

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque-book facility. The current account provides cheque book facility but no interest. Current account holders should 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 class 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.
 - e) Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
```

```
class Account {
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account (String customerName, int account  
    Number, String accountType, double balance) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = balance;  
    }
```

```
    void deposit (double amount) {
```

```
        balance += amount;
```

```
        System.out.println ("Deposit of " + amount +  
            " successful");
```

```
}
```

```
    void displayBalance () {
```

```
        System.out.println ("Balance: " + balance);
```

```
    void withdraw (double amount) {
```

```
        if (balance <= amount < 0) {
```

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

```
            return;
```

```
}
```

```
        balance -= amount;
```

```
        System.out.println ("Withdrawal of " + amount);
```

```
+ "successful");
```

```
}
```

```
class SavingAccount extends Account {  
    SavingAccount (String customerName, int  
        accountNumber, String accountType, double balance,  
        { super (CustomerName, accountNumber,  
            accountType, balance);  
    }
```

```
void compoundInterest () {
```

```
    double rate = 0.05;
```

```
    double time = 1.0;
```

```
    double interest = balance * Math. powe  
        (1 + rate, time) - balance;
```

```
    balance += interest;
```

```
    System.out.println ("Interest of " +  
        interest + " added").
```

```
}
```

```
void withdraw (double amount) {
```

```
    if (balance - amount >= 0) {
```

```
        System.out.println ("Withdrawal of "  
            + amount + " successful");
```

```
}
```

```
class CurrentAccount extends Account {
```

```
    double minimumBalance = 1000;
```

```
    double serviceCharge = 50;
```

```
CurrentAccount (String customerName,
```

The McGraw-Hill Companies

UNIX

Concepts and Applications

FOURTH EDITION

Sumitabha Das



data



save

005.43 DAS



54049

**Over
250 K
copies
sold**

```

int accountNumber, string accountType,
double balance () {
    super (customerName, accountNumber,
          accountType, balance);
}
}

```

```

void withdraw (double amount) {
    if (balance - amount < minimumBalance) {
        System.out.println ("Insufficient balance");
        return;
    }
    balance -= amount;
    System.out.println ("Withdrawal of " +
                        amount + " successful");
}
}

```

```

void imposeServiceCharge () {
    if (balance < minimumBalance) {
        balance -= serviceCharge;
        System.out.println ("Service charge of "
                            + serviceCharge + " imposed");
    }
}
}

```

```

public class Bank {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter customer name: ");
        String customerName = scanner.nextLine();
        System.out.print ("Enter account Number: ");
        int accountNumber = scanner.nextInt();
        System.out.print ("Enter account type
}
}

```

(savings / current); ");

String accountType = scanner.next();

System.out.print("Enter initial balance: ");
double balance = scanner.nextDouble();

Account account;

if (accountType.equals("savings")){
account = new Savings Account

(customerName, accountNumber, account
Type, balance);

} else {

account = new Current Account

(customerName, accountNumber,
accountType, balance);

}

while (true) {

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

System.out.println("2. Display balance");

System.out.println("3. Compute and
deposit interest");

System.out.println("4. Withdraw");

System.out.println("5. Exit");

System.out.print("Enter choice: ");

int choice = scanner.nextInt();

switch (choice){

case 1:

System.out.print("Enter amount
to deposit");

System.out.print(" ");

double amount = scanner.nextDouble();

account, deposit (amount);
break;

case 2:

account, display Balance ();
break;

case 3:

if (account instance of Savings Account) {
((Savings Account) account). compound
Interest ();
} else {

System.out.println ("Interest not
available for current account");
}

break;

case 4:

System.out.print ("Enter amount to
withdraw : ");

amount = scanner.next Double ();

account.withdraw (amount);

if (account instance of Current Account) {
((Current Account) account). impose Service
Charge ();

}

break;

case 5:

System.exit (0);

}

}

}

Output

Enter customer name: Praveen

Enter account number: 198

Enter account type (savings / current):
Savings

Enter initial balance: 1700

1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit

Enter choice: 4

Enter amount to withdraw: 500

withdrawal of 500.0 successful

Enter choice: 2

Balance: 1800.0

3/10/24