به نام خدا

درس برنامه نویسی پیشرفته

تمرین ششم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۲ -۱۴۰۱

مهلت ارسال :

۱۴۰۲/۰۳/۱۹

مباحث :

جنریک – ابسترکت – اینترفیس

مسئول تمارین :

آریا شهسوار

فهرست

آداب نامه تمرینات

- پاسخ تمامی سوالات تنها به زبان های C# قابل قبول می باشد .
- علیرغم اعتماد کامل تیم تی ای به شما دانشجویان عزیز، تمامی کد های شما با سایر دانشجویان بصورت خودکار و توسط برنامه مقایسه خواهند شد . همچنین در طول ترم ، از تمامی پاسخ های شما ارائه گرفته خواهد شد و نحوه کار تمامی بخش های هر سوال از شما پرسیده خواهد شد، لذا از کپی نمودن کد دوستانتان خودداری کنید و تمامی پاسخ ها، کد خودتان باشد . همچنین از آنجایی که مشورت و هم فکری با سایر دوستان بسیار کار پسندیده و مفیدی است – برخلاف کپی کردن کد (: – در صورت هم فکری با دانشجوی دانشجویان ، نام وی را بصورت کامنت شده در ابتدای کد خود بنویسید.
- برای ارسال تمارین در طول ترم، در مجموع ۷ روز می توانید تاخیر داشته باشید و در صورتی که جمع تاخیر دانشجویی بیشتر از ۷ روز شود، تمرین وی قابل قبول نخواهد بود لذا تلاش کنید تمرینات را در زمان مقرر در سامانه آپلود کنید.
- در تمامی تمرینات سعی شده است که سوالات ساده تر در ابتدا و سوالات دشوار تر در انتهای فایل قرار گیرند (از ساده به دشوار مرتب شده اند) .
- در صورت وجود هرگونه سوال در مورد تمرینات ، سعی کنید تا جایی که امکان دارد سوال خود را در گروه بپرسید چرا که شاید سوال شما، سوال دوستتان نیز باشد و دوستانتان نیز بتوانند از پاسخ سوال شما بهره ببرند.

نکات تمرین سری ششم

- **سوالات را در سامانه کوئرا و در قسمت تمرین ششم آپلود نمایید .**
- **در تمامی سواالت باید مدیریت خطا بطور کامل و دقیق صورت گیرد و بخشی از نمره هر سوال به** Exception Handling **درست و مناسب تعلق میگیرد .**
- **در این سری از تمارین نیازی به نوشتن** Unit test **برای هیچ یک از سوالات نیست.**
- **در پیاده سازی کلاس ها ، دقت کنید که تمام مواردی که در سوال از شما خواسته شده است با دقت پیاده سازی شود.**
- **از آنجایی که هر سوال توسط یک تی ای طرح شده است ، تنها تی ای طراح آن سوال می تواند شما را بصورت دقیق راهنمایی کند به همین منظور طراح هر سوال در زیر نوشته شده است تا در صورت ابهام و پرسش در مورد هر سوال، در صورتی که نیاز به پرسش سوال بصورت انفرادی در پیوی هست ، به تی ای مربوطه مراجعه بفرمایید.**
  - **سوال ۱ . آقای درپوش**
  - **سوال ۲ . آقای اوشنی**
  - **سوال ۳. آقای درپوش**

تمرین ۱ . شبیه ساز کامپیوتر

در این سوال قصد داریم یک شبیه ساز برای ساخت کامپیوترهای شخصی سازی شده بسازیم .
در این سیستم هر کامپیوتر از یک پردازنده، مموری و نمایشگر تشکیل شده است. هر قطعه از طریق یک رابط کاربری با کامپیوتر ارتباط برقرار میکند. در انتها یک برنامه را با کامپیوتر شخصی سازی خود پیاده سازی می کنید.

رابط های مورد نیاز :

IProcessor :

تابع Compute :

این تابع به این صورت کار میکند که دو رشته به عنوان ورودی میگیرد و یک رشته خروجی میدهد. نحوه پردازش این دو ورودی به کلاسی که شامل این اینترفیس میباشد بستگی دارد.

IMemory :

تابع Load :

این تابع یک رشته به عنوان ادرس ورودی میگیرد و به عنوان خروجی رشته ای که در ادرس مورد نظر حافظه وجود دارد را برمیگرداند.

تابع Store :

این تابع یک رشته به عنوان ادرس و یک رشته به عنوان دیتا میگیرد و دیتا ورودی را در خانه ای از حافظه با ادرس ورودی ذخیره میکند. این تابع خروجی ندارد.

IDisplay :

تابع Display :

این تابع یک بیت ارور به همراه یک رشته ورودی می گیرد و آن را در کنسول نمایش میدهد. در صورتی که بیت ارور برابر یک باشد عبارت “ :Error” به ابتدای رشته خروجی اضافه میشود.

کلاس های مورد نیاز :

IntegerCPU :

این کلاس شامل اینترفیس IProcessor است . در تابع Compute این کلاس جمع دو ورودی داده شده خروجی داده میشود .(توجه شود که ورودی های تابع Compute از نوع رشته هستند پس در ابتدا باید این ورودی ها به معادل عددی خود تبدیل شوند).

StringCPU :

این کلاس شامل اینترفیس IProcessor است . در تابع Compute این کلاس رشته ورودی دوم به آخر رشته ورودی اول چسبانده میشود و در نهایت خروجی داده میشود .

StaticMemory :

این کلاس شامل یک آرایه از استرینگ ها است که سایز آن درون Constructor ورودی گرفته شده و به عنوان حافظه استفاده میشود. همچنین این کلاس شامل اینترفیس IMemory نیز میباشد.

نکته ۱ : توجه شود که اگر در متد های Load یا Store ادرس حافظه نامعتبر وارد شد باید خطا ایجاد شود .

DynamicMemory :

این کلاس شامل یک لیست از استرینگ ها است که قابلیت رشد دارد و به عنوان حافظه استفاده میشود. همچنین این کلاس شامل اینترفیس IMemory نیز میباشد.

نکته ۲ : توجه شود که اگر در متد های Load یا Store ادرس حافظه نامعتبر وارد شد باید خطا ایجاد شود .

RGBDisplayer :

این کلاس دارای یک ابجکت ConsoleColor است که در Constructor ورودی گرفته میشود و رنگ خروجی را در کنسول را مشخص میکند. همچنین شامل اینترفیس IDisplayer است. در صورتی که بیت ارور فعال بود خروجی باید به رنگ قرمز چاپ شود.

BlackAndWhiteDisplayer:

این کلاس شامل اینترفیس IDisplayer است .

Computer:

این کلاس شامل یک آبجکت از نوع IProcessor , IMemory , IDisplayer میباشد .

تابع RunCommand :

این تابع دو رشته به عنوان عملگر و یک رشته به عنوان ادرس ورودی میگیرد. نحوه کار این تابع به این صورت است که ابتدا دو عملگر را به تابع Compute پردازنده میدهد و خروجی را در ادرس مشخص شده از ممووری ذخیره میکند.

متد Main :

در این متد یک کامپیوتر شخصی سازی شده برای خود بسازید و با استفاده از تابع RunCommand عبارت 1111222233334444 را چاپ کنید.

تمرین ۲ . بازی شطرنج

در این سوال شما باید یک شبیه ساز شطرنج تک نفره را طراحی کنید . به این منظور شما میبایست یک کلاس ابسترکت به نام مهره تعریف کنید .
این کلاس دارای متد است که با توجه به حرکات قانونی مهره ها در بازی اصلی برای هر مهره متفاوت است.

کلاس های مورد نیاز دیگر :

مهره رخ :

توانایی : مهره رخ ما برخلاف مهره رخ در شطرنج تنها توانایی حرکت رو به جلو به اندازه دو واحد یا حرکت به سمت طرفین به اندازه یک واحد را دارد.

مهره فیل :

توانایی : مهره فیل مانند مهره شطرنج توانایی حرکت به صورت اریب را داراست .

مهره اسب :

توانایی : می‌تواند مانند بازی شطرنج به صورت L حرکت کند .

مهره وزیر :

توانایی : این مهره می‌تواند مانند فیل و یا رخ حرکت کند.

قوانین بازی :

برخلاف شطرنج واقعی در این بازی ما فقط ۴ مهره متمایز از مهره های بالا داریم. صفحه بازی ما ۸*۴ است که در ابتدا هر مهره در ستون اول و در ردیف های مختلف قرار دارند.

نکته ۱ : در صورتی که بخواهیم از یک نوع مهره بیش از ۱ عدد داشته باشیم خطای مناسب نمایش داده شود.

نکته ۲ : در صورتی که یک مهره با یک حرکت از صفحه خارج شود لازم است که خطای مناسب نمایش داده شود .

طریقه انجام حرکت :

در ورودی دو عدد طول و عرض را برای یک مهره دریافت کرده و در صورتی که در بازه حرکتی مهره نباشد خطای مناسب را نمایش دهید .

نکته ۳ : در صورتی که در خانه ای که پس از حرکت مهره ما می رود، مهره دیگری قرار داشته باشد، مهره ای که در آن مکان قرار داشته، به خانه اولیه خود منتقل می شود.

شرط پایانی بازی :

بازی ما زمانی به پایان میرسد که یکی از مهره های ما به ستون آخر برسد.

نکته ۴ : ترتیب حرکت مهره ها بر اساس ساخت آن هاست و برای این که یک دور تمام شود تمام مهره ها باید حرکت کنند.

تمرین ۳. ساختمان داده Queue

در این سوال از شما خواسته شده است که ساختمان داده صف را به صورت جنریک پیاده سازی بکنید .

این نوع داده ای به صف مانند است و بر روی آن دو عمل Dequeue و Enqueue تعریف میشود .
به این صورت که ، Dequeue اولین عضو را خارج میکند و Enqueue یک عضو جدید را به انتهای صف اضافه میکند .

کلاس مورد نیاز :

MyQueue:

Constructor کلاس جنریک MyQueue را به گونه ای بنویسید، که دو ویژگی Size از نوع Integer و Values از نوع لیستی از همان نوع داده ای که قرار است، تشکیل شود .فیلد سایز باید از بیرون کلاس قابل خواندن باشد ولی قابل تغییر نباشد .

تابع Enqueue :

این متد را به گونه ای پیاده سازی کنید که یک ورودی از نوع داده ای مناسب بگیرد، و اولا Size را یک واحد افزایش دهد، دوماً این نوع داده ای را به لیست Values اضافه کند.

تابع Dequeue :

این متد را به گونه ای پیاده سازی کنید که اولا Size را یک واحد کاهش دهد، دوماً اولین داده ای که اضافه شده است را حذف کند. دقت کنید که پس از انجام این دستور، آن داده باید از لیست Values نیز حذف شود .

تابع Print :

این متد را بگونه ای پیاده سازی کنید، که داده های صف را در قالب یک رشته برگرداند که ترتیب شان، همان ترتیب خروجشان از صف است .همچنین دقت کنید که در این متد شما مجاز به استفاده از مقادیر Values نیستید و برای به دست آوردن مقادیر ذخیره شده، باید از متد های Dequeue و Enqueue استفاده کنید .

موفق باشید :)