**Day 5: 22-10-2025: OOPs concept and exception handling**

**Access specifiers and packages**

Java provided totally 4 types of access specifiers which expose the visibility or accessibility of class, constructor, variable and method within a same class, outside class, same package as well as other package.

1. private

visibility : within a same class

private we can use with instance variable, static variable, constructor, non static method static method but we can’t use with class as well as local variable.

1. default (nothing)

visibility : within a same package

default we can use with all.

1. protected

visibility : within a same package as well as other package if it is sub class.

protected we can use with instance variable, static variable, constructor, non static method static method but we can’t use with class as well as local variable.

1. public

visibility : within a same package as well as other package.

public we can use with all but can’t use with local variable.

package : package is a collection of classes and interfaces. Package help us to organized the classes and interfaces in proper manner. Using package we can create more than one classes as well as interface which have same name but different purpose. Package is like a directory or folder.

Package mainly divided into 2 types.

1. User defined package
2. Pre defined or built in package.

To create the package we need to use package keyword with name of the package.

package com;

**education**

**school college**

Attendance.java Attendance.java

**pre defined or built in packages.**

Java provided lot of pre defined packages.

Java provided mainly 2 root packages

java javax

root packages

lang sql

io net

util swing

sql servlet

net ejb

awt jms

etc etc

by default every java program imported lang package. without importing lang package we can use all classes and interfaces part of lang package.

by default every java program it may be pre defined or user defined extends Object class. Object is super class for all java classes.

class A {

}

class B extends A{

}

**Exception Handling:**

Exception is a pre defined class part of lang package. Exception is an object which generate occurs when unexcepted things or abnormal condition occurs during the execution of a program.

Using some technique you need to handle generated exception that is known as exception handling.

Java Program

Compile the program Run the program

javac java

compiler interpreter

compile time error

syntax error or typo

error

Run time error

Error Exception

Error : The error which generate at run time which we can’t handle it ie Error. JVM crash, software or hardware issue, out of memory etc.

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

Object

Throwable

Error Exception

Checked exception un checked exception

RuntimeException

IOException ArithmeticExeption

SQLException NumberFormatException

To handle both checked as well as unchecked exception java provided totally 5 keywords.

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

**unchecked exception**

**try catch block**

syntax

try {

try block

}catch(Exception e) {

catch block

}

In java we can write try with multiple catch block

Syntax

This code handle 3 types of exception and base upon exception it generate it execute that particular try block.

try{

}catch(ArithmeticException e) {

}catch(NumberFormatException e) {

}catch(ArrayIndexOutOfBounds e) {

}

Or

try{

}catch(ArithmeticException e) {

Specific

}catch(NumberFormatException e) {

Specific

}catch(ArrayIndexOutOfBounds e) {

Specific

}catch(Exception e) {

Generic exception

}

try{

} catch(Exception e) {

Generic exception

}catch(ArithmeticException e) {

Specific

}catch(NumberFormatException e) {

Specific

}catch(ArrayIndexOutOfBounds e) {

Specific

}

Always sub class must be on top and super class must be in bottom

try{

}

catch(SQLException e) {

} catch(RuntimeException e) {

}catch(Exception e) {

}

**Finally block:**

**try block**: the code which generate the exception. It may be one line code or multi line code you need to keep in try block.

**catch block:** this block execute only if any exception generate. No exception no catch block.

Finally is a block which execute 100% sure doesn’t matter exception generate or not.

try

catch catch catch catch finally

catch finally catch

finally

**throw** : throw keyword is use to throw or raise or generate pre defined or user defined exception with custom logic.

Syntax

throw new Exception()

or

throw new ExceptionSubClass()

**throws keyword :** throws keyword is use to throw the exception to caller method. throws keyword we use to method signature. It throw to caller method.

returnType methodName() throws Exception, ExceptionSubClass {

}

**Checked exception**

Un checked exception we can avoid with some extends. But checked exception you can’t avoid. It always check at compile time.

Checked exception check twice ie compile time as well as run time. checked exception we can’t avoid it we need to handle using try-catch or throws mandatory.

**Collection framework (Data Structure)**

int a=10;

a=20;

in java variable is use to store only one value. If store any other value previous value lost.

Array concept : in Array we can store more than one value of same types.

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

structure : it is use to store more one value of different or same types. C or C++ or C# support structure concept. But java doesn’t support structure.

Class : class is user defined data types which is use to store more than one value of same as well as different data types.

class Employee {

int id;

String name;

float salary;

}

Employee emp = new Employee();

emp.id=100;

emp.name=”Ravi”;

emp.salary=45000;

emp.id=200;

array object.

syntax

className objectRefName[]=new ClassName[size];

int abc[]=new int[100]; default value 0

Employee employees[]=new Employee[100]; default value is null

employees[0]=new Employee(); 1 employee object created and assign on 0 index position.

0 or 1 or 100 of type Employee;

It create 0 object of type of Employee it create array object created not employee object.

Service layer : which contains business logic.

add, delete, update, retrieve, sort, search etc.

**Business requirement for this project**

1. addEmployee()
   1. employee id must be unique
   2. employee salary must be > 8000 then only add else don’t add
2. deleteEmployee() :delete employee information using.
3. updateEmployeeSalary() search employee using and update salary If that employee present in array
4. searchEmployee() details using empId

etc

Limitation of an array of primitive or object types.

1. Array is known as fixed in memory.
2. It allow to store same data types values like int, float, string or user defined object.
3. It doesn’t provide any method which help add, search, update and retrieve very easily.

Collection Framework : Collection framework provide lot of pre defined classes and interfaces which help to store collection of object or element or data of any types like int, float, char, String or user defined objects.

It provided lot of pre defined methods which help to add, remove, search, iterate very easily.

**Collection Framework hierarchy**

Iterable -🡪 interface 🡪 lang package

Collection -🡪 Interface -🡪 util package

Set List Queue Map

All 4 are interfaces Set, List, Queue internally extends Collection but Map doesn’t.

**Set:** Set is an interface which is use to store collection of elements of any type. Set doesn’t allow duplicate value. Set doesn’t provide index concept. Under set few classes when we store it maintain the order or unorder or sorted by default ascending order.

**Set classes :** These classes directly or indirectly implements Set interface.

**HashSet :** HashSet super class.

**LinkedHashSet:** LinkedHashSet sub class which internally extends HashSet.

**TreeSet :** TreeSet internally implements SortedSet interface and that interface extends Set interface.

**List**: List is an interface which allow to store collection of elements of any type. List allow index. List maintain the order. List allow duplicate.

List classes : These below classes directly or indirectly implements List interface

**ArrayList**

**LinkedList**

**Vector**

**Stack**

**Queue:** Queue is one the data structure. Which support features as First In First Out.

**Queue classes:** These below classes directly or indirectly implements Queue interface.

**PriorityQueue**

**LinkedList**

**ArrayDeque**

**Map:** it allow to store key-value pairs. Key is unique and value can be duplicate.

**Map classes:** These below classes directly or indirectly implements Map interface.

**HashMap**

**LinkedHashMap**

**TreeMap**

**Hashtable**

**Wrapper classes**

Wrapper classes primitive data types

Byte byte

Short short

Integer int

Long long

Float float

Double double

Character char

Boolean boolean

Use of wrapper classes

1. Type casting : converting to object to primitive and vice-versa.
2. It provided lot of method which can apply on those values.