

```

package com.company;

import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        ArrayList<Customer> customers = new ArrayList<>();

        customers.add(new Customer("Adam"));
        customers.add(new Customer("Sara"));
        customers.add(new Customer("Billy"));
        new BankLogic(customers);
    }
}

package com.company;

import java.util.ArrayList;
import java.util.Scanner;

public class BankLogic {

    public BankLogic(ArrayList<Customer> customers) {
        ArrayList<Account> currentAccounts = new ArrayList<>();

        do {
            System.out.println("Please enter your customer ID here: ");
            int customerId = userInt();

            for (int i = 0; i < customers.size(); i++) { //Searching through
the customers list
                if (customers.get(i).getId() == (customerId)) { //customer
is found
                    Customer current = customers.get(i);
                    currentAccounts = current.accounts;
                    System.out.println("Welcome to the Bank " +
current.getName() + "! Hope you are having a wonderful day!");
                    menuManager(currentAccounts);
                    break;
                }
            }
            System.out.println("Customer not found, please try again");
        } while(true);
    }

    public static void menuManager(ArrayList<Account> currentAccounts){
        int menuOption = -1;
        do {
            displayMenu();
            menuOption = userInt();

            switch (menuOption) {
                case 1:
                    //add a new savings Account
                    currentAccounts = addSavingsAcc(currentAccounts);
                    break;
                case 2:
                    //add a new credit Account
                    currentAccounts = addCreditAcc(currentAccounts);

```

```

        break;
    case 3:
        //show all accounts
        showAccounts(currentAccounts);
        break;
    case 4:
        //deposit money
        currentAccounts = deposit(currentAccounts);
        break;
    case 5:
        //withdraw money
        currentAccounts = withdraw(currentAccounts);
        break;
    case 0:
        //end
        System.out.println("Thank you for visiting the Bank!");
        break;
    default:
        System.out.println("Choose a menu option between 1-
5.");
        break;
    }
} while (menuOption != 0);
}

public static void displayMenu() {
    System.out.println(
        "\nPlease choose one of the options below: \n" +
        "1. Create a saving account \n" +
        "2. Create a credit account \n" +
        "3. View all accounts \n" +
        "4. Deposit money \n" +
        "5. Withdraw money \n" +
        "0. Exit \n");
}

public static ArrayList<Account>
deposit(ArrayList<Account>currentAccounts){
    System.out.println("Please enter the account number of the account
you want to deposit money to: ");
    int accNr = userInt();
    System.out.println("Please enter how much money you want to
deposit: ");
    double deposit = userInt();
    for(Account account: currentAccounts){
        if (account.getAccountNumber() == (accNr)) {
            account.depositMoney(deposit);
        }
    }
    return currentAccounts;
}

public static void showAccounts(ArrayList<Account>currentAccounts){
    for(Account account: currentAccounts) {
        System.out.println(account.toString());
    }
}

```

```

        public static ArrayList<Account>
withdraw(ArrayList<Account>currentAccounts){
    double withdraw;

    System.out.println("Please enter the account number of the account
you want to withdraw money from: ");
    int accNr = userInt();
    System.out.println("Please enter how much money you want to
withdraw: ");
    withdraw = userInt();

    for(Account account: currentAccounts){
        if (account.getAccountNumber() == (accNr)) {
            account.withdrawMoney(withdraw);
        }
    }
    return currentAccounts;
}

    public static ArrayList<Account>
addCreditAcc(ArrayList<Account>currentAccounts){
    int accNr = generateAccNr();
    double balance = 0;

    currentAccounts.add(new CreditAccount(balance, accNr));
    System.out.println("A Credit account with the number - " + accNr +
        " has been created and currently has a balance of " +
balance + " SEK");
    return currentAccounts;
}

    public static ArrayList<Account>
addSavingsAcc(ArrayList<Account>currentAccounts){
    int accNr = generateAccNr();
    System.out.println("Please enter the amount of money you want to
deposit onto your new account: ");
    double balance = userInt();

    currentAccounts.add(new SavingsAccount(balance, accNr));
    System.out.println("A Savings account with the number - " + accNr +
        " has been created and currently has a balance of " +
balance + " SEK");
    return currentAccounts;
}

    public static int userInt(){
        do {
            try {
                Scanner sc = new Scanner(System.in);
                return sc.nextInt();
            } catch (Exception e) {
                System.out.println("Something went wrong, try again with a
positive whole number! ");
            }
        }while(true);
    }
}

```

```

        public static int generateAccNr() {
            int accNr = (int) (Math.random() * 10000);
            return accNr;
        }
    }
}

package com.company;

public abstract class Account {
    private double balance;
    private int accountNumber;

    public Account(double balance, int accountNumber) {
        setBalance(balance);
        setAccountNumber(accountNumber);
    }

    public double getBalance() {
        return balance;
    }

    public int getAccountNumber() {
        return accountNumber;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void depositMoney(double depositAmount) {
        this.balance += depositAmount;
    }

    public void withdrawMoney(double withdrawAmount) {
        this.balance -= withdrawAmount;
    }

    @Override
    public String toString() {
        return "Account{" +
            "balance=" + balance +
            ", accountNumber=" + accountNumber +
            '}';
    }
}

package com.company;

public class SavingsAccount extends Account {
    private double interestRate;
    private String accountType = "Savingsaccount";

    public SavingsAccount(double balance, int accountNumber) {
        super(balance, accountNumber);
        this.interestRate = 1; // in %
    }

    public void depositMoney(double depositAmount) {

```

```

        double newBalance = getBalance() + depositAmount;
        setBalance(newBalance);
    }

    public void withdrawMoney(double withdrawAmount){
        if((getBalance()-withdrawAmount) > 0) {
            double newBalance = getBalance() - withdrawAmount;
            setBalance(newBalance);
        }
        else{
            System.out.println("Not enough money on this account, try
again!");
        }
    }

    public double calcInterest(){
        double interest = getBalance() * (interestRate/100);
        return interest;
    }

    @Override
    public String toString() {
        return ("\nInformation about the savings account: " +
            "\naccount number: " + getAccountNumber() +
            "\nbalance: " + getBalance() +
            "\naccount type: " + accountType +
            "\ninterest rate: " + interestRate + "%" +
            "\ninterest on your balance right now: " + calcInterest());
    }
}

package com.company;

public class CreditAccount extends Account {
    private int creditLimit;
    private double interestRate;
    private String accountType = "Creditaccount";

    public CreditAccount(double balance, int accountNumber){
        super(balance, accountNumber);
        this.creditLimit = -5000;
        this.interestRate = 0.5; // in %
        balance = 0;
    }

    public void depositMoney(double depositAmount){
        double newBalance = getBalance() + depositAmount -
(depositAmount*(interestRate/100));
        setBalance(newBalance);
    }

    public void withdrawMoney(double withdrawAmount){
        if((getBalance()-withdrawAmount) < creditLimit) {
            double newBalance = getBalance() - withdrawAmount;
            setBalance(newBalance);
        }
        else{
            System.out.println("Not enough credit, try again!");
        }
    }
}

```

```

        @Override
        public String toString() {
            return ("\nInformation about the savings account: " +
                    "\naccount number: " + getAccountNumber() +
                    "\nbalance: " + getBalance() +
                    "\naccount type: " + accountType +
                    "\ninterest rate: " + interestRate + "%");
        }
    }
}

package com.company;

import java.util.ArrayList;

public class Customer {

    private String name;
    private int id;
    public ArrayList<Account> accounts; //List with Account classes
    private static int countCustomers; //Every time a new object is
    created it is counted. Helps us with the user id's.

    public Customer(String name){
        setName(name);
        setId();
        accounts = new ArrayList<>();
    }

    public String getName() {
        return name;
    }

    public ArrayList<Account> getAccounts() {
        return accounts;
    }

    public int getId() {
        return id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAccounts(ArrayList<Account> accounts) {
        this.accounts = accounts;
    }

    public void setId() {
        this.countCustomers = countCustomers+1;
        this.id = countCustomers;
    }
}

```