(به نام خداوند بخشنده‌ی مهربان)

درس برنامه‌نویسی پیشرفته

تمرین سری دوم

دانشکده‌ی مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۰-۱۳۹۹

مهلت ارسال: ۱۳۹۹/۱۲/۳۰

مبحث تمرینات:

مقدمات شی گرایی

۱.همایش‌های موفقیت

در اوایل ترم پس از گفتن جمله‌ی«قویا براشون فشردگی ترم معنی نداره.» توسط یکی از دانشجویان، در بچه‌ها شور و اشتیاق زیادی برای درس خواندن ایجاد شد. با مشاهده‌ی ایجاد این انگیزه در بچه‌ها توسط مسئولین دانشگاه و بازگویی این موضوع برای افراد مختلف دیگر، چند تن از مسئولین برگزاری همایش‌های موفقیت با این دانشجو جلسه گذاشتند و از او برای برگزاری سلسله همایش‌های موفقیت دعوت به عمل آوردند.

پس از برگزاری چند همایش از جمله همایش چگونه یک شبه صاحب بنز شوید، رفته رفته به تعداد شرکت کنندگان در این همایش‌ها افزوده شد. با بیشتر شدن تعداد شرکت کنندگان، مدیریت این مراسم‌ها سخت‌تر شد. برای همین قرار شد تا سیستمی برای مدیریت این همایش‌ها نوشته شود.

کلاس Conference

فیلدهای کلاس: اسم همایش از نوع string، اسم سالن برگزاری همایش از نوع string، ظرفیت سالن از نوع int، اعضایی که در این همایش ثبت نام کردند که آرایه‌ای از کلاس Participant است که در جلوتر توضیح داده می‌شود.

تمامی این فیلدها باید به عنوان پارامتر ورودی توسط سازنده ی کلاس دریافت و مقدارد هی شوند.

از این رو دو مدل سازنده برای این کلاس باید بنویسید.

سازنده اول همه فیلدها به غیر از آرایه از participent را به عنوان پارامتر ورودی میگیرد.

سازنده دوم همه فیلدها را از جمله آرایه شرکت کنندگان می گیرد.

متد AddParticipant: این متد برای اضافه کردن افراد شرکت کننده در همایش است. با یک بار صدا کردن این متد می‌توان به تعداد دلخواهی از شرکت کنندگان(participant) را به لیست شرکت کنندگان همایش اضافه کرد.

در واقع شما باید این امکان را فراهم کنید که با یکبار فراخوانی این متد هم امکان اضافه کردن یک participant باشد، هم امکان اضافه کردن پنج participant، هم امکان اضافه کردن ده participant. در همه‌ی موارد بالا این متد یکبار صدا زده می‌شود.

اگر ظرفیت سالن همایش پر شده باشد، نباید امکان اضافه کردن شرکت کننده‌ی جدید به کاربر داده شود. و در صورت رخ دادن این موضوع، باید پیغام مناسب در کنسول به کاربر نمایش داده شود.

کلاس participant

این کلاس دارای فیلدهای زیر است:

یک متغیر از نوع string برای نگهداری نام، یک متغیر از نوع string برای نگهداری نام خانوادگی، یک متغیر از نوع int برای نگهداری ID، متغیری از جنس int برای مشخص کردن قیمت بلیت شرکت در همایش

این کلاس باید دو سازنده داشته باشد. سازنده‌ی اول تنها پارامتر ID را به عنوان ورودی دریافت می‌کند.

سازنده‌ی دوم نام، نام خانوادگی، ID و وضعیت سلامتی فرد را به عنوان ورودی دریافت می‌کند.

متد استاتیک CalculatePrice: با دیدن تمایل افراد به شرکت در این همایش‌ها، مدیران این همایش‌ها تصمیم گرفتند که باتوجه به تعداد شرکت کنندگان، قیمت بلیت‌ها را افزایش دهند. شما باید این متد را طوری بنویسید که هر دفعه نسبت به دفعه‌ی قبل که فراخوانده می‌شود، مقدار بیشتری را برگرداند. در اولین بار که این متد فراخوانی می‌شود، باید عدد ۱۰۰۰ را برگرداند. در دفعه‌های بعدی باید عددی بیشتر از دفعه‌های قبل و در محدوده‌ی int را به عنوان خروج برگرداند. توجه شود که این متد را حتما باید به صورت استاتیک تعریف کنید.

متد CountParticipants: این متد استاتیک به عنوان خروجی تعداد کل عضوهای ساخته شده را برمی‌گرداند.

۲.کلاس دایره

در این سوال، کلاس دایره به نام Circle را تعریف کنید.

این کلاس شامل ویژگی های شعاع و مختصات طول و عرض مرکز دایره است. ویژگی های دایره از خارج آن قابل دسترسی نیستند.

متد هایی که لازم است برای این کلاس پیاده سازی شود به شرح زیر است:

- سازنده کلاس که شعاع، طول و عرض مرکز دایره را به عنوان پارامتر ورودی می گیرد.
- متد برای محاسبه محیط دایره.
- متد برای محاسبه مساحت دایره.
- متد برای تعیین فاصله یک نقطه تا مرکز مختصات (این متد مربوط به کلاس است نه یک شی)
- متد تعیین فاصله نقطه ای از مرکز دایره
- متدی که به عنوان ورودی یک نقطه بگیرد و تعیین کند آن نقطه در داخل یا خارج یا روی دایره قرار دارد.

همچنین در قسمت Main برنامه خود، باید شعاع و عرض و طول مرکز دایره را از کاربر گرفته و شی ای از کلاس دایره بسازید. سپس، با استفاده از متدهایی که در کلاس دایره نوشته اید (فراخوانی آنها) موارد زیر را (برای دایره ای که ساخته اید) در خروجی به کاربر نمایش دهید:

محیط دایره، مساحت دایره،(مختصات نقطه ای را در ورودی از کاربر بگیرد) اندازه فاصله‌ی نقطه تا مرکز دایره، و اینکه نقطه در چه قسمتی از دایره قرار دارد (داخل ، خارج ، روی دایره) که به صورت یک عبارت نمایش داده شود.

بخش دوم:

برای هر دایره یک جفتی وجود دارد که عرض مرکز مختصات آن یک واحد بیشتر، طول مرکز مختصات آن دو واحد کمتر و شعاع آن از دوبرابر شعاع دایره‌ی قبلی سه واحد کمتر است. کلاس

دایره باید دارای یک متد کپی باشد که با فراخوانی آن جفت متناظر دایره ساخته شود و به عنوان خروجی برگردانده شود.

در Main برنامه، ابتدا تعداد دایره ها را از کاربر بگیرید.

در خطهای بعدی، به ترتیب در هر خط اطلاعات مربوط به یک دایره (به وسیله کاربر) وارد خواهد شد.

پس از دریافت ویژگی های مربوط به هر دایره، باید اطلاعات گفته شده در بخش اول مربوط به آن دایره و دایره جفتش را چاپ کنید. (دقت کنید ابتدا باید دایره را بسازید و با استفاده از متد کپی آن، جفتش را نیز بسازید.)

پس از چاپ اطلاعات مربوط به یک دایره، در خط بعدی اطلاعات دایره‌ی بعدی را دریافت کنید..

در انتها پس از دریافت و نمایش تمام اطلاعات دایره‌ها، لیست مرتب شده‌ای از موارد زیر را باید به کاربر نمایش دهید

لیست مرتب شده‌ی دایره‌ها بر اساس محیط، لیست مرتب شده‌ی دایره‌ها بر اساس مساحت، لیست مرتب شده‌ی دایره‌ها بر اساس فاصله‌ی مرکز دایره‌ها از مبدأ مختصات. در این بخش باید اطلاعات دایره‌های کپی هم در نظر گرفته شود.

در انتها باید همه‌ی آنچه را که در کنسول به کاربر نمایش داده‌اید در یک فایل circle.txt نیز ذخیره کنید.

توجه شود که شعاع دایره نمی‌تواند عددی منفی یا صفر باشد. در صورتی که کاربر به عنوان ورودی بخش غیرمجازی داد، شما باید از کاربر درخواست کنید تا دوباره ورودی‌های خود را وارد کند. همچنین برای دایره‌های کپی، اگر شعاع این دایره‌ها صفر شد، شما شعاع یک را برای آن‌ها در نظر بگیرید. اگر شعاع این دایره‌ها عددی منفی به دست آمد، شما قدرمطلق این مقدار را به عنوان شعاع در نظر بگیرید.

۳.مدیریت کارخانه

مدیر یک کارخانه به شیوه ای که در ادامه توضیح داده می شود، کارخانه خود را اداره می کند. برنامه ای طراحی کنید که به او اجازه دهد صرفا با نوشتن دستوراتش، این کار را انجام دهد.

کارمندان در کارخانه ی معرفی شده، به چهار درجه(Degree) با حقوق ماهانه ی زیر تقسیم می شوند:

$ حقوق ۱۰۰ :Worker1- درجه ی

$ حقوق ۳۰۰ :Foremanدرجه ی ۲-

$ حقوق ۷۰۰ :Supervisorدرجه ی ۳-

$ حقوق ۹۰۰ :Leaderدرجه ی ۴-

*(میتوانید اخراج شده (Fired) را درجه ۰ و بازنشسته(Retired) را درجه ۵ درنظر بگیرید.)

*برنامه ما تنها دستور های زیر و دستور("End Career") را به عنوان دستور های valid قبول دارد. درصورت ورود هر رشته دیگر و یا رشته های زیر خارج از این ترتیب, برنامه ارور " Input is not valid" را چاپ کرده و از مدیر می خواهد دوباره دستور را وارد کند.

کلاسی با نام Employee تعریف کنید که دارای متغیر های Degree (درجه) و Balance (مانده حساب) و سه متغیر bool با نام (کارمند ویژه)Special و (قبلا وام گرفته)Loaned و hired(یک بار استخدام شده در شرکت) با مقدار اولیه false است. این کارخانه حداکثر دارای ۱۰۰ کارمند است. پس ابتدا در قست main برنامه آرایه ای ۱۰۰ تایی از شی Employee بسازید که دارای متغیر hired=false هستند.

متد های مناسب برای اجرای دستور های زیر پیاده سازی کنید:

hire [name] [degree]:

مدیر با این دستور کارمندی را با نام name و درجه degree به استخدام شرکت در می آورد. با استخدام هر کارمند، برای او حسابی باز می شود که حقوق های دریافتی او در آن قرار می گیرد.

(یعنی شی ای از کلاس Employee با name و degree داده شده و مقدار Balance 0 و متغیر Special – False Loaned-False و hired-trueساخته می شود.)

به نکات زیر توجه نمایید:

• حساب کارمندان در ابتدا خالی بوده و هیچ پولی در آن قرار ندارد.

• نام کارمندان یکتا و با حروف کوچک است. (متدی تعریف کنید که این شرط را هنگام گرفتن ورودی بررسی کند و در صورت برقرار نبودن شرط پیام خطا چاپ کند و دوباره ورودی بگیرد.)

• تضمین می شود که کارمندی با نام تکراری استخدام نمی شود. (متدی تعریف کنید که این شرط را هنگام گرفتن ورودی بررسی کند و در صورت برقرار نبودن شرط پیام خطا چاپ کند و دوباره ورودی بگیرد.)

• مدیر در مدت ریاست خود، حداکثر صد نفر را استخدام می کند. اگر کسی اخراج یا بازنشسته شود، مدیر کسی راجایگزین او نمی کند.

pay [name]:

زمان پرداخت حقوق هر یک از کارمندان بستگی به تصمیم مدیر کارخانه دارد. یعنی در یک زمان مشخص ممکن است یک کارمند سه بار حقوق دریافت کند درحالی که کارمند دیگری هیچ حقوقی دریافت نکرده باشد. مدیر باید امکان پرداخت حقوق به یک کارمند را با دستور بالا داشته باشد. در نتیجه ی این دستور، موجودی حساب فردی با حقوق ماهانه ی او جمع می شود.(متدی پیاده سازی کنید که در شی ها دنبال شی ای با این name بگردد و حقوق او را باتوجه به درجه حقوقش به Balanceاش اضافه کند)

get [name] [quantity]:

افراد کارخانه می توانند از پول حساب خود استفاده کنند و این کار با دستور بالا انجام می شود. در این صورت به میزان quantity از موجودی او کم می شود. اگر موجودی او کافی نباشد، پیام NotEnoughMoney

چاپ می شود.(متدی پیاده سازی کنید که در شی ها دنبال شی ای با این name بگردد و از Balanceاش به میزان quantity کم کند)

special [name]:

یک کارمند در صورتی که اقدام شایسته ای انجام دهد به کارمند ویژه تبدیل می شود. این مزیت براساس تصمیم مدیر است و با دستور بالا انجام می شود. مدیر می تواند چندین بار این دستور را برای یک نفر اعمال کند، ولی تاثیر آن تنها در دفعه ی اول است. (متدی پیاده سازی کنید که در شی ها دنبال شی ای با این name بگردد و متغیر Special را true کند.)

loan [name]:

کارمندان می توانند درخواست وام دهند. درخواست درصورتی پذیرفته می شود که تا به حال وام نگرفته باشند و جزو کارمندان ویژه باشند. اگر درخواست پذیرفته شود، به اندازه ی سه برابر حقوق ماهانه اش به حساب او اضافه می شود و پیام accepted چاپ می شود .در غیراین صورت پیام rejected چاپ می شود. (متدی پیاده سازی کنید که در شی ها دنبال شی ای با این name بگردد و اگر متغیر loaned آن شی false بود, به Balanceاش به میزان سه برابر حقوق(با توجه به degree اضافه کند) و متغیر loaned را true کند.)

promote:

هر زمانی که مدیر تصمیم بگیرد، تمام کسانی که شرط (کارمند ویژه بودن) را داشته باشند، ترفیع می گیرند. مثلا worker ها تبدیل به Foreman و Leader ها بازنشسته (Retired) می شوند. به این عملیات promote گفته می شود و با دستور بالا قابل انجام است.(حساب فرد بازنشسته محفوظ مانده و می تواند از آن برداشت نماید.)

همچنین اگر حداقل یک Leader در عملیات promote بازنشسته شود، در کارخانه جشنی برگزار می شود و همه کارمندان ویژه که قبلا وام گرفته اند، بعد از آن می توانند دوباره وام بگیرند.

(متدی پیاده سازی کنید که تمام شی های کارمندان را بررسی کند و درصورتی که متغیر Special آن ها true بود degree شان را یک درجه افزایش دهد.(درجه retired آخرین

درجه است) و در هنگام انجام این دستور اگر یکی از شی ها به درجه بازنشستگی رسید, تمام شی ها را دوباره بررسی کنید و اگر شی ای هر دو متغیر loaned و special اش true بود متغیر loaned را false کنید.)

regress:

هر زمانی که مدیر تصمیم بگیرد، تمام کسانی که شرط(کارمند ویژه نباشند) را داشته باشند، تنزل مقام می گیرند. مثلا Foreman ها تبدیل به Worker می شوند و Worker ها اخراج می شوند. به این عملیات regress گفته می شود و با دستور بالا قابل انجام است. حساب فرد اخراج شده محفوظ مانده و می تواند از آن برداشت نماید.

(متدی پیاده سازی کنید که تمام شی های کارمندان را بررسی کند و درصورتی که متغیر Special آن ها false بود degree شان را یک درجه کاهش دهد.(درجه fired آخرین درجه است.)(

report [name]:

مدیر میتواند گزارشی از کارمندی خاص بخواهد. برای این کار دستور بالا را وارد می کند. گزارش شامل افراد اخراج شده(Fired) و بازنشسته(Retired) هم می شود. کارمندان ویژه (حتی کارمندان سابق) با لفظ special که قبل از نامشان می آید شناخته می شوند.

(متدی پیاده سازی کنید که تمام شی ها را بررسی کند و شی ای با name داده شده بیابد و در خروجی اطلاعات مربوط به آن کارمند را مطابق دستور زیر چاپ کند.)

[name] [degree] [credit]

دقت کنید:

اگر کسی از شرکت بازنشته یا اخراج شود، تنها می تواند از پول خود برداشت نماید و دیگر نمی تواند حقوق یا وام یا ...بگیرد. (درخواستی غیر از دریافت حقوق، برای چنین اشخاصی، داده نمی شود.) متغیر hired در این افراد true می ماند. چون این متغیر نشان دهنده یک بار استخدام شدن در شرکت است.

(برنامه همواره تا زمان ورود دستور "End Career" ادامه پیدا میکند.)

۴. اعداد مختلط:

در این برنامه قصد داریم با اعداد مختلط سر و کار داشته باشیم. ما می‌دانیم که هر عدد مختلط به صورت a + bj تعریف می‌شود.(a جز حقیقی و b جز موهومی است.) باتوجه به روابطی که از اعداد مختلط می‌دانید(جمع، تفریق، ضرب و تقسیم) ابتدا کلاسی با نام TwoComplex تعریف کنید. این کلاس ۴ مولفه با نام‌های مثلا a,b,c,d از نوع int که در واقع دو عدد مختلط ما را که به فرم (a,b) و (c,d) بیان می‌کند، دارد.

در این کلاس متدی با نام start() از نوع void وجود دارد که ۴ مولفه‌ی مذکور را به عنوان ورودی از کاربر دریافت می‌کند. در واقع نیازی به گرفتن ورودی در main برنامه نداریم.

حال به همین ترتیب متدهایی با نام‌های add, sub, mul, div وجود دارند که عملیات مربوطه را با روابطی که در اعداد مختلط می‌دانید انجام داده و چاپ می‌کند.

همچنین باید یک متد به نام changeNumbers ایجاد کنید. در این متد، کاربر مقادیر a,b,c,d را می‌تواند تغییر دهد. به طوری که بعد از فراخوانی این متد، به ترتیب از کاربر خواسته می‌شود تا مقادیر جدید a,b,c,d را وارد کند. پس از اینکار، در صورت فراخوانی متدهای add,sub,mul,div عملیات‌های مربوطه باید با اعداد جدید که ساخته شده‌اند انجام شود.

در قسمت main برنامه ابتدا یک شی از کلاس TwoComplex باید ساخته شود. سپس برای دریافت ورودی‌ها متد start فراخوانی شود. در مرحله‌ی بعد، شما باید منویی طراحی کنید که با انتخاب گزینه‌ی مناسب، حاصل عملیات مربوطه را از طریق فراخوانی متدهای تعریف شده در کلاس Complex نمایش می‌دهد.علاوه بر آن باید برای متدchangeNumbers هم گزینه‌ی جدا در نظر بگیرید تا کاربر در صورت نیاز بتواند اعداد را تغییر دهد. همچنین باید یک گزینه برای خروج از برنامه در نظر بگیرید و تا زمانی که دستور مربوط به خارج شدن از برنامه وارد نشده باشد، برنامه باید ادامه پیدا کند.

مثلا اگر add گزینه‌ی ۱ منو باشد، در صورتیکه ۱ را انتخاب کنیم باید حاصل جمع دو عدد مختلط نمایش داده شود.

۵. کتابخانه

در این تمرین باید یک برنامه برای مدیریت یک کتابخانه بنویسید.

- کتابخانه آرایهای از اعضا و آرایهای از کتاب ها (و تعداد موجود از هر کتاب) دارد.
- هر کتاب یک نام و یک شناسه (id) منحصر به فرد دارد.
- هر عضو یک نام و یک id منحصر به فرد و آرایهای از کتاب هایی که امانت گرفته دارد.

در این سوال شما باید کتاب‌ها و اعضا را به صورت شی از کلاس خودشان ذخیره کنید.

مسئول کتابخانه با اجرای این دستور ها کتابخانه را مدیریت میکند (در هر خط یک دستور وارد میشود):

addBook [id] [name] [count]

این دستور تعداد count جلد ازکتابی با مشخصات name و id به مخزن کتاب اضافه می‌کند. (فرض کنید بیش از ۵۰ عنوان کتاب به مخزن اضافه نمی‌شود.)

addMember [id] [name]

این دستور عضوی با مشخصات name و id به فهرست اعضا اضافه می‌کند. (فرض کنید بیش از ۵۰ عضو اضافه نمی‌شود.)

get [member_id] [book_id]

با اجرای این دستور، به کاربری با شناسه id_member کتابی با شناسه id_book امانت داده میشود، در صورتی که تعداد کتاب هایی که امانت گرفته از سقف مجاز (۵ جلد) بیشتر نباشد؛ در غیر این صورت پیام

MaxReached : [member_name] [member_id]

چاپ می‌شود؛ و این که حداقل یک جلد از کتاب مربوطه موجود باشد؛ در غیر این صورت پیام

NotAvailable : [book_name] [book_id]

چاپ میشود. بدیهی است که تعداد جلد های موجود از کتاب فوق و آرایه کتاب های عضو مربوطه باید آپدیت شوند.

return [member_id] [book_id]

کاربری با شناسه id_member کتابی با شناسه id_book را پس میدهد.

bookStat

با این دستور، خلاصه از وضعیت کتاب ها با فرمت زیر گزارش میشود.

[name1] [id1] [count1]

[name2] [id2] [count2]

.

.

.

memberStat

با این دستور، خلاصه از وضعیت اعضا (شامل آرایه کتاب هایی که امانت گرفته اند) با فرمت زیر گزارش میشود.

[member_name] [id] [[book1_name] [book1_id] - [book2_name] [book2_id] - ...]

[member_name] [id] [[book1_name] [book1_id] - [book2_name] [book2_id] - ...]

.

.

.