



دانشگاه بو علی سینا

نام و نام خانوادگی دانشجو: محمدامین احمدی رشته: کامپیوتر شماره دانشجویی: 9912358001

نام استاد: خانم بطحائیان

موضوع: تمرین دوم

سورة الفاتحة

فهرست مطالب

4	شرح تمرین
5	مقدمه
6	class Account
8	Main

شرح تمرین

در pdf مربوط به تمرین دوم به وضوح و جزئیات بالا امور مربوط به بانک را شرح داده و من در اینجا به برخی از آن ها اشاره میکنم:

- افتتاح حساب
- اضافه کردن آی پی
- تمدید حساب
- انجام تراکنش
- واریز
- انتقال
- برداشت
- سود بانکی .
- وام
- بازپرداخت وام
- اعلام ورشکستگی
- مشاهده اطلاعات

و ...

مقدمه

من در این تمرین سعی کردم از توان خودم در راه پیاده سازی کامل کد این تمرین قدم بردارم و بتوانم که خواسته های تمرین را برآورده سازم.

با توجه به سنگینی دروس ترم جاری من توانستم که به اندازه ی کافی برای این تمرین وقت گذاشته و برخی از موارد خواسته شده را پیاده سازی کنم.

موارد پیاده سازی شده عبارتند از:

- افتتاح حساب
- اضافه کردن آی پی
- انجام تراکنش
- واریز
- انتقال
- برداشت
- مشاهده اطلاعات

همچنین نکات ذکر شده در تمرین نیز رعایت شده:

1. یک آی پی توسط دو نفر نمیتواند استفاده شود.
2. تراکنش ها در وکتور ذخیره میشوند.
3. برنامه تا زمانی که کاربر دستور خروج بدهد ادامه پیدا میکند و منتظر دستورات جدید است.
4. برنامه ماژولار است.
5. از گیت و cmake هم استفاده شده.
6. ریپازیتوری در گیت هاب قرار داده شده.
7. کامنت گذاری به وفور صورت گرفته.

class Account

کلاس اکانت شامل توابعیست که هریک از این توابع به شکل خاصی به بهبود برنامه کمک میکنند که در ادامه به آنها میپردازیم:

اما قبل از آن به سراغ متغیر های پرایوت این کلاس میرویم یعنی:

(1) یک متغیر از نوع استرینگ جهت ذخیره ی نام مشتری در آن :

```
std::string name;
```

(2) یک متغیر اینتی برای ذخیره ی آی دی مشتری:

```
int id;
```

(3) یک متغیر از نوع دابل برای ذخیره ی موجودی مشتری:

```
double balance;
```

یک لیست برای ذخیره:

```
static std::vector<Account> accountDatabase;
```

حال میرویم به سراغ توابع این کلاس:

یک کانستراکتور که به جای متغیر ای عددی صفر و به جای متغیر استرینگ رسته ی خالی میگذارد:

```
Account();
```

(2) چند تابع برای برگشت دادن اطلاعات به ترتیب نام مشتری، آی دی مشتری و موجودی مشتری:

```
std::string getName() const;
```

```
int getId() const;
```

```
double getBalance() const;
```

(3) تابعی برای گرفتن تعیین نام مشتری که عملا ولیدیشن خاصی در آن صورت نمیگیرد چون بالاخره اسم های عجیب و غریب زیادی را میتوان در سراسر جهان مشاهده کرد(به خصوص در آفریقا) :

```
void setName(std::string);
```

4) برای تعیین و تنظیم آی دی مشتری تابع عضوی وجود دارد که چهارچوبی برای آن در نظر گرفته شده. در این تابع آی دی وارد شده بررسی میشود که اول صفر یا منفی نباشد و دوم اینکه لیست آی پی های وارد شده را چک میکند که آی پی جدید مشابه آن ها نباشد. در صورت رعایت شدن موارد ذکر شده آی پی برای مشتری ثبت میشود و در غیر این صورت اکسپشن خاصی در نظر گرفته شده که اخطار مناسب را نمایش می دهد:

```
void setID(int);
```

5) در این تابع برنامه به تنظیم بالانس مشتری میپردازد و در نظر دارد که هرگز یک موجودی نمیتواند منفی باشد و در این باره اکسپشن خاصی طراحی شده است:

```
void setBalance(double);
```

6) در تابع addAccount ما با استفاده از دستور `accountDatabase.push_back(Account);` حساب کاربری جدید را در وکتور مطرح شده اضافه و ذخیره میکنیم:

```
std::vector<Account> Account::accountDatabase;
```

```
void addAccount (Account);
```

7) تابعی برای برداشت از حساب که چک میکند میزان برداشت منفی نباشد که اگر باشد اخطار می دهد:

```
void withdraw(double);
```

9) تابعی برای سپرده گذاری که چک میکند میزان سپرده منفی نباشد که اگر باشد اخطار می دهد:

```
void deposit(double);
```

هدر:

```
#include <iostream>
```

```
#include "Account.h"
```

```
#include <string>
```

تابع main :

در این تابع ما `MenuSelection();` را فرا خوانی میکنیم که از طریق آن توابع مناسب فراخوانی میشوند ولی قبل از آن باید در بانک حساب باز کنید .

بعد از اینکه حساب شما در سیستم با موفقیت ثبت شد شما میتوانید از امکانات بانک استفاده کنید.

به شما یک منو نمایش داده میشود که با استفاده از آن میتوانید عملیات برداشت، سپرده گذاری، انتقال را انجام دهید و همچنین از مشخصات حساب خود و میزان موجودی با خبر شوید.

در پناه خدا

شاد و پیروز باشید