

پروژه پایانی ساختمان داده

سامانه داروخانه

استاد درس: استاد حسین رحمانی

بکتابش انصاری

محمد اصولیان

مقدمه :

پروژه ی سامانه ی داروخانه پیاده سازی یک سامانه بصورت گرافیکی است که با استفاده از یک ساختار داده بهینه، داده هایی شامل : نام و قیمت دارو ها و نام بیماری ها و تداخل و تاثیر دارو ها بر روی هم و بر روی بیماری ها را ساختار بندی میکند. هدف از این پروژه ایجاد سامانه ای برای دسترسی ساده تر پزشکان و داروساز ها به داروها و ایجاد یک مرجعی برای ثبت تداخلات دارویی و یا استفاده از داده های از پیش ثبت شده.

پیاده سازی :

دو عنصر مهم در پیاده سازی این سامانه برای ما 1 : بینه سازی در حافظه 2 : بهینه سازی در سرعت بود که بتوانیم از داده ساختاری استفاده کنیم که تعادل بین این دو عنصر حفظ شود.

عنصر مهم بعدی پیاده سازی رابط گرافیکی بود که بتواند با وجود داده هایی با حجم بالا بهینگی را حفظ کند و حافظه ی زیادی اشغال نکند .

مواردی که در این پروژه پیاده سازی شده است عبارت اند از :

1. کلاس داده ساختار
2. توابع مورد نیاز آن کلاس (CRD ها)
3. رابط گرافیکی با استفاده از Visual Studio و ایجاد فایل نصبی برای سامانه
4. ایجاد فایل test برای اطمینان از عملکرد برنامه
5. محاسبه مدت زمان پردازش ها و نمایش آن ها به کاربر

داده ساختار استفاده شده :

در تعیین داده ساختار برای این پروژه، سرعت در اولویت در نظر گرفته شده. داده ساختار اصلی انتخاب شده برای این پروژه Dictionary است.

دیکشنری داده ساختاری مشابه با HashTable در C# است با این تفاوت که در هش تبیل لزومی به تعیین نوع تایپ کلید و ولیو ها نیست اما در دیکشنری کلید و ولیو ها نوع ثابتی دارند و باید نوع آن ها مشخص شود.

جدول زیر پیچیدگی زمانی اعمال اصلی دیکشنری را نشان می دهد:

Operation	Time complexity
Add	$O(1)$, $O(n)$

Delete	$O(1)$
Get	$O(1)$
Contains Key	$O(1)$

به علت ثابت بودن پیچیدگی زمانی عملیات ها در دیکشنری این داده ساختار به عنوان سریع ترین داده ساختار انتخاب شد و در این پروژه استفاده شد.

ساختار کلاس های پروژه:

در این پروژه کلاس اصلی کلاس Store است که برای کار با داروخانه کافیت یک instance از این کلاس ساخته شود و محل دیتاست ها هم به آن داده شود. باقی عملیات ها با استفاده از instance ساخته شده قابل دسترسی هستند.

در کلاس Store دو کلاس کمکی دیگر به نام های Drug و Disease تعریف شده.

1- کلاس Drug:

کلاس Drug حاوی اطلاعاتیست که لازم است از هر دارو داشته باشیم. در هر شیئی از این کلاس، نام دارو، قیمت، تداخلات دارویی آن و اثرات مثبت و منفی آن بر بیماری های مختلف ثبت شده.

با توجه به این که هر Effect بین دو داروی مختلف مشترک هست و در دیتاست مربوط به Effect به نوعی هر افکت دوبار ذخیره شده میتوانستیم کلاسی از نوع افکت هم بسازیم تا ابجکت های افکت را جداگانه ذخیره کنیم تا مجبور نباشیم هر افکت را در دو داروی مختلف ذخیره کنیم، اما بین حافظه و سرعت در این قسمت سرعت را انتخاب کردیم و افکت ها را در خود دارو ها ذخیره کردیم و حاصل این کار هم این بود که توانستیم همه عملیات های CRD را با پیچیدگی زمانی $O(1)$ انجام دهیم. (البته قطعا به جز خواندن و

نوشتن داده ها چرا که لازم است تک تک خطوط خوانده شود و از $O(n)$ کم تر نمی شود)

2- کلاس Disease

در این کلاس اطلاعاتی که لازم است هر بیماری داشته باشد ذخیره میشود. شامل: نام بیماری و اثرات مثبت و منفی دارو ها بر آن بیماری.

3- کلاس Store

در این کلاس تمامی داده ها ذخیره شده و عملیات ها پیاده سازی شده. یک لیست از Drug و یک لیست از Disease، بیماری ها و دارو ها ذخیره شده اند. برخی توابع مهم این کلاس به شرح زیر هستند:

3-1- تابع ReadFromDataSet

این تابع اطلاعات را از روی دیتاست میخواند و وارد مموری میکند. خود این تابع از چهار تابع کوچک تر دیگر تشکیل شده که فایل ها را یکی یکی میخواند. اما دسترسی دیگر فایل ها private است و قرار نیست توسط کاربر استفاده شود.

3-2- تابع WriteToDataSet

این تابع اطلاعات را در فایل ها مینویسد و در واقع داده را از رم به هارد دیسک منتقل میکند. این تابع هم از چهار بخش کوچک تر تشکیل شده که هر کدام برای یکی از فایل های دیتاست است.

3-3- توابع GetDrugInfo, AddDrug, DeleteDrug

این توابع همان عملیات های CDR برای دارو هستند. پیچیدگی زمانی هر سه این عملیات ها به کمک استفاده بیشتر از مموری، $O(1)$ است و در کم تر از یک میلی ثانیه انجام میشوند.

البته تابع Add با توجه به ساختار دیکشنری گاهی از اردر $O(n)$ است. اما به طور متوسط از همان $O(1)$ هستند.

3-4- توابع GetDiseaseInfo, AddDisease, DeleteDisease

این توابع هم همان عملیات های CRD برای بیماری ها هستند. پیچیدگی زمانی هر سه این عملیات ها هم مانند عملیات های دارو، $O(1)$ است. استثنا مربوط به تابع Add در بیماری ها نیز مانند تابع Add در دارو ها برقرار است.

در عملیات های Delete برای دارو ها و بیماری ها هم اثر DeleteCascade برقرار است.

تست ها :

برای تست درستی و سرعت این توابع هم یک کلاس تست نوشته شده که در آن با دیتای زیادی کار شده تا دقت بهتری داشته باشد.

توابع تست نوشته شده به شرح زیر است:

1- تابع TestReadDataset

این تابع اطلاعات را از روی فایل ها میخواند و بررسی میکند که خواندن فایل ها با اروری مواجه نشود. برای تست خواندن فایل ها هم از یک دیتاست پر استفاده شده هم یک دیتاست خالی درون تست تولید و خوانده میشود تا کلاس در صورت خالی بودن دیتاست به مشکل نخورد.

2- تابع TestWriteToDataset

این تابع نیز اطلاعات را درون فایل ها مینویسد و بررسی میکند که نوشتن اطلاعات با اروری مواجه نشود.

3- توابع TestAddDrug, TestAddDisease

این توابع تعداد زیادی بیماری و دارو را ایجاد میکند. برای بررسی درستی دارو ها و بیماری های ساخته شده از آنجا که Effect ها و Allergy ها به صورت رندوم انتخاب شده تنها نمیتوان خود آن ها را بررسی کرد. اما با توجه به تابع AddDrug و AddDisease نوشته شده در کلاس Store، برای هر دارو و بیماری تعداد ثابتی Effect و Allergy اضافه میشود. بنابراین پس از ساخته شدن دارو ها چک شده که تعداد افکت ها و آلرژی ها برای بیماری ها و دارو های ساخته شده درست باشد.

4- توابع TestDeleteDrug, TestDeleteDisease

این توابع عمل دلیت را برای دارو ها و بیماری ها تست می کنند. بلافاصله پس از دلیت بررسی می شود که اثر delete cascade به درستی اعمال شده باشد. تمام دارو ها و بیماری های اضافه شده تا آخر دلیت می شوند و درست بودن این اثر برای همه آن ها بررسی می شود.

Exception Handling:

برای هندل کردن اکسپشن ها در پیاده سازی توابع در صورت ورود داده نا مناسب به تابع در تمامی حالات اکسپشن ArgumentException پرتاب شده و این ارور ها در بخش GIU، catch شده اند و پیغام مناسب به کاربر نمایش داده شده.

معرفی رابط گرافیکی و نحوه استفاده از آن (راهنمای استفاده):

در این بخش می‌خواهیم رابط گرافیکی سامانه را معرفی کنیم و بخش های مختلف را نمایش دهیم .

بعد از نصب سامانه و اجرای آن اولین صفحه ای که نمایش داده میشود به شکل زیر است و در این مرحله صرفاً یک فرم نمایش داده شده و پردازشی صورت نمیگیرد.

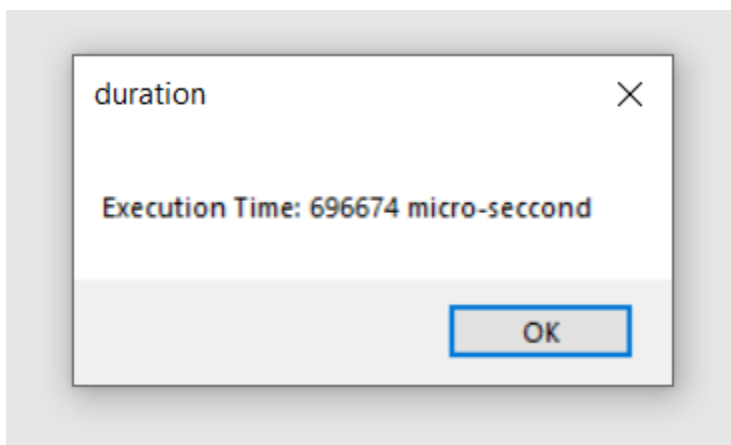


در این بخش ابتدا باید مسیر پوشه ی داده ها انتخاب شود که این کار با کلیک کردن بر روی دکمه "کلیک کنید" انجام میشود و بعد از انجام این کار صفحه ای باز خواهد شد تا پوشه انتخاب شود.

بعد از انتخاب شدن پوشه، بر روی دکمه "شروع" کلیک میکنیم و با این کار تمام داده های فولدر داخل هارد بر روی رم سیستم قرار خواهند گرفت .

دقت کنید که در صورت وارد کردن مسیر اشتباه یا وارد نکردن مسیر دوباره باید این کار انجام داده شود و با خطایی که کنترل شده است مواجه میشوید و دکمه شروع عمل نخواهد کرد.

بعد از انجام این کار صفحه کوچکی نمایش داده میشود که مدت زمان انجام این انتقال از هارد بر روی رم را بر حسب میکروثانیه نمایش میدهد.



که عدد نمایش داده شده به عوامل زیادی وابسته است.

و سپس دو فرم جدید یکی برای بخش های جست و جو و پاک کردن و اضافه کردن داده و یکی به عنوان سبد خرید برای جمع قیمت با در نظر گرفتن نرخ تورم باز میشود که به شکل زیر نمایش داده شده است :

sabadkharid

— □ ×

نام دارو را وارد کنید

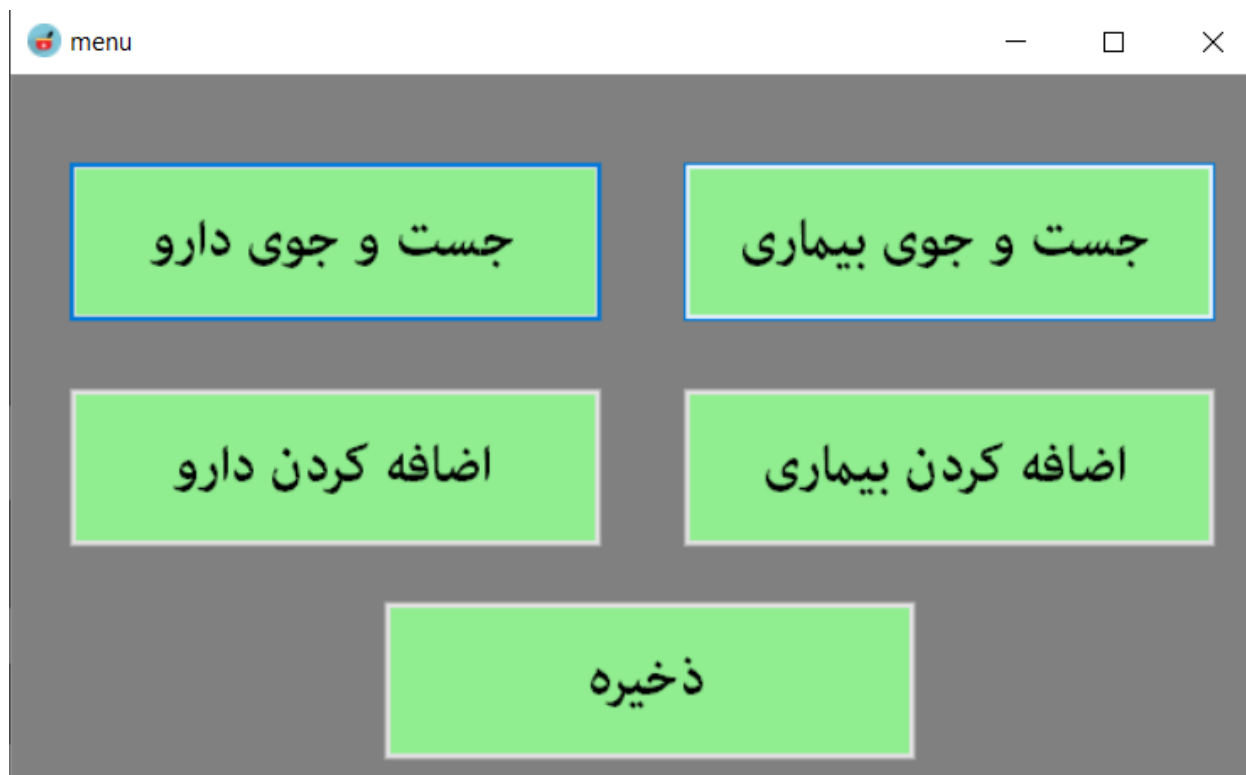
نرخ تورم

اضافه کن

قیمت

لیست خرید

Duration



سبد خرید :

با جست و جوی نام دارو و وارد کردن نرخ تورم داروی مورد نظر به لیست خرید اضافه میشود و قیمت نهایی لیست خرید نیز نمایش داده خواهد شد.

فیلد **Duration** مدت زمان پردازش برای جست و جوی دارو داخل داده ساختار را به میکروثانیه نمایش میدهد.

در بخش menu دکمه های : جست و جوی دارو, جست و جوی بیماری, اضافه کردن دارو, اضافه کردن بیماری و ذخیره مشاهده میشود.

ذخیره :

در این بخش بعد از انجام تغییرات بر روی داده های (اضافه کردن یا پاک کردن بیماری و دارو و.....) با زدن این دکمه تمام تغییرات بر روی هارد ذخیره شده و فایل داده های تغییر میکند که با زدن این دکمه زمان پردازش ذخیره ی این داده ها نیز نمایش داده خواهد شد.

اضافه کردن دارو و بیماری :

با زدن این دکمه ها صفحه های جدیدی به شکل های زیر باز میشود :

The screenshot shows a web application window titled 'adddrug'. It contains several input fields and buttons for adding drugs and diseases. The interface is in Persian.

- Top left: A small icon and the text 'adddrug'.
- Top right: Window control buttons (minimize, maximize, close).
- Form fields:
 - A text input field with the label 'نام دارو :' (Drug Name) to its right.
 - A text input field with the label 'قیمت دارو :' (Drug Price) to its right.
 - A large text area with the label 'اثرات دارو بر بیماری ها و اثرات مخرب و نوع اثر به صورت رندوم از دیتابیس انتخاب میشود' (Effects of drugs on diseases and destructive effects and type of effect are randomly selected from the database).
 - A text input field with the label 'Duration' to its right.
- Buttons:
 - A red button labeled 'اضافه کن' (Add) at the bottom left.

که با وارد کردن نام دارو و بیماری بصورت رندوم از داده های داخل فایل ها، ویژگی هایی به این دارو و یا بیماری اختصاص داده میشود و داخل داده ها قرار میگیرد.

مدت زمان انجام هر یک از این فرآیند ها نیز در بخش Duration نمایش داده خواهد شد.

نکته : در صورت وجود دارو و یا بیماری از پیش، با خطای کنترل شده مواجه خواهید شد .

جست و جوی دارو و یا بیماری :

با زدن این دکمه ها صفحه های زیر مشاهده خواهند شد :

SearchDrug

نام دارو را وارد کنید

جست و جو Clear

نام دارو :

قیمت دارو :

اثرات مثبت دارو بر بیماری ها :

اثرات منفی دارو بر بیماری ها :

اثرات مخرب و نوع اثر :

پاک کردن دارو Duration

searchdisease

نام بیماری را وارد کنید

جست و جو Clear

نام بیماری :

دارو ها با اثر مثبت :

پاک کردن Duration

در صفحه های باز شده شما با وارد کردن نام دارو و یا بیماری و با جست و جوی آن در داده ها میتوانید اطلاعات مربوط به دارو و یا بیماری را مشاهده کنید و در صورت نیاز میتوانید آن ها را پاک کنید .

با زدن دکمه clear تمامی فیلد ها خالی خواهند شد.

در فیلد duration نیز زمان پروسه ی جست و جو نمایش داده خواهد شد.

راهنمای فایل های پروژه :

DrugStore(setup) : فایل نصبی سامانه

DrugStore : کد های کامل پروژه

Datasets : داده ها

DrugStore(test) : پروژه به همراه تست

نکته : ابتدا test ها به همراه کد خام پروژه (بدون GUI) در فولدر DrugStore(test) زده شد و سپس پس از تست برنامه کد خام در پوشه اصلی پروژه (Drugstore) به همراه کد های GUI در قالب یک پروژه آماده شد.