

به نام خدا



درس برنامه نویسی پیشرفته

تمرین ششم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۰-۱۴۰۱

مهلت ارسال :

۱۴۰۱ / ۲ / ۲۵

مبحث:

اینترفیس - ابسترکت

مسئول تمارین :

آریا شهنسوار

فهرست

- آداب نامه تمرینات ۳
- نکات تمرین سری ششم ۴
- تمرین ۱. بیمارستان ۵
- تمرین ۲. فروشگاه ۹

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 تمرین 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 پرسید چرا که شاید سوال شما ، سوال دوستان نیز باشد و دوستانتان نیز بتوانند از پاسخ سوال شما بهره ببرند .

نکات تمرین سری ششم

- سوالات را در سامانه کوئرا و در قسمت تمرین سری ششم 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 پیاده سازی کلاس ها ، دقت کنید که تمام مواردی که در سوال از شما خواسته شده است با دقت پیاده سازی شده باشد
- در هر دو سوال ، فیلدهای که اسم ها به صورت Fieldname (با underline) آمده است باید منحصر به فرد باشند .
- در بخش امتیازی سوال دوم تمرین نیاز به دانش کافی در مورد WPF است . به این منظور در کانال درس ، ویدیوهای آموزشی مربوطه برای شما دانشجویان عزیز قرار داده شده است تا بتوانید با استفاده از آنها امتیاز این بخش از تمرین را دریافت کنید . همچنین لینک های زیر در کنار ویدیوهای موجود در کانال و آموزش های کامل WPF میتوانند به شما در مسیر یادگیری این مبحث کمک شایانی کنند :

<https://www.youtube.com/watch?v=OJygSefHvr0&t=298s>

<https://www.youtube.com/watch?v=MWVfsLOhUXM>

- در صورت استفاده از WPF برای نمایش منو به کاربر دیگر نیازی به طراحی منو در کنسول نیست
 - 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 آریا شهنسوار - آقای محمد علی فخاری
 - سوال ۲ : آقای آریا شهنسوار

تمرین ۱ . بیمارستان

در این تمرین قصد داریم برنامه ای جهت مدیریت بیمارستان پیاده سازی کنیم تا بتوانیم اطلاعات بخش های مختلف را به طور صحیح ذخیره کنیم.

توجه ۱ : استفاده از مباحث مختلف شی گرای که تا به حال فراگرفته اید و همچنین مباحث مربوط به اینترفیس در این تمرین ضروری است.

اجزای برنامه :

رابط ها (interfaces) :

۱. IPerson

این واسط برای پیاده سازی افراد مختلف جامعه به کار می رود که شامل ویژگی های:

- Firstname: (نام) از نوع **string**
- Lastname: (نام خانوادگی) از نوع **string**
- SSN: (شماره ملی) از نوع **string**

است. پس هر شخصی که وجود داشته باشد، الزاماً دارای ویژگی های ذکر شده هست.

۲. IDoctor

این واسط برای پیاده سازی پزشکان بیمارستان به کار می رود که شامل ویژگی های:

- Field: (زمینه کاری) از نوع **string**
- Salary: (حقوق ماهیانه) از نوع **long**
- University: (دانشگاه) از نوع **string** (این فیلد را به گونه ذخیره کنید که بخش اول رشته نام دانشگاه و بخش دوم آن سطح دانشگاه باشد و این دو بخش با یک فاصله از هم جدا شده اند.)
- Patients: (بیماران) از نوع **List<Patient>**

و **متد work()** است. پس هر پزشکی که وجود داشته باشد، الزاماً دارای ویژگی های ذکر شده است.

کلاس ها (Classes) :

۱. Patient

هر بیمار منطقاً یک شخص حقیقی است و دارای ویژگی های مربوط به IPerson است. هم چنین علاوه بر آنها دارای دو ویژگی:

- Disease : (زمینه بیماری) از نوع **string**
- Recovered : (بهبودی) از نوع **enum** که دارای سه درجه (سالم، تحت مراقبت، بیمار) میباشد.

۲. General Practitioner

هر پزشک عمومی منطقاً یک شخص حقیقی و یک پزشک است و دارای ویژگی های مربوط به واسط های گفته شده است.

متد ها :

+ Operator : این عملگر را به گونه ای تعریف کنید که یک بیمار را به لیست بیماران تحت نظر آن پزشک اضافه کنید. بیماری واجد شرایط است که زمینه بیماری اش، شامل کلید واژه هایی مانند

Cough ، Sneezing و Sore throat باشد.

</> Operator : این عملگر را به گونه ای پیاده سازی کنید که دو پزشک را از نظر سطح دانشگاهی که در آن تحصیل کرده اند، مقایسه کند.

GraduatedFrom : این متد را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با نام و نام خانوادگی پزشک شروع شود، سپس رشته **is graduated from** باشد و در انتها نیز دانشگاهی که آن پزشک در آن تحصیل کرده بیايد.

Work : این متد که از IDoctor گرفته شده، را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با رشته **This General Practitioner works on** شروع شود و در انتها نیز زمینه کاری آن پزشک مشخص شود.

۳. Dentist

هر دندان پزشک منطقاً یک شخص حقیقی و یک پزشک است و دارای ویژگی های مربوط به واسط های گفته شده است.

متد ها :

Operator + : این عملگر را به گونه ای تعریف کنید که یک بیمار را به لیست بیماران تحت نظر آن پزشک اضافه کنید. بیماری واجد شرایط است که زمینه بیماری اش، شامل کلید واژه هایی مانند **Toothache**، **Teeth** و **Dental** باشد.

Operator >/< : این عملگر را به گونه ای پیاده سازی کنید که دو پزشک را از نظر میزان درآمدشان ، مقایسه کند.

GraduatedFrom : این متد را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با نام و نام خانوادگی پزشک شروع شود، سپس رشته **is graduated from** باشد و در انتها نیز دانشگاهی که آن پزشک در آن تحصیل کرده بیاید.

Work : این متد که از **IDoctor** گرفته شده، را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با رشته **This Dentist works on** شروع شود و در انتها نیز زمینه کاری آن پزشک مشخص شود.

۴. Surgeon

هر جراح منطقاً یک شخص حقیقی و یک پزشک است و دارای ویژگی های مربوط به واسط های گفته شده است.

متد ها :

Operator + : این عملگر را به گونه ای تعریف کنید که یک بیمار را به لیست بیماران تحت نظر آن پزشک اضافه کنید. بیماری واجد شرایط است که زمینه بیماری اش، شامل کلید واژه هایی مانند **Cancer** ، **Appendix** و **Kidney** باشد.

Operator >/< : این عملگر را به گونه ای پیاده سازی کنید که دو پزشک را از نظر تعداد بیمارانشان ، مقایسه کند.

GraduatedFrom : این متد را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با نام و نام خانوادگی پزشک شروع شود، سپس رشته **is graduated from** باشد و در انتها نیز دانشگاهی که آن پزشک در آن تحصیل کرده بیاید.

Work : این متد که از **IDoctor** گرفته شده، را بگونه ای پیاده سازی کنید که یک خروجی از نوع **string** داشته باشد که؛ با رشته **This Surgeon works on** شروع شود و در انتها نیز زمینه کاری آن پزشک مشخص شود.

حال پس از پیاده سازی بخش ها گفته شده، مسئولین بیمارستان از شما میخواهند که اطلاعات پزشکان را نگهداری کنید. بنابراین شما باید کلاسی با عنوان **Doctors** پیاده سازی کنید که شامل ویژگی های زیر باشد:

۵. **Doctors** :

این کلاس را به گونه ای پیاده سازی کنید که یک ویژگی به نام **DoctorList** از نوع لیست (لیست را به گونه ای تعریف کنید که بتوان همزمان پزشکان از کلاس های مختلف را به لیست اضافه کرد. برای این کار میتواند از مباحث ارث بری استفاده کنید) داشته باشد، سپس برای آن، دو متد زیر را پیاده سازی کنید :

۱ - **ListOfRecoveredPatient** : این متد را به گونه ای پیاده سازی کنید که بین بیماران همه پزشکان که در ویژگی **Doctor** ذخیره شده اند، را بگردد و اسم کامل بیمارانی که سلامتی خود را به طور کامل به دست آورده اند را بازگرداند.

۲ - **SortDoctors** : این متد را به گونه ای پیاده سازی کنید که پزشکان ذخیره شده در **Doctors** را مرتب کند. شیوه مرتب سازی هم به اینگونه است که ابتدا پزشکان بر حسب نسبت تعداد بیماران بهبود یافته به کل بیماران آن پزشک مرتب میشوند، و اگر دو پزشک با نسبت یکسان وجود داشت، آنها را بر حسب نام آن پزشک مرتب کند.

به طور مثال اگر پزشک شماره ۱، در مجموع پنج بیمار داشت که دو بیمار بهبود یافته باشند، و پزشک شماره ۲، در مجموع سه بیمار داشت که دو بیمار بهبود یافته داشت، در خروجی نام و نام خانوادگی پزشک دوم زودتر از اولی می آید.

توجه ۲ + راهنمایی : برای گرفتن نمره این بخش، این مقایسه باید به کمک واسط **IComparable** انجام شود. به این صورت که برای مقایسه دو دکتر **d1** و **d2** بجای آنکه بطور مستقیم ویژگی های آن دو را مقایسه کنید، تابع **CompareTo** موجود در این اینترفیس را به گونه ای پیاده سازی کنید که این مقایسه را انجام دهد و شما صرفا با فراخوانی متد **CompareTo** بر روی یکی از آنها بتوانید آن دو را با هم مقایسه کنید.

توجه ۳: پس از پیاده سازی بخش های مختلف لازم است منوی طراحی کنید تا بتوان به کمک آن بخش های مختلف برنامه را اجرا کرد. در صورتی که برنامه منو نداشته باشد نمره ای از این بخش به شما تعلق نمیگیرد

تمرین ۲ . فروشگاه

فروشنده فروشگاهی که شما سامانه ای جهت مدیریت کالاهایش آماده کرده بودید ، اعلام کرده است که رضایت کافی از برنامه شما داشته است. مدیریت کالاهایش در ایام قرنطینه آسان شده و حال از شما مجدد میخواهد که این سامانه را گسترش دهید. این سامانه جدید مدیریتی به گونه ای است که علاوه بر نظارت فروشنده بر روی کالاهایش ، مشتریان نیز برای خریداری کالاها میتوانند از این سامانه استفاده کنند.

همچنین فروشنده نیز توانسته علاوه بر کتاب، به فروش مجلات و فیلم های آموزشی نیز پردازد

من به عنوان مشاور شما طرح کلی این سامانه جدید را توضیح خواهم داد تا بتوانید این سامانه جدید را با کیفیت مطلوب تری به فروشنده فروشگاه عرضه کنید. انجام این کار باید در چند بخش صورت بگیرد تا نتیجه این برنامه باکیفیت باشد.

فاز اول : دسترسی ها و لاگین کاربران

برنامه جدیدی که شما برای این فروشگاه باید بنویسید به شرح توضیحات زیر میباشد :

این برنامه دو مدل دسترسی خواهد داشت.

دسترسی فروشنده ها به صورت **Admin**

دسترسی مشتریان به صورت **User**

توضیحات لازم در مورد نحوه دسترسی کاربران که در فوق گفته شده است :

نحوه این دسترسی ها میبایست با تعریف یک Enum صورت بگیرد.

دسترسی Admin :

این دسترسی فقط برای فروشندگان این فروشگاه است. آنها برای دسترسی به قسمت Admin کافی است عبارت **Admin** را وارد کنند . سپس باید ایمیل خود را به عنوان نام کاربری و عبارت MyShop1234\$ را به عنوان پسورد خود وارد کنند .

دسترسی User :

این دسترسی برای خریداران این فروشگاه مدنظر است. هر مشتری باید عبارت User را وارد کند تا بعد از وارد شدن به قسمت User ، مشخص کند از چه نوع خریداری هست . خریداران این فروشگاه به سه دسته تقسیم میشوند:

۱ - دانش آموزان و دانشجویان

۲ - معلمان و اساتید

۳ - مردمان عادی (یعنی غیر از موارد فوق)

دقت کنید دسترسی Admin/User صرفاً به عنوان ورودی main شما برای مشخص کردن نوع آن میباشد پس در واقع 4 نوع کاربر برای استفاده از سامانه وجود دارد که به شرح زیر است:

کلاس Seller :

مشخص است هر فروشنده از طریق دسترسی به Admin میتواند از این سامانه استفاده کند. هر فروشنده کافی است اطلاعات زیر را به عنوان نام کاربری و پسورد وارد کرده تا بتواند از سامانه استفاده کند. در واقع این اطلاعات مولفه های این کلاس هستند:

نام کاربری (باید نام کاربری فرمت ایمیل را داشته باشد) [فرمت ایمیل](#)

پسورد (به صورت پیشفرض باید عبارت \$MyShop1234 باشد).

توجه ۱ : متدی برای بررسی درست بودن فرمت ایمیل در این قسمت لازم است .

کلاس Student :

دانش آموزان یا دانشجویان بعد از وارد شدن به بخش User از طریق منو هویت خود را به عنوان Student انتخاب کرده تا بتواند وارد قسمت لاگین خود شود و با وارد کردن اطلاعات خود میتواند از سامانه استفاده کند. این اطلاعات که مولفه اصلی کلاس هستند به شرح زیر است:

نام کاربری (کافی است نام خود را وارد کنند)

شماره دانش آموزی/دانشجویی (یک عبارت 8 رقمی به فرمت 9xxxxxxx) فرمت شماره دانشجویی

توجه ۲: متدی برای بررسی درست بودن فرمت شماره دانش آموزی/دانشجویی لازم است

نکته ۱: متدی برای ذخیره اطلاعات در یک فایل به نام CustomersInfo.txt به صورت دو خط لازم است. خط اول نام کاربری و خط دوم شماره دانش آموزی/دانشجویی است.

کلاس Teacher :

معلمان و اساتید بعد از وارد شدن به بخش User از طریق منو هویت خود را به عنوان Teacher انتخاب کرده تا بتواند وارد قسمت لاگین خود شود. با وارد کردن اطلاعات خود میتواند از سامانه استفاده کند. این اطلاعات که مولفه اصلی کلاس هستند به شرح زیر است :

نام کاربری (کافی است نام خود را وارد کنند)

موسسه/مدرسه/دانشگاه (جایی که تدریس میکنند)

نکته ۲: متدی برای ذخیره اطلاعات در یک فایل به نام CustomersInfo.txt به صورت دو خط لازم است. خط اول نام کاربری و خط دوم موسسه/مدرسه/دانشگاه است.

کلاس Customer :

مشتریان معمولی بعد از وارد شدن به بخش User از طریق منو هویت خود را به عنوان Customer انتخاب کرده تا بتواند وارد قسمت لاگین خود شود. با وارد کردن اطلاعات خود میتواند از سامانه استفاده کند. این اطلاعات که مولفه اصلی کلاس هستند به شرح زیر است:

نام کاربری (کافی است نام خود را وارد کنند)

کدملی (باید ده رقمی باشد) فرمت کدملی

متدی برای ذخیره اطلاعات در یک فایل به نام CustomersInfo.txt به صورت دو خط لازم است. خط اول نام کاربری و خط دوم کدملی کاربر است.

توجه ۳ : اطلاعاتی که در فایل CustomerInfo.txt ذخیره میشوند نباید در هر دفعه ورود کاربر و ذخیره اطلاعاتش در فایل، اطلاعات قبلی فایل را پاک کند. بلکه باید اطلاعات به ترتیب و به صورت دو خط دوخط وارد فایل شوند. نام کاربری ها در واقع خطوط فرد (خط اول، خط سوم و ...) این فایل هستند.

خلاصه ای از فاز اول:

در این بخش ابتدا کاربر مشخص میکند از نوع Admin یا User است. اگر کاربر Admin باشد با وارد کردن عبارت Admin وارد قسمت لاگین ادمین میشود و بعد از آن کافی است با وارد کردن ایمیل و پسورد وارد فضای ادمین شود. اگر کاربر User باشد باید بعد از وارد کردن عبارت User مشخص کند از کدام نوع خریدار است که در فوق شرح داده ایم. پس از وارد کردن اطلاعاتش میتواند از سامانه استفاده کند.

نکات فاز اول :

- ۱ - برای وارد کردن عبارتی به یک فرمت خاص مثلا وارد کردن شماره دانشجویی یا وارد کردن ایمیل باید از عبارات منظم (Regular expression) استفاده کنید .
- در تمرین سری سوم با عبارات منظم آشنا شدید . با این حال برای مرور مجدد این مبحث میتوانید از لینک های زیر استفاده کنید :

<https://www.geeksforgeeks.org/what-is-regular-expression-in-c-sharp/>

<https://docs.microsoft.com/en-us/dotnet/api/system.text.regularexpressions.regex?view=netframework-4.8>

<https://zetcode.com/csharp/regex>

دو عبارت منظمی که لازم است استفاده کنید به صورت زیر خواهد بود:

الف - فرمت مربوط به ایمیل

A@B.C

ب - فرمت مربوط به شماره دانشجویی یا دانش آموز

9xxxxxxxx

۲ - فرمت مربوط به کد ملی

میدانیم کد ملی ۱۰ رقمی است. برای وارد کردن کد ملی باید به شرح زیر عمل کنید:

حتما با Extension Methods یا همان متدهای تعمیم دهنده آشنایی دارید. این متدها به شما این اجازه را خواهند داد که بدون اینکه از یک کلاس مشتق شده استفاده کنید یا اینکه یک کامپایل مجدد داشته باشید و یا اینکه یک کلاس را به شکل دیگری اصلاح کنید بتوانید به یک کلاس موجود، متدهایی را اضافه کنید .

حال میخواهیم یک متد تعمیم دهنده به کلاس string اضافه کنید بطوریکه شماره ملی را به عنوان پارامتر دریافت کرده و بررسی کند که آیا این فرمت برای کد ملی درست است یا خیر .

الگوریتم تشخیص صحت کدملی به صورت زیر است:

سه متغیر با نام **a** و **b** و **c** در نظر بگیرید.

متغیر a را دهمین رقم شماره ملی را در نظر بگیرید.

متغیر b برابر است با:

(اولین رقم * 10) (+) (دومین رقم * 9) (+) (سومین رقم * 8) (+) (چهارمین رقم * 7) (+) (پنجمین رقم * 6) (+) (ششمین رقم * 5) (+) (هفتمین رقم * 4) (+) (هشتمین رقم * 3) (+) (نهمین رقم * 2)

متغیر c را برابر $11\%b$ قرار دهید.

اکنون میتوانید با بررسی حالات زیر صحت شماره ملی را تشخیص دهید:

الف - اگر متغیر **c** برابر با صفر باشد و مقدار **a** برابر **c** باشد، کدملی صحیح است.

ب - اگر متغیر **c** برابر با 1 باشد و مقدار **a** برابر 1 باشد، کدملی صحیح است.

ج - اگر مقدار **c** بزرگ تر از 1 باشد و مقدار **a** برابر قدر مطلق $11 - c$ باشد، کدملی صحیح است.

د - لازم به ذکر است که شماره ملی ای که همه ارقام آن مثل هم باشد نامعتبر است به عنوان مثال

شماره ملی 9999999999 نامعتبر است.

۳ - تمامی خطاها Exceptions های ممکن باید در طول برنامه مدیریت شوند.

۴ - هر آنچه از مفاهیم شی گرای که تا به الان خوانده اید باید استفاده کنید. (کلاس و ارث بری و اینترفیس و) هر آنچه که فکر میکنید در روند برنامه شما تاثیر بهتر و بهینه ای میگذارد به شرط آنکه از چارچوب تعیین شده خارج نشود، قابل قبول است

فاز دوم : فروشگاه

فروشنده از طریق دسترسی Admin وارد صفحه لاگین خود شد . سپس با وارد کردن اطلاعات خود(نام کاربری و پسورد) میتواند فعالیت خود را در این برنامه آغاز کند.

فروشنده باید بتواند کالاهای خود را نظارت و مدیریت کند. پس بهتر است ابتدا بدانیم فروشنده برای مدیریت و نظارت کالاها به چه مواردی نیاز دارد. همچنین لازم است بدانیم کالاهای این فروشنده چه چیزهایی هستند. همانطور که گفته شد فروشنده این فروشگاه کتاب، فیلم های آموزشی و مجلات را میفروشد. پس موارد مربوط به این کالاها را بررسی میکنیم:

کالاهای فروشگاه نیز به دسته کلی تقسیم میشوند:

۱ - کتاب ها (دانشگاهی، آموزشی، داستان، رمان و ...)

۲ - فیلم های آموزشی (برنامه نویسی، فتوشاپ، آشپزی، طراحی و ...)

۳ - مجلات (روزنامه ها، مجله ها، مقاله ها و ...)

با این حساب لازم است کلاس های زیر را داشته باشید:

کلاس Media :

این کلاس اینگونه است که سه کلاس مربوط به کالاهای این فروشگاه Books, Video ,Magazines که جلوتر توضیح خواهیم داد ، از این کلاسی ارث بری خواهند کرد.

در این کلاس مولفه های زیر مدنظر هستند:

نام کالا (منظور نوع کتاب یا اسم مجله و یا غیره میباشد)

قیمت واقعی کالا برای فروشنده (که بدون مالیات ارزش افزوده است)

شناسه کالا

کلاس Books :

این کلاس به عنوان کلاس کتاب ها معرفی میشود. در این کلاس علاوه بر فیلد های کلاس Media مولفه های زیر را خواهیم داشت :

نویسنده کتاب

ناشر کتاب

متدی برای محاسبه مالیات ارزش افزوده باید وجود داشته باشد به این صورت که برای هر کتاب باید ۱۰ درصد بر روی قیمت آن بیاید.

کلاس Videos :

این کلاس به عنوان کلاس فیلم های آموزشی معرفی میشود. در این کلاس علاوه بر فیلد های کلاس Media مولفه های زیر را خواهیم داشت :

مدت زمان فیلم آموزشی

تعداد سی دی های درون بسته

متدی برای محاسبه مالیات ارزش افزوده باید وجود داشته باشد. به ازای هر سی دی درون پکیج آموزشی ۳ درصد و به ازای هر ۶۰ دقیقه آن ۵ درصد مالیات ارزش افزوده در نظر گرفته میشود.

کلاس Magazines :

این کلاس به عنوان کلاس مجلات معرفی میشود . در این کلاس علاوه بر فیلد های کلاس Media مولفه های زیر را خواهیم داشت:

ناشر مجله

تعداد صفحات مجله

متدی برای محاسبه مالیات ارزش افزوده باید وجود داشته باشد . به ازای تعداد صفحات آن این مالیات در نظر گرفته شود. اگر تعداد صفحات بین ۱ تا ۲۰ بود ۲ درصد ، اگر بین ۲۱ تا ۵۰ بود ۳ درصد و در غیر اینصورت ۵ درصد مالیات در نظر گرفته میشود.

کلاس Library :

این کلاس برای نگهداری این اشیا استفاده میشود. این کلاس باید قابلیت های زیر را داشته باشد:

متد AddMedia برای اضافه کردن Media

در این متد باید هر شی ساخته شده در یک فایل ذخیره میگردد. طرز ذخیره شدن آن در فایل باید به این صورت باشد که به اندازه مولفه های آن شی (که اندازه آن ۵ است)، در ۵ خط در فایل ذخیره گردد

متد DelMedia برای حذف کردن Media

در این متد باید بر اساس شناسه هر شی، میتوان آنرا از فایل حذف نمود. حذف شی موردنظر به معنای حذف 5 خط مربوط به آن در فایل است

متد SearchMedia برای جستجوی Media

میدانیم هر شی یک شناسه دارد. در نتیجه جستج و شی مورد نظر باید بر اساس شناسه آن صورت بگیرد. بر اساس آن شناسه باید اطلاعات مربوط به آن شی به فروشنده نشان داده میشود. این اطلاعات شامل ۵ خطی است که در فایل ذخیره شده است (اضافه کرده اید)

منو :

تمامی موارد مورد نیاز برای کالاها گفته شده است. حال نحوه دسترسی که فروشنده باید مشخص شود. چیزی که فروشنده باید در حین کار با سامانه سروکار داشته باشد در قالب یک منو میباشد که این منو به شرح زیر خواهد بود:

گزینه ADD

گزینه DELETE

گزینه SEARCH

گزینه SHOWCUSTOMERS

گزینه CHANGEPASS

گزینه EXIT

برای سه گزینه اول یعنی اضافه کردن، حذف کردن و جستجو کردن توضیحات لازم در فوق گفته شده است.

گزینه SHOWCUSTOMERS: با انتخاب این گزینه فروشنده میتواند اطلاعات کاربرانی که وارد سامانه شده اند را ببیند. این کار با استفاده از اطلاعات ذخیره شده کاربران واقع در فایل CustomersInfo.txt امکان پذیر است.

گزینه CHANGEPASS: در این گزینه به فروشنده این قابلیت را میدهد که بتواند Password خود را که به صورت پیشفرض MyShop1234\$ بوده است را تغییر دهد. همچنین لازم است ساعت و تاریخی که پسورد تغییر داده شده به فروشنده نشان داده شود. اگر فروشنده مجدد بخواهد پسورد خود را تغییر دهد باید قبل آن، برنامه آخرین باری که پسور خود را تغییر داده به او نشان دهد و بعد از تغییر پسورد، تاریخ و ساعت زمانی که پسورد را تغییر داده به او نشان داده شود. برای آشنایی به نحوه کار با تاریخ و ساعت سیستم از لینک زیر میتوانید کمک بگیرید:

<https://docs.microsoft.com/en-us/dotnet/api/system.datetime?view=netframework-4.8>

<http://zetcode.com/csharp/datetime>

گزینه EXIT: بعد از انتخاب این گزینه سیستم باید به حالتی برود که دسترسی USER/ADMIN باید مشخص شود.

خلاصه ای از فاز دوم:

پس از ورود به دسترسی ادمین و وارد کردن نام کاربری و پسورد، فروشنده باید از طریق منوی که دارد انتخاب کند که قصد انجام چه کاری را دارد. اضافه کردن، حذف کردن، جستجو کردن، تغییر پسورد و خروج. تمامی اطلاعات لازم از کالاها نیز در فوق شرح داده شده است.

نکات فاز دوم:

۱ - دقت داشته باشید ذخیره اطلاعات در فایل به نوعی حکم بانک اطلاعاتی برای شما خواهد داشت. از نحوه کارکرد آن اطمینان حاصل کنید.

۲ - تمامی خطاها Exceptions های ممکن باید در طول برنامه مدیریت شوند.

۳ - هر آنچه از مفاهیم شی گرای که تا به الان خوانده اید باید استفاده کنید. (کلاس و ارث بری و اینترفیس و) هر آنچه که فکر میکنید در روند برنامه شما تاثیر بهتر و بهینه ای میگذارد به شرط آنکه از چارچوب تعیین شده خارج نشود، قابل قبول است.

فاز سوم : مشتریان

مشتریان از طریق دسترسی User ، با انتخاب یکی از گزینه های Student ، Teacher و Customer میتوانند از این سامانه استفاده کنند.

تمامی موارد مربوط به هر یک از کلاس های Student ، Teacher و Customer در بخش اول توضیح داده شده اند. موارد دیگری نیاز است در مورد هر یک از این گزینه ها بدانیم.

معلوم است هر یک از مشتریان از سه دسته فوق باید امتیازی متفاوت داشته باشند تا از یکدیگر متمایز باشند.

مشتریان دانش آموز/دانشجو(Student) :

این دسته از مشتریان امتیاز ویژه ای خواهند داشت. آنها میتوانند برای خرید اقلام مورد نیاز خود در پایان 20 درصد تخفیف دریافت کنند.

مشتریان معلمان/اساتیدTeacher :

این دسته از مشتریان برای خرید اقلام مورد نیاز خود میتوانند به ازای خرید بیشتر از 3 کالا تخفیف 15 درصد تخفیف دریافت کنند.

مشتریان عادی(Customer) :

این دسته از مشتریان تنها با خرید بیش از 5 کالا میتوانند تخفیف 5درصدی دریافت کنند.

موارد کلی برای مشتریان:

تمامی مشتریان به هنگام خرید در سامانه یک سبد خرید دارند. این سبد خرید ظرفیت محدودی دارد لذا تنها میتوانند به اندازه ۲۰ عدد کالا خریداری کنند.

بدیهی است برای استفاده از این سبد خرید لازم است کالاهای مورد نیاز خود را به آن اضافه کنند.

برای انجام این کار کافی است نام کالای خود را جستجو کنند. اطلاعات کالای جستجو شده در صورت موجودیت در فایلی که فروشنده به آن اضافه کرده است، به مشتری نشان داده میشود و بعد از تایید توسط مشتری به سبد کالا اضافه خواهد شد.

دقت کنید هر بار اضافه کردن به سبد کالا باید قیمت کلی بدون تخفیف را به مشتری نشان دهد. تخفیف پس از تایید نهایی مشتری اعمال میشود.

مشتریان امکان حذف کالای اضافه شده به سبد کالا را خواهند داشت.

مشتریان مجاز به دسترسی اطلاعات ذخیره شده در فایل اطلاعات کالاها هستند تا بتوانند طبق آن کالای مورد نیاز خود را خریداری کنند.

منو :

چیزی که مشتری به هنگام ورود به سامانه رو به رو خواهد شد منوی است که میتواند با آن فعالیت کند. گزینه های این منو به شرح زیر خواهد بود:

SELECT گزینه

EDIT گزینه

BUY گزینه

CHANCE گزینه

EXIT گزینه

گزینه SELECT :

با انتخاب این گزینه، مشتری میتواند کالای مورد نظر خود را به سبد خرید اضافه کند. در ابتدا با انتخاب این گزینه تمامی نام کالاهای موجود در فایل اطلاعات کالاها به مشتری نشان داده میشود و طبق آن مشتری فقط کافی است نام کالای خود را وارد کند. در صورتیکه نام کالا در فایل اطلاعات کالا موجود بود به سبد خرید اضافه خواهد شد و در غیر اینصورت پیامی در قالب ناموجود بودن این کالا باید به مشتری نشان داده شود. دقت کنید نام هر کالا در فایل اطلاعات ذخیره شده در خط هایی با ضریب ۵ قرار دارند.

گزینه EDIT :

با انتخاب این گزینه مشتری میتواند کالاهای اضافه شده به سبد خرید را به همراه قیمت آنها (بدون تخفیف و همان قیمت مالیات ارزش افزوده) بررسی کند و در صورتیکه بخواهد کالایی را حذف کند فقط کافی است نام آن کالا را وارد کند تا آن کالا از سبد خرید آن حذف شود.

گزینه BUY :

با انتخاب این گزینه قیمت نهایی با اعمال تخفیف آن به مشتری نشان داده میشود و مشتری پس از تایید آن میتواند آنها را به طور کلی خریداری کند.

گزینه CHANCE :

این یک گزینه شانس برای مشتریان است! این شانس به معنای خرید با تخفیف هیجان انگیز برای مشتری است. بدین صورت است که عددی رندوم بین اعداد ۰ و ۲ و ۳ و ۵ و ۷ و ۱۰ و ۱۵ و ۲۵ و ۳۰ مشخص میشود و به صورت یک تخفیف شانس برای مشتری در نظر گرفته میشود. این تخفیف علاوه بر آن تخفیفی خواهد بود که مشتری مطابق آن از قبل ممکن است داشته باشد. دقت کنید این تخفیف ها با هم جمع خواهند شد.

گزینه EXIT :

با انتخاب این گزینه، کاربر به صفحه مربوط به انتخاب دسترسی USER/ADMIN می رود.

خلاصه ای از فاز سوم:

در این بخش با نحوه فعالیت مشتریان در سامانه آشنا شده اید. آنها بعد از ورودی و لاگین به یک منو با گزینه های توضیح داده شده برخورد خواهند داشت و مطابق آن میتوانند اقدام مورد نیاز خود را توسط سامانه خریداری کنند.

نکات فاز سوم:

۱ - شما اکنون به عنوان یک فرد با تجربه توانایی تشخیص و انجام هر یک از کارهای گفته شده مطابق با توضیحات فوق در کلاسها را دارید. تمامی کلاس ها و متدهای مورد نیاز به صورت کلی در این سه فاز گفته شده است. شما مجاز هستید هر یک از موارد گفته شده را به شرط رعایت اصول و نکات گفته شده، در قالب کلاس ها و یا متدهای جدید تعریف کنید.

۲ - برای پیاده سازی این برنامه میتوانید از لیست ها برای نگهداری اطلاعات استفاده کنید.

۳ - دقت داشته باشید ذخیره اطلاعات در فایل به نوعی حکم بانک اطلاعاتی برای شما خواهد داشت. از نحوه کارکرد آن اطمینان حاصل کنید.

۴ - تمامی خطاها Exceptions های ممکن باید در طول برنامه مدیریت شوند.

۵ - هر آنچه از مفاهیم شی گرایی که تا به الان خوانده اید باید استفاده کنید. (کلاس و ارث بری و

اینترفیس و ...) هر آنچه که فکر میکنید در روند برنامه شما تاثیر بهتر و بهینه ای میگذارد به شرط آنکه از چارچوب تعیین شده خارج نشود، قابل قبول است

امتیازی :

در صورت استفاده از WPF برای نمایش منو برنامه به شما نمره امتیازی تعلق خواهد گرفت . هر چه برنامه شما داری المان های مناسب بیشتری باشد و ظاهر زیباتری را به کاربر نمایش دهد نمره شما نیز بیشتر خواهد بود .

موفق باشید :