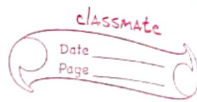


Aishwarya V  
1BM19CS008

Week 8

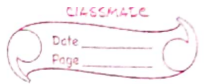


- 4) Class Bank → two kinds of accs -  
Savings - provides C.I, withdrawal facilities, no checkbook.  
Current - provides checkbook, no interest.  
    → shld maintain minimum balance,  
    if not service charge imposed.

Class Account - customer name, accno., type of acc.  
    → class Curr-acc  
    → class Sav-acc

methods - accept deposit from user and update balance.  
    - display balance.  
    - compute and deposit interest  
    - permit withdrawal and update balance.  
    - check min balance, impose penalty, update balance.

4) class Account  
{  
    String name, accno, type;  
    float bal;  
    int option;  
    float deposit, rate, years, withdrawal=0;  
    Account()  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter the name, acc no, Type, balance");  
    name = sc.next();  
    accno = sc.nextInt();  
    type = sc.next();  
    
$$CI = P \left( 1 + \frac{R}{100} \right)^n \text{ annually}$$
  
    import java.lang.Math;  
    Math.pow(a, b);      a      b  
}



```
bal = sc.nextFloat();  
}  
void display()  
{  
    System.out.println("the name = " + name + "acc. no = " + accno. +  
        " type of bank account = " + type + "balance" + bal);  
}  
}   
class sav-acc extends account  
{  
    float ci = 0;  
    boolean checkbook/facility = false;  
    void sav_activity()  
    {  
        Scanner sc = new Scanner(System.in);  
        do {  
            System.out.println("choose 1. deposit money  
            2. withdraw 3. check balance 4. checkbook 5. exit");  
            option = sc.nextInt();  
            switch (option)  
            {  
                case 1: {System.out.println("Enter the deposit amt, rate of interest,  
                no. of years");  
                    deposit = sc.nextFloat();  
                    rate = sc.nextFloat();  
                    years = sc.nextFloat();  
                    ci = deposit * ((float) Math.pow((1 + (rate/100)), years));  
                    bal = bal + ci;  
                    break;  
                }            }        }    }}
```

```

Case 2: { System.out.println("the enter amt to withdraw");
        withdrawal = sc.nextFloat();
        bal = bal - withdrawal;
    } break;
Case 3: { System.out.println("the remaining balance is =Rs'+bal);
        break;
Case 4: { if (checkbookfacility)
        System.out.println("Check book provided in savings account");
        else System.out.println(" " not provided ");
    } break;
Case 5: break;
default: System.out.println("invalid option");
} while (option != 5);
}
}
class curr_acct extends account
{
    boolean checkbookfacility = true;
    void curr_activity()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the minimum balance and penalty");
        float min = sc.nextFloat();
        float penalty = sc.nextFloat();
        do { System.out.println("choose 1. deposit 2. withdraw
        3. checkbalance 4. checkbook 5. exit");
            option = sc.nextInt();

```

switch (option).

```

{
    Case 1: { System.out.println("Enter the deposit");
            deposit = sc.nextFloat();
            bal = bal + deposit;
        } break;
    Case 2: { System.out.println("enter the amt you want to withdraw");
            withdrawal = sc.nextFloat();
            bal = bal - withdrawal;
        } break;
    Case 3: { if (bal < min)
            { bal -= penalty;
              System.out.println("penalty = " + penalty);
            }
            System.out.println("the remaining balance is =Rs." + bal);
        } break;
    Case 4: { if (checkbookfacility)
            System.out.println("check book provided");
            else System.out.println("checkbook not provided");
        } break;
    Case 5: break;
    default: System.out.println("invalid option");
} while (option != 5);
}
}

```

class bank {

```
public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the type 1. savings 2. current");
    int type = sc.nextInt();
    if (type == 1)
    {
        sav_acct s = new sav_acct();
        s.sav_activity();
    }
    if (type == 2)
    {
        curr_acct c = new curr_acct();
        c.curr_activity();
    }
}
```

4.5.2) Abstract Class Shape → 2 integers and entity method printArea();  
 Classes extend Shape → rectangle, triangle and Circle.  
 Each contain only method printArea(), that prints the area of the shape.

```
1. import java.util.Scanner;
abstract class Shape
{
    int a, b;
    abstract void printArea();
}
```

class rectangle extends shape

```
{
    void printArea()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the sides of rectangle");
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println("The area of rectangle = " + (a * b));
    }
}
```

class triangle extends shape

```
{
    void printArea()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter radius of circle");
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println("The area = " + ((1/2) * a * b));
    }
}
```

class circle extends shape

```
{
    void printArea()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter radius");
        a = sc.nextInt();
        System.out.println("area = " + (3.14 * a * a));
    }
}
```



```
{ class Shapes
{ public static void main (String args[])
{ rectangle r = new rectangle();
  r.printarea();
  triangle t = new triangle();
  t.printarea();
  circle c = new circle();
  c.printarea();
}
```