class Player {

private String name;

private int age;

public Player(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

}

class CricketPlayer extends Player {

private String cricketRole;

public CricketPlayer(String name, int age, String cricketRole) {

super(name, age);

this.cricketRole = cricketRole;

}

public String getCricketRole() {

return cricketRole;

}

}

class FootballPlayer extends Player {

private String position;

public FootballPlayer(String name, int age, String position) {

super(name, age);

this.position = position;

}

public String getPosition() {

return position;

}

}

class HockeyPlayer extends Player {

private String hockeyPosition;

public HockeyPlayer(String name, int age, String hockeyPosition) {

super(name, age);

this.hockeyPosition = hockeyPosition;

}

public String getHockeyPosition() {

return hockeyPosition;

}

}

public class PlayerHierarchy {

public static void main(String[] args) {

CricketPlayer cricketPlayer = new CricketPlayer("Sachin", 45, "Batsman");

FootballPlayer footballPlayer = new FootballPlayer("Ronaldo", 36, "Forward");

HockeyPlayer hockeyPlayer = new HockeyPlayer("Dhyan Chand", 40, "Midfielder");

System.out.println("Cricket Player: " + cricketPlayer.getName() + ", Role: " + cricketPlayer.getCricketRole());

System.out.println("Football Player: " + footballPlayer.getName() + ", Position: " + footballPlayer.getPosition());

System.out.println("Hockey Player: " + hockeyPlayer.getName() + ", Position: " + hockeyPlayer.getHockeyPosition());

}

}

You can create a **Player** class and then inherit the **CricketPlayer**, **FootballPlayer**, and **HockeyPlayer** classes from it in Java. Here's an example:

In this example, we have a **Player** class with a name and age. The **CricketPlayer**, **FootballPlayer**, and **HockeyPlayer** classes inherit from the **Player** class and add specific attributes for each sport.

You can create instances of these player types and access their respective attributes.