

~~WAP on using generics, show static class -
using 5 integers and 5 double~~

LAB-5

Create a class Account that stores customer name, account number, type of account. From this derive the class Current-Acc and overload the methods to make them more specific to their requirements.

Perform the following tasks.

- (a) Accept deposit from user and update balance.
- (b) Display the balance.
- (c) compute and deposit interest.
- (d) Permit withdrawal and update balance.

import java.util.*;

class Input

```
{ Scanner sc = new Scanner(System.in); }
```

class Account extends Input

{ String name;

int accNo;

double balance;

void getDetails()

```
{ System.out.println("Enter Name"),
```

```
name = sc.nextLine();
```

```
System.out.println("Enter ac No"),
```

```
accNo = sc.nextInt();
```

}

```
void deposit()
```

{

```
System.out.println("Enter amount to  
deposit"); double amt = sc.nextDouble();  
balance += amt;
```

```
void withdraw()
```

{

```
System.out.println("Enter amount");  
double amt = sc.nextDouble();  
if (balance >= amt)  
{ balance -= amt;  
System.out.println("Amount Withdrawn");  
}
```

else

```
System.out.println("Insufficient balance");  
}
```

```
void display()
```

```
{ System.out.println("Name: " + name);  
System.out.println("Account Number: " + accno);  
System.out.println("Balance: " + balance);  
}
```

}

```
class Current extends Account
```

{

```
double minbal = 500;
```

```
double penalty = 100;
```

```
void withdraw()
```

```
{ super.withdraw();  
checkMinBal();  
}
```

}

```
private void checkMinBalance()
{
    if (balance < MinBal)
        balance -= penalty;
    System.out.println("Penalty applied");
}
```

y y

class Savings extends Account

{

```
double interestRate = 0.04;
```

```
void computeInterest()
```

```
{
    double interest = balance * interestRate;
    balance += interest;
}
```

```
System.out.println("Interest: " + interest);
}
```

y y

class Bank extends Account

{ public static void main(String[] args)

{

```
Saving ob1 = new Saving();
```

```
Current ob2 = new Current();
```

```
ob1.getDetails();
```

```
ob2.getDetails();
```

```
int choice, String ac;
```

```
System.out.println("Menu");
```

```
System.out.println("1. Deposit In");
```

```
2. Withdraw In 3. Display In
```

```
4. Compute Interest (saving Only) In
```

~~5. Exit~~

~~while(1)~~

~~System.out.println("Enter your choice:");~~

~~choice = sc.nextInt();~~

~~System.out.println("Enter acc type");~~

~~acc = sc.next();~~

switch (choice)

{ case 1:

if (acc.equals ("Savings"));

ob1.deposit();

else

ob2.deposit();

break;

case 2:

if (acc.equals ("Savings"))

ob1.withdraw();

else withdraw();

ob2.withdraw();

break;

case 3:

if (acc.equals ("Savings"))

ob1.display();

else

ob2.display();

break;

case 4:

ob1.computeInterest();

break;

case 5: visit();

def

y

3

Output :

Enter customer name : Krishnan

Enter acc number : 1

~~~~ Account Type ~~~~

Menu

1. Deposit
2. Withdraw
3. Display
4. Compute interest (Savings Only)
5. Exit

Enter your choice : 1

Enter acc type : Savings

Enter deposit amount : 1000

Enter your choice : 2

Enter acc type : Savings

Enter withdrawal amount : 500

Enter choice : 3

Enter account type : Savings

Name : Krishnan

Account number : 1

Balance : 500

Enter choice : 2

Enter account type : Current

Enter deposit amount : 500

Enter choice : 2

Enter withdraw : 450

Penalty .

Enter your choice : 2

Enter acc type : Saving

Enter withdraw amount : 1050

Insufficient.

~~001 1/2~~