Unit : 2 Lab : 5

DOMS | Page No.
Date 09 / 01 / 2024

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account pro-vides compound interest and withdrawal fa-cilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders shou-ld also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks.

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
public class Account {
    String customer_name;
```

```
int account_no;
String type_acc;
double balance = 0;
Scanner sc;

Account (String customer_name, int account_
  no, String type_acc) {
    this.customer_name = customer_name;
    this.account_no = account_no;
    this.type_acc = type_acc;
   sc = new Scanner(System.in);
}

void diposite(){
   System.out.println("Enter the deposit
     amount:");
   int depo = sc.nextInt();
   balance += depo;
}

void withdrawl() {
   System.out.println("Enter the withdrawl
     amount :");
   int with = sc.nextInt();
   if(with > balance) {
       System.out.println("Insufficient
         Balance");
}
```

```
        else {
            balance -= with;
        }
    }

    void display() {
        System.out.println("Customer name: "
                + customer_name);
        System.out.println("Account number: "
                + account_no);
        System.out.println("Type of Account: "
                + type_acc);
        System.out.println("Balance: "+balance);
    }

    void applyinterest() { }
}

public class Cur_acct extends Account {
    Cur_acct(String customer_name, int account_
            no, String type_acc) {
        super(customer_name, account_no,
                type_acc);
    }

    void withdrawl() {
        System.out.println("Enter the withdrawl
            amount: ");
```

```
        int with = sc.nextInt();
        if(balance <= 2000){
            double pen = balance/(0.06);
            System.out.println("Insufficient ba-
          -lance penalty to be paid :"+pen);
            balance += pen;
        }
        else{
            balance += with;
        }
    }
}

public class Sav_acct extends Account{
    Sav_acct(String customer_name, int account
        no, String type_acc){
        super(customer_name, account_no,
            type_acc);
        }

    void applyinterest(){
        System.out.println("Enter the interest
            rate:");
        int rate = sc.nextInt();
        double interest = balance*(rate/100);
        balance += interest;
        System.out.println("Balance after
            interest :"+balance);
    }
}
```

```
import java.util.Scanner;
public class Bank {
  public static void main (String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter customer name:
      ");
    String cus_name = sc.nextLine();
    System.out.println("Enter account number
      : ");

    Account ca = new Cur_acct(cus_name,
        acc_no, "Current Account");
    Account sa = new Sav_acct(cus_name,
        acc_no, "Saving Account");

    int choice;
    while (true) {
      System.out.println("----MENU----");
      System.out.println("1. Deposite \n 2.
      Withdrawl \n3. Compute interest for
      saving account \n 4. Display account
      details \n 5. Exit");
      choice = sc.nextInt();

    switch (choice) {
    case 1:
      System.out.println("Enter the type
      of account: \n1. Saving Account \n
      2. Current Account");
```

```
int acc = sc.nextInt();
if (acc==1){
    sa.diposite();
}
else {
    ca.diposite();
    }
 break;
 case 2:
    System.out.println("Enter the type of
    account:\n1. Saving Account \n2. Current
     Account");
    int acc1= sc.nextInt();
    if (acc1==1){
        sa.withdrawl();
    }
    else {
        ca.withdrawl();
    }
     break;
  case 3:
     sa.applyinterest();
     break;
  case 4:
     System.out.println("Enter the type of
     account:\n1. Saving Account \n2. Curr-
     -ent Account");
     int acc2 = sc.nextInt();
```

```
if (acc2 ==1) {
    sa.display();
  }
else {
    ca.display();
  }
break;
case 5:
    break;
default:
    System.out.println("Invalid choice");
    break;
  }
if(choice == 5){
    break;
  }
 }
}
}
```

Output :

```
Enter custome name :
Shreya
Enter account number :
22
 --MENU--
1. Deposite
2. Withdrawl
3. Compute interest for Saving account
```

4. Display account details
5. Exit
1
Enter the type of account:
1. Saving Account
2. Current Account
1
Enter the deposit amount:
5000
2
Enter the type of account:
1. Saving Account
2. Current Account
1
Enter the withdrawal amount:
2000
4
Enter the type of account:
1. Saving Account
2. Current Account
1.
Customer name: Shreya
Account Number: 22
Type of Account: Saving Account
Balance: 3000.0
2.
Enter the type of account:
1. Saving Account
2. Current Account

DOMS | Page No.
Date / /

2.

Enter the withdrawal amount :
6000.
Insufficient balance. Penalty to be paid:
8333.3334

9/11/24

DOMS | Page No.
Date 16 / 01 / 24

19. Write a Java program to create an abstract class Bird with abstract method fly() and makeSound(). Create subclasses Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies and makes a sound

```
abstract class Bird {
    abstract void fly();
    abstract void makessound();
}

class Eagle extends Bird {
    void fly() {
        System.out.println("Eagle can fly very high");
    }
    void makessound() {
        System.out.println("Eagle makes a screech sound");
    }
}

class Hawk extends Bird {
    void fly() {
        System.out.println("Hawk can fly moderatlly high");
    }
```

```
    void makesound(){
        System.out.println("Hawk makes
          a shrill sound");
    }
}

public class Main {
    public static void main(String[] args)
    {
        Bird bird1 = new Eagle();
        bird1.fly();
        bird1.makesound();

        Bird bird2 = new Hawk();
        bird2.fly();
        bird2.makesound();
    }
}
```

Output:
Eagle can fly very high
Eagle makes a screech sound.
Hawk can fly moderatlly high
Hawk makes a shrill sound.

DOMS | Page No.
Date / /

Generics

Write a Java program to create a generic class Stack which holds 5 integers and 5 double values.

```
import java.util.EmptyStackException;
public class Stack<T>{
    private int maxSize;
    private int top;
    private Object[] stackArray;

    public Stack(int size){
        maxSize = size;
        stackArray = new Object[maxSize];
        top = -1;
    }

    public void push(T value){
        if(top<maxSize-1){
            top++;
            stackArray[top] = value;
        }else{
            throw new RuntimeException
                ("Stack is full");
        }
    }

    @SuppressWarnings("unchecked")
    public T pop(){
```

```
if (!isEmpty()) {
    return (T) stackArray[top--];
} else {
    throw new EmptyStackException();
  }
}

public boolean isEmpty() {
    return (top == -1);
}

public int size() {
    return top+1;
}

public static void main(String args[])
  {
    Stack<Integer> intStack = new
        Stack<>(5);
    Stack<Double> doubleStack = new
        Stack<>(5);

    for (int i=0; i<5; i++){
        intStack.push(i);
        doubleStack.push((double) i);
    }
    System.out.println("Integer Stack:");
    for (int i=0; i<5; i++){
        System.out.println(intStack.pop()
```

```
System.out.println("Double Stack:");
for (int i=0; i<5; i++) {
    System.out.println(doubleStack.
        pop());
    }
}
```

Output:

Integer Stack:

4

3

2

1

0

Double Stack:

4.0

3.0

2.0

1.0

0.0

6/1/24

DOMS | Page No.
Date / /

1. Demonstrate various string constructor with proper java programs

```
class String {
    public static void main (String args[]) {
        char C[] = {'J', 'a', 'v', 'a'};
        String s1 = new String(c);
        String s2 = new String(s1)
        System.out.println(s1);
        System.out.println(s2);
```

Output:

~~JAV~~ Java
Java

8. Demonstrate startswith() to give output true or false.

```
public class Main {
    public static void main (String args[]) {
        String test = "testString";
        String pattern = "te";
        System.out.println (test.startswith
                (pattern));
        pattern = "est";
        System.out.println(test.startswith
                (pattern));
        }
    }
```

DOMS | Page No.
Date / /

Output :
True
False