



به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران  
مبانی کامپیوتر و برنامه‌نویسی



استاد: دکتر مرادی

پروژه ۲ - فاز ۲

نیمسال دوم ۹۸-۹۹

## یوتی کالای پیشرفته!

### • مقدمه:

در فاز اول نقش‌های افراد، اضافه شدن آن‌ها به سامانه و روابط میان کاربران را با استفاده از حافظه‌ی پویا<sup>۱</sup> و ساختارها پیاده‌سازی کردید. در این فاز از پروژه باید بخش‌های دیگری را به پیاده‌سازی اولیه خود اضافه کرده و قسمت‌های مورد نیاز را تغییر دهید. از جمله‌ی این تغییرات، پیاده‌سازی لیست‌ها با استفاده از لیست پیوندی<sup>۳</sup>، ذخیره‌سازی اطلاعات در فایل و جستجو و مرتب‌سازی کالاهای می‌باشد.

**نکته:** تمامی دستورات فاز اول باید در این فاز نیز به درستی قابل انجام باشند.

### • توضیح پروژه و ویژگی‌های مورد نظر:

در این فاز از پروژه نیز مانند فاز ۱، باید ویژگی‌های خواسته شده را در محیط خط فرمان<sup>۴</sup> پیاده‌سازی نمایید. دقت کنید که باید پس از اجرای هر کدام از دستورات زیر به صورت موفقیت‌آمیز، پیغامی مناسب به کاربر نشان دهید و در صورت بروز خطا، با پیغامی مناسب کاربر را از خطای رخ داده مطلع سازید.

### • بخش اول: ذخیره‌سازی اطلاعات در فایل

تمامی اطلاعات فروشگاه از جمله مشخصات کاربران و کالاهای باید در یک یا چند فایل ذخیره شوند و در هنگام اجرای برنامه، ابتدا اطلاعات قبلی (در صورت وجود) از فایل(ها) خوانده شده و سپس ورودی‌ها در خط فرمان دریافت شوند.

<sup>۱</sup> Dynamic Memory

<sup>۲</sup> Structure

<sup>۳</sup> Linked List

<sup>۴</sup> Command Line

بنابراین پس از خاتمه دادن به برنامه و اجرای مجدد آن، نباید هیچ اطلاعاتی از دست رفته و یا تغییر کند و برنامه می‌بایست از همان وضعیتی که در آن خاتمه یافته، شروع به کار کند.

### راهنمایی:

می‌توانید اطلاعات کاربران و کالاها را در فایل‌های `users.txt` و `goods.txt` بصورت جداگانه ذخیره کنید. در این بخش به توضیح اجمالی درباره‌ی نحوه‌ی ذخیره سازی اطلاعات کاربران در فایل `users.txt` می‌پردازیم. در هر سطر از این فایل اطلاعات مربوط به یک کاربر را ذخیره کرده و در هر سطر، بخش‌های مختلف اطلاعات را با استفاده از یک حرف<sup>۵</sup> خاص از سایر بخش‌ها جدا کنید. برای مثال اگر دو کاربر `first` و `second` داشته باشیم، محتویات فایل می‌تواند بصورت زیر باشد:

**first,first\_pass,first\_role,...**

**second,second\_pass,second\_role,...**

و منظور از ... ، سایر اطلاعات کاربر است که نیاز به ذخیره سازی داشته‌اند.

در اینجا از ویرگول<sup>۶</sup> بعنوان حرف جدا کننده استفاده کردیم. هر حرف دیگری می‌تواند بعنوان جدا کننده مورد استفاده قرار گیرد، مشروط بر اینکه آن حرف در هیچ بخشی از اطلاعات اصلی استفاده نشده باشد. همچنین برای ذخیره‌سازی کالاها نیز می‌توانید از صورت مشابهی استفاده کنید.

## • بخش دوم: لیست‌های پیوندی

### (۱) استفاده از لیست پیوندی:

در این فاز از پروژه باید تمامی آرایه‌های پویایی که برای ذخیره سازی کاربران و کالاهای فروشگاه استفاده می‌کنید را با استفاده از لیست‌های پیوندی پیاده سازی کنید. بنابراین، عملیات مورد نیاز بر روی لیست‌ها (مانند حذف و یا جستجوی یک کاربر یا کالا که در ادامه توضیح داده خواهد شد)، باید بر روی لیست‌های پیوندی ممکن باشد.

---

<sup>۵</sup> Character

<sup>۶</sup> Comma

## (۲) استفاده از درخت جستجوی دودویی<sup>۷</sup> (امتیازی)

اطلاعات کاربران و کالاها در لیست‌های پیوندی را می‌توان بصورت مرتب شده ذخیره کرد. این کار سرعت مرتب سازی و جستجو در لیست را افزایش می‌دهد.

لیست کاربران و کالاها را بر اساس نام آن‌ها در درخت جستجوی دودویی ذخیره کرده و عملیات درج، حذف و جستجو را با استفاده از این درخت انجام دهید.

### • بخش سوم: دستورات جدید

#### (۱) حذف یک کالا:

فروشنده‌ی یک کالا می‌تواند با دستور زیر کالای خود را از فروشگاه حذف کند.

**remove\_goods goods\_name**

**Example: remove\_goods watch**

با حذف یک کالا، تمامی اطلاعات مربوط به آن از جمله امتیازات ثبت شده برای آن کالا، از بین می‌رود.

در صورت بروز هرگونه خطا از جمله اشتباه بودن نام کالا، باید پیغام مناسب چاپ شود.

#### (۲) تغییر قیمت کالاها:

کاربری که با نقش **فروشنده** در سامانه ثبت نام کرده است، می‌تواند قیمت اجناس را تغییر دهد. برای این

کار باید پس از ورود به حساب کاربری خود در نقش فروشنده، به صورت زیر ابتدا دستور مورد نظر و سپس نام کالا و قیمت جدید آن را وارد کند.

در صورتی که کاربر مورد نظر کالایی با نام وارد شده نداشته باشد، باید با پیغام خطا مواجه شود.

**change\_goods\_price <goods\_name> <new\_price>**

**Example: change\_goods\_price watch 50000**

#### (۳) امتیازدهی به کالاها

هر کاربر می‌تواند تنها یکبار و فقط به کالاهایی که آن‌ها را خریداری کرده است، امتیاز دهد.

---

<sup>7</sup> Binary Search Tree

این امتیاز باید یک عدد صحیح بین ۱ تا ۵ باشد.

نحوه امتیازدهی بصورت زیر است:

**rate\_goods <goods\_name> <seller\_username> <rating>**

**Example: rate\_goods watch third\_person 3**

در صورت بروز هرگونه خطا از جمله امتیازدهی تکراری، اشتباه بودن نام کالا و یا نام فروشنده، و...، باید پیغام مناسب چاپ شود.

#### (۴) غیرفعال شدن کالاهای با امتیاز پایین

در صورتی که میانگین امتیازهای یک کالا که متعلق به فروشنده‌ای خاص می‌باشد کمتر از ۲ شود (درحالی‌که حداقل ۵ خریدار به آن کالا امتیاز داده باشند)، آن فروشنده دیگر نمی‌تواند کالای ذکر شده را به فروش برساند و در صورت درخواست یک مشتری برای خرید آن، کالای مورد نظر باید ناموجود در نظر گرفته شود.

در صورتی که تعداد خریدارانی که برای کالا امتیازی را ثبت کرده اند کمتر از ۵ نفر باشند، آن کالا غیرفعال نخواهد شد.

#### فعال شدن مجدد کالاهای:

در صورتی که میانگین امتیازهای کالایی که پیش از این غیرفعال شده است، به واسطه‌ی امتیازدهی خریداران قبلی آن کالا که هنوز امتیاز خود را ثبت نکرده اند، بیشتر یا مساوی ۲ شود، کالای مورد نظر به حالت عادی بازمی‌گردد و تا زمانی که مجدداً امتیاز آن کمتر از ۲ نشود، می‌تواند به فروش برسد.

#### (۵) مشاهده‌ی کالاهای موجود

هر کاربر باید بتواند اطلاعات تمامی کالاهای قابل فروش سیستم را با وارد کردن دستوری مانند نمونه، مشاهده کند. این اطلاعات شامل موارد زیر هستند:

- نام فروشنده
- نام کالا
- قیمت واحد کالا
- موجودی کالا
- میانگین امتیازهای کالا

**توجه:** کالاها باید بر اساس نام کالا به ترتیب حروف الفبا مرتب شده باشند.

**show\_goods**

**Example: show\_goods**

**(۶ جستجوی کالاها)**

کاربران می‌توانند کالاهای قابل فروش سیستم را بر اساس یک یا چند ویژگی خاص، جستجو و مشاهده کنند.

نحوه‌ی وارد کردن دستور بصورت زیر است:

**search <number\_of\_conditions>**

**<goods\_attribute> <attribute\_value>**

...

در خط اول ابتدا نام دستور و سپس تعداد ویژگی‌های مورد جستجو وارد می‌شود.

سپس به تعداد **<number\_of\_conditions>**، زوج رشته‌های نام ویژگی و مقدار ویژگی وارد می‌شود بطوری که در هر خط دو رشته وجود خواهد داشت.

رشته‌های مورد قبول برای **<goods\_attribute>**، ۴ رشته‌ی **name, min\_price, max\_price, seller\_username** می‌باشد.

در نهایت تمامی کالاهایی که دارای ویژگی‌های تعیین شده می‌باشند، به همراه جزئیات ذکر شده در بخش ۶، نمایش داده خواهند شد.

**توجه:** کالاها باید بر اساس نام کالا به ترتیب حروف الفبا مرتب شده باشند.

**نمونه ورودی:**

**Example: search 2**

**name watch**

**min\_price 10000**

## • نکات پایانی

- (۱) استفاده از هرگونه توابع مربوط به ++C ممنوع است و در صورت استفاده، نمره‌ای به آن تعلق نمی‌گیرد.
- (۲) شما موظف هستید که انواع خطاهای ورودی را بررسی کنید و تشخیص موارد آن بر عهده‌ی شماست.
- (۳) در این پروژه حافظه‌ها باید به صورت پویا گرفته شوند و نباید هیچگونه محدودیت عددی در تعداد کاربرها، نام کاربرها و مواردی از این دست وجود داشته باشد (تعریف آرایه با طول بزرگ قابل قبول نیست).
- (۴) شما موظف هستید تمامی حافظه‌های گرفته شده را با دستور free آزادسازی نمایید.
- (۵) تمامی لیست‌ها باید با استفاده از لیست‌های پیوندی پیاده سازی شوند.
- (۶) در این پروژه مجاز به استفاده از اشاره‌گرها و ساختارها می‌باشید.
- (۷) هرگز مجاز به استفاده از متغیرهای Global و دستور System نیستید.
- (۸) در این پروژه، مانند پروژه‌ی قبل می‌بایست بخش‌های مختلف کد خود را که عملکرد مشخصی دارند، به صورت تابع پیاده‌سازی نمایید. البته جداسازی فایل‌ها و نوشتن کتابخانه نیز توصیه می‌شود.
- (۹) در صورتی که تنها بخش مربوط به ذخیره‌سازی در فایل را به فاز قبلی اضافه کنید، نمره‌ی بخش مربوطه را خواهید گرفت.

## • شیوه‌ی نمره‌دهی:

نمره	عنوان
۵	نامگذاری مناسب و اصولی متغیرها
۵	استفاده از تمام ورودی‌های تابع در آن و نامگذاری مناسب توابع
۵	عدم وجود قطعه کد تکراری
۵	دریافت صحیح ورودی در خط فرمان
۵	رسیدگی به خطاها و نمایش پیغام مناسب
۵	نمایش پیام‌های مناسب در هر رفتار موفقیت‌آمیز
۵	کامنت‌گذاری مناسب در هر بخشی که نیاز به مستندسازی دارد
۳۵	ذخیره‌سازی و خواندن اطلاعات از فایل
۳۵	به‌کارگیری صحیح و مناسب لیست‌های پیوندی

۳۰	پیاده‌سازی درخت جستجوی دودویی (امتیازی)
۱۰	حذف کالا
۱۰	تغییر قیمت کالا
۱۰	امتیازدهی به کالا
۱۰	غیرفعال شدن کالاهای با امتیاز پایین
۱۰	فعال شدن مجدد کالاها
۱۰	مشاهده‌ی مرتب شده‌ی کالاهای موجود
۲۰	جستجو در میان کالاها
۱۵	پاسخ‌دهی مناسب هنگام تحویل و تسلط بر کد

توجه: مجموع نمرات ۲۳۰ بوده و نمره‌ی کامل ۲۰۰ می‌باشد.

#### • نحوه‌ی تحویل پروژه:

۱. تنها فایل‌های با فرمت “c” (و در صورت وجود “h”) را در یک فایل زیپ با فرمت “zip” و با نام CA2\_P2\_SID.zip قرار دهید که SID همان شماره‌ی دانشجویی شماست. برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۰۰۰ باشد باید نام فایل خود را CA2\_P2\_810198000.zip قرار دهید.
۲. تمامی پروژه‌ها برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی می‌شوند و انجام آنها به صورت انفرادی خواهد بود. همچنین، در صورت شباهت میان دو پروژه (که به وسیله‌ی نرم افزارهای مربوطه بررسی می‌شود) برای هر دو نفر نمره‌ی صفر در نظر گرفته خواهد شد.
۳. در صورت وجود هرگونه سوال و یا ابهام می‌توانید پرسش‌های خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید و یا به [ut.icsp98.ca@gmail.com](mailto:ut.icsp98.ca@gmail.com) ایمیل بزنید.
۴. به تمامی مواردی که در فروم درس یا توجیهی پروژه مطرح می‌شوند استناد خواهد شد و شما موظف هستید این موارد را بررسی کرده و آنها را رعایت نمایید.

پیروز باشید

محمدرضا عظیمی، محمدجواد ربیعی کاشانکی، حمید سالمی