پروژه اول درس برنامه نویسی پیشرفته

ساخت فروشگاه اینترنتی

نام و نام خانوادگی و شماره دانشجویی اعضای گروه:

حسین شادمان 400103443

سامان ابراهیمی 400103092

سیستم ثبت سفارش:

در کلاسی به نام Product برای اطلاعات محصولات ، سه متغیر نام کالا ، قیمت آن کالا و همچنین مشخصه های مختلف هر کالا را تعریف می‌کنیم .

در ادامه کلاسی به نام Cart برای اطلاعات سبد خرید تعریف می‌کنیم . لیست خالی‌ای به نام items تعریف می‌کنیم که بتوانیم با دستور add_item محصولی را با ذکر نام محصول ، تعداد خرید از آن محصول و مشخصه آن به آن لیست اضافه کنیم .

همچنین در این کلاس دو ویژگی به نام های remove_item و get_total_price به ترتیب برای حذف کردن آیتم خریدی از لیست خریدمان و برای بدست آوردن جمع کل قیمت خرید سبد خریدمان تعریف می‌کنیم .

و درنهایت کلاسی به نام Order برای نگهداری اطلاعات سفارش و پرداخت ایجاد می‌کنیم . در این کلاس متغیر های سبد خرید ، آدرس مشتری ، شماره تلفن مشتری ، نام و نام خانوادگی مشتری ، زمان تحویل ، شیوه تحویل و شماره سفارش تعریف می‌کنیم . زمان تحویل و شیوه تحویل را باید از ماژول لجستیک فراخوانی کنیم .(توضیح در ادامه...) همچنین شماره سفارش ، یک عدد 11 رقمی تصادفی است که با ایمپورت کردن لایبری رندم در ابتدای کد آن را تخصیص می‌دهیم .

در این کلاس ویژگی‌ای به نام checkout تعریف می‌کنیم که فایل تاییدیه درگاه را در فایلی به نام "confirmation.txt" ذخیره کند . و همچنین فاکتور را در فایلی به نام "invoice.txt" ذخیره کند .

سیستم انبارداری:

در کلاسی به نام Product برای اطلاعات محصولات ، سه متغیر کد کالا ، نام کالا و موجودی آن کالا را تعریف می‌کنیم .

همچنین کلاسی به نام warehouse برای نگهداری اطلاعات انبار و موجودی محصولات ایجاد می‌کنیم .

در این کلاس لیستی برای محصولات انبار در نظر می‌گیریم . برای اضافه کردن محصول جدید به انبار از ویژگی add_product استفاده می‌کنیم .

همچنین برای آپدیت کردن موجودی پس از سفارش تابع update_stock را تعریف می‌کنیم . این تابع به این صورت است که با گرفتن کد محصول و تعداد خریداری شده از آن محصول ، در سیستم انبارداری موجودی خود را آپدیت می‌کند .

حال برای آپدیت کردن موجودی توسط مدیر فروشگاه (به دو روش مختلف) این گونه عمل کردیم:

- باتعریف تابع manager_update_stock_from_textfile مدیر فروشگاه می‌تواند با یک فایل تکست به عنوان ورودی موجودی انبار خود را آپدیت کند . و همچنین با تعریف تابع manager_update_stock_from_csvfile مدیر فروشگاه می‌تواند با یک فایل csv به عنوان ورودی موجودی انبار خود را آپدیت کند . (برای این کار نیاز است در ابتدای کد لایبری پانداس را ایمپورت کنیم...)
- با تعریف تابع manager_update_stock_manually مدیر فروشگاه می‌تواند به صورت دستی و با وارد کردن کد کالا و موجودی جدید (به صورت ایمپوت) موجودی انبار خود را آپدیت کند .

اکنون تابع check_availability را تعریف می‌کنیم . این تابع با گرفتن کد محصول و تعداد مورد نیاز به ما می‌گوید که آیا در انبار این تعداد موجود هست یا خیر...

و در نهایت برای گرفتن خروجی از انبار ، ابتدا تابع get_stock را تعریف می‌کنیم . این تابع یک لیست از لیست هایی را برمی‌گرداند که هر لیست شامل کد محصول، موجودی فعلی و کد انبار (پیش فرض یک) است.

برای آن که فروشنده بتواند هر زمان که خواست یک خروجی از انبار خود در یک فایل تکست بگیرد ، می‌تواند از تابع get_stock_in_textfile استفاده کند . با استفاده از این تابع خروجی (که شامل اطلاعات سه ستون کد کالا ، موجودی فعلی و کد انبار است) در فایلی به نام "getstock.txt" ذخیره می‌شود .

و همچنین برای آن که فروشنده بتواند هر زمان که خواست یک خروجی از انبار خود در یک فایل csv بگیرد ، می‌تواند از تابع get_stock_in_csvfile استفاده کند . با استفاده از این تابع خروجی (که شامل اطلاعات سه ستون کد کالا ، موجودی فعلی و کد انبار است) در فایلی به نام "getstock.csv" ذخیره می‌شود .(در اینجا مجددا از لایبری پانداس استفاده می‌کنیم که در ابتدای کد ایمپورت کرده ایم...)

سیستم حسابداری:

در ابتدا کلاس اصلی به نام Successful_orders درست می‌کنیم . در این کلاس لیست خالی ای به نام orders ایجاد می‌کنیم و برای اضافه کردن سفارش ها به این لیست تابع add_orders را تعریف می‌کنیم .

برای استفاده از این تابع ابتدا باید تعداد کالا ، قیمت خالص سفارش و قیمت حمل و نقل را به آن بدهیم . همچنین این تابع دو متغیر به نام شماره سفارش ایجاد شده در ماژول سفارش گیری (برای این کار می‌بایست در ابتدای کد از فایل سیستم ثبت سفارش کلاس Order را ایمپورت کنیم) و مالیات (قیمت خالص سفارش ضربدر 9 درصد) نیز دارد . این تابع اطلاعات هر سفارش را به صورت لیستی (به ترتیب اطلاعات: تعداد کالا ، شماره سفارش ، قیمت خالص سفارش ، قیمت حمل و نقل ، مالیات) به لیست orders اضافه می‌کند .

برای آنکه فروشنده بتواند هر زمانی که نیاز داشت با یک دستور خروجی سیستم حسابداری از ابتدا تا لحظه درخواست را در قالب یک فایل csv یا تکست دریافت کند دو تابع csv_output و text_output را تعریف می‌کنیم .

با استفاده از تابع csv_output خروجی (که شامل اطلاعات پنج ستون تعداد کالا ، شماره سفارش ، قیمت خالص سفارش ، قیمت حمل و نقل و مالیات است) در فایلی به نام "output.csv" ذخیره می‌شود . (در اینجا از لایبری پانداس استفاده می‌کنیم که در ابتدای کد ایمپورت کرده ایم...)

و درنهایت با استفاده از تابع text_output خروجی (که شامل اطلاعات پنج ستون تعداد کالا ، شماره سفارش ، قیمت خالص سفارش ، قیمت حمل و نقل و مالیات است) در فایلی به نام "output.txt" ذخیره می‌شود .

سیستم لجستیک:

در این سیستم ابتدای یک کلاس اصلی برای آدرس ایجاد می‌کنیم که چهار متغیر کد شهرستان ، کد شهر ، جزئیات سفارش و کد پستی دارد .

تابعی به نام check_address برای برسی صحیح بودن و یا ناصحیح بودن آدرس تعریف می‌کنیم . این تابع اینگونه عمل می‌کند که اگر کد شهرستان مقادیر 1 یا 2 یا 3 (یک برای تهران ، دو برای اصفهان و سه برای تبریز) داشته باشد و کد شهر مقادیر 1 یا 2 (دلخواه) اختیار کند و کد پستی نیز ده رقمی باشد ، خروجی می‌دهد که آدرس صحیح است و در غیر این صورت خروجی تابع این است که آدرس ناصحیح است .

برای شیوه تحویل و تخصیص سفارش به پیک یا پست تابعی به نام delivery_method تعریف می‌کنیم . در این تابع اگر کد شهرستان 1 (تهران) باشد سفارش را به پیک تخصیص می‌دهد و در غیر این صورت سفارش را به پست تخصیص می‌دهد .

درنهایت برای زمان تحویل تابعی به نام choose_delivery_time تعریف می‌کنیم . در ابتدا نیاز است دو متغیر به نام‌های afternoon_delivery_capacity برای ظرفیت سفارش ظهر و evening_delivery_capacity برای ظرفیت سفارش عصر در __init__ کلاس ایجاد کنیم که مقدار هردو نیز در ابتدا سه است . سپس لیست خالی‌ای به نام available_delivery_times در تابع درست می‌کنیم و به این لیست زمان صبح را اضافه می‌کنیم .(چون صبح محدودیت ندارد...) سپس اگر مقدار afternoon_delivery_capacity و evening_delivery_capacity بیشتر از صفر بود به لیستمان به ترتیب زمان‌های ظهر و عصر را نیز اضافه می‌کنیم . حال از کاربر

می‌خواهیم که به صورت اینپوت بین زمان‌های قابل دلیوری زمانی را برای ارسال خود انتخاب کند . و خروجی تابع همان زمان ارسالی است که کاربر انتخاب کرده است .

همچنین در این تابع باید این نکته را لحاظ کنیم که اگر کاربر زمان ظهر و یا عصر را انتخاب کرد به ترتیب از مقدار afternoon_delivery_capacity و evening_delivery_capacity یک واحد باید کم شود .

و در انتهای پروژه نیاز است به فایل سیستم ثبت سفارش برویم و با ایمورت کردن این فایل در ماژول سفارش گیری ، متغیر های زمان تحویل و شیوه تحویل را تخصیص دهیم . برای این کار در ابتدای کد سیستم ثبت سفارش از ماژول لجستیک کلاس آدرس را ایمپورت می‌کنیم . سپس به کلاس Order کد می‌رویم و متغیرهای زمان تحویل و شیوه تحویل را از ماژول لجستیک فراخوانی می‌کنیم .

و پایان پروژه...(: