;

}

Employee emp = new Employee();

emp.id = 100;

emp.name =“Ravi”;

emp.salary = 12000

array object

Employee emp[]=new Employee[100];

emp[0];

emp[99];

array which can hold primitive or user defined object.

limitation

fixed in memory size.

Array object doesn’t provide any pre-defined method which help to add, remove, search, iterate one by one.

Collection framework provide set of collection of classes and interface which help to add the collection of object or elements of any type (like int, float, char, string as well as user defined object).it provided lot of pre defined method which help to add, remove, search, iterate one by one very easily.

Collection hierarchy

All collection framework classes part of util package.

Collection --🡪 interface

Extends

List Set Queue Map🡪 interface

List : it maintain the order using index position. List allow duplicate.

List API (Application Programming Interface). It may classes and interface.

ArrayList

LinkedList

Vector

Stack

Set : it doesn’t allow duplicate. In Set few API maintain order, unorder or sorted.

HashSet

LinkedHashSet

TreeSet

Queue : Queue First in First Out operation we can do using Queue.

PriorityQueue

LinkedList

Map : it allow to store the data in key-value pairs. Key is unique and value may be duplicate.

HashMap

LinkedHashMap

TreeMap

Hashtable

Deque: it is a type of interface which extends Queue interface.

ArrayList : ArrayList is a type of list api we can add same as well as different types of values by default.

Normal Array is known as fixed memory size we can store same type of value. Adding and removing elements from normal array more complex.

ArrayList allow to store any types of values. It is known dynamic memory and we can add or remove elements very easily.

LinkedList : In Java LinkedList is a type of list API. Whenever we store the elements in LinkedList is use node concept.

LinkedList mainly divided into 3 types

Single LinkedList

Double LinkedList

Circular linked List

In Java by default Linkedlist consider as double linked list.

**Vector** : it is a type of legacy class. By default all method in Vector class are synchronized. Default size of vector 10. Vector once the size 10 cross it will increase by default 100% in ArrayList it will increase by default 50%

Stack : Stack is a type of data structure which help to do the operation as First In Last Out. Stack extends Vector. We can say Stack is a type of List API.

Queue : First in First Out

PriorityQueue : first in first out base up on priority

LinkedList : first in first out

Map : it allow to store the data in key-value pairs.

**HashMap :** Implements Map interface . it can allow null key but only one we can store more than one null value. In HashMap display element randomly.

**LinkedHashMap :** Is it sub class of HashMap as well as implements Map interface. It maintain the order. . it can allow null key but only one we can store more than one null value. In HashMap.

**TreeMap :** implements NavigableMap and this interface extends SortedMap interface. In TreeMap it display the element ascending order as a key. In TreeMap we can’t store key as null but it can allow null value. In TreeMap key must be same type because this class internally implements SortedMap interface and that interface provide sorting algorithm.

**Hashtable :** it implements Map interface. Hashtable is legacy API and all method in Hashtable are synchronized. In hashtable we can’t store null key as well as null value. It display the element in unorder.

**Set :** Set doesn’t allow duplicate. Set doesn’t provide index position.

Set API

HashSet : it implements Set interface and it hold element unorder. It allow null value as well as by default we can store any type of values.

LinkedHashSet: It extends HashSet interface and maintain the order. It allow null value as well as by default we can store any type of values.

TreeSet : TreeSet indirectly implements SortedSet interface. Because of this reason in TreeSet we have to store same type of values. It doesn’t allow null value.

Retrieving the value from collection API one by one

1. For each loop
2. Iterator interface : only forward direction
3. ListIterator : forward as well as backward direction.
4. Enumeration

Collection framework with generics

Day 10 : 25-11-2022

Collection framework with complex object.

Collection utilities classes

**Arrays**

**Collections**

Both these classes is known as utilities classes which prove set of static method help to do some operation in primitive array as well as list api.

Arrays which provide set of methods to do some operation on primitive array of int, char, float, double, string etc.

Collections.sort(listRef);

If that list reference hold primitive value as well as string value then we will not get any error.

Because by default all Wrapper as well as string class internally implements Comparable interface. Comparable is interface part of lang package. That interface contains compareTo method which is responsible to do sorting.

When we pass list reference sort method expect that class which hold by list must implements Comparable interface reference.

When we pass list reference and list hold the user defined class object then we will get the error because user defined class by default doesn’t implements Comparable interface. So to avoid the error no our user defined class must be implements Comparable interface and provide the body for compareTo method.

**Day 11 : 28-11-2022**

MySQL

show databases; this command is use to show all database present in our account.

use databaseName; this command is use to switch to existing database.

create database databaeName;

use databaseName;

show tables;

Oracle

In oracle database the login itself is consider as one database.

select \* from tab; is will show all tables present in current login account.

SQL (Structured Query Language).

DDL : Data Definition language

Employee

PK

Id name salary

Int string float

create table employee(id int primary key,name varchar(25), salary float);

**DML** : Data Manipulation language

Insert :

**insert into tableName values(v1,v1,v3);**

**delete query**

delete from tableName;

delete from tableName where clause

**delete from employee where id=4;**

update query

update tableName set columnName = value;

update employee set salary = 25000;

update employee set salary = 22000 where id=1;

update employee set salary = 24000 where salary = 12000;

update employee set salary = 32000 where name =’Raj’;

DRL or DQL : Data Query language or Data Retrieval language

select \* from tableName;

select columnName,columnName from tableName;

select \* from employee where id=1;

select \* from employee where name ='Mahesh';

select \* from employee where salary > 15000;

**JDBC :** Java Database Connectivity : JDBC provide classes and interfaces which help to connect any database like RDBMS (MySQL Or Oracle etc). after connect with help of Java program we can store, delete, update and retrieve records from table using Java technologies.

Steps.

1. We need to import the package

import java.sql\*; and import javax.sql.\*;

1. JDBC through check exception. So we need to write the code with exception handling ie try-catch or throws exception.
2. Load the Driver : Driver is a pre defined class provided by vendor whose database we are going to connect.

4 types driver

1. Type 1 driver or jdbc odbc bridge driver from Java8 onward removed.
2. Type 2 driver or jdbc native api driver
3. Type 3 driver or jdbc net protocol driver
4. Type 4 or jdbc thin or pure driver : it comes in the form of jar file. So need to download jar file and add in our project.

Class.forName(“driverName”)

Class is a predefined class names itself is a Class which contains forName static method which help to load the class or driver.

Driver dd = new DriverName();

1. Establish the connection :

Connection con = DriverManager.getConnection(url,username,password);

DriverManager is a pre defined class part of sql package. Which contains getConnection is static method. Which takes 3 parameter

1st URL

2nd username

3rd password

This method return type is Connection interface reference.

1. Statement : Statement is a pre defined interface which provide set of methods which help to insert, delete, update and retrieve records from database.

Syntax to create the reference of Statement interface.

1. DML Operation : Insert, Delete, Update

int res = stmt.executeUpdate(“DML Operation”);

return type is int if query executed successfully how many record get effect that return as a number.

1. DRL : select query

ResultSet rs = stmt.executeQuery(“select query”)

executeQuery() method return type is ResultSet interface reference. This reference point to before record if record present in table we need to use next() method to scan the value and then retrieve each cell value with respective their data types.

Statement and PreparedStatement

Both are interfaces which help to do some operation on tables.

In Statement is we execute any query again and again then each time query get compile in java side send to database and execute in database and we will get the acknowledgement.

In PreparedStatement query compiled only once and execute n number of times in database side.

PreparedStatement also known as pre compiled query.

So performance wise PreparedStatement is faster than Statement.

Using PreparedStatement we can pass dynamic value with help of parameterized query concept.

In PreparedStatement one reference we can use for only one purpose.

Maven tool : maven is known as build tool. Build tools means the tool is responsible to compile, run, creating jar, war or ear file, help to download the dependencies (external jar files), creating documentation for the project.

ANT tool

Maven

Gradle DevOps

NetBean IDE

Eclipse

Oxgen

Hellion

Indigo

Etc

MyEclipse

Inlijet J

RAD

JDeveloper

VS Code

All IDE support Maven as well as Gradle project.

Both tools follow their own project structure which support all IDE.

Downloading the external jar file.

Maven : is XML base build tool (eXtensible Markup Language).

Gradle : XML less build too ( we provide the project configuration details in build file in the form of JSON).

Maven provide us pom.xml file(Project Object Model) which contains complete details about our project as well as project configuration details.

So we will create Maven project through Command prompt.

Maven provide in build life cycle ( it will call automatically)

1. Default : it is responsible to build and deploy the project
2. Clean : clean old build project.
3. Site : it is use to create the document for the project.

Each build life cycle contains more than one phases and each phase is responsible to execute more than goal and each goal is responsible to execute specific task.

Maven goals

mvn validate : check pom.xl file syntax

mvn clean : it is uses to clean the project

mvn compile : it compile the project and it create target folder which contains all build files.

mvn test : it is use to test the project unit testing

mvn install : it is use to download the jar file. In local repository.

mvn package : it is use to create the jar file.

mvn archetype:generate

: this command use to create the project using command prompt.

Open the command prompt and write as

mvn archtype:generate

after download few plugin

then hit enter key.

Then once again hit enter key

Groupid is known as collection of more than one project.

ArtifactId is actual project name

Define value for property 'groupId': MyJavaProject

Define value for property 'artifactId': SimpleMavenProject

Define value for property 'version' 1.0-SNAPSHOT: : 1.0

Define value for property 'package' MyJavaProject: : com

Confirm properties configuration:

groupId: MyJavaProject

artifactId: SimpleMavenProject

version: 1.0

package: com

Y: :

Creating JDBC application with Maven project

Using Maven with the help of pom.xml file we can download required jar file.

Database Table

Employee 🡪 Id, Name, Salary (Column)

In Java Table must be map to Java class and that class is JavaBean (Entity).

In Application we need to create class with same name class table with variable name must be match with all columns name.

From Java 11 onward Java is not open source

1.8

11

Version

Java 8

**Day 30-11-2022**

In database Employee 🡪 Table

Id,name,salary

Table must be map to JavaBean class.

JDBC code always in DAO layer or class (Data Access Object).

EmployeeDao

CustomerDao

LoginDao

ManagerDao

To write business logic we need to use Service class.

If we write business logic and database logic in same class. In future if we want to change business logic base upon application requirement then database logic also get effected and vice-versa.

Resource class : this class is responsible to provide the resource details.

Resource details we can provide using

1. Using java classes
2. Using xml file
3. Using properties file

**Web Application**

Java

J2SE J2EE J2ME

JavaSE JavaEE JavaME

JSE JEE JME

With help of JSE Java Standard edition we can develop desktop or standalone application.

The application which running only one machine is known as desktop or standalone application.

AWS and Swing or JAXFX 🡪GUI (Graphical User Interface).

JDBC

MySQL Or Oracle

JEE : Java Enterprise Edition : it is use to create web application using Java technology.

https://[www.google.com](http://www.google.com) -🡪 URL

req(http)------------🡪

Client Server

🡨--------Res(http/https) html/html5

CSS/CSS3

Bootstrap

JavaScript

jQuery

JEE (Servlet/JSP/EJB)

Asp.net

Php

Python

Node

Servlet , JSP(Java Server Pages) and EJB(Enterprise Java Bean)

To write program using servlet, jsp and ejb we require server.

Server is mainly help us to deploy and run the application in their engine.

Mainly server are divided into 2 types

1. Web server : Tomcat (Apache company) and it is an open source server
2. Application server : Web Logic, JBoss, Glashfish etc.

In Servlet, JSP and EJB there is no main method. After created the application we deploy these application in server. So server contains container.

Container is a part of server or also known as engine. Container is responsible to load the classes, created the object, call the life cycle method and destroy the object. the complete execution of Servlet, JSP and EJB is taken care by container.

Container mainly divided into different types.

Web Container : if server is a type of web server it contains only one type of container ie Web Container. The Web container is responsible to execute servlet and jsp application.

EBJ Container if server is a type of application server it container different types of container like web container, ejb container , jms container etc. Web container part of application server responsible to execute servlet and jsp program and ejb container responsible to execute ejb program.

Application provide lot of great features. Connection pooling, thread management, resource management, security etc.

Servlet : Servlet is normal Java program which help to create dynamic web page on server side.

We need to create normal java class and extends or implements type servlet.

Servlet -🡪 interface

GenericServlet -🡪 abstract class

HttpServlet 🡪abstract class

init it will call only once at the beginning

service which contain two parameter request and response which help to take the request from a client and response is use to give response back to client.

destroy : it will only once at the last.

HttpServlet class provide the benefits of http protocol. This class provide some few extra method in the form of doXXX like doGet, doPost, doPut, doDelete. These method wrap service method.

<http://localhost:8080/ProjectName/URLPattern>

**Day 01-12-2022**

By default every form method is consider as get.

If method is get then it will send the data through URL using URL rewriting technique.

URL?key=value&key=value

In get method data is not secure.

Get method call doGet method of servlet program.

If we want to send the data using security then we need to use post method. If method is post then data will send through request body. And it will call doPost method.

Servlet provided pre defined API ie RequestDispatcher. RequestDisaptcher is an interface which provide set of methods which help to navigate from one page to another page.

RequestDispathcer rd = request.getRequestDispatcher(“path”);

Target page is servlet then path must target servlet page url pattern.

If target page is html or jsp then we need to provide pageName.html or pageName.jsp.

rd.forward(request,response);

we can see the output of target page only

Or

rd.include(request,response); if we use include we can see the output of source + target.

**JSP: Java Server Pages**

Limitation of Servlet

1. Servlet is normal java program if we do any changes in servlet program we need to re compile and re deploy the application in server.
2. In Servlet if we want to write any html or presentation logic we must be write inside pw.println(“<h1>Welcome to Servlet </h1>”); it is string consider. Java can’t understand html code. It must be in double quote.
3. Servlet is complex if we want to do any simple task using servlet. First we need to create normal java class then class must be implements or extends type of servlet then we need to override life cycle method or doGet or doPost method. Then create PrintWriter class object. and then we need to provide the servlet class configuration details using web.xml file or using annotation.

JSP (Java Server Pages). JSP is tag base object oriented scripting language which help to create the dynamic web page in server.

JSP provided lot of pre defined tags

1. Scripting tag
   1. Scriptlet tag

<% opening tag

Java coding…

%> closing tag

If we write any code inside a scriptlet tag it consider as the code inside a method like doGet or doPost.

* 1. Declarative tag

<%!

Variable declaration.

%>

* 1. Expression tag

<%= opening tag

%> closing tag

1. Jsp directive tags
2. Implicit object
   1. out : out is a pre-defined implicit object of PrintWriter class.
   2. request
   3. response
3. Jsp action tags

<jsp:forward

<jsp:include

**MVC : Model View Controller :**

MVC is a design pattern or architecture which help us to break code or classes base upon their functionality.

Model layer : Java Bean, Service, Dao and Resource layer.

Main class replace by Servlet : it is known a Controller layer

In core java project we are taking the value through keyboard in web application we are going to take the value through browser :

View : HTML or JSP.

If we write business logic in jsp ie normal business logic or database logic then data code not secure.

JSP is also type of servlet. So performance wise JSP is slower than servlet.

In JSP we can’t do code re-usability. JSP provide only include or forward so we can include whole page or forward whole page. We can’t include part of the page.

If we write business logic or database logic inside servlet in doGet or doPost method. That code we can’t do re-usability. doGet and doPost called by web container base upon URL path.

If we write database logic or business logic in servlet that code become local to that servlet that same code we can’t access in another servlet.

Model layer :

Form (html or JSP) ---🡪 JavaBean class ---🡪 Table

Username,password username,password username,password

Field variable column

Java Bean class

Service class

Insert and Retrieve operation using MVC style with help of Servlet, JSP and JDBC

If we want to share the value or object from servlet to jsp and vice-versa we need to use scope object concept.

1. request.setAttribute(“obj”,value);

request.getAttribute(“obj”);

Dao class

Resource class

JavaBean JavaBean

View ----🡪Controller ---🡪Service ---🡪Dao -🡪Resource

Day

12-05-2022

Spring Framework

Spring is open source layered architecture framework. Spring also known as onion architecture framework.

Framework : Framework contains lot of API (Application Programming interfaces). Which internally connected to each others. Framework follow standard. Design pattern. Design pattern is best practice or solution of repeating problem.

Core Java : creational design pattern, structural design pattern and behavioral design pattern.

Singleton design pattern : we want to create only one memory for that class or resource class.

If we develop any application using any framework. So Framework internally follow all design pattern base up type of framework.

Framework is implementation of design pattern. 70 to 80% task is taken care by framework. But framework in not final product is a protocol or template so we need to write 20 to 30% code to make final product.

JEE : Servlet, JSP and EJB

JSP (View )-------------🡪Controller (Servlet )-------------🡪

Web Container will create the objet and maintain the object or life of jsp and servlet But web container not maintaining model layer (JavaBean, Service, Dao, Resource layer).

To improve model layer we need to use EJB.

Enterprise Java bean.

To run ejb application we need to ejb container. To get ebj container we need application server.

EJB is heavy and complex.

MVC

Model Layer -🡪 EJB Container

View 🡪 HTML/JSP

Controller 🡪 Servlet

Collection Framework

Struts : Struts is one of open source web framework provided by Apache. Which internally follow MVC. Internally they provided front controller design pattern.

EmployeeController, Employee,EmployeeSerivce, EmployeeDao modules

FrontController -🡪 it is a type of servlet which control all controller.

ActionController

Struts is known as controller centric framework.

JSF : JSF is open source web framework provide by oracle. JSF is known View Centric framework.

Angular Framework or React JS Vs JSF

Hibernate/JPA : ORM (Object Relation Mapping ): this framework is use to improve DAO layer.

Spring is open source layered architecture framework. Spring provided lot of modules which help to improve all layer in application.

Spring code

Spring context

Spring MVC Model centric framework

EJB Vs Spring MVC

Spring DAO : Data Access Object

Spring ORM Hibernate or JPA or iBaties

Spring AOP Aspect Oriented Programming

Spring boot

Spring cloud

Spring security

Spring integration

Etc

IOC : Inversion of Control

DI : Dependency Injection

IOC : It is a concept. IOC is a design pattern or programming rather than we are creating any resource or maintaining any resource explicitly allow to create by container. If container will create it will maintain properly. Resource means object creation, database connection, security, file handling etc. Servlet, jsp, strut, jsf these object creation taken care by container and that container is web container and it is part of server. Normal class those container doesn’t create or maintain the life. Those container are part of server.

Spring framework creating and maintain the life of object for normal class ie POJO class. Plain Old Java Object. that class not to extends or implements any pre defined class.

Spring framework container is part of jar file with xml file configuration or annotation.

DI : DI is implementation of IOC. DI is a implementation of IOC. We can achieve DI in Spring framework using two ways

1. constructor base DI
2. setter base DI

If I am a container I will inject dependency using xml or annotation with constructor or setter base. As developer we need to pull it use it and leave it. Container maintain the life.

We need to configure using xml

Using annotation

With few jar file we need add it.

Auto wired: autowired is a spring framework features which help to inject complex property implicitly rather than explicitly using property ref or constructor ref. Using auto wired spring framework scan di for complex property.

If any POJO or any other layered class contains primitive property spring framework do di implicitly. If class contain complex property then we need to do di explicitly using property ref or constructor ref.

Auto wired we can use byType and byName

In byType in beans.xml file we need to provide only one DI for that class.

In byName we can provide more than DI for that class. In this option we need to map id name class reference name must be match.

DI using annotation with partial XML file as well as no xml file

@Component :generic annotation we use on pojo class.

@Autowired : to do auto wired for complex object.

@Scope : provide the scope for object ie singleton or prototype

@Value : this annotation is use provide default value for primitive data types.

By default @component annotation is not enable so we need to enable using xml or using java class with few more annotation.

Spring Framework with DAO Layer (going to improve model layer ) JavaBean class, Dao class, Service class and resource class.

First we need to properties tag to change the java version

3 dependencies

Spring Core

Spring Context

Spring jdbc

Spring Framework provided pre defined class ie

DriverManagerDatasource part of spring jdbc jar file which help to provide database connectivity using DataSource features.

DataSource : provide the database connectivity in a proper manner from application server before spring framework.

Now way day we can get same features from spring framework without depending upon any external server with help of few jar files.

@Service : this is known as Service layer specific annotation.

@Repository : This is known as Dao layer specific annotation

To improve DAO layer

1. Core JDBC
2. JdbcTemplate :
3. ORM (Object Relationship ) : Hibernate /JPA
4. Spring Data

Spring MVC with Spring boot

Spring boot with MVC to creating REST API

Spring framework and Spring boot

**Day 07-12-2022**

In DAO layer we use Core JDBC

But spring framework provided one of the pre defined API ie

JdbcTemplate : JdbcTemplate wrap core jdbc and provided extra method to improve jdbc coding.

Spring DAO layer

public static void dis(String args[]) {

}

String abc[]={“A”,”B”,”C”}

dis(abc);

Using var arguments

Public static void dis(String … args ) {

}

dis();

dis(“A”)

dis(“A”,”B”);

dis(“A”,”C”,”D”);

ORM : Object Relationship Mapping. ORM is a concept which help to interact with database through programming language like Java, Asp.net etc.

Limitation of JDBC.

1. Using jdbc we can’t store as well as we can’t retrieve java object. So we need to convert java object into sql and vice-versa in DAO layer.
2. JDBC use sql language and SQL is database language.
3. JDBC through checked exception so we need to uses exception handling mandatory which writing jdbc code in dao layer. Exception hierarchy provided by jdbc is database dependent.
4. Jdbc doesn’t support any relationship like is a as well as has a relationship.

According to ORM we need table and table must be map to JavaBean class mandatory. That java class in ORM is known as entity class.

Object Relation

class Employee { Employee

id ID

name NAME

salary SALARY

}

Mapping

Employee -🡪Employee

Id –ID PK

Name –NAME

Salary –SALARY

Old Version we are doing mapping using xml

New version we are doing mapping using annotation

Configuration : this file contains database details

Like drivername, url, username password, mapping file details.

In Java We can achieve ORM using two ways

Hibernate : Hibernate is a framework. Part of JBoss people. Hibernate is a implementation of JPA.

JPA (Java Persistence API ) : JPA is technologies. It is a part of Java people. It is a type of EJB. JPA is a specification as well as implementation.

Servlet and JSP with Hibernate

Servlet and JSP with JPA.

Spring framework with Hibernate as well as JPA.

(Spring ORM).

In Spring boot we have to use only JPA no hibernate because in Spring boot hibernate is deprecated.

ORM using JPA using annotation

Using ORM we can create the table also possible.

Using Java classes we can create tables or using tables we can create entity classes.

Step 1: Employee (id (PK), name, salary)

We will create the Maven project.

To provide database details in JPA we need to create persistence.xml file

To provide database details in Hibernate we need to create hibernate.cfg.xml file

@Entity : This annotation we have to use on java bean class to map to table.

@Id : This annotation we have to use on that variable for map to column with primary key.

By default ORM not auto commit.

Commit and rollback

DML Operation : Insert, Delete and Update : if we do this operation in database if everything done correctly we can save commit. (save permanently. ). If anything go wrong we can say rollback.

TCL (Transaction control language) commit, rollback etc.

Update account set amount = amoun- 500 where accno=1;

Update account set amount = amoun + 500 where accno=8;

Rollback.

If we do any DML Operation through JDBC by default jdbc to auto commit.

But if we want to do transaction through jdbc we have to have to enable transaction.

con.setAutoCommit(false);

but in ORM by default not auto commit.

**Day**

**08-12-2022**

ORM (JPA ) Provided JPQL (Java Persistence Query Language)

SQL (it is database dependent)

Select \* from employee; employee table name

Select \* from employee where salary > 12000;

JPQL (it is database independent).

Select e from Employee e Employee is entity class name and e is reference name

Select e from Employee e where e.salary > 12000;

JPA Relationship

4 types of relationship

1. One to one -🡪 Person -🡪 Passport PK (one) FK(many)
2. One to Many 🡪 Trainer -🡪 Student
3. Many to One 🡪 Employee 🡪 Project
4. Many to Many 🡪 Students 🡪 SkillSet

One to Many Relationship

Trainer and Student

Trainer

PK

TID TName Tech

100 Raj Java

101 Ravi Python

102 Ramesh Angular

Student

PK FK

SID SName Tech TSID

1 Seeta 21 1

2 Reeta 22 1

3 Meeta 23 2

4 Veeta 24 null

create table trainer(tid int primary key,tname varchar(20), tech varchar(20));

create table student(sid int primary key,sname varchar(20), age int, tsid int, foreign key(tsid) references trainer(tid));

PK : if column is primary key that column doesn’t allow duplicate value as well as null value.

FK : foreign key is use to refer to another or same type primary key. If column foreign that column allow only does value which present in primary key it can allow null value.

**Join**

**Without alias**

select tname,tech,sname from trainer inner join student on tid=tsid;

**with alias**

select t.tname,t.tech,s.sname from trainer t inner join student s on t.tid=s.tsid;

Day

09-12-2022

Spring MVC : Spring MVC (Model View Controller) which help to create dynamic web application. Spring MVC follow mvc design pattern. It provided front controller is DispatcherServlet. Which responsible to keep the tack about all controller flow.

Spring MVC Flow



View (HTML /JSP)----🡪 From view request will send to application 🡪 In Web.xml or java classes we need to configure front controller ie DispatcherServlet. Dispatcher Servlet receive the request and it load the spring configuration file. Old Spring spring framework we are using Handling mapping classes. Which are responsible to map the request. But in New version we need to enable controller classes with help of component scan tag.

In Spring MVC from version 3.x

We have to make normal class with @Controller annotation

@Controller

class MyController {

@RequestMapping(value=”Hi”)

public ModelAndView sayHello() {

// logic or call service layer

ModelAndView mav = new ModelAndView();

mav.setViewName(“display.jsp”); //RequestDispatcher

return mav;

}

}

Inside this class we can write user defined method and return type of method methods can be ModelAndView or String. ModelAndView is a pre-defined class which will keep the track about the model and view layer. Then DispatcherServlet get the response from controller method and base upon request if we want to give the output in different format base upon client requirement. Then we need take the help of ViewResolver classes. After view conversion then DispatcherServlet pass the request to specific view.

In spring MVC spring configuration file name must be follow some rules ie

servletname-servlet.xml (servletname in servlet tag in web.xml)

12-12-2022

While developing the application using spring framework we need to do more configuration like downloading individual jar file with help of dependencies, maintaining xml file configuration.

Spring boot : Spring boot is bootstrap for spring application which help develop ready to deploy the application.

Spring boot =all spring modules – no xml file + few annotation + inbuilt or embedded tomcat server.

Spring boot is standalone application which help to develop any type of application. (Spring boot provide main method).

Spring boot with

Maven : if we use maven we need to use pom.xm file. Pom.xml is part of maven tool not part of spring boot.

Gradle : Gralle is another build tool which help to develop using java. Gradle doesn’t use build tool. In Gradle we can configuration details in the form of json.

Spring boot components

1. Spring boot starter : Spring boot starter provided one dependencies which help download more than one jar file base upon type of application we are developing.

Spring boot starter provided set of all jar file with only one dependencies base upon type application.

Spring boot starter (non web starter)

Spring boot web starter

Spring boot jdbc starter

Spring boot jpa starter

Spring boot testring

Etc

Spring boot auto configuration : this features provide us to add automatically configuration for our application base upon the jar file added in classpath or spring project.

Spring boot configuration is replacement of xml file configuration.

If we added web starter it will configure for use front controller ie DispatcherServlet. No need to configure any xml file.

If we do spring boot with DI for non web application.

Develop web application using spring boot.

@SpringBootApplication : This annotation we need to write on main class. This annotation provide benefits of those annotation like

@SpringBootApplication = @Configuration + @ComponentScan + @AutoConfiguration

To start spring boot application Spring boot provided pre defined class ie SpringApplication which contains run method and this is static method which help to start the spring boot application.

Spring boot internally provide the tomcat and that tomcat run on default port number 8080.

Spring boot provided configuration file ie application.properties or application.yml

So now we create application.properties file and their we can provide all configuration details.

Now we will create the Controller.

Spring MVC we create normal class with @Controller annotation and inside that class we written more than one method with @RequestMapping.

And that method return type is ModelAnView class return which help to redirect to jsp view page.

Web Service Giving the service for web application when both the application running using different technologies.

req Spring boot

Client Amazon payment paytm asp.net

Res gpay python

Phone pay node js

Credit spring boot

Net banking

Etc

Java java(req) asp.net

HDFC XML/JSON HSBC

Asp.net(res)

Xml : xExtensible markup language

JSON : JavaScript Object Notation

Employee emp = new Employee();

emp.setId(100);

emp.setName(“Ravi”);

emp.setSalary(12000);

emp

to convert this object in xml format

<Employee>

<Id>100</Id>

<Name>Ravi</Name>

<Salary>12000</Salary>

</Employee>

Json

{“id”:100,”name”:”Ravi”,”salary”:12000}

Web Service

Types of web services

1. SAOP Web Service : Simple Object Access Protocol

In SOAP web service we can consume and produce data only the form of XML.

Everything we need go convert in xml format.

1. RESTFull Web Service : Representation State Transfer : We can consume and produce the data in any format base upon client requirement.

Like xml, html, plain text, json as well as other format.

In Restfull web service we expose our resource in any format base upon client requirement.

Resource like a entity or object.

If our class is normal Servlet or Spring MVC class with @Controller annotation then our view must be html or jsp.

If class is normal servlet or Spring MVC @Controller then view tightly coupled with that technologies.

Now @Controller annotation replace by @RestController

If We write normal class with @RestController annotation then we can produce and consume any data from any technologies in the form of xml or json or plain text or html or any other format.

View for RestController can be angular or react js or javascript or python or asp.net or php or any rest client api.

Rest API rules

1. Get method : this method is use to get the resources like Employee, Customer, Manager, Product, Order etc.

Example : getEmployeeDetails in the form json or xml.

Get text message in plain text, html, xml

Get employee object : json or xml

When client or rest api client calling rest full web service if they want to pass some value.

We can use two concept

1. Query param
2. URL?key=value passing single value
3. URL?key=value&key=value&key=value :passing multiple values

By default html form with method internally use query param technique.

1. Path param

Using these technique we can pass the value for rest full web service through get methods.

1. URL/value1 passing single value
2. URL/value1/value2/value3 passing multiple values

13-12-2022

Creating REST API for Employee. These REST API developing using Spring boot with Database connectivity.

Connecting database using Spring data.

Spring boot web starter : it is responsible to take for web application using rest api.

Spring jpa starter : it is use to achieve ORM using spring jpa as well as spring data.

Mysql connector jar

Spring boot search configuration file with name application.properties or application.yml

Dao layer

We can write the code using JDBC

DataSource features using Spring framework

JdbcTemplate using spring framework

Jpa using Spring framework

Spring Data : Spring data is a module provided by spring framework which internally use jpa and they wrap jpa features and provided one interface ie JpaRepository and that interface contains all default implementation of all crud method on entity class.

Spring data abstract complete jdbc logic or jpa logic.

Post man is tool which help to test the rest api.

Post method : This method is use to store the data or create the resource.

14-12-2022

{

    "id":100,

    "name":"Ravi",

    "salary":12000

}

In json we have to pass the data in curly braces. In the form of key – value pairs

Key must be in double quote. Value can be number, string, boolean, array, complex object. key name and variable name in Java class must be match.

Delete method : This method is use to delete the record or resource. Normally we delete record using primary key and that id we have to receive using path param.

Put and patch method : These method is use to update the existing resource.

Put : generally this method is use to update all property of existing object.

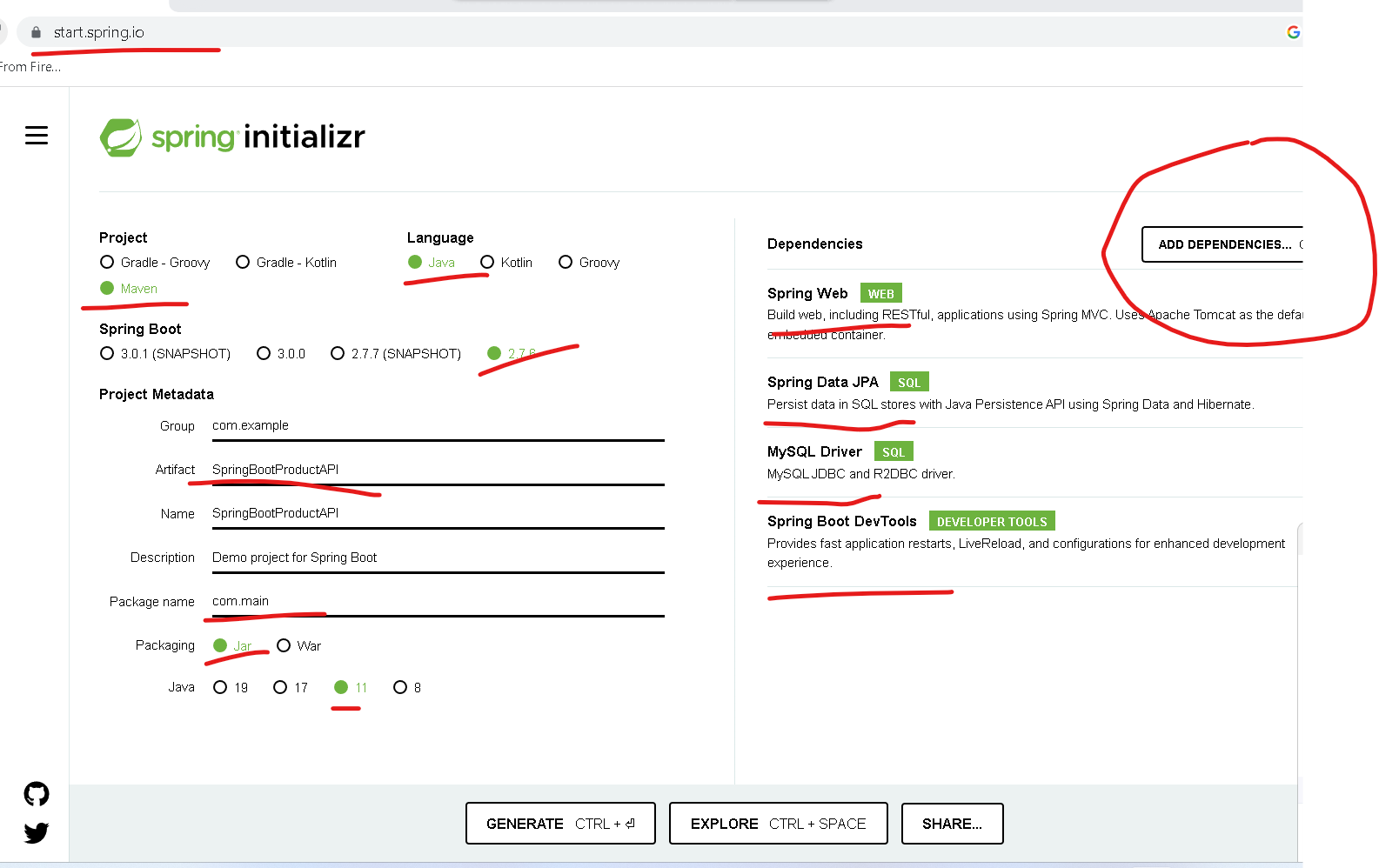
In employee class if we want to update all property like name and salary then use put method

Patch : this method is use to update partial object property .

In Employee class if we want to update may be salary or name then use patch method.

Spring framework provided spring initializer official website which help to create the project with all required starter.

<https://start.spring.io/>



Application.properties file

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/mydatabase

spring.datasource.username=root

spring.datasource.password=root@123

spring.jpa.hibernate.ddl-auto=update

spring.jpa.hibernate.ddl-auto this property help us to create the table in database base upon entity classes present in spring boot or orm.

@RequestMapping : This annotation we can use on method level as well as class level. If we use on class level to control the base path as well as we can use on method level with method attribute like get, post, put, delete and patch.

@GetMapping

@PostMapping

@PutMapping

@DeleteMapping

@PutchMapping

These annotation we can use on method level. And in these annotation no method attribute available because the name of the method itself consider as type of methods.

15-12-2022

@Controller

class MyController {

@RequsestMapping(value=”say”)

public @ResponseBody String sayHello() {

return “Welcome to Rest API”:

}

}

@ResController // @Controller + @ResponseBody

class MyController {

@RequsestMapping(value=”say”)

public String sayHello() {

return “Welcome to Rest API”:

}

}

Frontend technologies

JAX\_RS : Java API for XML (RestFull Web Service)

Making Servlet as a Web Service

Jersey plugin we can make servlet and jsp as RestFull Web Service.

Spring boot is rest full web service provider

Servlet and JSP we have to make rest consumer.

Spring boot can be consumer : RestTemplate help use to call rest full web service.

JSP As view

HTML,CSS and JavaScript :

fetch(“”): it is use to call backend technologies rest api.

In Angular HttpClient

In React JS axios or fetch

Cors : Cross Origin Resource Sharing

Lot of modules

LoginModule CustomerModule AccoutModule

ProductModule

After integrate all those module then we need to create jar or war file and deploy in production environment.

Git

Jar or war file

All these modules we can deploy independently we are developing using same technologies or different in same database or different database like oracle, mysql, db2, mongdb etc.

These all modules is known as micro service. These all modules or service can interact with each other using rest api.



Spring boot provided few modules like Spring cloud and Spring micro service

Spring boot provided one pre defined server eureka server.

So we need to create spring boot project with eureka server starter. And run project.

8176 default port number.

Eureka server help us to deploy more than one spring boot micro service project. In every project we need to add eureka client starter we need to add and enable that starter using some annotation.

rs.forward(req,res);

response.sendRedirect(“http://www.google.com”);

Trainer

PK

TID TName Tech

Student

PK FK

Sid SName Age TSID

Class Trainer {

Tid,tname,tech

List<Student> ts;

}

Class Student {

Sid,sname,age, tsid

}

19-12-2022



AWT and Swing

Monolithic architecture

: In this architecture we have develop whole application. While developing the application we required mode then one module. LoginModule, CustomerModule,ManagerModule, LoanModule, etc

Devops : Git repository

Build phase : combine more than one module and compile, run and create jar or war and deploy in server.

Micro Service is a concept. According to Micro service rather than creating whole application in one project and build and deploy all module. Create the n number of project as micro service project and inside each project develop the application for specific module. Those module or micro service project can be creating using java or any other technologies, database can be same or different and we can build and deploy independently.

To implements micro service in java we can use of the implementation ie spring boot.

Using spring boot we can create micro service project easily.

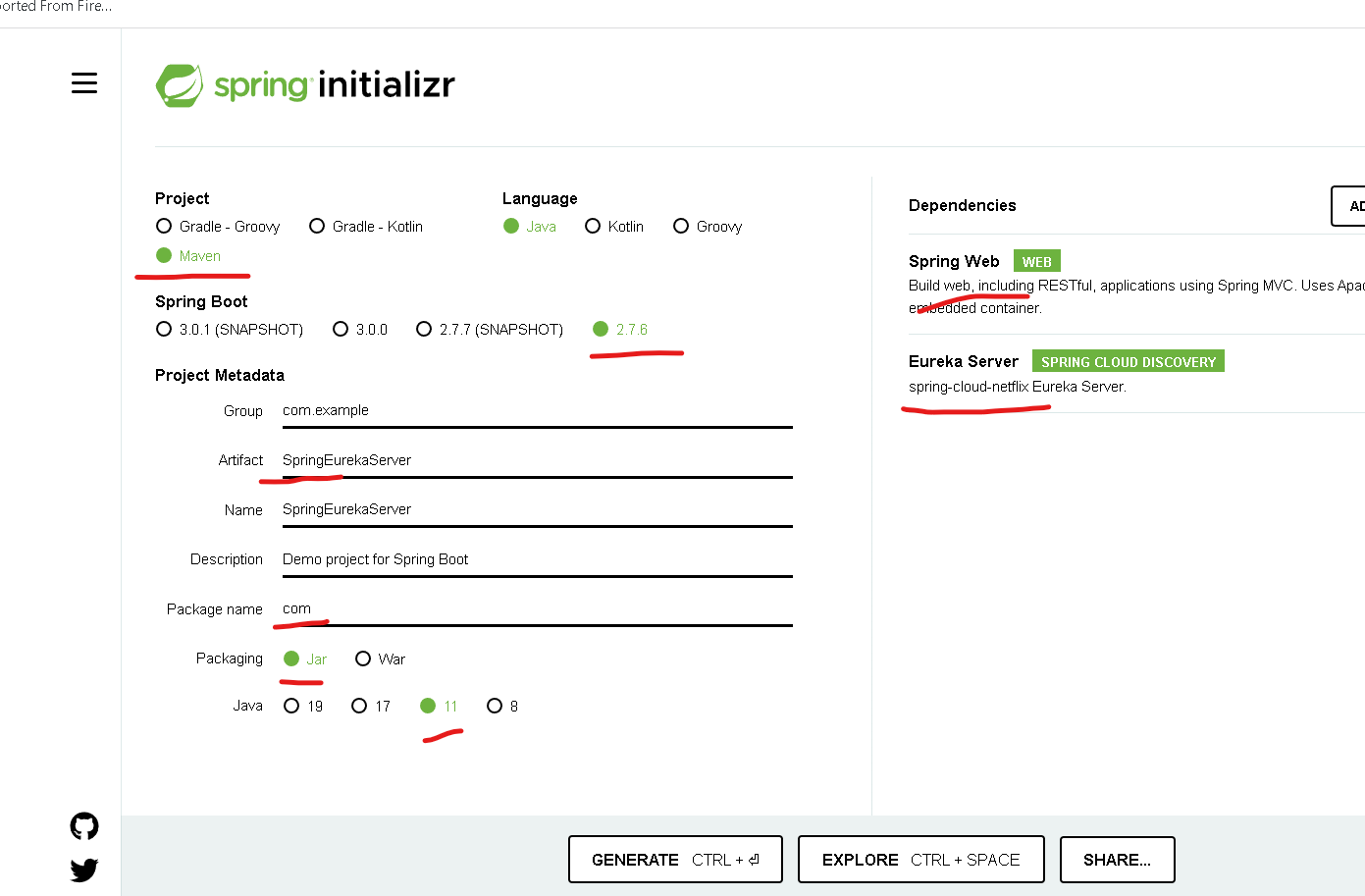
Spring boot and Spring cloud using these modules we can create micro service project using java technologies.

To deploy all micro service project spring cloud provided one of the open source server ie Eureka server. This Eureka server provide features to deploy spring micro service project.

We will develop micro service project using spring boot.

**First we will create Eureka Server project and we will run this project.**

**This project contains two starter ie 1. Web server 2. Eureka Server Starter**

****

@EnableEurekaServer: This annotation is use to enable eureka server in spring boot project.

**Spring boot we project contains by default tomcat server. That tomcat server by default run on default port number 8080.**

**Now eureka server also run on default port number 8080.**

**So whenever we create more than one micro service project and if we run those micro service project by default they project search eureka server on default port number 8761.**

**The project which contains eureka server in that project we need to change the port number from 8080 to 8761.**

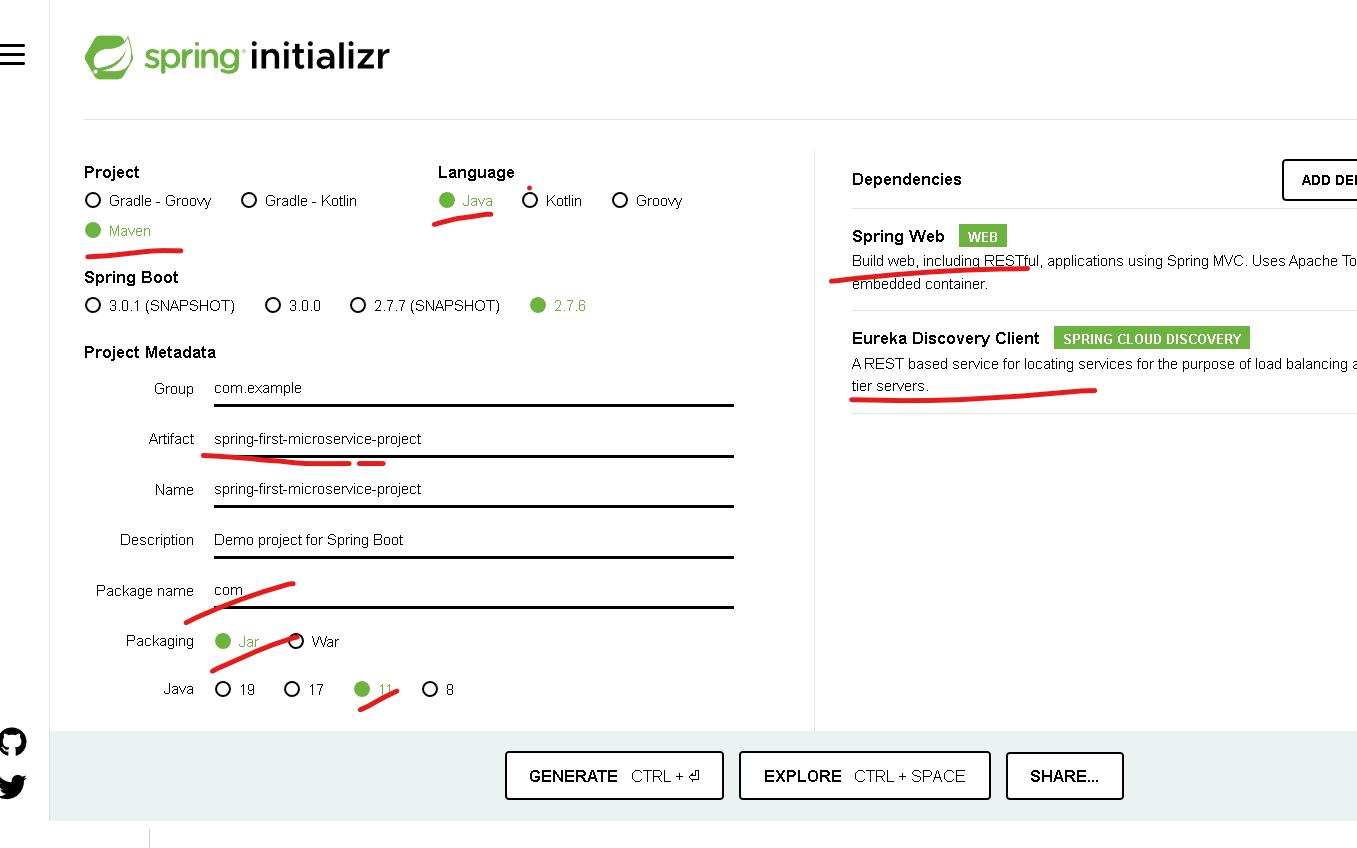
**To change the port number in application.properties.**

**Run the main class and check eureka server running or not**

[**http://localhost:8761**](http://localhost:8761)

**Now we will create the Spring boot micro service project.**

**First-micro-service**

****