```
java
import java.io.*;
import java.util.*;
class Account implements Serializable {
private static final long serialVersionUID = 1L;
String owner;
int balance;
List<String> history = new ArrayList<>();
public Account(String owner) {
this.owner = owner;
this.balance = 0;
history.add("Account created");
}
public void deposit(int amount) {
balance += amount;
history.add("Deposit: " + amount + " (balance: " + balance + ")");
}
public boolean withdraw(int amount) {
if (amount > balance) return false;
balance -= amount;
history.add("Withdraw: " + amount + " (balance: " + balance + ")");
return true;
}
}
public class BankApp {
private static final String DATA_FILE = "account.ser";
private Account account;
private Scanner scanner = new Scanner(System.in);
public static void main(String[] args) {
BankApp app = new BankApp();
app.load();
app.run();
}
private void run() {
if (account == null) {
System.out.print("Owner name for new account: ");
String name = scanner.nextLine().trim();
account = new Account(name.isEmpty() ? "Unknown" : name);
}
while (true) {
printMenu();
String choice = scanner.nextLine().trim();
switch (choice) {
case "1": showInfo(); break;
case "2": deposit(); break;
case "3": withdraw(); break;
case "4": showHistory(); break;
case "0": save(); System.out.println("Bye."); return;
default: System.out.println("Unknown menu.");
}
}
}
private void printMenu() {
System.out.println("\n==== Mini Bank ====");
System.out.println("1. Show account info");
System.out.println("2. Deposit");
System.out.println("3. Withdraw");
System.out.println("4. Show history");
System.out.println("0. Exit");
System.out.print("Select: ");
}
private void showInfo() {
System.out.println("Owner: " + account.owner);
System.out.println("Balance: " + account.balance);
}
private void deposit() {
System.out.print("Amount to deposit: ");
String v = scanner.nextLine().trim();
try {
int amount = Integer.parseInt(v);
if (amount <= 0) {
System.out.println("Must be positive.");
return;
}
account.deposit(amount);
System.out.println("Deposited.");
} catch (NumberFormatException e) {
System.out.println("Invalid number.");
}
}
private void withdraw() {
System.out.print("Amount to withdraw: ");
String v = scanner.nextLine().trim();
try {
int amount = Integer.parseInt(v);
if (amount <= 0) {
System.out.println("Must be positive.");
return;
}
if (!account.withdraw(amount)) {
System.out.println("Not enough balance.");
} else {
System.out.println("Withdrawn.");
}
} catch (NumberFormatException e) {
System.out.println("Invalid number.");
}
}
private void showHistory() {
System.out.println("\n[History]");
for (String h : account.history) {
System.out.println(h);
}
}
private void load() {
File f = new File(DATA_FILE);
if (!f.exists()) return;
try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(f))) {
account = (Account) ois.readObject();
System.out.println("Loaded account of " + account.owner);
} catch (Exception e) {
System.out.println("Failed to load account: " + e.getMessage());
}
}
private void save() {
try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(DATA_FILE))) {
oos.writeObject(account);
System.out.println("Saved account.");
} catch (IOException e) {
System.out.println("Failed to save account: " + e.getMessage());
}
}
}
```