**Ex No: Inheritance**

**Date:**

**AIM**

The aim of the provided Java programs is to implement different concepts of inheritance

**STEP 1: Single Inheritance:**

* Create a base class LivingBeing with methods breathe() and respond().
* Create a derived class Animal that inherits from LivingBeing and add methods walk() and noOfLegs().
* Instantiate an object of class Animal and demonstrate calling methods of both classes.

**STEP 2: Multilevel Inheritance:**

* Create a base class LivingBeing with methods breathe() and respond().
* Create a derived class Animal that inherits from LivingBeing and add methods walk() and noOfLegs().
* Create further derived classes Cat and Dog from Animal, adding methods meow() and bark() respectively.
* Instantiate objects of Cat and Dog classes and demonstrate calling methods of all classes.

**STEP 3: Messaging Service:**

* Create a base class User with fields for name, phone number, and status.
* Create a derived class Contact from User with an array of contacts and a method to add contacts.
* Create a class Message with fields for sender, receiver, and message content.
* Create a class Chat with arrays for participants and messages, and methods to add messages and display chat history.
* Instantiate objects for users, contacts, and messages, and demonstrate the functionality of the messaging service.

1. **Write a program to show single inheritance**

**Class LivingBeing**

**Methods: Breath() and Response()**

**Class Animal**

**Methods: Walk() and NoOfLegs()**

**CODE**

class LivingBeing {

public void breathe() {

System.out.println("I am breathing");

}

public void respond() {

System.out.println("I am responding");

}

}

class Animal extends LivingBeing {

public void walk() {

System.out.println("I am walking");

}

public void noOfLegs() {

System.out.println("I have 4 legs");

}

}

public class SingleMethode {

public static void main(String[] args) {

Animal animal = new Animal();

animal.breathe();

animal.respond();

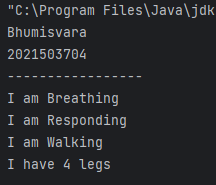
animal.walk();

animal.noOfLegs();

}

}

**OUTPUT**

****

1. **Write a program to show Multilevel inheritance**

**Class LivingBeing**

**Methods: Breath() and Response()**

**Class Animal**

**Methods: Walk() and NoOfLegs()**

**Class Cat**

**Methods: Meow()**

**Class Dog**

**Methods: Bark()**

**CODE**

class LivingBeing {

public void breathe() {

System.out.println("I am breathing");

}

public void respond() {

System.out.println("I am responding");

}

}

class Animal extends LivingBeing {

public void walk() {

System.out.println("I am walking");

}

public void noOfLegs() {

System.out.println("I have 4 legs");

}

}

class Cat extends Animal {

public void meow() {

System.out.println("Meow");

}

}

class Dog extends Animal {

public void bark() {

System.out.println("Woof");

}

}

public class MultiMethode {

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

System.out.println("--------------------");

Cat cat = new Cat();

cat.breathe();

cat.respond();

cat.walk();

cat.noOfLegs();

cat.meow();

Dog dog = new Dog();

dog.breathe();

dog.respond();

dog.walk();

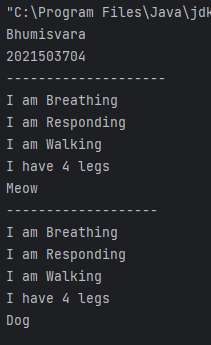
dog.noOfLegs();

dog.bark();

}

}

**OUTPUT**

****

1. **Write a program in Java to create messaging service like WhatsApp that uses single inheritance, multilevel inheritance, and hierarchical inheritance.**

**CODE**

class User {

protected String name;

protected String phoneNumber;

protected String status;

public User(String name, String phoneNumber, String status) {

this.name = name;

this.phoneNumber = phoneNumber;

this.status = status;

}

}

class Contact extends User {

protected User[] contacts;

protected int contactCount;

public Contact(String name, String phoneNumber, String status, int maxContacts) {

super(name, phoneNumber, status);

this.contacts = new User[maxContacts];

this.contactCount = 0;

}

public void addContact(User user) {

if (contactCount < contacts.length) {

contacts[contactCount++] = user;

System.out.println(name + " added " + user.name + " to contacts.");

} else {

System.out.println(name + "'s contact list is full.");

}

}

}

class Message {

protected User sender;

protected User receiver;

protected String messageContent;

public Message(User sender, User receiver, String messageContent) {

this.sender = sender;

this.receiver = receiver;

this.messageContent = messageContent;

}

}

class Chat {

private User[] participants;

private Message[] messages;

private int messageCount;

public Chat(User user1, User user2, int maxMessages) {

participants = new User[]{user1, user2};

messages = new Message[maxMessages];

messageCount = 0;

}

public void addMessage(Message message) {

if (messageCount < messages.length) {

messages[messageCount++] = message;

System.out.println("New message added to the chat.");

} else {

System.out.println("Chat is full. Cannot add more messages.");

}

}

public void displayChatHistory() {

System.out.println("Chat History:");

for (Message message : messages) {

if (message != null) {

System.out.println(message.sender.name + " to " + message.receiver.name + ": " + message.messageContent);

}

}

}

}

public class Whatsapp3704 {

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

System.out.println("---------------------------");

User alice = new User("Bhumi", "+1234567890", "Available");

User bob = new User("Mams", "+9876543210", "Away");

Contact aliceContacts = new Contact("Bhumi", "+1234567890", "Available", 10);

aliceContacts.addContact(bob);

Message message1 = new Message(alice, bob, "Hi, Mams!");

Message message2 = new Message(bob, alice, "Hello, Bhumi!");

Chat chat = new Chat(alice, bob, 100);

chat.addMessage(message1);

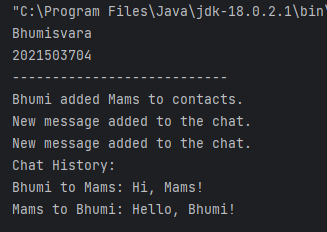
chat.addMessage(message2);

chat.displayChatHistory();

}

}

**OUTPUT**

****

**RESULT**

Thus, The Inheritance program has been successfully implemented.