به نام خدا

درس برنامه نویسی پیشرفته

تمرین سوم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۱-۱۴۰۰

مهلت ارسال :

۱۴۰۱ / ۱ / ۹

مبحث:

رشته ها - فایل - نوع شمارشی - مدیریت خطا

مسئول تمارین :

آریا شهسوار

فهرست

آداب نامه تمرینات

- پاسخ تمامی سوالات تنها به زبان 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
- برای هر تمرین در زمان ددلاین دانشجو باید درخواست خود را مبنی بر ارسال با تاخیر به تی ای ها اعلام نماید
- 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 بپرسید چرا که شاید سوال شما ، سوال دوستتان 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 می گیرد .
- در این سری از تمارین نیازی به نوشتن Unit test برای هیچ یک از سوالات نیست .
- در پیاده سازی کلاس ها ، دقت کنید که تمام مواردی که در سوال از شما خواسته شده است با دقت پیاده سازی شده باشد
- 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 : آریا شهسوار
  - سوال ۲ : کامیار مرادیان - ریحانه شاهرخیان
  - سوال ۳ : محمد علی فخاری - آریا شهسوار
  - سوال ۴ : محمد علی فخاری
  - سوال ۵ : محمد علی فخاری - آریا شهسوار

تمرین ۱ . کار با فایل مقدماتی

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

به این صورت که در فایل جدید تمامی فضاهای خالی بین کلمات تبدیل به * شده اند .

همچنین باید در خروجی کنسول

۱ - تعداد خطوط

۲ - تعداد اعداد

۳ - تعداد ستاره ها

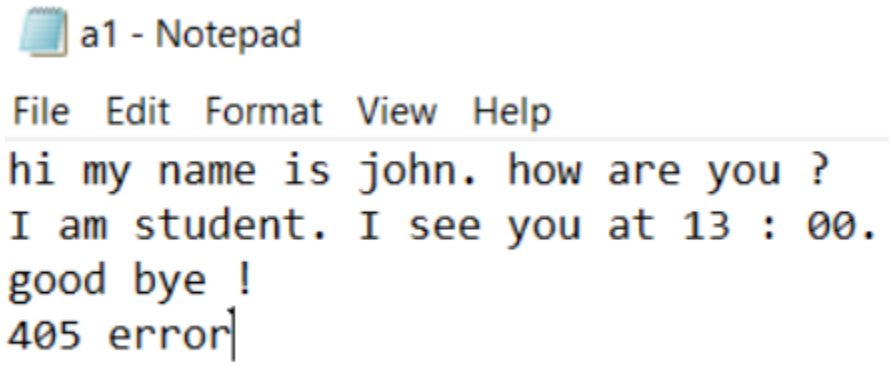
۴ - تعداد حروف صدادار فایل

۵ - تعداد کلماتی با 'a' شروع به 'e' ختم می شوند

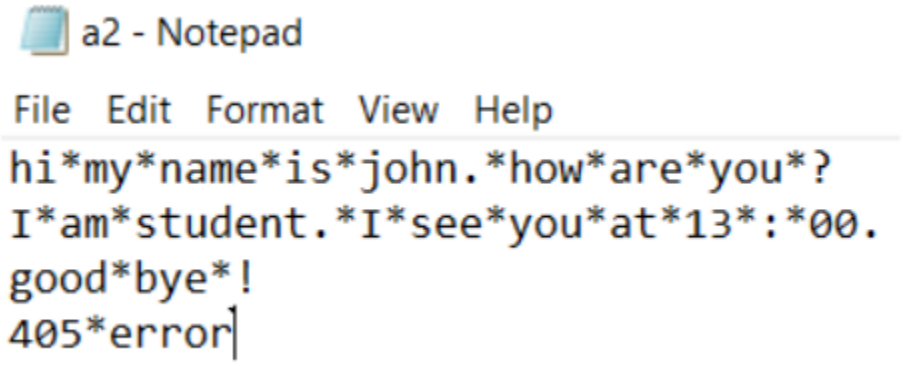
۶ - تعداد کلمه "student" در متن (بزرگی و کوچکی حروف را در نظر نگیرید به این معنا که اگر در متن هم کلمه "student" ، هم "Student" و هم "sTuDeNt" داشته باشیم ، خروجی باید عدد ۳ باشد)

نمایش داده شود.

مثال ۱ :



فایل خروجی :



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 باغ وحش

مسئول یک باغ وحش می خواهد برای نگهداری حیوانات از برنامه مدیریت اطلاعات استفاده کند و به کمک شما نیاز دارد. به همین منظور نیاز است کلاسی برای مدل کردن اطلاعات حیوانات بنویسید. هر حیوان یک شی از این کلاس میباشد. نوع حیوانات موجود در باغ وحش به شرح زیر است:

میمون (Monkey) – شیر(Lion) – فیل (Elephant) – خرس (Bear) – ببر(Tiger) – زرافه (Giraffe)

شما باید کلاسی به نام Zoo برای نگهداری اطلاعات حیوانات این باغ وحش طراحی کنید . اطلاعات هر حیوان شامل :

۱ - کد حیوان (ID)

۲ - نوع حیوان (type)

۳ - نام مستعار حیوان (name)

۴ - محل نگهداری (location)

۵ - نوع غذاهایی که میخورد (food)

است. همچنین یک فیلد مختص به کلاس وجود دارد که تعداد حیوانات موجود در باغ وحش را نشان میدهد، عنوان این فیلد (number) است. این اطلاعات فیلدهای کلاس Zoo را تشکیل می دهند.

توجه ۱ : ویژگی کد هر حیوان و همچنین تعداد حیوانات موجود در باغ وحش نباید خارج از کلاس در دسترس باشند.

توجه ۲ : نوع هر حیوان باید مقداری از یک نوع داده از enum باشند که خودتان تعریف کرده اید .

نکته ۱ : نام هیچ دو حیوانی یکسان نیست . به عبارتی در هنگام ساخت هر حیوان این مورد باید چک شده و در صورتی که چنین نباشد، خطایی با پیغام مناسب نمایش داده شود. این عمل با استفاده از IsValidName صورت میگیرد که در ادامه توضیح داده میشود.

نکته ۲ : کد هر حیوان همان تعداد حیوانات موجود در باغ وحش پس از اضافه شدن این حیوان است. به عنوان مثال، اگر ۱۰ حیوان در باغ وحش وجود داشته باشند، پس از اضافه شدن حیوان جدید، این حیوان کد ۱۱ را دریافت خواهد کرد.

نکته ۳ : در هیچ جایی از کد برنامه تان اسم حیوانات را به صورت رشته نگهداری نکنید و فقط از enum ای که تعریف کرده اید استفاده نمایید . حتی در هنگام دادن ورودی به عنوان پارامتر های ورودی سازندهٔ تابع این مورد را لحاظ کنید. در غیر این صورت نمرهٔ این بخش را از دست میدهید.

مقادیر عددی هر یک از موارد موجود در enum به شرح زیر است:

(میمون = ۱۰) – (شیر = ۱۱) – (فیل = ۱۲) – (خرس = ۱۳) – (ببر = ۱۴) – (زرافه = ۱۵)

توجه ۳ : در صورت نیاز می توانید فیلد های دیگر را نیز به کلاس اضافه کنید.

کلاس Zoo :

پارامتر های سازندهٔ کلاس، عبارت اند از : نوع حیوان (type)، نام حیوان (name) ، محل نگهداری (location) و همچنین نوع غذایی که میخورد (food) را بگیرد.

متد های موجود در کلاس :

۱- متد SaveToFile :

اطلاعات حیوانات را در فایل ذخیره میکند. برای انجام این کار بایستی به ازای هر نوع حیوان یک فایل جداگانه بسازید. به عبارتی فایل با عنوان 11 مربوط به حیواناتی از نوع شیر میباشد. در هر یک از این فایل ها، اطلاعات حیوان شامل آیدی، نام مستعار حیوان، محل نگهداری حیوان و نوع غذاهایی که میخورد ذخیره سازی میشوند. همچنین هر حیوان را با قرار دادن یک خط فاصله از حیوان بعدی جدا کنید. در صورتی که تعداد غذاهایی که یک حیوان می خورد از یک بیشتر باشد، هر مورد را با یک خط تیره از یکدیگر جدا کنید.

۲- متد IsValidName :

این متد جهت چک کردن معتبر بودن نام مستعار هر یک از حیوانات است. پارامتر ورودی این متد نام حیوان است. نام معتبر نامی است که تنها شامل حروف الفبای انگلیسی به صورت بزرگ یا کوچک باشد و همچنین تا کنون تکرار نشده باشد.

۳ -متد Change :

سه متد با این نام ولی با پارامتر های ورودی مختلف وجود خواهد داشت. در هر یک از آنها ورودی اولی نام حیوان خواهد بود. سایر پارامتر های متد ها را با توجه به نیاز آنها تعیین کنید.

متد اول : جهت تغییر محل نگهداری حیوان است.

متد دوم : جهت تغییر نوع غذاهایی که میخورد است.

متد سوم : جهت تغییر محل نگهداری به همراه نوع غذاهایی که میخورد.

نکته ۴ : تغییرات ایجاد شده در این متد روی فایل مربوط به حیوان نیز اعمال شود.

۴ -متد AllInfo :

که به صورت استاتیک تعریف می شود و به عنوان خروجی یک فایل با همین نام تولید میشود که در آن اطلاعاتی شامل :

* تعداد حیوانات برای هر نوع حیوان از حیوانات
* مجموع تعداد حیوانات موجود در باغ وحش
* تعداد حروف صدا دار موجود در نام های حیوانات موجود برای هر نوع حیوان

در قسمت main این سوال :

دقت کنید که باید پیغام مناسب برای راهنمایی کاربر هنگام دریافت ورودی و نمایش خروجی چاپ کنید.

ترتیب کارهایی که انجام می دهد به این صورت است :

در ابتدا تعداد حیواناتی که قرار است به باغ وحش اضافه کنید را از ورودی دریافت کنید. سپس در خطوط بعدی اطلاعات حیوانات از ورودی دریافت کنید.

پس از دریافت اطلاعات هر حیوان، شی متناظر از این حیوان ساخته می شود. پس از آن با صدا زدن تابع SaveToFile این حیوان را داخل فایل مربوطه ذخیره سازی کنید.

پس از دریافت تمام این اطلاعات، نوبت تغییر مقادیر مربوط به حیوانات میباشد. جهت انجام این کار از ورودی عددی را دریافت کنید. بدیهی است که مقدار این عدد نمیتواند از تعداد حیوانات موجود در باغ وحش بیشتر باشد. سپس در هر خط پس از آن نام های حیواناتی که قرار است اطلاعات آن ها را تغییر دهید به همراه مواردی که قرار است تغییر دهید را وارد کنید.

نکته ۵ : اگر در ورودی های مربوط به تغییر اطلاعات مربوط به یک حیوان هیچ یک از دو مقدار وارد نشود دوباره ورودی برای این حیوان وارد شود.

در انتها باید متد AllInfo را که صدا بزنید.

نکته ۶ : هر غذا را هم در هنگام اضافه کردن حیوان و هم در هنگام تغییر اطلاعات مربوط به آن ها با استفاده از یک comma (,) از یکدیگر جدا کنید.

نمونه ی ورودی و خروجی ها به صورت زیر است:

```
Enter the number of animals to save:
1
Enter the type:
Lion
Enter the Name:
Alex
Enter the location:
Block1
Enter the food:
Sushi,Steak
It’s time to change information.
Enter the number of animals you want to change their information:
2
Wrong! Total number of animals in the zoo is 1. Enter a smaller
number:
1
Enter the name:
Alex
Enter the new food: (if you do not want to change it press enter)
Enter the new location: (if you do not want to change it press enter)
None of the values has not entered. Enter the values again:
Enter the new food: (if you do not want to change it press enter)
Enter the new location: (if you do not want to change it press enter)
Block3
```

تمرین ۳ . سیستم بانکی

در این سوال قصد داریم تا سیستم بانکی ساده ای را پیاده سازی کنیم. این سیستم بانکی دارای اجزای زیر میباشد:

بانک : هر بانک یک نام مخصوص به خود دارد و خدمات افتتاح حساب و اعطای وام را ارائه میکند.

مشتری : هر مشتری یک اسم مخصوص به خود دارد و هیچ دو مشتری ای اسم یکسانی ندارند. نمره منفی ویژگی دیگری است که مشتری ها دارند (در مورد نمره منفی در قسمت وام بیشتر آشنا می شویم) . هر مشتری در ابتدا مقداری پول در گاوصندوق خود دارد. او میتواند یک مقدار از این پول را در هر بانکی در قالب یک حساب سپرده گذاری کند. با تمام شدن مدت تعیین شده برای حساب، مبلغ سپرده گذاری شده همراه با سود آن به صورت اتوماتیک به گاوصندوق مشتری برمیگردد.

اگر مشتری زودتر از مدت تعیین شده برای باز پس گرفتن سپرده اش به بانک مراجعه کند، تنها مبلغ اصلی(بدون سود) را دریافت میکند و به گاوصندوقش واریز میکند.

با پس گرفته شدن پول یک حساب یا به پایان رسیدن مدت تعیین شده ی حساب، آن حساب از بین میرود.

نکته ۱ : شما باید انواع مشتری را به صورت enum تعریف و استفاده کنید. به صورت پیش فرض همه مشتریان از نوع عادی میباشند اما در صورت گرفتن 5 نمره منفی تبدیل به مشتری بد حساب میشوند.

حساب : سه نوع حساب در بانک وجود دارد؛ حساب های کوتاه مدت، بلندمدت و ویژه (نوع حساب ها را به صورت enum ذخیره و استفاده کنید؛ در صورت استفاده از رشته، نمره ای از این قسمت به شما تعلق نمیگیرد). هر حساب شماره، میزان سپرده ی اولیه، درصد سود و مدتِ تعیین شده دارد. پس از سپری شدن زمانِ تعیین شده، درصد سود به علاوه ی ۱ شده و در میزان سپرده ی اولیه ضرب می شود، سپس به گاوصندوق صاحب حساب باز میگردد.

درصد سود حساب های کوتاه مدت، بلندمدت و ویژه به ترتیب ۱۰%، ۳۰% و ۵۰% است. نوع حساب (کوتاه مدت، بلند مدت و ویژه) و مدتِ حساب در هنگام ساختن حساب توسط مشتری تعیین میشود.

توجه ۱ : هر مشتری می تواند چندین حساب برای خود ایجاد کند.

نکته ی دیگر اینکه شماره ی هر حسابی که هر مشتری ایجاد میکند، یکی بیشتر از تعداد حساب هایی میشود که همان مشتری از ابتدا ساخته است؛ برای مثال اگر یک مشتری پنجمین حسابِ خود را ایجاد کند، شماره ی آن حساب ۵ میشود (یعنی ممکن است در بانکی دو حساب با شماره یکسان وجود داشته باشد ولی مطمئناً آن دو حساب متعلق به دو نفرِ مختلف هستند).

وام : هر بانک میتواند به مشتریان خود وام پرداخت کند. هر وام درصد سود و مقدار دارد. دو نوع وام داریم: وام 6 قسطه و 12 قسطه. بعد از ثبت درخواست، وام به گاوصندوق مشتری واریز میشود.

مکانیزم پرداخت اقساط به این صورت است که در هر واحد زمانی یک قسط به مقدار زیر از گاوصندوق مشتری کم میشود: (واحد زمانی یکی از گزینه های منو بوده و در ورودی آن را از کاربر میگیرید)

اگر مشتری در گاوصندوق خود به مقدار قسط پول نداشته باشد، هیچ مقداری از گاوصندوق او کم نشده، تعداد اقساط باقیمانده اش ثابت می ماند و یک نمره منفی میگیرد.

بدی نمره منفی این است که اگر مشتری ای حداقل پنج نمره منفی داشته باشد دیگر وامی به او تعلق نمیگیرد.

توجه ۲ : هر مشتری می تواند همزمان چند وام بگیرد. (در صورت برقرار بودن شرایط)

به این نکته هم توجه داشته باشید که ترتیبِ کم شدنِ اقساط وام ها هم بر اساس ترتیب به وجود آمدنِ وام ها است؛ یعنی مثلا اگر وام x زودتر از وام y در ورودی آمده باشد، قسط وام x زودتر از قسط وام y از حسابِ مشتریان کم می شود.

منوی برنامه :

- افزودن مشتری (add customer)
- افزودن بانک (add bank)
- افزودن حساب کوتاه مدت (add account)
- برداشت پول از حساب (get money)
- پرداخت وام (pay loan)
- آپدیت سیستم (update)
- نمایش اطلاعات مشتری (show info)
- خروج

توضیح روند برنامه :

افزودن مشتری : با اجرای این دستور و گرفتن نام مشتری و پول اولیه موجود در گاوصندوقش، در سیستم ثبت نام میشود.

افزودن بانک : با گرفتن نام بانک، آن را به سیستم اضافه میکنیم.

افزودن حساب کوتاه مدت : مشتری ای با نام مشخص و مقدار سپرده اولیه به مدت زمان معلوم در بانک مشخصی یک حساب سپرده گذاری ایجاد میکند.(کوتاه مدت، بلند مدت، ویژه)

توجه ۳ : در صورتی که نام بانک وارد شده وجود نداشت پیغام خطای مناسبی نمایش دهید.

توجه ۴ : اگر موجودی مشتری برای سپرده گذاری کافی نبود خطای مناسب نمایش داده شود.

به همین صورت اگر ورودی ها معتبر نبودند باید پیغام مناسب در کنسول چاپ نمایید.

برداشت پول از حساب : با گرفتن نام مشتری و شماره حساب مدنظر، مشتری میتواند تمام پول سپرده گذاری شده را برداشت نماید.

توجه ۵ : در صورت موجود نبودن مشتری و یا حساب، پیغام خطای مناسب چاپ نمایید.

پرداخت وام : با اجرای این دستور، یک وام با مقدار X و سود Y% و اقساط 6 یا 12 واحدی از بانک M به شخص A داده میشود.(تمامی مقادیر گفته شده باید از ورودی گرفته شوند)

توجه ۶ : در صورت وجود نداشتن بانک M و یا اگر مشتری از نوع مشتری بد حساب باشد(حداقل 5 نمره منفی داشته باشد) پیغام خطای مناسبی چاپ کنید و وام به شخص تعلق نخواهد گرفت.

آپدیت سیستم : با اجرای این دستور، کاربر باید واحد زمانی طی شده از زمان اجرای برنامه را وارد کند. به عنوان مثال با وارد شدن عدد 5 می فهمیم که پنج واحد زمانی از اجرای برنامه گذشته است و باید اطلاعات مربوط به وام ها و حساب ها را بروزرسانی کنیم.

نمایش اطلاعات مشتری : با گرفتن نام مشتری باید اطلاعات او همچون پول موجود در گاوصندوقش، نوع کاربری مشتری(نرمال-بد حساب) به همراه تعداد نمرات منفی کسب شده توسط مشتری و همچنین حساب های فعالش در بانک های مختلف(شماره حساب ها به همراه نام بانکی که در آن حساب دارد) را در کنسول نمایش دهید.

خروج : با اجرای این دستور برنامه پایان می پذیرد.

توجه ۷: تا زمانی که دستور خروج اجرا نشده باشد باید منوی برنامه پس پایان اجرای هر دستور نمایش داده شود.

توجه ۸ : پرداخت اقساط وام توسط مشتری و کم شدن از مدت تعیین شده برای حساب ها، در پایان هر واحد زمانی اتفاق می افتد(زمانی که کاربر دستور آپدیت سیستم را اجرا میکند) و همچنین سایر عملیات از جمله واریز سپرده حساب ها پس از تمام شدن مدت حساب نیز بلافاصله پس از اجرای این دستور انجام میشود.

تمرین ۴ . خبر رسان

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 استرینگ 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، 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 می بایست 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 می شود، 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 می شود.(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://docs.microsoft.com/en-us/dotnet/api/system.text.regularexpressions.regex?view=netframework-4.8

/https://regexone.com

افزودن مخاطبین : پس از ورود به برنامه آن 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 ، 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 میکند 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 را بر اساس زمان ارسال مرتب کرده و در خروجی نمایش دهید.

مرتب کردن بر اساس آیدی خبر : با انتخاب این گزینه می بایست 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.

توضیح کلاس ها :

کلاس رستوران : شامل کیف پول که درآمد 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 می شود 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.

ویژگی های 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 آیدی (Accept transaction)
- چاپ کردن لیست تراکنش ها (Print transaction list)
- چاپ کردن موجودی رستوران(Print income)
- خروج(Exit)

توضیح هر عملیات:

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 می شود . در نتیجه پس از اجرای برنامه برای اولین نفری که از رستوران خریدی انجام می دهد 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 ران می شود و مشتری ۱ از رستوران خریدی انجام می دهد . از آنجایی که این مشتری اولین نفری است که امروز از رستوران خرید میکند تخفیف ویژه برای وی اعمال می شود . 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 نمی گیرد . حال با دستور exit از برنامه خارج می شویم و برای بار دوم برنامه را ران 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 مناسب چاپ نمایید.

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 انبار 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 نمایید و از اضافه شدن آن صرف نظر کنید.

افزایش مقدار مواد اولیه موجود در انبار :

با گرفتن نام ماده موجود در انبار و مقدار مورد نیاز، موجودی آن را به اندازه داده شده افزایش میدهد.

مثال :

increase warehouse material

Material Name: Rice

Amount: 100

در صورتی که نام ماده داده شده در انبار یافت نشد پیغام خطای مناسب در خروجی نمایش دهید.

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 می شود.

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 همچین 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 نداشت 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 قبلا 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 می یابد.

موفق باشید :)