

به نام خدا



درس برنامه نویسی پیشرفته

تمرین سوم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۲ - ۱۴۰۱

مهلت ارسال :

۱۴۰۲ / ۰۱ / ۲۵

مباحث :

رشته ها - فایل - نوع شمارشی - مدیریت خطا

مسئول تمارین :

آریا شهسوار

فهرست

۳	<input type="checkbox"/> آداب نامه تمرینات
۴	<input type="checkbox"/> نکات تمرین سری سوم
۵	<input type="checkbox"/> تمرین ۱ . کار با فایل مقدماتی
۷	<input type="checkbox"/> تمرین ۲ . مدیریت سینما
۱۰	<input type="checkbox"/> تمرین ۳ . مدیریت کتابخانه
۱۶	<input type="checkbox"/> تمرین ۴ . خبرسان
۲۲	<input type="checkbox"/> تمرین ۵ . مدیریت رستوران

آداب نامه تمرینات

- پاسخ تمامی سوالات تنها به زبان های C# قابل قبول می باشد .
- علیرغم اعتماد کامل تیم تی ای به شما دانشجویان عزیز، تمامی کد های شما با سایر دانشجویان بصورت خودکار و توسط برنامه مقایسه خواهند شد . همچنین در طول ترم ، از تمامی پاسخ های شما ارائه گرفته خواهد شد و نحوه کار تمامی بخش های هر سوال از شما پرسیده خواهد شد، لذا از کپی نمودن کد دوستانتان خودداری کنید و تمامی پاسخ ها، کد خودتان باشد . همچنین از آنجایی که مشورت و هم فکری با سایر دوستان بسیار کار پسندیده و مفیدی است - برخلاف کپی کردن کد (: - در صورت هم فکری با دانشجوی دانشجویان ، نام وی را بصورت کامنت شده در ابتدای کد خود بنویسید.
- برای ارسال تمارین در طول ترم، در مجموع ۷ روز می توانید تاخیر داشته باشید و در صورتی که جمع تاخیر دانشجویی بیشتر از ۷ روز شود، تمرین وی قابل قبول نخواهد بود لذا تلاش کنید تمرینات را در زمان مقرر در سامانه آپلود کنید.
- در تمامی تمرینات سعی شده است که سوالات ساده تر در ابتدا و سوالات دشوار تر در انتهای فایل قرار گیرند (از ساده به دشوار مرتب شده اند) .
- در صورت وجود هرگونه سوال در مورد تمرینات ، سعی کنید تا جایی که امکان دارد سوال خود را در گروه پرسید چرا که شاید سوال شما، سوال دوستان نیز باشد و دوستانتان نیز بتوانند از پاسخ سوال شما بهره ببرند.

نکات تمرین سری سوم

- سوالات را در سامانه کوئرا و در قسمت تمرین سری سوم آپلود نمایید .
- در تمامی سوالات باید مدیریت خطا بطور کامل و دقیق صورت گیرد و بخشی از نمره هر سوال به Exception Handling درست و مناسب تعلق میگیرد .
- توجه داشته باشید که برخی موارد خواسته شده در سوالات من جمله مدیریت خطا در جلسات بعد از تعطیلات تدریس میشوند .
- از آنجایی که هر سوال توسط یک تی ای طرح شده است ، تنها تی ای طراح آن سوال می تواند شما را بصورت دقیق راهنمایی کند به همین منظور طراح هر سوال در زیر نوشته شده است تا در صورت ابهام و پرسش در مورد هر سوال، در صورتی که نیاز به پرسش سوال بصورت انفرادی در پیوی هست ، به تی ای مربوطه مراجعه بفرمایید.
- سوال ۱ . آقای زارعیان ، آقای شهسوار
- سوال ۲ . خانم کاوسی نژاد
- سوال ۳ . خانم عربعلی
- سوال ۴ . آقای ارشیا ، خانم عربعلی ، آقای شهسوار
- سوال ۵ . آقای ارشیا ، خانم عربعلی ، آقای شهسوار

تمرین ۱. کار با فایل مقدماتی

در این تمرین قصد تمرین اطلاعات یک فایل متنی را با کمی تغییرات به یک فایل جدید منتقل کنیم .

به این صورت که در فایل جدید تمامی فضاهاى خالی بین کلمات تبدیل به * شوند .

همچنین باید در خروجی کنسول :

۱. تعداد خطوط

۲. تعداد اعداد


۳. تعداد ستاره ها

۴. تعداد حروف صدا دار فایل

۵. تعداد کلماتی که با 'a' شروع و به 'e' ختم میشوند

۶. تعداد کلمه "student" در متن (بزرگی و کوچکی حروف را در نظر نگیرید) به این معنا که اگر در متن هم کلمه "student" هم "Student" و هم "sTuDeNt" داشته باشیم خروجی باید عدد ۳ باشد .

مثال :

 a1 - Notepad

File Edit Format View Help


hi my name is john. how are you ?

I am student. I see you at 13 : 00.

good bye !

405 error|

فایل خروجی :

 a2 - Notepad

```
File Edit Format View Help
hi*my*name*is*john.*how*are*you*?
I*am*student.*I*see*you*at*13*:*00.
good*bye*!
405*error|
```

خروجی کنسول :

Number of lines : 4

Number of stars : 20

Number of digits : 7

Number of vowel sounds : 25

Number of words start with 'a' and end with 'e' : 1

Number of "student" : 1

تمرین ۲. مدیریت سینما

مدیر یک سینما می خواهد فیلم های موجود در سینما را مدیریت کند و به کمک شما نیاز دارد. در ادامه کلاس و عملیات های مورد نیاز را به طور کامل توضیح خواهیم داد .

کلاس موجود در برنامه :

کلاس سینما : این کلاس را برای مدل کردن اطلاعات فیلم ها بنویسید. هر فیلم یک شی از این کلاس است.

اطلاعات هر فیلم :

۱. کد فیلم (ID)
۲. نام فیلم (Name)
۳. نام کارگردان (Director)
۴. نام نویسنده (Writer)
۵. ژانر (Genre)
۶. قیمت بلیط (Price)

توجه ۱: ویژگی های کد فیلم، نام نویسنده و قیمت بلیط نباید در خارج از کلاس قابل دسترسی باشند .

توجه ۲: ژانر هر فیلم باید مقداری از یک نوع داده از Enum باشد که خودتان تعریف کرده اید و آن را به صورت رشته ذخیره نکنید . ژانر ها به هشت نوع زیر تقسیم بندی میشوند :

۱. Horror (۱۱)
۲. Action (۱۲)
۳. Drama (۱۳)
۴. ScienceFiction (۱۴)
۵. Comedy (۱۵)
۶. Documentary (۱۶)
۷. History (۱۷)
۸. Fiction (۱۸)

نکته ۱: کد هر فیلم یک عدد سه رقمی است که ارقام صدگان و دهگان آن برابر کد ژانر و رقم یکان آن برابر تعداد فیلمهای آن ژانر با اضافه کردن این فیلم است. به عنوان مثال ما قبل از اضافه کردن فیلم x که دارای ژانر History است، ۳ فیلم از این ژانر داشتیم که کدهای آن ها به ترتیب ۱۷۱، ۱۷۲ و ۱۷۳ بود. کد فیلم x برابر با ۱۷۴ خواهد شد.

نکته ۲: برای هر ژانر حداکثر ۹ فیلم خواهیم داشت.

منوی برنامه :

- افزودن فیلم (Add Film)
- تعداد فیلم هر ژانر (Number of Films)
- ژانر کارگردان (Genre of Director)
- تغییر یکی از ویژگی های فیلم (Change Film Property)
- نمایش اطلاعات (Show Info)
- خروج (Exit)

توضیح روند برنامه :

افزودن فیلم : با اجرای این دستور با گرفتن **اطلاعات فیلم**، فیلم به لیست فیلم های موجود در سینما افزوده میشود .

توجه ۳: در صورتی که فیلم وارد شده در لیست فیلم های سینما **وجود داشت** پیام خطای مناسبی نمایش داده شود .

تعداد فیلم هر ژانر : این متد به عنوان ورودی یک ژانر را دریافت می کند و خروجی آن تعداد فیلم های آن ژانر است که در سینما موجودند .

ژانر کارگردان : این متد نام یک کارگردان را به عنوان ورودی دریافت می کند و بررسی می کند که این کارگردان بیشتر در کدام ژانر فیلم کارگردانی کرده است . در صورت برابر بودن تعداد ژانرها (مثلا کارگردان مورد نظر ۴ فیلم دارد که ۲ فیلم از ژانر Horror و ۲ فیلم دیگر از ژانر Action هستند) یکی از آن ها را به صورت دلخواه بر میگرداند.

تغییر یکی از ویژگی های فیلم (Change) : سه متد با نام Change اما با ورودی های مختلف وجود خواهند داشت. ورودی اول در هر سه متد، نام فیلم است .

- متد تغییر نویسنده فیلم
- متد تغییر قیمت بلیت فیلم
- متد تغییر ژانر فیلم

نمایش اطلاعات (ShowInfo) : دو متد با نام ShowInfo با ورودی های مختلف وجود خواهند داشت. در یکی از متد ها ورودی فیلم است و فقط اطلاعات فیلم مورد نظر چاپ می شود. متد دیگر ورودی ندارد و خروجی نمایش اطلاعات تمامی فیلم های موجود در سینما است .

خروج (Exit) : در انتها با وارد کردن عبارت Exit اجرای برنامه پایان میابد .

در قسمت Main این سوال :

توجه کنید که باید پیام مناسب برای راهنمایی کاربر در هر مرحله را چاپ کنید. پیشنهاد میشود که یک منو مانند زیر برای برنامه در نظر بگیرید که بر اساس عنوان یا شماره دستور، دستور مورد نظر اجرا شود .

نکته ۳ : دقت کنید که تا زمانی که دستور خروج وارد نشده، برنامه باید ادامه پیدا کند و بعد از هر دستور بخش راهنمایی نمایش داده شود .

نکته ۴ : یک فایل به نام Cinema.txt داریم که بعد از اضافه کردن هر فیلم، اطلاعات آن به انتهای فایل اضافه میشوند .

تمرین ۳. مدیریت کتابخانه

در این سوال قصد داریم تا سیستم یک کتابخانه را پیاده سازی کنیم. اجزای این سیستم به شرح زیر است :

کلاس های موجود در برنامه :

کلاس کتابخانه : هر کتابخانه یک نام مخصوص به خود دارد و خدمات عضویت و خرید و امانت گرفتن کتاب را ارائه میکند. همچنین شامل لیستی از کتاب های کتابخانه است .

کلاس کاربر : هر عضو یک نام کاربری مخصوص به خود دارد. با توجه به فعالیت کاربر نوع حساب کاربری او متفاوت میشود . هر کاربر دارای کیف پول که میتواند آن را جهت خرید و عضویت در کتابخانه شارژ کند .

توجه ۱ : در صورت تکراری بودن نام کاربری وارد شده در سیستم کتابخانه پیام خطای مناسب چاپ شده و از کاربر تا زمان وارد کردن نام کاربری غیر تکراری ورودی بگیرد .

توجه ۲ : انواع کاربر رو به صورت Enum تعریف کنید . به صورت پیش فرض کاربرها از نوع فعال میباشند اما در صورت عدم تحویل کتاب امانت گرفته شده در موعد مقرر به کاربر بد حساب تبدیل میشوند در صورت عدم فعالیت (عدم خرید یا به امانت گرفتن کتاب) بعد از ۳۰ روز عضویت او به اتمام میرسد و حساب کاربری او از بین میرود . (هر ۱ ثانیه را ۱ روز در نظر بگیرید) .

توجه ۳ : هر کاربر میتواند نهایتا ۳ کتاب را به صورت همزمان و به مدت ۱۰ روز به امانت بگیرد .

کلاس کتاب : هر کتاب نام مخصوص به خود را دارد و همچنین یک ژانر مخصوص و ردیف محل قرارگیری کتاب در کتابخانه . امکان موجود بودن چند نمونه از یک کتاب وجود دارد .

توجه ۴ : شما باید اطلاعات مربوط به کتاب ها را که شامل نام , نام نویسنده , قیمت , ژانر و ردیف محل نگهداری در کتابخانه است را در زمان اضافه کردن هر کتاب در فایل Books.txt اضافه کنید .

توجه ۵ : شما باید انواع ژانر کتاب را به صورت Enum تعریف کنید . انواع ژانرها به شرح زیر است :

۱. Romance
۲. Classic
۳. Horror
۴. Science
۵. Poetry
۶. Novel
۷. History

توجه ۶: در طول برنامه هر جایی نیاز به استفاده از ژانر کتاب ها داشتید از رشته استفاده کنید .

منوی اصلی برنامه :

- دسترسی به عنوان ادمین (Admin)
- دسترسی به عنوان کاربر (User)
- خروج (Exit)

توضیح روند برنامه :

این برنامه دو مدل دسترسی خواهد داشت شامل :

۱. دسترسی به صورت Admin
۲. دسترسی به صورت User

- **دسترسی Admin:** ادمین برای دسترسی به این قسمت کافایت عبارت Admin را وارد کرده و سپس عبارت "DM!N@" را به عنوان پسورد خود وارد کند. پس از وارد شدن به عنوان ادمین منوی زیر نمایش داده میشود :

منوی ادمین :

- افزودن کتابخانه (Add Library)

- افزودن کتاب (Add New Book)
- افزودن عضو (Add Member)
- جستجوی کتاب (Search Book)
- نمایش کتابها (Display Books)
- تغییر اطلاعات کتاب (Change Book Info)
- بازگشت به منوی اصلی (Return)

- **دسترسی User:** هر کاربر برای دسترسی به این قسمت کفایت عبارت User را وارد کند سپس نام کاربری ثبت شده درسیستم کتابخانه را وارد کند. پس از وارد شدن به عنوان یوزر منوی زیر نمایش داده میشود :

منوی کاربر:

- جستجوی کتاب (Search Book)
- خرید کتاب (Buy Book)
- امانت گرفتن کتاب (Borrow Book)
- شارژ کردن کیف پول (Charge Account)
- بازپسگیری کتاب (Return Book)
- نمایش کتابها (Display Books)
- شارژ کردن کیف پول (Charge Account)
- بازپسگیری کتاب (Return Book)
- بازگشت (Return)

- نکته ۱:** دسترسی User/Admin صرفاً به عنوان ورودی Main شما برای مشخص کردن نوع آن میباشد .
- نکته ۲:** در صورتی که ادمین یا کاربر کلمه Exit را وارد کنند برنامه به پایان میرسد .
- نکته ۳:** در صورت وارد کردن عبارتی به غیر از ۳ مورد ذکر شده دوباره از کاربر ورودی بگیرد .

توضیح روند برنامه :

افزودن کتابخانه: با گرفتن نام کتابخانه آن را به سیستم اضافه میکند .

افزودن عضو: با اجرای این دستور و گرفتن نام کاربر و مبلغ اولیه کیف پول و همچنین نام کاربری مدنظر او در سیستم عضو میشود . همچنین زمان عضویت کاربر نیز با توجه به زمان سیستم ذخیره گردد. در هنگام عضویت مبلغ اولیه عضویت از کیف پول او کسر میشود . (عدد ثابتی برای مبلغ اولیه عضویت در نظر بگیرید).

توجه ۷: در صورتی که نام کاربری مد نظر شخص در سیستم توسط شخص دیگری ثبت شده باشد پیام خطای مناسبی نمایش داده شود و تا زمانی که نام کاربری تکراری نباشد از شخص ورودی گرفته شود.

توجه ۸: اگر موجودی وارد شده برای کیف پول او جهت عضویت کافی نبود پیام خطای مناسبی نمایش داده شود و مبلغ دیگری برای کیف پول از کاربر گرفته شود و در صورتی که کاربر کلمه QUIT را وارد کرد فرایند عضویت کاربر به پایان میرسد و در صورت کافی بودن موجودی پیام مناسبی جهت موفقیت آمیز بودن ثبت نام چاپ شود .

افزودن کتاب جدید: با وارد کردن این دستور توسط ادمین ، اطلاعات ذکر شده برای کتاب از ادمین گرفته میشود و کتاب در کتابخانه ثبت میگردد .

جستجوی کتاب: کاربر میتواند بر اساس ژانر ، اسم و نویسنده کتاب ها را جستجو کند .

- **جستجو بر اساس ژانر :** تمامی کتاب های موجود در آن ژانر همراه با ردیف قرارگیری در کتابخانه نمایش داده شود .
- **جستجو بر اساس نام کتاب :** موجود بودن یا عدم موجودی آن کتاب نمایش داده شود و در صورت موجود بودن محل قرارگیری هم نمایش داده شود .
- **جستجو بر اساس نام نویسنده :** نام تمامی کتاب های موجود آن نویسنده همراه با ردیف قرارگیری در کتابخانه نمایش داده شود .

خرید کتاب : کاربر برای خرید کتاب ابتدا نام کاربری سپس نام کتاب مد نظر را وارد میکند .

توجه ۹ : در صورت عدم موجودی کافی کیف پول یا موجود نبودن کتاب و یا موجود نبودن نام کاربری پیام مناسب چاپ شود و در صورت کافی بودن موجودی قیمت کتاب از کیف پول کسر میگردد و در صورت ناکافی بودن کاربر میتواند در صورت تمایل حساب خود را شارژ کند .

امانت گرفتن کتاب : کاربر برای امانت گرفتن کتاب ابتدا نام کاربری سپس نام کتاب مد نظر را وارد میکند . همچنین زمان امانت گرفتن کتاب نیز با توجه به زمان سیستم ذخیره گردد.

توجه ۱۰ : در صورت موجود نبودن کتاب و یا موجود نبودن نام کاربری پیام مناسب چاپ شود .

توجه ۱۱ : در صورتی که ۳ کتاب را همزمان به اختیار گرفته باشد نمیتواند کتاب دیگری به امانت بگیرد .

شارژ کیف پول : کاربر برای شارژ کیف پول ابتدا نام کاربری خود سپس مبلغ مد نظر را وارد میکند .

توجه ۱۲ : در صورت موجود نبودن نام کاربری پیام مناسب چاپ شود .

بازپسگیری کتاب : با اجرای این دستور کاربر نام کاربری خود و سپس نام کتاب را مورد نظر را وارد میکند .
بعد از اجرای این دستور کتاب های اختیار گرفته شده توسط کاربر و موجودی کتاب های کتاب خانه و نوع کاربر با توجه به زمان سیستم بروز رسانی میشود .

توجه ۱۳ : در صورت موجود نبودن نام کاربری پیام خطای مناسب چاپ شود . در صورتی که نام کتاب وارد شده در لیست کتاب های اختیار گرفته شده توسط کاربر نبود پیام مناسب چاپ شده و دوباره از کاربر نام کتاب گرفته شود .

نمایش کتابها : با اجرای این دستور، تمامی کتاب های موجود در کتابخانه همراه با قیمت و تعداد آنها نمایش داده خواهد شد .

بازگشت به منوی اصلی : با اجرای این دستور به منوی اصلی برمیگردیم .

تمرین ۴. خبررسان

هدف این است که برنامه ای برای کاربران طراحی کنیم تا بتوانند در آن خبر های روز را با یکدیگر به اشتراک بگذارند. خبر هایی که در این برنامه میان کاربران رد و بدل میشود به شش نوع:

۱- اقتصادی (۱۲)

۲- اجتماعی (۲۳)

۳- حوادث (۳۴)

۴- تکنولوژی (۴۵)

۵- ورزشی (۵۶)

۶- هواشناسی (۶۷)

تقسیم بندی میشود که هر مورد کد مخصوص به خود را دارد و در نتیجه آخر باید هر کدام به همراه کد آن چاپ شود.

توجه ۱: نوع خبر ها را باید به صورت Enum نگهداری کنید و در طول برنامه هر جایی نیاز به استفاده از نوع اخبار داشتید آن ها را بکار ببرید و از string استفاده نکنید، در صورت استفاده از رشته نمره ای از این قسمت به شما تعلق نمیگیرد.

توجه ۲: در این تمرین شما باید مدیریت خطا نیز انجام دهید و در صورت ایجاد هر گونه خطا در حین اجرا، برنامه شما متوقف نشود.

توجه ۳: در این سوال شما باید سه فایل txt. به نام های Contact.txt، News.txt و User.txt داشته باشید تا بتوانید داده های مربوط به هر قسمت را در فایل مربوطه ذخیره کنید.

توجه ۴: در ابتدا لازم است چند کاربر را در این سامانه ثبت نام کنید تا بتوانید دستورات موردنظر را پیاده کنید.

کلاس های موجود در برنامه :

کلاس کاربر : هر کاربر شامل نام کاربری، پسورد (به صورت رشته نگهداری کنید)، لیستی از جنس string که شامل مخاطبین آن کاربر میباشد، لیستی از جنس کلاس خبر، که خبر های ارسالی کاربر را نمایش میدهد و همچنین لیستی از نوع کلاس خبر که شامل اخبار دریافتی از سایر کاربران میباشد .

کلاس خبر : این کلاس فیلد های **آیدی خبر** (برای هر خبر آیدی منحصر به فرد ثبت کنید) ، نام فرستنده، نام گیرنده، زمان ارسال خبر، متن خبر ارسالی (خبر ارسالی شامل سر تیتر خبر بوده و حداکثر یک خط میباشد) و نوع خبر را دارا هست .

منوی برنامه :

- ورود به برنامه (Log in)
- ثبت نام (Sign up)
- افزودن مخاطبین (Add contact)
- حذف مخاطب (Remove contact)
- ارسال خبر (Send news)
- ویرایش خبر (Edit news)
- نمایش اخبار ارسالی و دریافتی کاربر (Show news)
- مرتب کردن اخبار (Sort news)
- جستجو خبر (Search news)
- حذف خبر (Delete news)
- تغییر رمز (Change password)
- خروج (Exit)



توضیح روند برنامه :

نکته ۱: تمامی تغییرات اعمال شده در برنامه میبایست در فایل ها ذخیره شوند، به طوریکه با اجرای مجدد برنامه، تغییرات بر روی داده ها اعمال شده باشند .

ورود به برنامه : هر کاربر با استفاده از **نام کاربری** و رمز عبور خود وارد برنامه میشود، واضح است اگر کاربر موفق به ورود برنامه نشود (به دلیل وجود نداشتن آن کاربر یا وارد کردن نادرست رمز عبور و یا نام کاربری) امکان استفاده از برنامه را نخواهد داشت .

توجه ۵ : تا زمانی که کاربر وارد شده دستور خروج را اجرا نکند، شخص دیگری نمیتواند وارد شود، همچنین با ورود کاربر، گزینه ثبت نام غیر فعال خواهد بود .

ثبت نام : اگر کاربری در لیست کاربران وجود نداشته باشد با گرفتن نام کاربری و پسورد در برنامه ثبت نام میشود. (در صورت وارد شدن نام کاربری که از قبل در لیست وجود داشته باشد پیغام خطای مناسبی را نمایش دهید) .

توجه ۶ : پسورد موجود در فایل User.txt را به وسیله الگوریتمی دلخواه رمزنگاری کنید؛ به گونه ای که به شکل واضح قابل مشاهده نباشد (روش های متفاوتی برای رمزنگاری وجود دارد. از جمله ساده ترین آنها میتوان به cipher encoding اشاره کرد و در موارد پیچیده تر شامل رمزنگاری RSA و ... میشود) برای این کار میتوانید از کتابخانه موجود در سی شارپ استفاده کنید .

توجه ۷ : پسورد ورودی حتما باید از فرمت زیر پیروی کند :

۱. حداقل شامل یک حرف بزرگ
۲. حداقل یک حرف کوچک
۳. حداقل یک عدد
۴. و دارای حداقل ده کاراکتر باشد



نکته ۲: برای این قسمت باید از عبارات منظم (Regex) استفاده کنید. در ادامه چند لینک برای آشنایی با این موضوع آورده شده است:

<https://learn.microsoft.com/en-us/dotnet/api/system.text.regularexpressions.regex>

<https://regexone.com/>

افزودن مخاطب: پس از ورود به برنامه کاربر میتواند از میان کاربران موجود در برنامه به لیست مخاطبینش، افراد را اضافه کند. (با گرفتن نام کاربر، در صورتی که نام او در لیست کاربران وجود داشته باشد به فرمت زیر در فایل Contact.txt، شخص به لیست مخاطبین کاربر اضافه میگردد)

username : username_contact1, username_contact2, ...

حذف مخاطب: با استفاده از این دستور کاربر میتواند با وارد کردن نام مخاطب آن را از لیست مخاطبین خود حذف کند. توجه داشته باشید وقتی کاربری از لیست مخاطبین خود شخصی را حذف میبایست اخباری که به آن شخص فرستاده است را نیز از فایل News.txt حذف نمایید.

ارسال خبر: هر کاربر می تواند خبر هایی با موضوعات داده شده و متن خبر، آن را به یکی از کاربران موجود در لیست کاربران خود ارسال کند. اگر پیام را به کاربری که در لیست کاربریش نیست ارسال کند پیغام خطای مناسبی را در کنسول نمایش دهید.

توجه ۸: پیام های ارسالی به فرمت زیر در فایل News.txt ذخیره شوند:

SenderName : (Receiver1, NEWS, type_of_NEWS), (Receiver2, NEWS, type_of_NEWS), ...

ویرایش خبر: با اجرای این دستور، آیدی خبر مورد نظر را از ورودی بگیرید و سپس متنی که میخواهید به جای آن قرار دهید را وارد کنید.



نمایش اخبار ارسالی و دریافتی کاربر: با استفاده از لیست های اخبار ارسالی و دریافتی که در کلاس کاربر تعریف کرده بودیم در این قسمت متن اخباری که آن ها را به کاربران دیگر ارسال کرده و همچنین متن اخباری که دریافت کرده است نیز در کنسول نمایش دهید .

مرتب کردن اخبار: این گزینه شامل دو قسمت است :

- **مرتب کردن بر اساس زمان ارسال خبر:** با استفاده از فیلدی که در کلاس خبر برای نگهداری زمان ارسال خبر ساخته اید اخبار موجود در فایل News.txt را بر اساس زمان ارسال مرتب کرده و در خروجی نمایش دهید .
- **مرتب کردن بر اساس آیدی خبر:** با انتخاب این گزینه میبایست تمامی خبر ها بر اساس شناسه آنها مورد چینش قرار گرفته و نمایش داده شوند .

جستجوی خبر: این گزینه شامل دو قسمت است :

- **جستجو بر اساس نام فرستنده:** کاربر با وارد کردن نام فرستنده میتواند متن خبر ها و نوع آنهایی را که آن شخص فرستاده است، مشاهده کند .
- **جستجو بر اساس نوع خبر:** با وارد کردن نوع خبر می تواند خبر هایی با آن موضوع را مشاهده کند .

توجه ۹: اگر خبری با اطلاعات داده شده یافت نشد پیغام مناسبی نمایش دهید و سپس دوباره منوی برنامه نمایش داده شود .

حذف خبر: در این قسمت با وارد کردن نام فرستنده یا نوع خبر میتوانیم خبر ها را حذف کنیم. در صورت وارد کردن نام فرستنده تمام خبر هایی که آن شخص فرستاده است از فایل خبر حذف میشود و یا با وارد کردن نوع خبر، خبر هایی با آن موضوع را حذف میکند .

تغییر رمز: با این دستور کاربر با وارد کردن رمز قدیمی می تواند رمز عبور خود را تغییر دهد. (در این قسمت نیز باید نکات گفته شده در قسمت ثبت نام را رعایت کنید) .



خروج : با وارد کردن این دستور از بخش کاربری خارج میشویم ولی منو برنامه همچنان نمایش داده شود تا کاربر دیگری بتواند Log in کند، در صورتی که بخواهیم از برنامه به طور کامل خارج شویم باید بار دیگر این دستور را اجرا کنیم .



تمرین ۵. مدیریت رستوران

در این سوال قصد داریم برنامه ای برای مدیریت یک رستوران پیاده سازی کنیم. در ادامه کلاس ها و عملیات های مورد نیاز را به طور کامل توضیح خواهیم داد .

کلاس های موجود در برنامه :

کلاس رستوران : شامل کیف پول که در آمد حاصل از فروش غذاها به آن واریز می شود و همچنین فیلدی که تخفیف ها را به صورت زوج کد تخفیف و مقدار تخفیف (هر دو عدد صحیح می باشند) ذخیره می کند تا مدیر، آنها را در هنگام خرید مشتریان بتواند بر روی قیمت پرداختی آنان اعمال کند .

کلاس مشتری : هر مشتری با نام و آیدی در رستوران ثبت نام میشود. علاوه بر این در این کلاس فیلدی برای نگهداری مقدار موجودی حساب مشتری، فیلدی برای نگهداری تعداد استفاده از تخفیف ویژه (این مورد را جلوتر توضیح میدهیم)، فیلدی برای نگهداری مشتریان ثبت شده و نیز فیلدی که کد تخفیف مشتری را در خود نگهداری میکند .

کلاس غذا : هر غذا با گرفتن نام و قیمت و نیز گرفتن یک یا چند ماده اولیه به همراه مقدار مورد نیاز آن ماده ساخته میشود همچنین فیلدی برای نگهداری تعداد غذای ساخته شده نیاز داریم .

کلاس انبار : در این کلاس فیلدی برای اسم ماده و فیلدی برای نگهداری مقدار آن ماده نیاز است همچنین لیستی از کلاس انبار که در آن هر ماده غذایی که به انبار اضافه میشود به آن Add میکنیم .

کلاس تراکنش : هر تراکنش، یک شناسه، شناسه مشتری و مقداری پول دارد و میتواند تخفیف نیز داشته باشد. بعد از خرید موفق هر غذا توسط مشتری یک تراکنش ساخته میشود که در صورت تایید شدن این تراکنش پول از مشتری کم شده و به حساب رستوران واریز می شود. همچنین فیلدی برای نگهداری تراکنش های انجام شده از نوع خود کلاس تراکنش میباشد .

منوی برنامه :

- اضافه کردن مشتری (Add customer)
- شارژ کردن حساب مشتری (Increase balance of customer)
- افزودن موجودی انبار با ماده اولیه (Add warehouse material)
- افزایش مقدار مواد اولیه موجود در انبار (Increase warehouse material)
- افزودن غذا(تا کنون در رستوران وجود نداشته است) (Add food)
- اضافه کردن غذا (Increase food)
- افزودن کد تخفیف (Add discount code)
- افزودن کد تخفیف برای مشتری (Add discount code to customer)
- فروش غذا به مشتری (Sell food)
- قبول کردن تراکنش با شماره ID (Accept transaction)
- چاپ کردن لیست تراکنش ها (Print transaction list)
- چاپ کردن موجودی رستوران (Print income)
- خروج (Exit)

توضیح روند برنامه :

افزودن مشتری : در این قسمت با گرفتن نام و آیدی شخص آن را به مشتریان رستوران اضافه میکند .

توجه ۱: شماره شناسایی یک عدد طبیعی است. و هر مشتری شماره شناسایی منحصر به فردی دارد .

مثال :

Add customer

Id: 12

Name: Moradi

توجه ۲: پس از ثبت نام شدن مشتری در رستوران شما باید آیدی و نام مشتری را به همراه **تعداد استفاده** از تخفیف ویژه را در فایل **User.txt** ثبت نمایید تا با اجرای مجدد برنامه بتوانیم از اطلاعات ثبت شده استفاده نماییم.

فرمت ذخیره در فایل به صورت زیر باشد:

User1, id1, numOfSpecialDiscount

User2, id2, numOfSpecialDiscount

توضیح تخفیف ویژه: در هر روز به اولین نفری که در رستوران خریدی ثبت می کند تخفیفی ۵ درصدی اعمال میشود. در نتیجه پس از اجرای برنامه برای اولین نفری که از رستوران خریدی انجام میدهد تخفیفی ۵ درصدی بر روی کل قیمت خرید وی در نظر گرفته می شود و در فایل text مربوطه به عدد "numOfSpecialDiscount" وی یک واحد اضافه می گردد. دقت شود که تنها به اولین خریدار در هر بار اجرای برنامه تخفیفی اعمال شود.

توضیح تکمیلی: فرض کنید برنامه برای بار اول اجرا میشود و مشتری ۱ از رستوران خریدی انجام میدهد. از آنجایی که این مشتری اولین نفری است که امروز از رستوران خرید میکند تخفیف ویژه برای وی اعمال میشود. حال مشتری ۲ خرید دیگری ثبت می نماید. از آنجایی که این مشتری نفر اولی نیست که در امروز خرید ثبت میکند به وی تخفیفی تعلق نمیگیرد. حال با دستور Exit از برنامه خارج میشویم و برای بار دوم برنامه را اجرا میکنیم. در این بار اجرای برنامه نیز باید به نفر اولی که خریدی ثبت می کند تخفیف ویژه تعلق بگیرد.

توجه ۳: اگر مشتری با شماره شناسایی تکراری بخواهد اضافه شود پیغام مناسب در خروجی چاپ نمایید و از اضافه شدن آن صرف نظر کنید.

شارژ کردن حساب مشتری: با گرفتن مقدار پول و آیدی شخص پول را به حساب او واریز مینماید.

مثال:**Increase balance of customer****Id: 12****Amount: 40000****توجه ۴:** در صورتی که آیدی شخص موجود نباشد پیغام خطای مناسب را چاپ کنید.**فروش غذا به مشتری:** با گرفتن نام غذا و تعداد و آیدی مشتری، غذا را به فروش میرساند.

- اگر نام غذا در رستوران یافت نشد خطای مناسب را در کنسول چاپ نمایید.
- اگر تعداد غذای کافی در رستوران نبود پیغام خطای مناسب را نمایش دهید.
- اگر مشتری با این شماره شناسایی وجود نداشت پیغام خطای مناسب را چاپ کنید.
- اگر پول مشتری برای خرید غذا کافی نبود خطای مناسب را در کنسول چاپ نمایید.

توجه ۵: دقت کنید در صورت برخورد با اولین خطا، نیازی به چک کردن باقی خطاها نیست. در صورتی که هیچ خطایی وجود نداشته باشد، یک تراکنش ایجاد و پیام مناسب را چاپ کنید.**مثال:****Sell food****Food Name: rice and kebab****Amount: 3****Customer Id: 12****توجه ۶:** اولین تراکنش ساخته شده با شماره ۱ ساخته شده و تراکنش های بعدی نسبت به آن شماره گذاری میشوند. همچنین اگر مشتری کد تخفیف داشته باشد این کد در تراکنش لحاظ شده و کد تخفیف مشتری باطل میشود. اگر مشتری کد تخفیف نداشت به جای آن مقدار یک قرار دهید.**افزودن موجودی اولیه با ماده اولیه:** با گرفتن نام ماده و مقدار، آن را به انبار اضافه میکند.

مثال :**Add warehouse material****Material Name: Rice****Amount: 500**

در صورتی که ماده ای که می‌خواهیم آن را به انبار اضافه کنیم از قبل در انبار وجود داشته باشد پیغام مناسب چاپ نماییم و از اضافه شدن آن صرف نظر کنید .

افزودن غذا (تا کنون در رستوران وجود نداشته است) : با گرفتن اسم غذا (مقداری یکتا و منحصر به فرد میباشد) و قیمت، یک یا چند ماده اولیه اصلی و مقدار مورد نیاز آن مواد اولیه اصلی، غذای جدید به رستوران اضافه میشود .

مثال :**Add food****Name: rice and kebab****Price: 60000****Materials: rice 5, meat 2**

توجه ۷: با اضافه شدن غذای جدید در رستوران موجودی آن صفر میباشد که برای اضافه کردن مقدار آن از دستور بعدی استفاده میکنیم .

- اگر مواد اولیه اصلی غذای داده شده در انبار یافت نشد یا مقدار آن برای تهیه غذا کافی نبود پیغام خطای مناسبی در کنسول نمایش دهید .
- اگر چنین غذایی از قبل در رستوران موجود باشد پیغام خطای مناسبی به کاربر نمایش دهید .

اضافه کردن غذا: با گرفتن نام غذا و تعداد، موجودی غذا در رستوران به همان تعداد افزایش میابد .

توجه ۸: در هنگام افزایش غذا باید به همان تعداد داده شده مواد اولیه اصلی نیز از موجودی انبار کم کنیم و در صورت کمبود مواد اصلی، پیغام خطای مناسبی نمایش دهید .

- اگر هنگام افزایش موجودی غذا آن غذا از قبل موجود نباشد پیغام خطای مناسبی نمایش دهید .

مثال :

Increase food

Name: rice and kebab

Amount: 2

افزودن کد تخفیف : هر تخفیف از دو قسمت تشکیل شده است: کد تخفیف و مقدار تخفیف (هم کد تخفیف و هم مقدار تخفیف مقادیری طبیعی هستند) .

توجه ۹: اگر کد تخفیف از قبل وجود داشته باشد پیغام خطای مناسب را نمایش دهید .

مثال :

Add discount code

Code: 4378

Price: 15000

افزودن کد تخفیف برای مشتری : با گرفتن کد تخفیف و آیدی مشتری، تخفیف را به مشتری نسبت میدهیم . هر مشتری در هر لحظه فقط میتواند یک کد تخفیف داشته باشد .

توجه ۱۰: اگر کد تخفیفی داده شده در لیست کد های تخفیف رستوران وجود نداشته پیغام خطای مناسبی را نشان دهید .

توجه ۱۱: اگر مشتری با آیدی داده شده ثبت نام نشده بود، پیغام خطای مناسبی نمایش دهید .

مثال :**Add discount code to customer****Code: 4378****Customer Id: 12**

قبول کردن تراکنش با شماره ID : با گرفتن شماره تراکنش پول از حساب مشتری کم شده و به حساب رستوران واریز میشود، اگر مشتری کد تخفیف داشته باشد در این عملیات مقدار تخفیف از کل پول کم شده و سپس مقدار باقیمانده بعد از تخفیف از حساب مشتری کم شده و به حساب رستوران ریخته میشود.

توجه ۱۲: اگر تراکنشی با این شماره وجود نداشته باشد یا قبلاً قبول شده باشد پیغام خطای مناسبی را نمایش دهید .

مثال :**Accept transaction****Id: 1**

چاپ کردن لیست تراکنش ها : با اجرای این دستور تمام تراکنش هایی که قبول شده اند به ترتیب از اولین تراکنش ساخته شده به فرمت زیر در خروجی چاپ می شوند :

Transaction [Id]: [Customer id] [payment amount] [discount code] [final price]**Payment amount = food price * amount**

• final price قیمت نهایی بعد از اعمال کد تخفیف میباشد .

چاپ کردن موجودی رستوران : با اجرای این دستور مقدار پولی که در رستوران از فروش غذا به دست آمده است، چاپ میشود .

مثال:

Print income

100000

خروج: در انتها با وارد کردن عبارت Exit اجرای برنامه پایان میابد.

موفق باشید (:)