GitHub Link : https://github.com/Kalpit12/PART-2.git

|  |  |
| --- | --- |
| Name | Kalpit patel |
| id | 21ce094 |

|  |  |
| --- | --- |
| **Question1** | Design a class named Circle containing following attributes and behavior.•Onedouble data field named radius. The default valueis1.•A no-argument constructor that creates a default circle.•A Single argument constructor that creates a Circle with the specified radius.•A method named getArea() that returns area of the Circle.•A method named getPerimeter() that returns perimeterof it. |
| **Answer** | Code:  import java.util.Scanner;   public class Circle {         private double radius = 1;         public Circle() {           // radius = 2;       }         public Circle(int r) {           radius = (double) (r);       }         public double getArea() {           return radius \* 3.14 \* radius;       }         public double getPerimeter() {           return 2 \* 3.14 \* radius;       }         public static void main(String[] args) {           Circle circle1 = new Circle();           System.out.println("default area");           System.out.println(circle1.getArea());           System.out.println("default perameter");           System.out.println(circle1.getPerimeter());           System.out.println("enter radius");             Scanner sc = new Scanner(System.in);           int rad = sc.nextInt();           Circle circle2 = new Circle(rad);           System.out.println("area");           System.out.println(circle2.getArea());           System.out.println("perameter");           System.out.println(circle2.getPerimeter());         }   } |
| **Question** | Design a class named Account that contains:  •A private int data field namedid for the account (default 0).0  •A private double data field named balance for the account (default 500₹)  .•A private double data field named annualInterestRate that stores the currentinterest rate (default 7%). Assume all accounts have the same interest rate  .•A private Date data field named dateCreated that stores the date when theaccount was created.•A no-arg constructor that creates a default account.  •A constructor that creates an account with the specified id and initial balance  .•The accessor and mutator methods for id, balance, and annualInterestRate.  •The accessor method for dateCreated.  •A method named getMonthlyInterestRate() that returns the monthlyinterest rate.  •A method named getMonthlyInterest() that returns the monthly interest.  •A method named withdraw that withdraws a specified amount from theaccount.  •A method named deposit that deposits aspecified amount to the account. |
| **Answer** | Code:  import java.util.Date;  import java.util.Scanner;  import java.text.SimpleDateFormat;  public class Account {      private int id = 0;      private double balance = 500;      private double annualInterestRate = 0.07;      private Date date;      Account() {      }      Account(int a, double b, double c) {          id = a;          balance = b;          annualInterestRate = c;      }      public void getdata() {          Scanner sc = new Scanner(System.in);          id = sc.nextInt();          balance = sc.nextDouble();          annualInterestRate = sc.nextDouble();      }      public void putdata() {          System.out.println(id);          System.out.println(balance);          System.out.println(annualInterestRate);      }      public void getdate() {          SimpleDateFormat dateform = new SimpleDateFormat("dd/MM/yyyy");          Date newdate = new Date();          System.out.println(dateform.format(newdate));      }      public double getMonthlyIntrestRate() {          double l;          l = balance \* annualInterestRate;          System.out.println(l);          return l;      }      public void Withdraw() {          Scanner sc = new Scanner(System.in);          int l = sc.nextInt();          balance -= l;      }      public void Deposit() {          Scanner sc = new Scanner(System.in);          int l = sc.nextInt();          balance += l;      }      public static void main(String[] args) {          Account acc = new Account();  System.out.println("enter id balance and annual intrest rate");          acc.getdata();          System.out.println("data that you have entered");          acc.putdata();          System.out.println("date");          acc.getdate();          System.out.println("your monthly intrest");          acc.getMonthlyIntrestRate();          System.out.println("money you want to deposite");          acc.Deposit();          System.out.println("money you want to withdraw");          acc.Withdraw();          System.out.println("your final id balance and intrest rate");          acc.putdata();      }  } |
| **Question3** | Use the Account class created as above to simulate an ATM machine. Create 10 accounts with id AC001.....AC010 with initial balance 300₹. The system prompts the users to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, display menu with multiple choices.  1.Balance inquiry  2.Withdraw money [Maintain minimum balance 300₹]  3.Deposit money  4.Money Transfer  5.Create Account  6.Deactivate Account  7.Exit Hint: Use ArrayList, which is can shrink and expand with compared to Array. |
| **Answer** | Code:  import java.util.Scanner;  public class Accoutn1 {      public String id;      double balance = 300;      Accoutn1() {      }      public Accoutn1(String s) {          id = s;      }      public static void main(String[] args) {          Accoutn1[] acc = new Accoutn1[12];          Scanner sc = new Scanner(System.in);          for (int i = 1; i <= 10; i++) {              acc[i] = new Accoutn1();              acc[i] = new Accoutn1("AC" + i);              System.out.println(acc[i].id + " 100 ");          }          int k = 0;          int l;          int j;          do {              System.out.println("Enter valid ID number");              String t = sc.next();              for (j = 1; j <= 10; j++) {                  if (t.equals(acc[j].id)) {                      k++;                      break;                  } else {                  }              }              if (k == 0) {                  l = 10;              } else {                  l = 3;              }          } while (l != 3);          System.out.println("Hello");          System.out.println("ENter 1 for balance");          System.out.println("ENter 2 for withdraw");          System.out.println("ENter 3 for deposit");          System.out.println("ENter 4 for money transfer");          System.out.println("ENter 5 for create account");          System.out.println("ENter 6 for deactivate account");          System.out.println("Enter 7 for exiting");          int x = sc.nextInt();          switch (x) {              case 1: {                  System.out.println(acc[2].balance);                  break;              }              case 2: {                  int s = sc.nextInt();                  acc[j].balance -= s;                  System.out.println(acc[2].balance);                  break;              }              case 3: {                  int s = sc.nextInt();                  acc[j].balance += s;                  System.out.println(acc[2].balance);                  break;              }              case 4: {                  int s = sc.nextInt();                  System.out.print("To be transferred in :");                  int i = sc.nextInt();                  acc[i].balance += s;                  acc[j].balance -= s;                  System.out.println(acc[j].balance);                  System.out.println(acc[i].balance);                  break;              }              case 5: {                  acc[11] = new Accoutn1();                  System.out.println("Account created");                  break;              }              case 6: {                  acc[j].balance = 0;                  acc[j].id = "0A";                  break;              }          }      }  } |
| **Question4** | Subclasses of Account) In Programming Exercise 2, the Account class was defined to model a bank account. An account has the properties account number, balance, annual interest rate, and date created, and methods to deposit and withdraw funds. Create two subclasses for checking and saving accounts. A checking account has an overdraft limit, but a savings account cannot be overdrawn. Draw the UML diagram for the classes and then implement them. Write a test program that creates objects of Account, Savings Account, and Checking Account and invokes their to String() methods. |
| **answer** | import java.util.Scanner;  class Account  {      private int id = 0;      double balance = 500, annualInterest = 7, amount;      String dateCreated;      Account()      {          id = 0;          balance = 50000;          annualInterest = 7;      }      Account(int i, double bal)      {          id = i;          balance = bal;      }      void setdata(int i, double bal, double aInt, String dt)      {          id = i;          balance = bal;          annualInterest = aInt;          dateCreated = dt;      }      int getId()      {          return id;      }      double getBal()      {          return balance;      }      double getAnn()      {          return annualInterest;      }      double getMonthlyInterestRate()      {          return (annualInterest \* 100) / 12;      }      double getMonthlyInterest()      {          return balance \* (annualInterest \* 100) / 12;      }      String getDt()      {          return dateCreated;      }      void withdraw(double amount)      {          balance -= amount;          if (balance > 0)          {              System.out.println("The balance left after withdrawal of Rs " + amount + " is Rs. " + balance);          }          else          {              System.out.println("Withdrawal of Rs " + amount + " is not possible ");          }      }      void deposit(double amount)      {          balance += amount;          System.out.println("The balance left after deposit of Rs." + amount + " is Rs. " + balance);      }  }  class SavingAccount extends Account  {      SavingAccount(double a)      {          amount = a;          balance -= amount;      }      public String toString()      {          if (balance >= 3000)          {              return "The balance left after withdrawal of Rs " + amount + " is Rs. " + balance;          }          else          {              return "Minimum balance of Rs. 3000 is required.";          }      }  }  class CheckingAccount extends Account  {      CheckingAccount(double am)      {          amount=am;          balance-=amount;      }      public String toString()      {          System.out.println("Withdrawal successful");          return "Now the balance left is Rs. "+balance+" after the withdrawal of Rs. "+amount;      }  }  class pr4  {      public static void main(String[] args)      {          Account a1=new Account();          Account a2=new Account(123456,100000);          a2.setdata(1289031,100000,5.6,"2-8-2022");          System.out.println("Account Details: ");          System.out.println("Balance : "+a2.balance);          System.out.println("Annual Interest : "+a2.getAnn());          System.out.println("Monthly Interest Rate : "+a2.getMonthlyInterestRate());          System.out.println("Monthly Interest : "+a2.getMonthlyInterest());          System.out.println("Account was created on : "+a2.getDt());          a2.withdraw(12000);          a2.deposit(15000);          System.out.print("------------------------\n");          SavingAccount a=new SavingAccount(900);          CheckingAccount b=new CheckingAccount(1000);          System.out.println("For Saving Account : ");          System.out.println(a);          System.out.print("------------------------\n");          System.out.println("For Checking Account : ");          System.out.println(b);      }  } |
| **Question5** | Develop a Program that illustrate method overloading concept. |
| **Answer** | Code:  import java.util.Scanner;  public class Overloading {      static double area(int r) {          double t = 3.14 \* r \* r;          return t;      }      static float area(int a, int b) {          float t = a \* b;          return t;      }      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);           System.out.println("enter radius");          int cr = sc.nextInt();           System.out.println("enter length");          int rs = sc.nextInt();           System.out.println("enter breadth");          int rs2 = sc.nextInt();           System.out.println("area of circle");          System.out.println(area(cr));           System.out.println("area of rectangle");          System.out.println(area(rs, rs2));      }  } |