



تمرین دوم طراحی سیستم بانکی

برنامه سازی پیشرفته

پاییز 1400

استاد: مهندس نرگس بطحائیان
مهلت تحویل: تا ساعت 23:59 یکشنبه 30 آبان 1400
زبان ++c

طراحان تمرین:
حسام کوچکی - محسن حامی - علی زین الدینی



فهرست مطالب

2	بانک؟!.....
3	مقدمه
4	افتتاح حساب
5	آی پی
6	اضافه کردن آی پی
7	تمدید حساب
8	انجام تراکنش
9	واریز
10	انتقال
11	برداشت
12	سود بانکی
13	وام
14	بازپرداخت وام
15	اعلام ورشکستگی
16	مشاهده اطلاعات
16	مشتری
16	بانک
17	سایر نکات قابل توجه

بانک؟!!

پروفسور بعد از اینکه موفق به استفاده از پول های دزدیده شده اش نشد، تصمیم گرفت با اعتراف به کارزشت اش و همکاری با پلیس برای مجازاتش تخفیف بگیرد. وی بعد از دوران کوتاه حبس اش بلاخره آزاد شده و قصد دارد با به کارگیری تجربیاتش یک بانک تاسیس کند. اما از آنجایی که او برنامه نویسی را خوب بلد نیست، از شما خواسته تا در راه اندازی یک سیستم بانکی کمکش کنید!

مقدمه

در یک سیستم بانکی، چون با پول سروکار داریم همه چیز باید خیلی دقیق و منظم باشد. شما در این تمرین یک سیستم بانکی ساده را تولید میکنید که حساب‌های مشتریان را کنترل و ویرایش میکند، در صورت امکان وام میدهد و به مشتری‌ها سود بانکی میپردازد، و یا در صورت بروز فاجعه، اعلام ورشکستگی میکند.

این سیستم مانند خط فرمان عمل میکند و دستورهای ما را پردازش میکند. در ادامه دستورها به اختصار معرفی میشوند.

افتتاح حساب

اولین چیزی که یک بانک نیاز دارد مشتری است، و برای داشتن مشتری باید برای او یک حساب افتتاح کنیم، پس اولین چیزی که باید بانک ما داشته باشد قابلیت ایجاد حساب برای مشتریان جدید است.

پس نیاز به یک کلاس به نام مشتری داریم که اطلاعات مشتریان را در خود ذخیره میکند. این کلاس شامل متغیرهای مختلفی مانند **نام صاحب حساب** یا همان **یوزرنیم** (غیر تکراری)، **شماره کارت** (یک عدد 4 رقمی رندوم غیر تکراری)، **تاریخ افتتاح حساب**، **تاریخ انقضای کارت**، **مبلغ موجودی**، **لیست تراکنش های اخیر** (گردش حساب)، **لیست آی پی های مجاز** و سایر اطلاعات دیگر است.

تاریخ انقضای کارت ها 2 سال پس از تاریخ افتتاح ست میشوند.

واضح است یک لیست (یا وکتور) از کاربران داریم که اطلاعات آنها در آن ذخیره میشود و در صورت نیاز ویرایش میشود.

ساخت و افتتاح حساب برای مشتری ها با این دستور انجام میشود:

```
>>> create username:ip
```

با این دستور یک کاربر جدید با نام username ساخته میشود که دارای آی پی ip است، در مورد آی پی و ساختار آن در ادامه توضیح میدهیم.

توجه کنید که یک یوزر وقتی معتبر است که هم نام آن معتبر باشد، و هم آی پی آن.

یک نام معتبر، نامی است که فقط از حروف کوچک و بزرگ و اعداد انگلیسی تشکیل شده باشد. واضح است که اعداد در ابتدای نام نمیتوانند قرار بگیرند.

در صورتی که نام یا آی پی نامعتبر بودند باید پیغام خطایی نمایش داده شود و چیزی در سیستم ذخیره نشود.

موجودی فرد بعد از افتتاح حساب برابر 0 است.

آی پی

یک آی پی معتبر، عبارتی متشکل از 4 بخش اعداد است که با یک نقطه "." از یکدیگر جدا شده‌اند. اعداد هر بخش می‌توانند از 0 تا 255 متغیر باشند. در زیر دو نمونه آی پی معتبر و غیر معتبر را مشاهده می‌کنید:

```
192.168.15.15 Valid
255.0.142.241 Valid
123.280.23.245 Not Valid
44.280.23 Not Valid
```

تضمین می‌شود آی پی شامل اعداد منفی نمی‌باشد.

تعداد نقطه‌های آی پی می‌تواند متغیر باشد، ولی حتماً یک نقطه در آی پی وجود دارد. واضح است آی پی معتبر دقیقاً 3 نقطه و 4 بخش اعداد معتبر دارد.

در صورتی که نام یا آی پی نامعتبر بودند باید پیغام خطایی نمایش داده شود و چیزی در سیستم ذخیره نشود.

اضافه کردن آی پی

گفتیم که امنیت خیلی در بانک مهم است، برای همین یک کاربر برای انجام تراکنش فقط با آن آی پی میتواند تراکنش انجام دهد که در ابتدای افتتاح حساب با آن وصل بوده یا بعداً آن را به صورت دستی به سیستم اضافه کرده است. برای اضافه کردن آی پی برای یک حساب از دستور زیر استفاده میکنیم:

```
>>> add_ip username:new_ip
```

با این دستور یک آی پی با آدرس new_ip به لیست سفید آی پی های کاربر username اضافه میشود، البته در صورتی که معتبر باشد.

امتیازی: بشود در یک دستور، بیش از چند آی پی اضافه کرد (تعداد نامشخصی از آی پی، پیاده سازی توسط پردازش ها)، مانند:

```
>>> add_ip username:new_ip:another:one_more
```

تمدید حساب

در صورتی که تاریخ انقضای کارتی فرا رسیده بود، نباید اجازه انجام هیچ تراکنشی داده شود.

کاربر میتواند با تایید خودش، مبلغی را پرداخت نموده و حساب خودش را 2 سال دیگر تمدید کند. (مبلغ با توجه به اینکه برنامه را ریالی نوشته اید یا دلاری یا ... میتواند متغیر باشد، بر عهده خودتان)

```
>>> renewal username:ip
```

در صورت تایید برای تمدید، مبلغ را از حساب او کم کنید.

امتیازی: کاربر در دو حالت بتواند حساب خودش را تمدید کند، یک حالت با استفاده از روش بالا، و یک حالت دیگر هم بعد از اینکه تمایل داشت تراکنشی انجام دهد و سیستم به او خطا داد؛ در این حالت باید بعد از نمایش دادن خطا، از او سوال شود که آیا مایل به تمدید حساب خودش در ازای مبلغ مشخص هست یا خیر؟

امتیازی: در صورت تایید تمدید، اما نبود موجودی کافی، موجودی را 0 کنید و باقی مانده را به عنوان وام در نظر بگیرید، در ادامه با وام آشنا میشوید.

انجام تراکنش

انجام تراکنش‌هایی که در ادامه گفته میشود باید در صورتی انجام بگیرد که تاریخ انقضای حساب هنوز فرا نرسیده باشد. در صورتی که حساب فرد منقضی شده باشد باید پیغام خطا چاپ شود.

در سیستم بانکی ما نیاز به انجام تراکنش داریم. تراکنش‌ها به دو صورت انجام میشوند، واریز و یا انتقال.

واریز

واریز حالتی است که کاربر پول نقد به بانک میدهد و بانک موجودی حساب او را افزایش میدهد. واریز برای یک کاربر با دستور زیر انجام میشود:

```
>>> deposit username:ip:money
```

برای مثال، دستور زیر یک مثال معتبر برای انجام تراکنش واریز است:

```
>>> deposit professor:144.123.63.48:25000
```

البته دستور بالا در صورتی معتبر است که قبلاً پروفسور یک افتتاح حساب موفق در بانک داشته باشد، آی پی او در لیست سفید آی پی های اکانت خودش موجود باشد و عددی صحیح و مثبت را به عنوان رقم وارد کرده باشد. در غیر این صورت باید با خطا مواجه شود.

انتقال

مانند همه بانک‌ها، باید بتوانیم انتقال از حسابی به حساب دیگر هم انجام بدهیم، برای انجام این کار از دستور زیر استفاده میکنیم:

```
>>> transfer sender_username:sender_ip:receiver_username:money
```

در واقع اینجا از حساب فرستنده کم میکنیم و به حساب گیرنده اضافه میکنیم. نحوه اعتبارسنجی این تراکنش به این صورت است که باید مقدار پول در حساب فرد فرستنده موجود باشد، آی‌پی او درست باشد و فرد گیرنده قبلاً در بانک حسابی افتتاح کرده باشد.

امتیازی: با دستور زیر هم بتوان انتقال انجام داد:

```
>>> transfer sender_username:sender_ip:receiver_ip:money
```

برداشت

مسئله بعد از واریز باید بتوان برداشت هم کرد، برداشت با دستور زیر انجام میشود:

```
>>> withdraw professor:144.123.63.48:25000
```

اعتبارسنجی این تراکنش هم باید به درستی انجام شود.

موجودی فرد نمیتواند منفی شود، یعنی فرد نمیتواند بیشتر از موجودی خودش برداشت کند.

سود بانکی

یکی از بخش‌های جذاب برای سرمایه‌داران، سودی است که بانک به آن‌ها می‌پردازد. البته در دنیای واقعی، میزان سودی که در یک سال از بانک‌ها می‌گیریم از میزان تورم در آن سال کمتر است و این یعنی ما با نگه داشتن پول مان در بانک ضرر کرده‌ایم، ولی در هر حال برای کسانی که آمادگی یا سواد کافی برای سرمایه‌گذاری در بازار بورس و یا سایر بازارهای مالی را ندارند تا از تورم جلو بزنند و ارزش پول خود را حفظ کنند سود بانکی گزینه وسوسه‌کننده‌ای است.

سود بانکی در بانک ما به صورت درصدی از کل مبلغ سپرده مشتری تعلق می‌گیرد. سود بانکی به صورت دستی و توسط شما برای مشتریان لحاظ می‌شود، پس یک دستور بنویسید که با گرفتن نام مشتری، برای او سود بانکی‌اش را به حسابش اضافه کند.

سود بانکی به صورت درصدی (مثلاً 10 درصد) از میانگین موجودی حساب مشتری در 7 روز گذشته محاسبه می‌شود.

سود بانکی در صورتی برای مشتری اضافه می‌شود که او در 30 روز گذشته سودی دریافت نکرده باشد.

اگر مشتری در حال حاضر وامی گرفته باشد و هنوز آن را تسویه نکرده باشد سود بانکی به او تعلق نمی‌گیرد. 30 روز بعد از تسویه کامل آخرین وام، سود می‌تواند برای او لحاظ شود.

```
>>> add_profits username
```

امتیازی: چند درصد سود بیشتر برای کسانی که در یک هفته گذشته، از حساب خود برداشت نکرده باشند (واریز ایرادی ندارد).

امتیازی: سود بیشتر برای کسانی که پول بیشتری در بانک دارند. میزان آن برعهده شما.

هرکسی در زندگی ممکن است نیاز به وام داشته باشد، پس بانک ما هم جهت مشتری مداری قصد دارد وام کم بهره به مشتری ها بدهد. مشتری میتواند از بانک درخواست وام کند و اگر بانک قادر بود به او وام دهد. یک مشتری با دستور زیر از بانک طلب وام می کند:

```
>>> get_loan username:ip:money
```

بانک در صورتی به مشتری وام می دهد که مشتری حداقل $\frac{3}{4}$ مبلغ وام را در حسابش داشته باشد. توجه کنید که بانک ها، از خودشان پولی ندارند، و از پول هایی که دیگران در بانک گذاشته و نگهداری می کنند به افراد دیگر وام می دهند (پس به همین دلیل است در بانک های واقعی، بانک ها برای سپرده های بلند مدت سود بیشتری در نظر می گیرند، چون بانک میخواهد شما را ترغیب کند که سپرده خود را از بانک بیرون نبرید تا بتواند به افراد بیشتری وام بدهد، و البته بهره های وام های بیشتری دریافت کند).

پس قبل از وام دادن به کسی باید دو موضوع را چک کنید، یک اینکه آیا او در حسابش به اندازه $\frac{3}{4}$ مبلغ وام، پول دارد؟ و دوم هم اینکه آیا خود بانک میتواند مبلغ وام را به مشتری پرداخت کند؟

اگر شرایط مهیا نبود، به مشتری پیغامی مبنی بر رد درخواست وی برای وام نمایش داده شود.

مشتری تا زمانی که تمام وامش را به بانک پس نداده باشد، نمیتواند از حسابش برداشت کند، در نتیجه باید مشخص کنید که مشتری در حال حاضر وامی گرفته یا خیر. (راهنمایی: بهتر است یک متغیر در کلاس مشتری تعریف کنید و در آن میزان بدهی مشتری را نگهداری کنید، اگر مشتری 10 دلار وام گرفت و بهره وام 10 درصد است، پس در آن متغیر 11 دلار ذخیره کنید و تا زمانی که این متغیر صفر نشده باشد، اجازه برداشت ندهید)

طبیعتاً میزان بهره وامی که تعیین میکنید تا از مشتری بگیرید، باید بیشتر از میزان سودی باشد که به مشتری ها پرداخت میکنید.

امتیازی: چاپ پیغام دقیق، که مشتری متوجه شود به کدام یک از دو دلیل بالا نمیتواند وام بگیرد.

بازپرداخت وام

اگر مشتری وام فعالی داشت و دستور زیر را وارد کرد، از بدهی اش کم کنید، توجه کنید مشتری میتواند در چند مرحله اقدام به بازپرداخت وام کند. اگر مشتری بیشتر از مبلغ وام واریز کرد، بدهی را تسویه، و باقی مانده را به موجودی اش اضافه کنید.

```
>>> pay_loan username:money
```

اعلام ورشکستگی

بانک‌ها اگر بی‌رویه وام‌های بدون پشتوانه بدهند، خود را در خطر ورشکستگی قرار می‌دهند.

ورشکستگی یک بانک زمانی اتفاق می‌افتد که بانک، اکثر پول‌های مشتریان خود را به دیگران وام داده، و ناگهان، یک یا چند مشتری قصد برداشتن سپرده‌های خودشان را میکنند. در این هنگام چون بانک در حال حاضر پول آن‌ها را نزد خود ندارد (چون آن‌ها را وام داده) نمیتواند به آن‌ها پولی بدهد و اعلام ورشکستگی میکند. در برنامه باید اگر زمانی مشتری ای درخواست برداشت کرد و شما پولی در بانک نداشته بودید و آن‌ها را وام داده بودید، باید پیغامی مبنی بر اعلام ورشکستگی چاپ کنید.

امتیازی: کسی بعد از اعلام ورشکستگی بانک، نتواند حساب جدید باز کند، یا مشتریان قبلی نتوانند پول جدید واریز یا منتقل کنند، و موارد مشابه.

مشاهده اطلاعات

مشتری

یک مشتری باید بتواند اطلاعات حساب خودش را مشاهده کند، برای این کار میتواند از دستورهای زیر استفاده کند:

```
>>> ip
Or
>>> card_number
```

یعنی شخص با وارد کردن آی پی خودش یا نام کاربری اش یا شماره کارتش باید بتواند اطلاعات حساب خودش را مشاهده کند.

امتیازی: کاربر بتواند با نوشتن نام کاربری هم اطلاعاتش را مشاهده کند:

```
>>> username
```

بانک

با دستور زیر، مشخصات بانک نظیر تعداد مشتری ها، مبلغ کل پول موجود در بانک، مبلغ کل وام های اعطاء شده، و سایر اطلاعات مفید را چاپ کنید:

```
>>> bank
```

سایرنکات قابل توجه

- یک آی پی نمیتواند توسط دو نفر استفاده شود.
- تمامی تراکنش ها باید ثبت و ذخیره شوند، یک بار در یک لیست (یا وکتور) کلی مربوط به بانک که تمامی تراکنش هارا به ترتیب اجرا در خودش ذخیره میکند، و یک بار هم در مشخصات حساب خود شخص مورد نظر، به صورت اشاره گری به تراکنش مربوطه در لیست کلی بانک. به عبارت دیگر اگر 2 تراکنش در لیست کلی بانک است، باید برای کاربران، یک اشاره گر به تراکنشی در لیست اصلی که مربوط به آنها است ذخیره شود.
- برنامه باید تا زمانی که کاربر دستور خروج نداده باشد ادامه پیدا کند و منتظر دستورات جدید باشد.
- برنامه باید مازولار نوشته شود، یعنی از توابع استفاده کنید.
- استفاده از گیت و cmake اجباری است.
- تحویل به صورت لینک در گیت هاب انجام میگیرد.
- کامنت و گزارش اجباری است.
- راه حل ها و یا موارد جدید ابداعی و خلاقانه که در متن تمرین ذکر نشده اند ولی جالب باشند نمره مثبت خواهند داشت.
- همه بخش های مورد نیاز به صراحت در متن تمرین ذکر نشده اند، پیاده سازی همه بخش هایی که در تمرین گفته شده است ضروری است ولی ممکن است کافی نباشد، در صورت نیاز بخش های مورد نیاز را اضافه کنید.



موفق باشید..!