



دانشگاه صنعتی شریف

دانشکده مهندسی برق

برنامه نویسی شی گرا

گزارش شریف بانک

تهیه کننده :

آرمان لطفعلی خانی 99109166

کلاس Bank :

ابتدا چند توضیح درباره عملکرد بانک ضروری است:

در تراز بانک هم بدهی ها و هم سرمایه بانک ثبت می شوند و از تفاضل این ها تراز بدست می آید. در نتیجه با افتتاح حساب تراز بانک عوض نمی شود. منابع افزایش تراز خالص، سرمایه گذاری به صورت درصدی از سرمایه بانک و سود وام هاست. منابع کاهش آن نیز پرداخت سود سپرده ها تعیین شده است. در ضمن، شرط ورشکستگی بانک منفی شدن تراز خالص آن در نظر گرفته شده و نه تمام شدن سرمایه آن. همچنین، در تراز بانکی تعهدات لحظه ای بانک در نظر گرفته شده اند و تعهدات آتی (سود سپرده ها در ماه های آینده) در این محاسبه منظور نمی شوند.

در مورد محاسبه سود یا قسط وام، حذف بخش اعشاری ممکن است نتایج را اندکی با برخی روش های محاسباتی متفاوت کند که باید به آن در تست کیس ها دقت شود.

در هنگام ورشکستگی بانک، روند به روز رسانی به این صورت است:

الف) بانک با باقی مانده پول اجازه سرمایه گذاری ندارد.

ب) مبلغ حساب های جاری نه از طریق قسط وام و نه از طریق سود سپرده عوض نخواهد شد.

ج) در مدت ورشکستگی بانک، سپرده ها به روال قبل جلو رفته و در تاریخ خود منقضی می شوند اما پول پرداخت نشده توسط بانک به عنوان بدهی بانک به حساب جاری ثبت شده سپس به هنگام بازگشایی بانک (پس از دریافت پول کافی) پرداخت خواهد شد.

د) مهلت وام با گذشت هر ماه کامل، یک ماه تمدید خواهد شد. بانک همچنان به کسر قسط از حساب های جاری ادامه خواهد داد، ولی در این مدت حساب های هیچ شخصی به علت پرداخت نکردن وام مسدود نخواهد شد و جریمه تاخیر نیز تعلق نخواهد گرفت. همچنین در حالت هایی که یک حساب جاری هم سود سپرده دارد و هم قسط از آن کم می شود، قسط در صورت لزوم می تواند از بدهی بانک به حساب (متغیر bankDebt) پرداخت شود. این پرداخت نیز در لیست پرداختی های وام ثبت خواهد شد.

توضیح توابع مختلف بانک:

```
private Bank(BankList b)
private Bank(BankList b, long initialBalance)
```

کانستراکتورهای بانک که به ترتیب با سرمایه اولیه پیش فرض و سرمایه اولیه تعیین شده کار می کنند.

```
public static boolean newBank(String name)
public static boolean newBank(String name, long initialBalance)
```

enum در وجود بانک (وجود در لیست بانک‌های معتبر) این‌جا پیاده‌سازی شده است.

```
public static boolean setInvestmentProfit(String bankName, int investmentProfit)
public static boolean setShortTimeProfit(String bankName, int shortTimeProfit)
public static boolean setLongTimeProfit(String bankName, int longTimeProfit)
public static boolean setMoney(String bankName, long money)
```

توابع setter برای تنظیم نرخ سودهای بانک و افزایش سرمایه. در تابع افزایش سرمایه شرطی برای بازگشایی بانک (در صورت ورشکستگی) نیز وجود دارد.

```
public static void showBankInterest(String bankName)
public static void showBankBalance(String bankName)
```

توابع نمایش تراز و سود سرمایه‌گذاری بانک.

```
private static String primaryChecks(String bankName, String ID)
```

بررسی‌های اولیه معتبر بودن اسم بانک، ورشکسته نبودن بانک و مسدود نبودن عملیات بانکی. در توابع دیگر به کرات استفاده می‌شود.

```
public static String createCurrentAccount(String bankName, String ID, long money)
```

پس از انجام بررسی‌های اولیه و بررسی شرط سنی، حساب جاری برای شخص در بانک باز خواهد شد. اطلاعات تمام مپ‌های بانک مرکزی و بانک نیز به روز خواهند شد. در میان این فرایند یک کارت نیز برای شخص تولید می‌شود.

```
public static String createDepositAccount(String bankName, String ID, String duration, long money)
```

با دریافت نوع حساب سپرده از جهت مدت زمانی، پس از بررسی‌های اولیه شامل شرط سنی، اقدام به تشکیل حساب سپرده خواهد کرد. اگر شخص حساب جاری در آن بانک نداشته باشد، با تابع `createCurrentAccount` ابتدا برایش حساب جاری ساخته می‌شود.

```
public static void depositMoney(String bankName, long ID, long money)
```

پرداخت پول به حساب (از طریق مراجعه به بانک) را انجام می‌دهد.

```
public static void withdraw(String bankName, long  
accountNum, String ID, long money)
```

برداشت (نامحدود) از حساب را مدیریت می کند. خطاهای معمول نیز همانند قبل در آن بررسی می شوند.

```
public static void accountTransact(String bankName, long  
senderNum, String ID, long receiverNum, long amount)
```

انتقال از یک حساب به حسابی دیگر را مدیریت می کند. درون آن، تابعی با همین نام از بانک مرکزی صدا می شود که بقیه کار را به عهده می گیرد.

```
public static boolean cardExtend(String bankName, long cardNum,  
String ID)
```

این تابع، تمدید تاریخ انقضای کارت را انجام می دهد. بررسی های اولیه اینجا انجام شده ادامه کار در تابعی در کلاس Card به نام extend انجام می شود.

```
public static String removeAccount(String bankName, String ID,  
long accountNum)
```

حذف حساب از این طریق انجام می شود. حساب مورد نظر نباید به حساب سپرده یا وامی مرتبط باشد در غیر این صورت پیام خطا چاپ خواهد شد.

```
public static String getNewLoan(String bankName, String ID, long  
money)
```

گرفتن وام توسط این دستور انجام می شود. بررسی شرط سنی و حداکثر مبلغ نیز در این متد پیاده سازی شده اند.

```
private static int ifExists(String bankName)
```

تابعی داخلی و پرکاربرد برای سرچ یک اسم بانک در لیست بانک های ایجاد شده است. بیشترین استفاده را در primaryChecks دارد.

```
public void show()
```

چاپ اطلاعات هر بانک

```
private boolean deduct(long money)
```

کسر مبلغ از بانک را انجام می دهد. در صورت عدم توانایی این کار، پیام ورشکستگی بانک چاپ می شود.

```
void addOrDeduct(long money)
```

تابعی کوتاه برای نظم پیدا کردن تغییرات تراز بانک.

```
public void update()
```

تمامی به روز رسانی ها به صورت روزانه در این تابع انجام می شود. ابتدا سود سرمایه گذاری در آخر ماه (به صورت پله ای) واریز می شود سپس قسط وام ها دریافت شده و در صورت تسویه وام، بانک آن را حذف می کند. بعد

از آن به سود سپرده‌ها رسیدگی می‌شود. پیاده سازی به روز رسانی در شرایط ورشکستگی نیز در این تابع پیاده سازی شده است.

```
public void reopen()
```

بازگشایی بانک پس از ورشکستگی (در صورت دریافت پول جدید) را پیاده سازی می‌کند. بدهی همه حساب‌هایی که از بانک طلب داشته‌اند به آنها پرداخت شده سپس بانک آغاز به کار دوباره می‌کند.

```
private void removeLoan(Loan loan)
```

پیاده سازی حذف وام در متد update.

کلاس CentralBank:

اطلاعات همه حساب‌ها، وام‌ها و کارت‌ها در این کلاس قرار دارند.
توابع موجود:

```
public static CentralBank getInstance()
```

طراحی سینگلتون برای این کلاس به جهت یکتایی بانک مرکزی

```
public static long IDGenerator(Bank b)
```

ساخت شماره حساب طبق دستورالعمل صورت سوال

```
public static void CardGenerator(Card c)
```

تولید رمز (اول) و CVV2 برای کارت

```
public void cardTransact(long senderId, long receiverID, long amount)
```

بخش مربوط به کارت دریافت کننده در انتقال کارت به کارت اینجا پیاده سازی شده است.

```
public boolean accountTransact(long senderId, long receiverID, long amount)
```

همانند تابع قبلی، بخش مربوط انتقال پول بین حساب‌ها از سمت بانک مرکزی را پیاده سازی می‌کند.

```
public static void getCardMoney(long cardNum, int password)
```

نمایش موجودی حساب جاری متصل به کارت در صورت معتبر بودن شماره کارت

```
public void update()
```

به روز رسانی همه بانک‌ها

```
public static void showLoans()
```

```
public static void showAccounts()  
public static void showBanks()  
public static void showCentralBankBalance()
```

توابع نمایش همه بانک‌ها، وام‌ها، حساب‌ها و تراز بانک مرکزی.

کلاس Account:

کلاس پدر برای حساب‌های جاری و سپرده به همراه یک تابع مشترک بین کلاس‌ها.

```
abstract long close();  
void show()
```

تابع show برای چاپ اطلاعات است.

کلاس CurrentAccount:

کلاسی برای اطلاعات حساب‌های جاری که از Account ارث می‌برد.

```
CurrentAccount(Member holder, long money, Bank bank)
```

کانستراکتور

```
void addLoan(Loan l)
```

اضافه کردن وام به مشخصات دارنده حساب و حساب

```
boolean withdraw(long money)
```

برداشت پول از حساب

```
long close()
```

پیاده سازی تابع بسته شدن حساب

```
void add(long money)
```

واریز به حساب

```
void show()
```

نشان دادن اطلاعات حساب. از تابع show کلاس پدر نیز استفاده می‌کند.

کلاس DepositAccount:

کلاسی برای ذخیره اطلاعات حساب‌های سپرده است که از کلاس **Account** ارث می‌برد.

```
DepositAccount(Member holder,String duration, long money, Bank bank, CurrentAccount currentAccount)
```

کانستراکتور حساب سپرده

```
long close()
```

پیاده سازی منطق بسته شدن حساب با احتساب جریمه احتمالی در برداشت زود هنگام. مبلغ حساب را با کسر جریمه احتمالی پس می‌دهد.

```
long getPay()
```

getter برای محاسبه سود سپرده

```
void show()
```

نمایش اطلاعات حساب سپرده را انجام می‌دهد. درون خود از متد کلاس پدر استفاده می‌کند.

کلاس **Member**: (abstract class)

اعضای جامعه را به همراه ویژگی‌های آنها ذخیره می‌کند. برخی از متغیرهای کلاس:

```
long debt;  
long money;  
LocalDate startDay;  
boolean isBanned;
```

بدهی: هرگاه شخص وام بگیرد به بدهی‌اش اضافه می‌شود. تاریخ آغاز نیز بسته به نوع عضو، تاریخ تولد یا تاریخ تاسیس شرکت می‌باشد.

توابع:

```
boolean addAccount(Account intended)  
boolean closeAccount(Account intended)
```

توابع متناظر با افتتاح و بستن حساب در این کلاس می‌باشند. مواردی مانند حذف و اضافه به لیست حساب‌های هر آبجکت از کلاس‌های فرزند **Member** را مدیریت می‌کنند.

```
void show()
```

نمایش اطلاعات هر فرد یا شرکت

```
static boolean payLoan
```

تابع پرداخت وام به صورت دستی. برنامه پرداخت وام دستی از این جا آغاز می‌شود.

```
public static void showLoan
```

پیاده سازی دستور نمایش وام با کد ملی (یا کد شرکت)

```
public static void showAccounts
```

پیاده سازی دستور نمایش حساب با کد ملی یا کد شرکت

کلاس Company:

از کلاس member ارث می برد.

```
private Company(String ID, Person manager)
```

کانستراکتور

```
static String createCompany(String name, String nationalCode)
```

دستور ساخت شرکت است که از کانستراکتور داخل کلاس بهره می برد

```
public static String createID(String name)
```

این دستور اسم شرکت را گرفته مطابق الگو آیدی تولید می کند.

```
void show()
```

اطلاعات شرکت را نمایش می دهد. داخل آن از تابع کلاس پدر نیز استفاده می شود.

```
static void showCompanies()
```

نمایش اطلاعات تمام شرکت ها

کلاس Person:

```
private Person(String firstName, String lastname, String  
nationalCode, String date)
```

کانستراکتور

```
static boolean createPerson(String firstName, String lastname,  
String nationalCode, String date)
```

دستور ساخت فرد است که از کانستراکتور داخل کلاس بهره می برد

```
void show()
```

اطلاعات فرد را نمایش می دهد. داخل آن از تابع کلاس پدر نیز استفاده می شود.

```
static void showPersons()
```

نمایش اطلاعات تمام افراد

کلاس MyDate:

مدیریت زمان در برنامه را به عهده دارد.

```
public static LocalDate getGlobalDate()
```

هدف این تابع یکتا ماندن آبجکت از این کلاس است (طراحی سینگلتون). در ضمن، در آغاز برنامه تاریخ را از سیستم گرفته بعد آن با دستورات ما جلو می‌رود.

```
public static void goDay()  
public static void goMonth()  
public static void goYear()  
public static void goDays(int n)  
public static void goMonths(int n)  
public static void goYears(int n)  
public static void goToDate(LocalDate d)
```

دستورات مربوط به گذر زمان.

کلاس Card:

اطلاعات یک کارت بانکی در آن ذخیره شده‌اند.

```
Card(CurrentAccount currentAccount )
```

کانستراکتور کارت

```
public LocalDate getExpiry()
```

getter برای دریافت تاریخ انقضای کارت

```
boolean extend()
```

تابع تمدید تاریخ انقضا

```
public static boolean transact(long ID,int firstPassword, long  
receiverID, long amount)
```

تابع انتقال کارت به کارت. برنامه انتقال کارت به کارت از اینجا شروع شده درون آن تابع متناظر بانک مرکزی صدا زده می‌شود

```
public static void cardWithdraw(long ID, int firstPassword, int  
amount)
```

تابع برداشت مبلغ از طریق کارت. پیاده سازی مشابه تابع کارت به کارت دارد.

```
boolean withdraw(int amount)
```

تابع مدیریت کننده برداشت پول از کارت. با تابع `cardWithdraw` ارتباطی تنگاتنگ دارد

```
private boolean isActive()
```

تابعی داخلی برای بررسی فعال یا غیرفعال بودن کارت

```
static boolean changeFirstPassword(long CardNumber, int  
oldPassword, int newPassword)
```

تابع تغییر رمز اول

```
public static boolean setSecondPassword(long CardNumber, int  
firstPassword, int secondPassword)
```

تابع قرار دادن رمز دوم

```
public static boolean changeSecondPassword(long CardNumber, int  
firstPassword, int newPassword)  
public boolean changeSecondPassword(int secondPassword)
```

دو تابع با ارتباط تنگاتنگ برای تغییر رمز دوم

```
public String getBalance(int firstPassword)
```

دریافت تراز حساب جاری متناظر با کارت

```
public static boolean cardPrimaryChecks(long ID, int  
firstPassword)
```

تابعی داخلی برای بررسی‌های اولیه شناسه و رمز کارت

کلاس Loan :

توضیحاتی درباره عملکرد وام به صورت کلی:

الف) قسط وام با زمان متغیر است و عوامل جریمه و پرداخت دستی وام بیش از یک قسط در مقدار آن اثر دارند.

ب) رفع سواثر (ممنوعیت عملیات بانکی) منوط به پرداخت بدهی‌های قبلی است. متغیر `loanDebt` برای همین منظور در نظر گرفته شده است.

ج) مدت وام بجز در حالت ورشکستگی ثابت و 4 سال است در نتیجه بدهی شخص به بانک از حدی فراتر نخواهد رفت

د) رفع سواثر هم از طریق پرداخت دستی و هم کسر خودکار از حساب ممکن است.

ه) در لیست پرداخت، پرداخت‌های اتوماتیک نیز ثبت خواهند شد.

و) تاریخ انقضا، آخرین روز اعتبار وام است و از فردای آن وام دیگر بررسی نخواهد شد. اگر شخص مبلغ را پرداخت نکرده باشد وام از بین نخواهد رفت.

برخی از متغیرها که نیاز به توضیح دارند:

```
long amount;  
long paid;  
long loanDebt;  
int interest;  
int penalty;
```

به ترتیب کل مبلغ لازم به پرداخت (با احتساب سود و جریمه)، مقدار پرداخت شده، بدهی (احتمالی و در صورت پرداخت نشدن قسط) از بابت وام، سود وام (به درصد) و جریمه (به درصد) می‌باشند.

```
void show()
```

کارکردش نمایش دادن اطلاعات مربوط به وام است.

```
void check()
```

بررسی و پردازش خودکار وام در زمان‌های پرداخت قسط. کلاس Bank از آن استفاده می‌کند.

```
void manualPay(long amount)
```

پرداخت بخشی از وام به صورت دستی را پردازش می‌کند. ورودی آن پول پرداختی است.

```
void enable(long amount)
```

رفع سوءاثر از وام را انجام می‌دهد.

کلاس InputProcessor :

صرفاً یک تابع برای هر دور اجرای برنامه دارد. رشته‌های ابتدای کد صرفاً از جهت خوانایی کد به آن شکل وجود دارند و هر کدام فقط یک بار استفاده می‌شوند.

اینام BankList :

لیست بانک‌های مجاز به همراه اسم آن‌ها به انگلیسی در این جا آمده است.

یک نمونه تست کیس با حداقل یک دستور از هر رده در زیر آمده که نتایج آن نیز در عکس‌ها موجود است.

Add person Arman LotfaliKhani 1234567891 2001/11/02

Add bank ALI

Add bank MELLI

Add bank SADERAT 4000000000

Add company Myco 1234567891

Add company Myco 0024683256

Set bank income percent MELLI 15

Set bank interest percent SADERAT 7

Show date

Open deposit account SADERAT 20210524_Myco long-term 2000

Open deposit account SADERAT 20210524_Myco long-term 10000

Open current account MELLI 1234567891 100

Change card password (cardnum) (password) 1111

Deposit money MELLI (cardnum) 900

Receive loan MELLI 1234567891 1000

Receive loan SADERAT 1234567891 10000

Withdraw money (cardnum) 1111 999

Transfer money to another account MELLI (accountnum) 1234567891 (receivernum) 1000

Get account balance (cardnum) 1111

Go for 2 months

Open current account MELLI 1234567891 200

Pay off the loan MELLI 1234567891 600

Show all loans

Open current account MELLI 1234567891 200

Close account MELLI 1234567891 (old)

Close account MELLI 1234567891 (new)

Show bank balance MELLI

Show bank balance SADERAT

Show central bank balance

Show all companies

Show all banks

exit

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Q2 [D:\Aarmaan\Sharif\O. O. programming\HW3] - Bank.java

HW3 | src > Bank > showBankBalance

Project | Loan.java | CurrentAccount.java | Company.java | BankList.java | InputProcessor.java | Bank.java | CentralBank.java | Person.java | Card.java | DepositAccount.java

Run: | Unnamed |

"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.3.2\lib\idea_rt.jar=4748:C:\Program Files\JetBrains\In
Add person Arman LotfaliKhan1 1234567891 2001/11/02
Add bank ALI
Add bank MELLI
Add bank SADERAT 4000000000
Add company Myco 1234567891
Add company Myco 8024683296
Set bank income percent MELLI 15
Set bank interest percent SADERAT 7
Show date
Open deposit account SADERAT 20210524,Myco long-term 2000
Open deposit account SADERAT 20210524,Myco long-term 10000
Open current account MELLI 1234567891 100Person was created successfully
Bank name is invalid or it already exists
Bank MELLI was created successfully
Bank SADERAT was created successfully
Company was created successfully
Error: Manager doesn't exist!
Investment profit changed successfully
Short-term deposit account profit changed successfully
2021-05-24
Error: Your proposed amount must not be lower the the minimums for opening deposit accounts
Current account was created successfully: Your account number is:6037691500000013
Card password and cvv2: 7260 619
Deposit account was successfully created. Account number:6037698600000025 Linked current account ID6037691500000013

Current account was created successfully: Your account number is:6037999400000011
Card password and cvv2: 4256 453
Change card password 6037999400000011 4256 1111
Card password has changed to 1111

Run | TODO | Problems | Build | Terminal
All files are up-to-date (moments ago)
Search for anything
8:45 AM 5/24/2021
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Q2 [D:\Aarmaan\Sharif\O. O. programming\HW3] - Bank.java

HW3 | src > Bank > showBankBalance

Project | Loan.java | CurrentAccount.java | Company.java | BankList.java | InputProcessor.java | Bank.java | CentralBank.java | Person.java | Card.java | DepositAccount.java

Run: | Unnamed |

Investment profit changed successfully
Short-term deposit account profit changed successfully
2021-05-24
Error: Your proposed amount must not be lower the the minimums for opening deposit accounts
Current account was created successfully: Your account number is:6037691500000013
Card password and cvv2: 7260 619
Deposit account was successfully created. Account number:6037698600000025 Linked current account ID6037691500000013

Current account was created successfully: Your account number is:6037999400000011
Card password and cvv2: 4256 453
Change card password 6037999400000011 4256 1111
Card password has changed to 1111
Deposit money MELLI 6037999400000011 900
Money was deposited successfully
Receive loan MELLI 1234567891 1000
Receive loan SADERAT 1234567891 10000Your have received your loan successfully

Error:There is already a loan for this ID in the banking system
Withdraw money 6037999400000011 1111 999
Money was successfully withdrawn. Here is the money:999$
Transfer money to another account MELLI 6037999400000011 1234567891 6037691500000013 1000
Transaction was successfully done. Account number: 6037999400000011 amount: 1000 receiver account number: 6037691500000013
Get account balance 6037999400000011 1111
The balance is:1
Bo for 2 months
Open current account MELLI 1234567891 200
Pay off the loan MELLI 1234567891 6002 months have passed
You haven't paid a loan and are thus banned from any action

Loan was paid off successfully. Date: 2021-07-24
|

Run | TODO | Problems | Build | Terminal
All files are up-to-date (3 minutes ago)
Search for anything
8:47 AM 5/24/2021
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Q2 [D:\Aarmaan\Sharif\O. O. programming\HW3] - Bank.java
HW3 src > Bank > showBankBalance
Loan.java x CurrentAccount.java x Company.java x BankList.java x InputProcessor.java x Bank.java x CentralBank.java x Person.java x Card.java x DepositAccount.java x
Run: Unnamed x
Go for 2 months
Open current account MELLI 1234567891 200
Pay off the loan MELLI 1234567891 600 2 months have passed
You haven't paid a loan and are thus banned from any action

Loan was paid off successfully. Date: 2021-07-24
Show all loans
Open current account MELLI 1234567891 200 Bank name:MELLI
Account: 6037999400000011
Total amount which has to be paid: 1200
Paid: 600
Total interest: 20
Date received: 2021-05-24
Expiry: 2025-05-24
Payments:
Payment amount: 600
Date: 2021-07-24

Current account was created successfully: Your account number is:6037999000000023
Card password and cvv2: 6645 365
Close account MELLI 1234567891 6037999400000011
Error: Cannot close an account with a loan
Close account MELLI 1234567891 6037999000000023
Account was closed successfully. Here is the money in the account: 200
Show bank balance MELLI
Show bank balance SABERAT
Show central bank balanceBalance is: 3967499277
Balance is: 5760001840

Balance is: 9727501117
|

Run TODO Problems Build Terminal
All files are up-to-date (4 minutes ago) Event Log 73:1
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Q2 [D:\Aarmaan\Sharif\O. O. programming\HW3] - Bank.java
HW3 src > Bank > showBankBalance
Loan.java x CurrentAccount.java x Company.java x BankList.java x InputProcessor.java x Bank.java x CentralBank.java x Person.java x Card.java x DepositAccount.java x
Run: Unnamed x
Balance is: 9727501117
Show all companies
Show all banks
exit ID: 20210524_Myco
Manager name: Arman Lotfalikhani
Founded on: 2021-05-24
Net worth: 14000
Card IDs:
6037691500000013
Account IDs:
Current account ID:
6037691500000013
Money in the account
4000
Deposit account ID: 6037698600000025
Money in the account
10000
Loan
No Loan
ID:603799
Net balance is:3967499277
Interest rate for short-term accounts:10
Interest rate for long-term accounts:15
Monthly investment interest rate:15
Bankruptcy state: false
Total debt:1
ID:603769
Net balance is:5760001840
Interest rate for short-term accounts:7
Interest rate for long-term accounts:15
Monthly investment interest rate:20
Bankruptcy state: false

Run TODO Problems Build Terminal
All files are up-to-date (4 minutes ago) Event Log 105:1
```

