

```
prgm -5
```

```
import java.util.Scanner;

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 deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful. Updated balance: $" + balance);
    }

    public void displayBalance() {
        System.out.println("Account Balance: $" + balance);
    }
}
```

```
}
```

```
class CurAccount extends Account {  
    double minBalance;  
    double serviceCharge;  
  
    public CurAccount(String customerName, long accountNumber, double balance, double  
minBalance, double serviceCharge) {  
        super(customerName, accountNumber, "Current", balance);  
        this.minBalance = minBalance;  
        this.serviceCharge = serviceCharge;  
    }  
  
    public void checkMinBalance() {  
        if (balance < minBalance) {  
            balance -= serviceCharge;  
            System.out.println("Minimum balance not maintained. Service charge of $" +  
serviceCharge + " imposed.");  
            displayBalance();  
        }  
    }  
  
    public void withdraw(double amount) {  
        if (amount > balance) {  
            System.out.println("Insufficient funds. Withdrawal failed.");  
        } else {
```

```
        balance -= amount;

        System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);

        checkMinBalance();

    }

}

}
```

```
class SavAccount extends Account {

    double interestRate;

    public SavAccount(String customerName, long accountNumber, double balance, double
interestRate) {

        super(customerName, accountNumber, "Savings", balance);

        this.interestRate = interestRate;

    }

    public void computeInterest() {

        double interest = balance * (interestRate / 100);

        balance += interest;

        System.out.println("Interest computed and deposited: $" + interest);

        displayBalance();

    }

}
```

```
public void withdraw(double amount) {

    if (amount > balance) {
```

```
        System.out.println("Insufficient funds. Withdrawal failed.");
    } else {
        balance -= amount;
        System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);
    }
}

}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        CurAccount currentAccount = new CurAccount("John Doe", 123456789, 1000, 500, 10);
        SavAccount savingsAccount = new SavAccount("Jane Doe", 987654321, 2000, 5);

        int choice;
        do {
            System.out.println("\nSelect an option:");
            System.out.println("1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Compute Interest (Savings Account only)");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
        }
```

```
switch (choice) {  
    case 1:  
        System.out.print("Enter amount to deposit: ");  
        double depositAmount = scanner.nextDouble();  
        System.out.print("Select account (1. Current, 2. Savings): ");  
        int accountType = scanner.nextInt();  
        if (accountType == 1) {  
            currentAccount.deposit(depositAmount);  
        } else if (accountType == 2) {  
            savingsAccount.deposit(depositAmount);  
        } else {  
            System.out.println("Invalid account type.");  
        }  
        break;  
    case 2:  
        System.out.print("Select account (1. Current, 2. Savings): ");  
        int accType = scanner.nextInt();  
        if (accType == 1) {  
            currentAccount.displayBalance();  
        } else if (accType == 2) {  
            savingsAccount.displayBalance();  
        } else {  
            System.out.println("Invalid account type.");  
        }  
        break;
```

```
case 3:  
    if (savingsAccount instanceof SavAccount) {  
        ((SavAccount) savingsAccount).computeInterest();  
    } else {  
        System.out.println("Invalid option for current account.");  
    }  
    break;  
  
case 4:  
    System.out.print("Enter amount to withdraw: ");  
    double withdrawAmount = scanner.nextDouble();  
    System.out.print("Select account (1. Current, 2. Savings): ");  
    int accTyp = scanner.nextInt();  
    if (accTyp == 1) {  
        currentAccount.withdraw(withdrawAmount);  
    } else if (accTyp == 2) {  
        savingsAccount.withdraw(withdrawAmount);  
    } else {  
        System.out.println("Invalid account type.");  
    }  
    break;  
  
case 5:  
    System.out.println("Exiting the program. Thank you!");  
    break;  
  
default:  
    System.out.println("Invalid choice. Please enter a valid option.");  
}
```

```
} while (choice != 5);

scanner.close();
}

}
```

Output :

Select an option:

- 1. Deposit
- 2. Display Balance
- 3. Compute Interest (Savings Account only)
- 4. Withdraw
- 5. Exit

Enter your choice: 1

Enter amount to deposit: 5000000

Select account (1. Current, 2. Savings): 2

Deposit of \$500000.0 successful. Updated balance: \$502000.0

Select an option:

- 1. Deposit
- 2. Display Balance
- 3. Compute Interest (Savings Account only)
- 4. Withdraw
- 5. Exit

Enter your choice: 2

Select account (1. Current, 2. Savings): 2

Account Balance: \$502000.0

Select an option:

- 1. Deposit
- 2. Display Balance
- 3. Compute Interest (Savings Account only)
- 4. Withdraw
- 5. Exit

Enter your choice: 3

Interest computed and deposited: \$25100.0

Account Balance: \$527100.0

Q. 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.

→ import java.util.Scanner;

```
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 deposit (double amount) {  
    balance += amount;  
    System.out.println ("Deposit of $" + amount  
    + " successful. updated balance : $" + balance);
```

```
}
```

system.out.print("Enter service charge for user account :");
double service_charge = scanner.nextDouble();

userAccount = new NYAccount(customerName,
System.currentTimeMillis(), initialAmount, service_charge);
{ else if (accountType == 2) {

System.out.print("Enter interest rate for the savings account :");
double interestRate = scanner.nextDouble();

userAccount = new SAVAccount(customerName, System.currentTimeMillis(), initialAmount, interestRate);
{ else {

System.out.println("Invalid account type choice.
Exiting the program.");

scanner.close();

return;

}

int choice;

do {

System.out.println("\nSelect an option:

System.out.println("1. Deposit\n2. Withdraw
Balance\n3. Compute Interest Savings Only
4. Withdraw\n5. Exit\n")

Enter choices") ;

choice = s.nextInt() + in.nextInt();

for the
double();
Name,
, min,

the savings
balance;
choice.
an option:
display
by
if
else
switch (choice) {
case 1:
System.out.print("Enter amount to deposit:");
double depositAmount = scanner.nextDouble();
System.out.print("Select account (1-current, 2-savings);
int accountType = scanner.nextInt();
if (accountType == 1) {
currentAccount.deposit(depositAmount);
} else if (accountType == 2) {
savingsAccount.deposit(depositAmount);
}
break;
case 2:
System.out.print("Select account (1-current, 2-
savings);");
int accType = scanner.nextInt();
if (accType == 1) {
currentAccount.displayBalance();
}
else if (accType == 2) {
savingsAccount.displayBalance();
}
else {
System.out.println("invalid account type");
}
break;
case 3:
if (savingsAccount instanceof SavingsAccount)
{
(SavingsAccount) savingsAccount).computeInterest();
}
else {
System.out.println("invalid option for current
account");
}
break;

```
*[public void deposit (double amount) {  
    balance += amount;  
    System.out.println("de")]
```

```
public void displayBalance () {  
    System.out.println("Account Balance : " + balan
```

```
public void withdraw (double amount) {  
    System.out.println("Withdrawal not success  
for this account type.");
```

```
}
```

```
}
```

```
class curAccount extends Account {  
    double minBalance;  
    double serviceCharge;
```

```
public curAccount (String customerName, long accountNumber, double minBalance, double serviceCharge)  
super (customerName, accountNumber, "current",  
this.minBalance = minBalance;  
this.serviceCharge = serviceCharge;
```

```
}
```

```
public void checkMinBalance () {
```

```
if (balance < minBalance) {  
    balance -= serviceCharge;
```

~~scraper~~

```
System.out.println("Minimum balance not met  
service charge + " imposed.");
```

```
displayBalance();
```

```
}
```

```
@override
```

```
        balance) {  
    System.out.println("Insufficient funds. Withdraw  
-aval failed.");
```

```
} else {
```

```
    balance -= amount;
```

```
    System.out.println("Withdrawal of $" + amount  
+ " successful. Updated balance : $" + balance);
```

```
}
```

```
public class Bank {
```

```
    public static void main (String [] args) {
```

```
        Scanner scanner = new Scanner (System.in);
```

```
        System.out.print ("Enter your name:");
```

```
        String customerName = scanner.nextLine();
```

```
int
```

```
{
```

```
else:
```

```
        System.out.print ("Enter initial amount:");
```

```
        double initialAmount = scanner.nextDouble();
```

```
        System.out.print ("Select account type (1. current, 2.  
savings):");
```

```
        int accountTypeChoice = scanner.nextInt();
```

```
        Account userAccount = null;
```

```
        if (accountTypeChoice == 1) {
```

```
            System.out.print ("Enter minimum balance for  
the current account:");
```

```
            double minBalance = scanner.nextDouble();
```

```

case 4:
    System.out.print("Enter amount to withdraw:");
    double withdrawAmount = scanner.nextDouble();
    System.out.print("Select account (1. current, 2. savings");
    int accTyp = scanner.nextInt();
    if (accTyp == 2) {
        savingsAccount.withdraw(withdrawAmount);
    } else {
        System.out.println("Invalid account type");
    }
    break;
case 5:
    System.out.println("Exiting the program.");
    System.out.println("Thank you!");
    break;
default:
    System.out.println("Invalid choice. Please enter a valid option.");
}
} while (choice != 5);
scanner.close();
}

```

→ Output:

select an option : 1. deposit

- 2. display balance
- 3. compute interest
- 4. withdraw
- 5. exit

Enter your choice : 1.

Enter amount to deposit : 500

select acc
deposit

- 1. select acc
- 2. deposit
- 3. display balance
- 4. compute
- 5. withdraw

Enter your choice :

select

Account

- 1. deposit
- 2. display balance
- 3. compute
- 4. withdraw
- 5. exit

Enter your choice :

Date _____
Page _____

select account (1. current, 2. savings) : 1
deposit of 500 successful.

select an option :

1. deposit
2. display balance.
3. compute interest
4. withdraw
5. Exit

Enter your choice : 2.

Select account (1. current, 2. savings) : 1

Account Balance : 15000.

select an option :

1. deposit
2. display balance
3. compute interest
4. withdraw
5. Exit

Enter your choice : 5

Exiting the program

Thank you!