| **Name:** | Mahadev Balla |
| --- | --- |
| **UID:** | 2023300010 |
| **Experiment No.** | 6B |

| **AIM:** | Implement a program to demonstrate method overriding. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Jayesh works at ABC Company. He noticed that different roles (positions) have different salaries and bonuses. The 1st Role is an ‘Intern’ which has 3/4th of the base salary of an Employee. Then there is ‘Clerk’ which has ½ of base salary. And then there are ‘Manager’ who have twice the base salary of that of an Employee. Help him write a program in Java using method overriding concept. Follow the following rules/conditions.  # Create a class ‘Employee’ which has a method named ‘getSalary’ which  returns a base salary of Rs. 10,000. It also has methods named ‘getBonus’  which returns the bonus amount for that role(initially set to Rs. 0).  # Make 3 subclasses for different roles which inherit the ‘Employee’ class  and each has functions named ‘getSalary’ and ‘getBonus’.(You can assume  values for ‘getBonus’ method)  # Display the output for all cases. Also print the total salary received by each Employee after getting the bonus. |
| **PROGRAM:** | import java.util.\*;  class Employee{  private double salary=10000;    public double getSalary(){  return salary;  }    public double getBonus(){  return 0;  }  }  class Intern extends Employee{  public double getSalary(){  return 0.75 \* super.getSalary();  }    public double getBonus(){  return 500;  }  }  class Clerk extends Employee{  public double getSalary(){  return 0.5 \* super.getSalary();  }    public double getBonus(){  return 1000;  }  }  class Manager extends Employee{  public double getSalary(){  return 2 \* super.getSalary();  }    public double getBonus(){  return 2500;  }  }  class salary{  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  int x=0;  do{    System.out.print("1. Employee\n2. Intern\n3. Clerk\n4. Manager\n5. Exit\nEnter your choice : ");  x = sc.nextInt();  sc.nextLine();  switch(x){  case 1: System.out.print("Enter your details -\nName : ");  String nameE = sc.nextLine();  System.out.print("Age : ");  int ageE = sc.nextInt();  if(ageE>=18){  Employee e = new Employee();  System.out.println("Salary : " + e.getSalary() + "\nBonus : " + e.getBonus() + "\nTotal Salary : " + (e.getSalary() + e.getBonus()));  }  else if(ageE>0 && ageE<18){  System.out.println("Age requirement not met!! You're not eligible for a job at ABC company.");  }  else if(ageE>0 && ageE<18){  System.out.println("Age requirement not met!! You're not eligible for a job at ABC company.");  }  else{  System.out.println("Invalid input!!");  }  break;    case 2: System.out.print("Enter your details -\nName : ");  String nameI = sc.nextLine();  System.out.print("Age : ");  int ageI = sc.nextInt();  if(ageI>=18 && ageI<=30){  Intern i = new Intern();  System.out.println("Salary : " + i.getSalary() + "\nBonus : " + i.getBonus() + "\nTotal Salary : " + (i.getSalary() + i.getBonus()));  }  else if(ageI>0 && ageI<18 || ageI>30 ){  System.out.println("Age requirement not met!! You're not eligible for an internship at ABC company.");  }  else{  System.out.println("Invalid input!!");  }  break;      case 3: System.out.print("Enter your details -\nName : ");  String nameC = sc.nextLine();  System.out.print("Age : ");  int ageC = sc.nextInt();  if(ageC>=18){  Clerk c = new Clerk();  System.out.println("Salary : " + c.getSalary() + "\nBonus : " + c.getBonus() + "\nTotal Salary : " + (c.getSalary() + c.getBonus()));  }  else if(ageC>0 && ageC<18){  System.out.println("Age requirement not met!! You're not eligible for a job at ABC company.");  }  else{  System.out.println("Invalid input!!");  }  break;    case 4: System.out.print("Enter your details -\nName : ");  String nameM = sc.nextLine();  System.out.print("Age : ");  int ageM = sc.nextInt();  if(ageM>=18){  Manager m = new Manager();  System.out.println("Salary : " + m.getSalary() + "\nBonus : " + m.getBonus() + "\nTotal Salary : " + (m.getSalary() + m.getBonus()));  }  else if(ageM>0 && ageM<18){  System.out.println("Age requirement not met!! You're not eligible for a job at ABC company.");  }  else{  System.out.println("Invalid input!!");  }  break;    case 5: System.out.println("Thank you!!");  break;    default: System.out.println("Invalid choice!!");  }  }  while(x!=5);  }  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Consider a class Product with data members barcode and name of the product. Create the appropriate constructor and write getter methods for the individual data members. Add a method called print() to print product details.  Derive 2 classes from Product, 1st class is PrepackedFood and 2nd class is FreshFood. the PrepackedFood class should contain the unit price and the FreshFood class should contain a weight and a price per kilo as data members. Add appropriate constructors. Override the print() method in the FreshFood and PrepackedFood classes. Write Tester class which contains main(), which creates an array of Product objects and stores either FreshFood or PrepackedFood products in the array. Print the details of products. Write a method in the Tester class which finds the cheapest product which is of type FreshFood by iterating through the products array. |
| **PROGRAM:** | /\*Program to find cheapest product acc to its MRP and not total amt….to find cheapest product acc to total amt ‘getAmt’ method shld be replaced wherever getuPrice and getPricepk methods are called. I’ve printed the cheapest food product acc to MRP as taking no. of units or kilos of product as i/p isn’t mentioned in the problem statement. \*/  import java.util.\*;  class Product{  private String name, barcode;  Product(String name, String barcode){  this.name = name; this.barcode = barcode;  }  public String getName(){  return name;  }  public String getBarcode(){  return barcode;  }    void print(){  System.out.println("Name of the product : " + name + "\nBarcode : " + barcode);  }  }  class PrepackedFood extends Product{  private double uPrice;  private int qty;  PrepackedFood(String name, String barcode, double uPrice, int qty){  super(name,barcode);  this.qty = qty;  this.uPrice = uPrice;  }  public double getuPrice(){  return uPrice;  }  public double getAmt(){  return qty\*uPrice;  }    @Override  void print(){  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n" ,super.getName(), super.getBarcode(), uPrice, qty, (qty\*uPrice));  }  }  class FreshFood extends Product{  private double pricepk, wt;  FreshFood(String name, String barcode, double pricepk, double wt){  super(name,barcode);  this.wt = wt;  this.pricepk = pricepk;  }  public double getPricepk(){  return pricepk;  }  public double getAmt(){  return wt\*pricepk;  }  @Override  void print(){  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", super.getName(), super.getBarcode(), pricepk, wt, (wt\*pricepk));  }  }  class food1 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  int x, qty, n1, n2, m1=0, m2=0; double uPrice, pricepk, wt, min1=0, min2=0;  do{  System.out.print("Enter number of prepacked food products : ");  n1 = sc.nextInt();  if(n1<0){  System.out.println("Enter a valid input!!");  }  }  while(n1<0);  sc.nextLine();  PrepackedFood p1[] = new PrepackedFood[n1];  for(int i=0; i<n1; i++){  System.out.print("Enter details of product " + (i+1) + " -\nName : ");  String name = sc.nextLine();  System.out.print("Barcode : ");  String barcode = sc.nextLine();  do{  System.out.print("Price per unit : ");  uPrice = sc.nextDouble();  if(uPrice<=0){ System.out.println("Enter a valid input!!"); }  }  while(uPrice<=0);  do{  System.out.print("Number of units : ");  qty = sc.nextInt();  if(qty<=0){ System.out.println("Enter a valid input!!"); }  }  while(qty<=0);  p1[i] = new PrepackedFood(name,barcode,uPrice,qty);  sc.nextLine();  }  do{  System.out.print("Enter number of fresh food products : ");  n2 = sc.nextInt();  if(n2<0){  System.out.println("Enter a valid input!!");  }  }  while(n2<0);  sc.nextLine();  FreshFood p2[] = new FreshFood[n2];  for(int i=0; i<n2; i++){  System.out.print("Enter details of product " + (i+1) + " -\nName : ");  String name = sc.nextLine();  System.out.print("Barcode : ");  String barcode = sc.nextLine();  do{  System.out.print("Price per kilo : ");  pricepk = sc.nextDouble();  if(pricepk<=0){ System.out.println("Enter a valid input!!"); }  }  while(pricepk<=0);  do{  System.out.print("Weight : ");  wt = sc.nextDouble();  if(wt<=0){ System.out.println("Enter a valid input!!"); }  }  while(wt<=0);  p2[i] = new FreshFood(name,barcode,pricepk,wt);  sc.nextLine();  }  if(n1>0){  min1 = p1[0].getuPrice();  for(int j=0; j<n1; j++){  if(p1[j].getuPrice()<min1){  min1 = p1[j].getuPrice();  m1 = j;  }  }  }    if(n2>0){  min2 = p2[0].getPricepk();  for(int k=0; k<n2; k++){  if(p2[k].getPricepk()<min2){  min2 = p2[k].getPricepk();  m2 = k;  }  }  }    if(n1+n2>0){  System.out.println("Details of the products -");  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per unit/kilo", "Number of units/Weight", "Total Price");  for(int j=0; j<p1.length; j++){  p1[j].print();  }  for(int j=0; j<p2.length; j++){  p2[j].print();  }    if(n1>0){  System.out.println("Details of cheapest prepacked food product -");  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per unit", "Number of units", "Total Price");  p1[m1].print();  }  if(n2>0){  System.out.println("Details of cheapest fresh food product -");  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per kilo", "Weight", "Total Price");  p2[m2].print();  }  System.out.println("Details of cheapest food product product -");  if(n1==0){  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per kilo", "Weight", "Total Price");  p2[m2].print();  }  else if(n2==0){  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per unit", "Number of units", "Total Price");  p1[m1].print();  }  else{  if(min1<min2){  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per unit", "Number of units", "Total Price");  p1[m1].print();  }  else if(min1==min2){  System.out.println("Details of cheapest products are same as that of the above two!!");  }  else{  System.out.printf("%-25s %-25s %-25s %-25s %-25s\n", "Name", "Barcode", "Price per kilo", "Weight", "Total Price");  p2[m2].print();  }  }  }  }  } |
| **RESULT:** | |
| **RESULT FOR TEST CASES:** | |
| **CONCLUSION:** | Studied the implementation of method overriding to solve the given problems. |