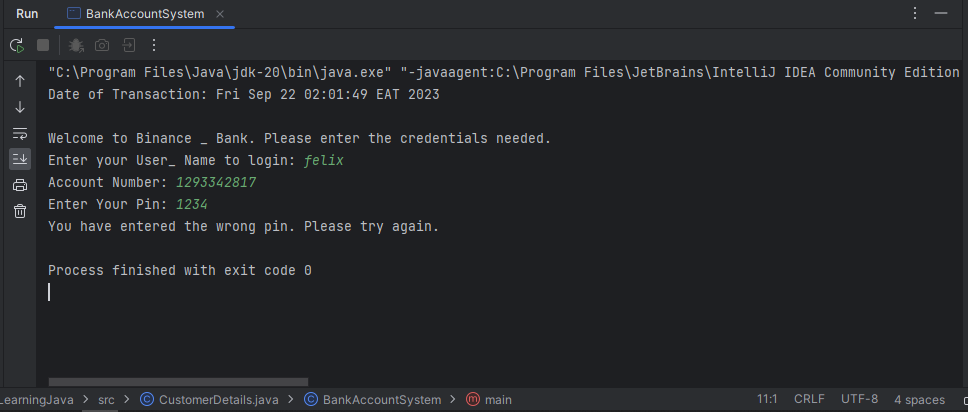
if (details.getPin() == CustomerDetails.*defaultPin*) {  
 details.currentBalance();  
  
 TransactionsMenu menu = new TransactionsMenu();  
 menu.userMenu();  
  
 BankAccountSystem system = new BankAccountSystem();  
 system.setNumber(system.getNumber());  
  
 switch (system.getNumber()) {  
 case 1: {  
 menu.depositCash(menu.getDepositedCash());  
 break;  
 }  
 case 2: {  
 menu.sendMoney(menu.getAmountSent(), menu.getRecipientAccNumber(), menu.getReceiverName());  
 break;  
 }  
 case 3: {  
 menu.withdrawCash(menu.getCashWithdrawn());  
 break;  
 }  
 case 4: {  
 menu.savingsAccount(menu.getSavingsAmount());  
 }  
 case 5: {  
 menu.getAccountBalance();  
 break;  
 }  
 case 6:  
 break;  
 default:  
 System.*out*.println("Enter appropriate value...");  
 }  
}  
else {  
 System.*out*.println("You have entered the wrong pin. Please try again. ");

Nikirun hii program ya hii code, inaniletea hii… Unaona wapi ni wrong, siradi. Inafaa inilete hizo options zikokwa switch juu niliset default pin 1234. Kama vle nimeandika hapo.



import java.text.NumberFormat;  
import java.util.Date;  
import java.util.Scanner;  
  
public class CustomerDetails {  
 static final short *defaultPin* = 1234;  
 String customerName;  
 private int accountNumber;  
 public short pin;  
  
 Scanner scan = new Scanner(System.*in*);  
  
 public CustomerDetails () {  
 Date today = new Date();  
 System.*out*.println("Date of Transaction: " + today + "\n");  
 System.*out*.println("Welcome to Binance \_ Bank. Please enter the credentials needed.");  
  
 }  
  
 public void setDetails (String customerName ,int accountNumber) {  
 System.*out*.print("Enter your User\_ Name to login: ");  
 customerName = scan.nextLine();  
 System.*out*.print("Account Number: ");  
 accountNumber = scan.nextInt();  
 this.customerName = customerName;  
 this.accountNumber = accountNumber;  
 }  
 public void setPin (short pin) {  
 System.*out*.print("Enter Your Pin: ");  
 pin = scan.nextShort();  
 }  
  
 public String getCustomerName() {  
 return customerName;  
 }  
  
 public int getAccountNumber() {  
 return accountNumber;  
 }  
  
 public short getPin() {  
 return pin;  
 }  
 public void currentBalance () {  
 System.*out*.println("Your current balance is: Ksh 0.00 \n");  
 }  
}  
  
class TransactionsMenu extends CustomerDetails{  
 private double accountBalance = 0.00;  
 static final float *overdraft* = 50\_000;  
 private double totalBalance = *overdraft* + accountBalance;  
 private double savings;  
 double cashDeposit;  
 double amtSend;  
 int recipientAccNumber;  
 String receiverName;  
 double cashWithdrawn;  
 double savingsAmount;  
  
  
 public TransactionsMenu() {  
 System.*out*.println("Welcome to your Binance Digital Menu " + getCustomerName());  
 }  
 public void userMenu () {  
 System.*out*.println("To navigate through the bank menu, select the number of your choice.\n" + " For example: [1] or [2]... You get the point.");  
 String [] transactions = {"1. Deposit Cash", "2. Send Money", "3. Withdraw Cash", "4. Savings", "5. Check Balance", "6. Exit Application"};  
 for(String transaction : transactions) {  
 System.*out*.println(transaction + "\n");  
 }  
 }  
 public void depositCash (double cashDeposit) {  
 System.*out*.print("Enter amount to deposit in your account: ");  
 this.cashDeposit = scan.nextDouble();  
 accountBalance += cashDeposit;  
 System.*out*.println("Congratulations, you have successfully deposited " + NumberFormat.*getCurrencyInstance*().format(cashDeposit) + " Your account balance is: " + NumberFormat.*getCurrencyInstance*().format(accountBalance));  
  
 }  
 public double getDepositedCash () {  
 return this.cashDeposit;  
 }  
 public void sendMoney (double amtSend, int recipientAccNumber, String receiverName) {  
 System.*out*.print("Please enter the name of the recipient: ");  
 receiverName = scan.nextLine();  
 System.*out*.print("Account Number: ");  
 recipientAccNumber = scan.nextInt();  
 System.*out*.println("Enter the amount to send: ");  
 amtSend = scan.nextDouble();  
 if (amtSend <= accountBalance) {  
 accountBalance -= amtSend;  
 System.*out*.println("Congratulations, you have successfully sent " + NumberFormat.*getCurrencyInstance*().format(amtSend) + " to " + receiverName + ". Acc/ No: " + recipientAccNumber + ". Your balance is: " + NumberFormat.*getCurrencyInstance*().format(accountBalance));  
 }  
 else {  
 accountBalance = 0.00;  
 totalBalance -= amtSend;  
 System.*out*.println("Congratulations, you have successfully sent " + NumberFormat.*getCurrencyInstance*().format(amtSend) + " to " + receiverName + ". Acc/ No: " + recipientAccNumber + ". Your balance is: " + NumberFormat.*getCurrencyInstance*().format(accountBalance));  
  
 }  
 }  
 public double getAmountSent () {  
 return this.amtSend;  
 }  
 public int getRecipientAccNumber () {  
 return this.recipientAccNumber;  
 }  
 public String getReceiverName () {  
 return this.receiverName;  
 }  
 public void withdrawCash (double cashWithdrawn) {  
 System.*out*.print("Enter amount to withdraw: ");  
 cashWithdrawn = scan.nextDouble();  
 if (cashWithdrawn <= accountBalance) {  
 accountBalance -= cashWithdrawn;  
 System.*out*.println("Congratulations, you have successfully withdrawn " + NumberFormat.*getCurrencyInstance*().format(cashWithdrawn) + " to your account. " + ". Your balance is: " + NumberFormat.*getCurrencyInstance*().format(accountBalance));  
 } else {  
 accountBalance = 0.00;  
 totalBalance -= cashWithdrawn;  
 System.*out*.println("Congratulations, you have successfully withdrawn " + NumberFormat.*getCurrencyInstance*().format(cashWithdrawn) + " to your account. " + ". Your balance is: " + NumberFormat.*getCurrencyInstance*().format(accountBalance));  
  
 }  
 }  
 public double getCashWithdrawn () {  
 return this.cashWithdrawn;  
 }  
 public void savingsAccount (double savingsAmount) {  
 System.*out*.print("Enter amount you wish to save: ");  
 savingsAmount = scan.nextDouble();  
 if (savingsAmount <= accountBalance) {  
 accountBalance -= savingsAmount;  
 savings = savingsAmount;  
 System.*out*.println("Congratulations you've successfully deposited " + NumberFormat.*getCurrencyInstance*().format(savingsAmount) + " into your savings accounts. Saving balance is = " + NumberFormat.*getCurrencyInstance*().format(savings) );  
  
 }  
 else{  
 System.*out*.println("You have insufficient funds to complete the transaction.\tFirst, deposit funds...");  
  
 }  
 }  
 public double getSavingsAmount () {  
 return this.savingsAmount;  
 }  
 public void getAccountBalance () {  
 System.*out*.println("Your account balance is as follow: \n Account Balance : " + NumberFormat.*getCurrencyInstance*().format(accountBalance) + "\n" + " Savings balance: " + NumberFormat.*getCurrencyInstance*().format(savings));  
 }  
}  
  
class BankAccountSystem extends CustomerDetails{  
 public byte number;  
 public void setNumber (byte number) {  
 System.*out*.print("Enter Your preferred selection: ");  
 number = scan.nextByte();  
 this.number = number;  
 }  
 public byte getNumber () {  
 return this.number;  
 }  
 public static void main (String [] args) {  
 CustomerDetails details = new CustomerDetails();  
 details.setDetails(details.getCustomerName(), details.getAccountNumber());  
 details.setPin(details.getPin());  
 details.currentBalance();  
  
 TransactionsMenu menu = new TransactionsMenu();  
 menu.userMenu();  
  
 BankAccountSystem system = new BankAccountSystem();  
 system.setNumber(system.getNumber());  
  
 switch (system.getNumber()) {  
 case 1: {  
 menu.depositCash(menu.getDepositedCash());  
 break;  
 }  
 case 2: {  
 menu.sendMoney(menu.getAmountSent(), menu.getRecipientAccNumber(), menu.getReceiverName());  
 break;  
 }  
 case 3: {  
 menu.withdrawCash(menu.getCashWithdrawn());  
 break;  
 }  
 case 4: {  
 menu.savingsAccount(menu.getSavingsAmount());  
 }  
 case 5: {  
 menu.getAccountBalance();  
 break;  
 }  
 case 6:  
 break;  
 default:  
 System.*out*.println("Enter appropriate value...");  
 }  
 }