| **Name:** | Mahadev Balla |
| --- | --- |
| **UID:** | 2023300010 |
| **Experiment No.** | 5B |

| **AIM:** | Implement a program to demonstrate heirarchical inheritance. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Give the definition of four classes, Person, Doctor, Patient and Billing, whose objects are records for a clinic. Class Doctor will be derived from the class Person. A doctor have name and Date (inherited from the class Person), it’s speciality; fees and income; Patient will be derived from the class Person. A Patient record has the patient’s name and Date (inherited from the class Person) and a Doctor object. A Billing object will contain a Patient object, a Doctor object, Date of bill using date object and an amount due of type double. Be sure your classes have a reasonable complement of constructors, copy constructor, override equals(check equality of object) and toString methods and member functions. First write a driver program to test all your member functions, and then write a test program that creates at least two patients, at least two doctors, and at least two Billing records, then prints out the total income from the Billing records. At the time of Billing the Patient’s Doctor name and Billing object’s doctor name equality should be checked. Total bill generated will be no of days the patient admitted ( current date-admitted date ) \* 2000+ doctore fees \* no of days from the date of billing ( current date-admitted date ). Update the Doctors income to no of days from the date of billing ( current date-admitted date ).  To Calculate Days:  D2 = new Date();  days = (D2.getTime() - D.getTime())/(1000\*60\*60\*24); |
| **PROGRAM:** | import java.util.\*;  import java.util.Date;  class Person {  String name;  Date dateOfJoining;  Person(String name, Date dateOfBirth) {  this.dateOfJoining = dateOfBirth;  this.name = name;  }  public Person(Person person) {  this.dateOfJoining = new Date(person.dateOfJoining.getTime());  this.name = person.name;  }  @Override  public String toString() {  return name + " " + dateOfJoining;  }  }  class Doctor extends Person {  String speciality;  double fees;  double income;  Doctor(String name, Date dateOfJoining, String speciality, double fees, double income) {  super(name, dateOfJoining);  this.speciality = speciality;  this.fees = fees;  this.income = income;  }  public Doctor(Doctor doctor) {  super(doctor);  this.speciality = doctor.speciality;  this.fees = doctor.fees;  this.income = doctor.income;  }  @Override  public String toString() {  return super.toString() + " " + speciality + " " + fees + " " + income;  }  }  class Patient extends Person {  Doctor attendingDoctor;  long phone;  Patient(String name, Date dateOfJoining, Doctor attendingDoctor, long phone) {  super(name, dateOfJoining);  this.attendingDoctor = attendingDoctor;  this.phone = phone;  }  public Patient(Patient patient) {  super(patient);  this.attendingDoctor = new Doctor(patient.attendingDoctor);  }  @Override  public String toString() {  return super.toString() + " " + attendingDoctor.name;  }  }  class Billing {  Patient patient;  Doctor doctor;  Date billingDate;  double amountDue;  Billing(Patient patient, Doctor doctor, Date billingDate, double amountDue) {  this.patient = patient;  this.doctor = doctor;  this.billingDate = billingDate;  this.amountDue = amountDue;  }  public Billing(Billing billing) {  this.patient = new Patient(billing.patient);  this.doctor = new Doctor(billing.doctor);  this.billingDate = new Date(billing.billingDate.getTime());  this.amountDue = billing.amountDue;  }  @Override  public String toString() {  return patient.name + " " + doctor.name + " " + billingDate + " " + amountDue;  }  }  public class clinic {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  Date D2 = new Date();  System.out.print("Enter the number of doctors : ");  int n = sc.nextInt();  sc.nextLine();  Doctor d[] = new Doctor[n];  for (int i = 0; i < n; i++) {  System.out.print("Enter details of doctor " + (i + 1) + " -\nName : ");  String name = sc.nextLine();  System.out.print("Speciality : ");  String speciality = sc.nextLine();  System.out.print("Fees : ");  double fees = sc.nextDouble();  System.out.print("Income : ");  double income = sc.nextDouble();  sc.nextLine();  d[i] = new Doctor(name, new Date(), speciality, fees, income);  }  System.out.print("Enter the number of patients : ");  int m = sc.nextInt();  sc.nextLine();  Patient p[] = new Patient[m];  Billing bill[] = new Billing[m];  double temp=0;  for (int i = 0; i < m; i++) {  System.out.print("Enter details of patient " + (i + 1) + " -\nName : ");  String name = sc.nextLine();  System.out.print("Enter your phone number : ");  long phone = sc.nextLong();  sc.nextLine();  System.out.print("Enter the name of the doctor : ");  String docName = sc.nextLine();  Doctor doc = null;  for (int j = 0; j < n; j++) {  if (d[j].name.equals(docName)) {  doc = d[j];  temp = d[j].fees;  break;  }  }  if (doc == null) {  System.out.println("Doctor with name " + docName + " not found!");  return;  }  p[i] = new Patient(name, new Date(), doc, phone);  System.out.print("Enter no. of days the patient was admitted : ");  int days = sc.nextInt();  double amount = days\*2000 + temp;  sc.nextLine();  bill[i] = new Billing(p[i], doc, new Date(), amount);  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (bill[j].doctor.equals(d[i])) {  d[i].income += bill[j].amountDue;  break;  }  }  }  System.out.printf("Doctors' details as on current date -\n%-20s %-35s %-20s %-20s %-20s\n", "Name", "Day & Date of Joining", "Speciality", "Fees", "Income");  for (int i = 0; i < n; i++) {  System.out.printf("%-20s %-35s %-20s %-20s %-20s\n", d[i].name, d[i].dateOfJoining, d[i].speciality, d[i].fees, d[i].income);  }  double totalIncome = 0;  System.out.println("Billing Details - ");  System.out.printf("%-20s %-20s %-20s %-20s\n", "Name", "Phone Number", "Doctor's Name", "Amount Due");  for (int i = 0; i < m; i++) {  System.out.printf("%-20s %-20s %-20s %-20s\n", p[i].name, p[i].phone, p[i].attendingDoctor.name, bill[i].amountDue);  totalIncome += bill[i].amountDue;  }  System.out.println("Total income from billing records: " + totalIncome);  }  } |
| **RESULT:** | |
| **CONCLUSION:** | Studied the implementation of heirarchical inheritance and method overriding to solve the given problem. |