

بسمه تعالی

سوالات امتحان پایان ترم برنامه‌نویسی پیشرفته

دانشکده مهندسی کامپیوتر – دانشگاه علم و صنعت ایران

خرداد ۱۴۰۱

این فایل حاوی ۴ سوال است. برای پاسخ به این ۴ سوال ۱۲۰ دقیقه فرصت دارید.

سوال ۱- برنامه‌نویسی شیء‌گرا (وراثت) (۱.۲۵ نمره)

یک برنامه بنویسید که مخزن موجودی منابع سخت افزاری یک شرکت را ذخیره کند. هر منبع در برنامه شما باید به صورت یک شیء از یک کلاس مدلسازی شود. منابع مورد استفاده در این شرکت شامل CPU، HDD و SSD است. در این شرکت هر منبع ممکن است در یکی از وضعیت های آزاد، در حال استفاده یا از رده خارج باشد. همچنین ممکن است در هر لحظه منابع جدیدی به مخزن شرکت اضافه شود (خریداری شود)

یک کلاس پایه به نام Resource داریم. در این کلاس باید عملکردهای مشترک همه منابع واقعی پیاده‌سازی شود و سایر کلاسها باید از این کلاس ارث ببرند. هیچ شیء مستقیمی از این کلاس ساخته نمی شود. فرض کنید در این شرکت CPU، HDD و SSD داریم. این کلاس باید دارای ویژگی های زیر باشد:

name	نام نمونه کلاس Resource (مثلا Intel Core i9-9900K)
manufacturer	سازنده نمونه Resource (مثلا Nvidia)
total	تعداد موجودی از آن نوع قطعه
allocated	تعداد قطعاتی از آن نوع که در حال استفاده است
__str__	تنها نام نمونه را برگرداند.
__repr__	نمایش با جزئیات بیشتر
claim(n)	متودی که n تا نمونه از مخزن را برای استفاده برمی‌دارد. (البته در صورت موجود بودن آن از آن منبع)
freeup(n)	متودی که n تا نمونه را به مخزن برمی‌گرداند
died(n)	متودی که n تا منبع را به صورت دائمی از مخزن خارج می‌کند (n طبیعتاً از تعداد کل موجودی کمتر یا مساوی است)
purchased(n)	متودی برای اضافه کردن n منبع به مخزن
category	یک computed property که اسم کلاس را (به حروف کوچک) برمی‌گرداند.

سپس کلاس‌های فرزند CPU، HDD و SDD را هم تعریف کنید.

برای کلاس CPU:

cores	تعداد هسته ها (مثلا ۸)
socket	نوع سوکت سی.پی.یو (مثلا AM4)
power_watts	توان سی.پی.یو (مثلا ۹۴)

برای HDD و SDD باید یک کلاس دیگر هم به نام Storage تعریف کنید که دارای property زیر باشد:

capacity_GB	مثلا ۱۲۰
-------------	----------

و کلاس HDD از کلاس Storage ارث می برد و دارای property های اضافه زیر است:

size	مثلا 2.5"
rpm	چرخش در دقیقه (مثلا 7000)

کلاس SSD از کلاس Storage ارث می برد و دارای property اضافه زیر است:

interface	نوع رابط که ssd به وسیله آن به سیستم وصل میشود (مانند PCIe NVMe 3.0 x4)
-----------	--

برای همه کلاسها باید سازنده ای تعریف کنید که property ها را مقداردهی کند، باید مقادیر عددی را اعتبارسنجی کنید و همچنین \_\_repr\_\_ مناسبی پیاده سازی کنید.

total و allocated در کلاس Resource باید property های فقط خواندنی باشند، attribute های متناظر آنها باید private باشند. مقادیر آنها در متوذهای claim، return، died و purchased تغییر می کند. سایر attribute ها مانند manufacturer و ... باید فقط خواندنی باشند.

در دایرکتوری tests در گیت هاب درس تعدادی تست وجود دارد که میتوانید به کمک آنها عملکرد برنامه خود را تست کنید.

سوال ۲- خزش وب و چندنخی! (۵.۱ نمره)

یک کلاس به نام Twitter تعریف کنید که نمونه های آن دارای دو ویژگی نام کاربری و پسوندد باشند. برنامه شما از این نام کاربری و پسورد برای لاگین کردن در اکانت توییتتر کاربر استفاده می کند.

۱. متودی به نام get\_account\_info در آن تعریف کنید که نام کاربری یک اکانت توییتتری را بگیرد و زمان پیوستن صاحب اکانت به توییتتر را به صورت یک رشته برگرداند. (در صورتی که به هر دلیل موفق به نوشتن این تابع نشدید، متودی بنویسید که اطلاعات موجود در bio این کاربر را برگرداند و ۵۰ درصد نمره این بخش را بگیرید)

۲. متودی به نام get\_accounts\_info بنویسید که نام یک فایل csv شامل تعدادی نام کاربری کاربران توییتتر را بگیرد و با استفاده از برنامه نویسی چندنخی (ایجاد یک Thread به ازای هر نام کاربری) و تابعی که در بخش اول نوشته اید اطلاعات مورد نظر از هر کاربر را به دست آورد و در یک دیتا فریم که ایندکس آن نام کاربری و ستون اول آن اطلاعات استخراج شده است، ذخیره کند. سپس این دیتا فریم را به صورت یک فایل اکسل به نام info.xlsx ذخیره کند.

۳. در قسمت main برنامه کد خود را برای تعدادی نام کاربری دلخواه امتحان کنید.

ممکن است توییتتر در فواصل زمانی کوتاه به درخواست های یک حساب کاربری پاسخ ندهد. برنامه شما در قسمت main باید بتواند با ایجاد thread های موازی و استفاده از حداقل دو نمونه مجزا از کلاس Twitter و همچنین wait های مناسب تا حدی مشکل را رفع کند. پاسخ خود را در یک فایل py. به نام twitter.py بنویسید و در کوئرا آپلود کنید.

سوال ۳- شی گرای (چندریختی) (۵.۱ نمره)

یک کلاس به نام Fraction ایجاد کنید. یک Fraction (کسر) به صورت نسبت a/b تعریف می شود به نحوی که مقدار b نباید صفر باشد. کلاس Fraction دارای دو attribute محافظت شده (صورت و مخرج کسر) و دو property برای خواندن و نوشتن آنهاست.

- ایجادکننده شی (سازنده شی)

یک شی Fraction ایجاد کند؛ a صورت کسر و b مخرج کسر باشد. مثلاً Fraction(5,6) برابر با کسر ۵/۶ است. a و b باید protected باشند و مقدار هر دو اعتبارسنجی شود. a و b باید حتما عدد صحیح باشند و b باید عدد صحیح غیر صفر باشد. در غیر این صورت exception مناسب raise شود و هیچ شیئی ساخته نشود.

توجه داشته باشید که کسرها باید به ساده ترین حالت ذخیره شوند، یعنی ۶/۴ باید به صورت ۳/۲ ذخیره شود.

- بازنمایی شی

کلاس باید دارای تابعی باشد که بتوانیم بازنمایی رشته ای نمونه های کلاس Fraction را از طریق توابع str و print داشته باشیم. مثلا برای کسر مثبت 4/5 باید رشته 4/5 را برگرداند. در صورتی که کسر منفی باشد، باید علامت منفی با یک فاصله پشت کسر درج شود. در صورتی که تابع برای یک کسر غیر معتبر فراخوانی شود باید رشته initialization Failed برگردانده شود.

## - محاسبات ریاضی

توابع مناسبی برای پشتیبانی از عملگرهای + , - , / , \* بین کسرها پیاده سازی کنید. مثلا Fraction(2,3)\*Fraction(2,5) باید Fraction(4,15) را برگرداند. تقسیم در صورتی که عملوند دوم صفر است باید منجر به exception با پیغام مناسب شود.

علاوه بر آن هر کدام از عملگرها باید برای حالتی که عملوند سمت چپ یا راست عدد صحیح یا عدد اعشاری باشند هم درست کار کنند.

## - عملگرهای مقایسه

متودهای مناسبی برای مقایسه دو کسر با استفاده از عملگرهای == , != , > , < , >= , <= تعریف کنید. برای == یک مقدار خطا در حد  $10^{-7}$  را در نظر بگیرید.

## - متود نمونه

یک متود نمونه به نام decimal تعریف کنید. این متود باید معادل دهمی کسر را با حداکثر ۷ رقم اعشار برگرداند. مثلا decimal().fraction(1/4) عدد اعشاری ۰.۲۵ را برمیگرداند.

## - متود کلاس

یک متود کلاس به نام fraction تعریف کنید که یک عدد اعشاری را به صورت دهمی بگیرد و کسر معادل آن از نوع Fraction را برگرداند. مثلا Fraction.fraction(2.5) باید Fraction(5,2) را برگرداند. اگر پارامتر ورودی این متود یک عدد صحیح است، مخرج کسر را برابر با ۱ در نظر بگیرید.

## - یونیت تست

برای تمام متودها و عملگرهای این کلاس حداقل یک تست با استفاده از unittest بنویسید.

## - duck typing

در قسمت main برنامه با یک مثال مفهوم duck typing را در پایتون نشان دهید.

سوال ۴ – برنامه نویسی تابعی (۰.۷۵)

- تکنیک memoization را در پایتون توضیح دهید و به سه روش برای پیاده سازی آن اشاره کنید. تابع فاکتوریل را با این تکنیک پیاده سازی کنید. (توضیحات خود را در قالب کامنت در فایل پاسخ بنویسید)
- یک دیکوریتور به نام progress تعریف کنید که در صورتی که به هر تابعی اعمال شود، زمان فراخوانی تابع را چاپ کند. همچنین مقدار تمام متغیرهای محلی تابع را هر ۱۰ ثانیه یک بار چاپ کند. در پایان اجرای تابع هم مقدار تمام متغیرهای محلی تابع را چاپ کند.

پاسخ این سوال را در یک فایل به نام Q4.py کپی کنید و در کوئرا آپلود نمایید.