



برنامه سازی پیشرفته سی شارپ

دکتر ملکی مجد

98-99

تمرین سری پنجم

مهلت ارسال: 13 دی

پس از پایان مهلت ارسال، تا ۲ روز به ازای هر روز تاخیر ۱۰ درصد از نمره تمرین کسر خواهد شد و پس از ۲ روز نمره ای به تمرین تعلق نخواهد گرفت.

سوال اول)

در این سوال می خواهیم متدی generic تعریف کنیم. این متد دارای یک پارامتر ورودی از نوع T، و خروجی از نوع **int** است. در بدنه این متد، باید برحسب نوع ورودی مقداری را برگردانید:

- ✚ اگر ورودی از نوع **int** بود، خود عدد را به عنوان خروجی برگردانید.
- ✚ اگر ورودی از نوع **double** بود، جذر عدد را به عنوان خروجی برگردانید.
- ✚ اگر ورودی از نوع **long** بود، قدرمطلق عدد را به عنوان خروجی برگردانید.
- ✚ اگر ورودی از نوع **string** بود، مجموع **ascii codes** را به عنوان خروجی برگردانید.
- ✚ و در نهایت اگر ورودی هیچ یک از انواع بالا نبود، **null** را به عنوان خروجی برگردانید.

سوال دوم)

در این سوال می خواهیم با کلاس generic بیشتر آشنا شویم. در این سوال شما باید یک **interface** و چهار **class** تعریف کنید.

Interface:

IMove را به گونه ای تعریف کنید که شامل دو **property**: نام از نوع **string** و سرعت از نوع **double** و یک متد به نام **Move**، بدون پارامتر ورودی و خروجی از نوع **string** باشد.

Class:

Car: در این کلاس رابط **IMove** را پیاده سازی کنید. متد **Move** رشته زیر را برمیگرداند:

Name **moves with speed of** Speed

هم چنین سازنده این کلاس دارای دو پارامتر ورودی یکی از نوع **string** و دیگری از نوع **double** است.

Ship: در این کلاس رابط **IMove** را پیاده سازی کنید. متد **Move** رشته زیر را برمیگرداند:

Name **sails with speed of** Speed

هم چنین سازنده این کلاس دارای دو پارامتر ورودی یکی از نوع **string** و دیگری از نوع **double** است.

Airplane: در این کلاس رابط **IMove** را پیاده سازی کنید. متد **Move** رشته زیر را برمیگرداند:

Name **flies with speed of** Speed

هم چنین سازنده این کلاس دارای دو پارامتر ورودی یکی از نوع **string** و دیگری از نوع **double** است.

Transportation: به صورت **generic** تعریف شود. این کلاس برای تایپ هایی که رابط **IMove** را پیاده سازی کرده باشند تعریف می شود پس **constraint** را به شکل مناسب قرار دهید.

در این کلاس یک **property** از نوع **T** وجود دارد. سازنده ای با یک پارامتر از نوع **T** نیز تعریف کنید. و در نهایت یک متد تعریف کنید که به عنوان خروجی، همان خروجی متد **Move** را برمیگرداند.

پس از پیاده سازی کلاس ها، در تابع **Main** برنامه خود 3 شی از نوع کلاس **Transportation<Car>**، **Transportation<Ship>** و **Transportation<Airplane>** تعریف کنید.

```
new Car("BMW", 150)
```

```
new Ship("Titanic", 100)
```

```
new Airplane("Airbus", 500)
```

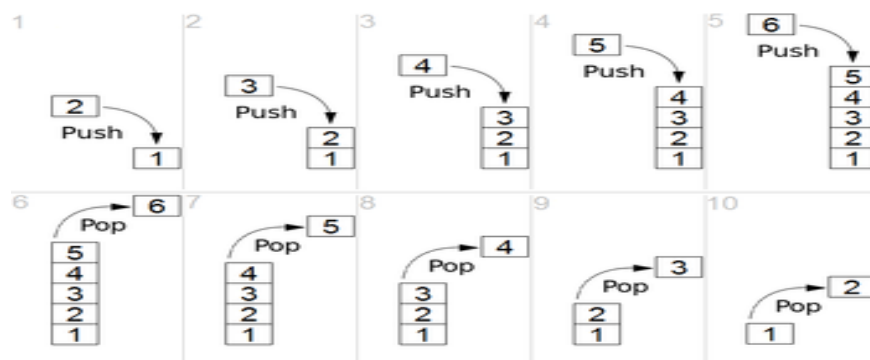
(3 مورد بالا، ورودی سازنده هستند.)

خروجی متد موجود در کلاس **Transportation** را برای هر یک از سه شی ساخته شده چاپ کنید.

سوال سوم)

آشنایی با Stack (پشته) :

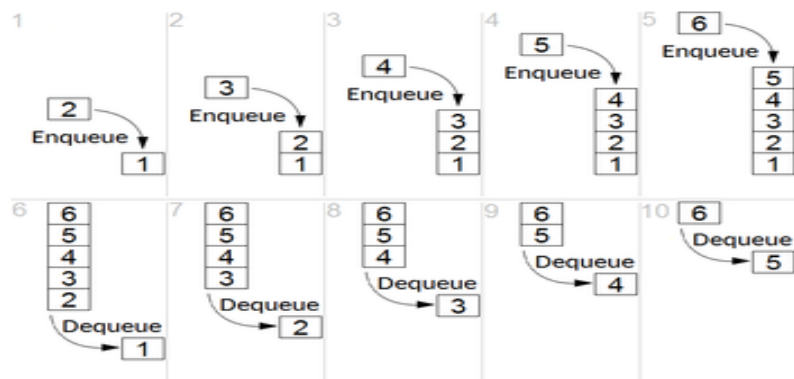
پشته مجموعه ای از عناصر است که عملیات اضافه کردن و حذف کردن عنصر از یک طرف آن انجام میشود به عبارت دیگر به صورت Last in First out (LIFO) عمل میکند. یعنی آخرین عنصر وارد شده ، اولین عنصری است که خارج میشود. در مثال زیر با مفهوم پشته بیشتر آشنا میشوید:



عمل push به معنای وارد کردن چیزی به پشته و عمل pop به معنای حذف کردن آن طبق تعریف گفته شده میباشد. در این مثال در ابتدا عدد 1 درون پشته قرار داشته و سپس عدد 2 در آن push میشود. به همین ترتیب شاهد push کردن تا عدد 6 هستیم. به نحوه قرار گیری اعداد در پشته دقت کنید. حال با عمل pop اولین عنصر در روی پشته که عدد 6 میباشد برداشته میشود و به همین ترتیب برای سایر اعداد روی پشته اتفاق میوفتد. در مثال واقعی تر میتوان به مجموعه ای از کتاب ها که روی هم قرار دارند، اشاره کرد.

آشنایی با Queue (صف) :

صف مجموعه ای از عناصر است که عناصر از یک طرف به نام انتهای صف (Rear) به آن اضافه و از یک طرف به نام ابتدای صف (Front) از آن حذف میشوند. به عبارت دیگر صف به صورت First in First out (FIFO) عمل میکند. در مثال واقعی مانند یک صف نانوايي میتوان اشاره کرد بدین صورت که اولین عنصر وارد شده اولین عنصری است که خارج میشود و عنصر جدیدی که وارد میشود به انتهای صف میرود. در مثال زیر بهتر متوجه خواهید شد:



در واقع Enqueue همان حکم push و Dequeue همان pop است.

عملیات Enqueue به منظور اضافه کردن عنصر جدید تا عدد 6 ادامه دارد. دقت کنید در عمل Dequeue عدد شماره 1 است که خارج میشود بر خلاف اینکه در پشته عدد 6 خارج شده بود.

حال در این برنامه میخواهیم نمرات دانشجویان یک کلاس ده نفره را یکبار با استفاده از پشته و بار دیگر با استفاده از صف پیاده سازی کنید. در ابتدا تمامی نمرات به پشته/صف اضافه میشوند. سپس عملیات برداشتن مقادیر از پشته/صف تا زمانی ادامه پیدا میکند که به اولین نمره 20 کلاس بر بخوریم. همچنین باید بررسی شود که در پشته/صف جدید ایجاد شده نمره 10 داریم یا خیر! (به صورت True یا False). اعداد باقی مانده در پشته/صف را در یک آرایه ذخیره و چاپ کنید. دقت کنید نمرات نباید از 20 بیشتر باشد و در صورت بیشتر بودن آنها را مدیریت کنید!

توجه: متدهای مهم Stack به صورت Push ، Pop ، Peek ، ToArray و Contains هستند. متدهای مهم Queue به صورت Clear ، Peek ، Dequeue ، EnQueue و ToArray و Contains میباشد. این موارد را مطالعه کنید 😊

هر دو برنامه شما باید در یک فایل زیپ قرار بگیرند.

سوال چهارم)

در این سوال میخواهیم با استفاده از پشته (Stack) برنامه ای بنویسید که یک عدد دهمی را به مبنای دو تبدیل کند.

سوال پنجم)

در ابتدا یک فایل با نام MyNote در مسیر برنامه خود بسازید و در آن یک متن دلخواه ذخیره کنید. حال به کمک برنامه ای با استفاده از Stack ، با دریافت نام فایل محتویات فایل را از انتها به ابتدا به صورت کاراکتر به کاراکتر بخواند و آن را در خروجی نمایش دهد:

Hi I Live In Tehran

➔ narheT nI eviL I iH

سوال ششم)

در این تمرین قصد داریم تا کلاس Student را با رابط [Comparable](#) پیاده سازی کنید.

هر Student شامل یک شماره دانشجویی به صورت int، نام به صورت string و معدل به صورت double است. با استفاده از متدی با نام [CompareTo](#) که یک متد انتزاعی از رابط Comparable است معدل دو دانشجو مقایسه میشود که معیاری برای کوچکتز، بزرگتر یا مساوی بودن دو Student است.

در ابتدا لازم است تعدادی دانشجو را به عنوان ورودی در نظر گرفته و آنرا در یک جنریک List قرار دهید. همچنین لازم است بررسی شود دانشجوی مورد نظر با معدل مورد نظر شما در این لیست وجود دارد یا خیر! در پایان خروجی شما به صورت زیر خواهد بود:

```
123 Mohsen 13/5
231 Ali 18/5
312 Zahra 16
132 Javad 11

Sorted List :
132 Javad 11
123 Mohsen 13/5
312 Zahra 16
231 Ali 18/5
```

حال بعد از sort کردن و نمایش آن یک شخص را به عنوان ورودی میدهیم تا ببینیم در لیست موجود است یا خیر:

222 Aryan 15.5

→ Aryan with Id 222 and average 15.5 does not exist!

متدهای مهم کلاس List به صورت Add ، Clear ، Insert ، Remove ، Sort ، Reverse ، Contains ، BinarySearch ، ToArray میباشد. در صورت نیاز میتوانید از متدهای ذکر شده استفاده کنید.

راهنمایی : برای چاپ اطلاعات به صورت آنچه در فوق میبینید از متد زیر استفاده کنید:

```
Public static void Display ( ... ) { ... }
```