

// Develop a Java program to create a class bank that maintains 10 accounts for some customer names, account numbers, type of account. Accept deposit from customer & update the balance. Display the balance, compute and deposit interest, permit withdrawal and update the balance, check for the minimum balance. Impose penalty if needed.

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

```
abstract class Account {
```

```
    String customerName;
    long accountNumber;
    String accountType;
    double balance;
```

```
    public Account (String customerName, long
                    accountNumber, String accountType,
                    double balance) {
```

```
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
```

```
        this.accountType = accountType;
```

```
        this.balance = balance;
```

```
}
```

```
public void displayBalance () {
```

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

```
}
```

// Abstract method for withdrawal

```
public abstract void withdraw (double
                            amount);
```

```
}
```

class Current extends Account {
 double minimumBalance;
 double serviceCharge;

public Current(String customerName, long accountNumber, double balance) {
 super(customerName, accountNumber, "Current Account", balance);
 this.minimumBalance = 1000; // set minimum balance
 this.serviceCharge = 50; // set service charge
}

@Override

public void withdraw(double amount) {
 if (balance - amount >= minimumBalance) {
 balance -= amount;
 System.out.println("Withdrawal Successful
 Remaining balance: \$" + balance);
 } else {
 System.out.println("Insufficient funds
 Service charge of \$" + serviceCharge
 applied.");
 balance -= serviceCharge;
 System.out.println("Remaining balance
 after service charge: \$" + balance);
 }
}

class SavAcc extends Account {
 double interestRate;

public SavAcc(String customerName, long accountNumber, double balance) {

super(customerName, accountNumber, accountType : "Savings Account", balance);
 this.interestRate = 0.005; } }

{ }

@override

```
public void withdraw(double amount) {
    if (balance - amount >= 0) {
        balance -= amount;
        System.out.println("Withdrawal successful.");
        Remaining balance: $" + bal);
    }
}
```

{ }

else {

```
System.out.println("Insufficient funds.");
cannot complete withdrawal.";
```

{ }

```
public void depositInterest() {
```

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

```
System.out.println("Interest deposited, update
: $" + balance);
```

{ }

```
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 no: ");
long accountNumber = scanner.nextInt();
```

```
System.out.print("Enter account type ");
String accountType = scanner.next();
```

Account a account;

```
if (accountType.equals("Current")) {
    String string = "current";
```

```
    account = new CurrentAccount(customerName, accountNumber, initialBalance);
```

```
else if (accountType.equals("Savings")) {
    String string = "savings";
```

```
    account = new SavingsAccount(customerName, accountNumber, initialBalance);
```

} else {

```
System.out.println("Invalid account
type. Existing program.");
```

return;

}

int choice;

do {

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

```
System.out.println("2. Display Balance");
```

```
System.out.println("3. Deposit");
```

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

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

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

choice = scanner.nextInt();

switch (choice) {

Case 1 :

```
System.out.print("Enter deposit amount:");
double depositAmount = Scanner.nextDouble();
account.balance += depositAmount;
System.out.println("Deposit successful");
break;
```

Case 2 :

```
account.displayBalance();
break;
```

Case 3 :

```
? if (account instanceof SavAcct) {
    ((SavAcct) account).depositInterest();
}
```

else

```
System.out.println("only Sav acc");
```

?

break;

Case 4 :

```
System.out.print("Enter withdrawal amt ");
double withdrawalAmount = Scanner.nextDouble();
account.withdraw(withdrawalAmount);
break;
```

Case 5 :

```
System.out.println("Exiting Program.Goodbye!");
```

break;

No fault

```
System.out.println("invalid choice.");
```

? while (choice != 5);

? Scanner.close();

Open File .

Dare
Page

Enter customer name : Fagner

Enter account numn : 1234

Enter initial balance : 500

Enter account type : savings

1. Deposit

2. Display Balance

3. Deposit + Interest

4. withdraw

5. Exit.

Enter your choice, 5

Deposit successful. updated bal : \$ 3000.0

1. Display Balance .

2. Deposit + Interest

3. withdraw

4. Deposit

5. Exit

Enter your choice, 2 .

ACount Balance: \$ 3000.0

1. Deposit

2. Display Balance

3. Deposit + Interest

4. withdraw

5. Exit .

Enter your choice 5

Exiting program. Goodbye .

✓
09/01/20