**Course 2 - Backend and Database Development**

**Day 6: 17 Dec 2024**

**Interface :** interface is a type of reference data types. Interface contains only constant variable and abstract methods.

Syntax

interface interfaceName{

variables;

methods;

}

By default all variables part of interface public static and final.

By default all methods in interface are public and abstract.

interface Abc { super interface

public static final int A=10;

public abstract void dis1();

}

interface Xyz { super interface

public static final int B=20;

public abstract void dis2();

}

interface Mno extends Abc,Xyz{ sub interface

public static final int C=20;

public abstract void dis3();

}

Like a class one interface can extends another interface even interface can extends more than one interface but class can’t. using interface we can achieve multiple inheritance.

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

interface Abc { super interface

public static final int A=10;

public abstract void dis1();

}

interface Xyz { super interface

public static final int B=20;

public abstract void dis2();

}

interface Mno extends Abc,Xyz{ sub interface

public static final int C=20;

public abstract void dis3();

}

class Demo implements Abc,Xyz{

public void dis1() { }

public void dis2() { }

}

Using interface we can provide the specification for our application. Class is use to provide the implementation base upon specification.

Using abstract class we can do partial abstraction. Because abstract class can contains normal as well as abstract methods. Using interface we can achieve 100% abstraction.

Abstraction : hiding the internal implementation without knowing background details. a

interface Bank {

public void withdraw(int accno, int amount);

public void deposit(int accno, int amount);

public void transfer(int from,int to, int amount);

}

public class Hsbc implements Bank {

provide the body for all 3 methods

}

public class BankOfAmerica implements Bank {

provide the body for all 3 methods

}

Access specifiers :

Using access specifiers we can control the visibility of class, variable and methods within a same class as well as same package as well as different package.

Java provided totally 4 access specifiers.

private : scope with a same class.

default (no access specifiers) : scope : within a same package

protected : scope : within a same package as well as other package if sub class.

public : scope : within a same package as well as other package.

package : package is a collection of class and interface which have same name but different functionality. Package is like a directory or folder.

Package mainly divided into two types.

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

education

school college

Attendance Attedance

Java provided mainly 2 root package

java javax -🡪 root package

x means extendible

package

lang language sql

util utility servlet

io input and output ejb

sql structured query language jms

net net

etc etc

import java.util.Scanner;

Scanner sc = new Scanner(System.in);

import java.sql.\*; // basic

import javax.sql.\*; // adv as well as extra functionality

by default java import lang package. So we can use all classes and interfaces part of lang package without importing.

By default java extends pre defined class ie Object. In Java Object is a pre defined class part of lang package. By default every java program may be pre defined or user defined internally extends Object class.

Banking Application

JavaBean class :

In JavaBean class all variable must be private. For each variable we need to provide two methods ie setter and getter. Setter method is use to set the value and getter method is use to get the value.

Service class

This class contains pure business logic.

Imp class and interface part of lang package.

Exception

Error

Types of Exception classes

Thread

Runnable interface

String

StringBuffer

StringBuilder

System

Etc