



بسم الله الرحمن الرحيم

برنامه سازی پیشرفته C#

دکتر ملکی مجد

تمرین شماره سوم

مهلت ارسال: 20 آبان

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.

اهداف این تمرین یادگیری موارد زیر می باشد:

- Array
- Method with array parameter
- Using params in method parameters
- Method overloading
- File
- Enum

1) ابتدا در **main**، فایل هایی که در پیوست آمده است

(**"numbers1.txt"** , **"numbers2.txt"** , **"numbers3.txt"**)

را بخوانید. در هر یک از این فایل ها، در خط اول تعداد اعداد (**n**) و در خط بعد دنباله ای از اعداد کمتر از 500 (به تعداد **n**) به شما داده شده است. این اعداد توسط کاما از یکدیگر جدا شده اند. (تضمین می شود اعداد تکراری نیستند) این دنباله اعداد را به کمک **Split**، در آرایه یا لیستی از **int** ذخیره کنید و سپس خروجی متد **NumbersOfFibonacci** را در فایل های

result1.txt , **result2.txt**, **result3.txt**

ذخیره کنید. (این فایل توسط کد باید ساخته شود)(خروجی مربوط به **numbers1** در **result1** ذخیره میشود و ...)

متد **NumbersOfFibonacci**:

ورودی این متد **int[]** و خروجی آن از نوع **int** است. در بدنه این متد ابتدا باید لیستی از اعداد فیبوناچی کمتر از 500 (با شروع از عدد 1) بسازید و سپس چک کنید که چه تعداد از اعداد ورودی، عدد فیبوناچی هستند. این متد باید تعداد اعداد فیبوناچی را به عنوان خروجی برگرداند.

توجه: تمامی فایل های **numbers** و **result** در فولدر اصلی برنامه قرار گیرند. برای آدرس دهی در **StreamWriter** و **StreamReader** می توانید از آدرس دهی نسبی کمک بگیرید. برای اطلاعات بیشتر می توانید **relative path** را در گوگل سرچ کنید.
(درستی کد شما هنگام مراجعه به تی ای توسط فایل های دیگر امتحان می شود.)

2) کلاسی با نام **Car** تشکیل دهید و در آن دو **Property**: مدل ماشین از نوع **string**، و سال ساخت آن از نوع **int** وجود دارد و سطح دسترسی هر دو را **public** قرار دهید. برای این کلاس سازنده ای با دو پارامتر مدل ماشین و سال ساخت تشکیل دهید.

سپس **متدی** تشکیل دهید که در ورودی آن **تعداد نامشخص variable** از نوع **Car** آمده و خروجی این متد از نوع **string** است. در خروجی شما باید مدل ماشینی که جدیدترین سال ساخت را دارد برگردانید. (در پارامتر ورودی امکان استفاده از **List<Car>** نیست)

در خط اول ورودی، تعداد ماشین ها (**n**) آمده و سپس در **n** خط بعدی، در هر خط مدل ماشین به همراه سال ساخت که با کاما از یکدیگر جدا شده اند، آمده است. (سال ساخت ماشین ها دو به دو متفاوت هستند) شما باید متد بالا را برای این ماشین ها صدا زده و نتیجه را در خروجی چاپ کنید.

نمونه ورودی:

3

Hyundai,2017

Saipa,2004

BMW,2015

خروجی:

Hyundai

نمونه ورودی:

4

Peugeot,1998

Maserati,2013

Toyota,2010

Ferrari,2018

خروجی:

Ferrari

3) یک فایل متنی با عنوان **Test.txt** در محل برنامه خود ذخیره کنید.

در این برنامه می‌خواهیم در ابتدا مسیر فایل را به عنوان ورودی داده (اگر در آن مسیر فایل موجود نبود می‌بایست توسط اکسپشن **FileNotFoundException** هندل شود) و سپس متنی را به عنوان ورودی در آن فایل ذخیره کنیم

حال با خواندن فایل متنی می‌بایست **تعداد حروف صدا دار، تعداد ارقام و تعداد فضاها**ی خالی موجود در آن فایل را شمارش کرده و در خروجی نمایش دهد.

حروف صدا دار شامل **حروف a و O و u و i و e** به هر دو صورت کوچک و بزرگ هستند.

به عنوان مثال فایل **Test.txt** در مسیر "D:\" ایجاد می‌شود

سپس در کنسول یک متن در آن ذخیره می‌کنیم مثلاً: **Hello World 2020 !!!**

پس از خواندن متن خروجی ما به صورت زیر خواهد بود:

Horoofsedadar: 3

Space: 3

Numbers: 4

(درستی کد شما هنگام مراجعه به تی‌ای توسط فایل‌های دیگر امتحان می‌شود.)

4) یک سامانه ای برای ذخیره اطلاعات بلیط مسافر باید آماده سازی کنید. این سامانه در ابتدا یک منویی برای مسافر نشان می دهد که بتواند نوع سفر خود را انتخاب کند. (یکطرفه و دوطرفه)
منو شما مثلا میتواند به صورت روبه رو باشد:

1)Two Way

2)One Way

اطلاعات ورودی کاربر فقط شامل کدملی، تاریخ رفت و تاریخ برگشت بسته به انتخاب نوع سفر است!
کاربر در انتها اطلاعات ورودی خود را به صورت یک فایل تحت عنوان **MyInfo.txt** دریافت میکند.
کلاسی با نام **Passenger** دارای **property** های زیر میباشد:

- کدملی به صورت **int** (نه به صورت **string**!)
- شماره صندلی به صورت **int** // تعداد صندلی ها به اندازه 50 نفر است
- تاریخ رفت به صورت **string** // فرمت ورودی آن به صورت مثلا 980202
- تاریخ برگشت به صورت **string** // فرمت آن همانند فرمت تاریخ رفت است.

متدهای این کلاس به شرح زیر است:

- **CheckTypeOfPassenger**:

این متد کدملی را میگیرد. اگر رقم اول از چپ کدملی عددی فرد بود مسافر از نوع Normal میباشد لذا شماره صندلی آن عددی Random از 21 تا 50 انتخاب میشود در غیراینصورت مسافر VIP محسوب شده و میتواند شماره صندلی خود را به دلخواه انتخاب کند.

- **NumToDate**: (میتوانید به روش دلخواه خود پیاده سازی کنید)
توسط این متد فرمت هر تاریخی (رفت یا برگشت) تغییر میکند مثلاً اگر تاریخی به صورت 980202 وارد شده بود به صورت 98/02/02 تغییر می یابد.

- **TicketInfo**: (overloading)
توسط این متد میبایست اطلاعاتی نظیر نوع مسافر (Normal یا VIP) و شماره صندلی و فرمت درست تاریخ (بسته به یکطرفه یا دوطرفه بودن سفر) در فایل مذکور (MyInfo.txt) ذخیره شود.

مواردی که باید بدانیم و رعایت کنیم:

- فرض میکنیم تعداد ظرفیت هر وسیله حمل و نقل 50 نفر میباشد لذا شماره صندلی ها نیز از 1 تا 50 شماره گذاری شده اند.
- شماره صندلی میتواند تکراری باشد.
- تاریخ برگشت نباید از تاریخ رفت کوچکتر و یا مساوی باشد.
- مسافران به دو دسته Normal و VIP تقسیم می شوند. (Enum)
- شماره صندلی 1 تا 20 برای مسافران VIP و از 21 تا 50 برای مسافران Normal است.
- در تعداد ارقام ورودی تاریخ رفت و برگشت دقت شود. (یک عدد 6 رقمی است).

نمونه های ذخیره شده در فایل:

File Edit Format View Help

Type : VIP

Departure Date : 98/05/15

Return Date : 98/05/20

Seat Number : 11

Type : Normal

Departure Date : 98/07/11

Return Date : 98/07/18

Seat Number : 21

File Edit Format View Help

Type : Normal

Date : 98/06/16

Seat Number : 35

File Edit Format View Help

Type : VIP

Date : 98/06/14

Seat Number : 17

(درستی کد شما هنگام مراجعه به تی ای توسط فایل های دیگر امتحان می شود.)

لطفا قبل از مراجعه با تی ای خود هماهنگی لازم جهت بررسی کد انجام شود
مهلت مراجعه برای تمرین سوم تا پایان روز شنبه 25 آبان می باشد.