



برنامه‌سازی پیشرفته
تمرین‌های سری چهارم

مدرس: سید صالح اعتمادی
طرح تمرین: امید میرزاجانی

مهلت ارسال:
دوشنبه ۲۶ اسفند ۹۸

فهرست مطالب

۲	۱	مقدمه
۲	۱.۱	موارد مورد توجه
۲	۲	آماده‌سازی‌های اولیه
۲	۱.۲	ساخت پروژه ی C#
۲	۲.۲	قواعد نام‌گذاری
۳	۳	لپ‌تاپ مناسب! C# Java
۳	۱.۳	پیاده‌سازی C#
۳	۲.۳	پیاده‌سازی Java
۴	۴	جابجایی لپ‌تاپ!
۴	۱.۴	کلاس Graphic C#
۴	۲.۴	کلاس Memory C#
۴	۳.۴	کلاس Cpu C#
۴	۴.۴	جابجایی C#
۵	۵	ساختار حافظه!
۵	۱.۵	آشنایی با Stack C#
۵	۲.۵	آشنایی با Heap C#

۱ مقدمه

۱.۱ موارد مورد توجه

- توجه داشته باشید که برای کسب نمره‌ی قبولی درس کسب حداقل نصف نمره‌ی هر سری تمرین الزامی می‌باشد.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام‌شده است. توصیه می‌شود نوشتن تمرین را به روزهای نهایی موکول نکنید.
- همکاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ ارسالی از سوی شماست؛ بنابراین ارسال پاسخ در ریپازیتوری گیت شما به این معناست که پاسخ آن تمرین، توسط شما نوشته شده است. در صورت تقلب یا اثبات عدم نوشتار پاسخ حتی یک سوال از تمرین، برای هر دو طرف تقلب‌گیرنده و تقلب‌دهنده نمره‌ی مردود برای درس در نظر گرفته خواهد شد.
- توجه داشته باشید که پاسخ‌ها و کدهای مربوط به هر مرحله را بایستی تا قبل از پایان زمان مربوط به آن مرحله، در سایت [Azure DevOps](#) (طبق توضیحات کارگاه‌ها و کلاس‌ها) بفرستید. درست کردن `Pull request` و `Complete` کردن `Pull request` و انتقال به شاخه‌ی `master` پس از تکمیل تمرین فراموش نشود!
- پس از پایان مهلت ارسال تا ۲ روز به ازای هر روز تاخیر ۱۰ درصد از نمره مربوط به تمرین کسر خواهد شد و پس از ۲ روز نمره‌ای به تمرین تعلق نخواهد گرفت.
- بعضی از قسمت های تمرین نیاز به پیاده سازی بر روی هر چهار زبان `C#` ، `Python` ، `C++` و `Java` را دارند بعضی هم خیر. بنابراین روبروی هر سوال زبان های مورد نیاز برای پیاده سازی مشخص شده است.

۲ آماده سازی های اولیه

۱.۲ ساخت پروژه ی C#

برای ایجاد پروژه C# کافی است کد زیر را در ترمینال خود اجرا کنید:

```
۱ mkdir A4_cs
۲ cd A4_cs
۳ dotnet new sln
۴ mkdir A4_cs
۵ cd A4_cs
۶ dotnet new console
۷ cd ..
۸ dotnet sln add A4_cs\A4_cs.csproj
۹ mkdir A4_cs.Tests
۱۰ cd A4_cs.Tests
۱۱ dotnet new mstest
۱۲ dotnet add reference ..\A4_cs\A4_cs.csproj
۱۳ cd ..
۱۴ dotnet sln add A4_cs.Tests\A4_cs.Tests.csproj
```

۲.۲ قواعد نام گذاری

قواعد نام‌گذاری تمرین را از جدول ۱ مطالعه کنید.

جدول ۱: قراردادهای نام‌گذاری تمرین

Naming conventions		
Branch	Directory	Pull Request
fb_A4	A4	A4

* در کل یک دیرکتوری داخل Assignments به نام A۴ بسازید و داخل آن، دو دیرکتوری به نام های A۴_cs و A۴_java داشته باشید و فایل های مربوط به هر یک را داخل دیرکتوری مربوطه بگذارید.

۳ لپ تاپ مناسب! C# Java

دیمو^۱ به تازگی به کامپیوتر علاقه پیدا کرده و قصد دارد خود را برای مسابقات برنامه نویسی آماده کند. برای دیمو فقط ویژگی های حافظه، پردازنده و گرافیک از یک لپتاپ مهم است. او حافظه (Memory) یک لپ تاپ را مهم ترین قسمت آن میداند. حال برای شروع کار، میخواهد او را در خرید لپ تاپ کمک کنید و متد `ChooseBest` را پیاده سازی کنید.

۱.۳ پیاده سازی C#

یک `enum` به نام `Config` تعریف کنید که شامل مقادیر `Graphic`، `Memory` و `Cpu` است. میتوانید از قطعه کد زیر به استفاده کنید:

```
1 public enum Config
2 {
3     Graphic=/*TODO*/,
4     Memory =/*TODO*/,
5     Cpu =/*TODO*/
6 }
```

متد `ChooseBest` را در کلاس `Program` قرار دهید که ورودی آن یک `Config` باشد و خروجی مناسب داشته باشد. همانطور که گفتیم حافظه برای دیمو از هر چیزی مهم تر است؛

- اگر این لپتاپ دارای هر سه ویژگی گرافیک، حافظه و پردازنده بود، `Excellent` را خروجی دهد.
- اگر حافظه را به همراه فقط یکی از پردازنده و گرافیک داشت، `Very Good` را خروجی دهد.
- اگر فقط حافظه داشت، `Good` را خروجی دهد.
- اگر فقط گرافیک داشت، `Not Bad` را خروجی دهد.
- اگر فقط پردازنده داشت، `Not Bad` را خروجی دهد.

پس از پیاده سازی صحیح، تست `ChooseBest_Tests` پاس خواهد شد.

۲.۳ پیاده سازی Java

مانند پیاده سازی سی شارپ یک `enum` به نام `Config` با همان ویژگی های مذکور به همراه سازنده اش تعریف کنید که آن را در متغیر `ConfigValue` ذخیره کند. برای مفهوم شدن سوال، میتوانید از قطعه کد زیر استفاده کنید:

```
1 public enum Config
2 {
3     Graphic(/*TODO*/),
4     Ram(/*TODO*/),
5     Cpu(/*TODO*/);
6
7     public int ConfigValue;
8     public int Value;
9
10    private Config(int configValue)
11    {
12        this.ConfigValue = configValue;
13    }
14
15    public void ChangeValue(int v)
16    {
17        // TODO
18    }
19 }
```

^۱Dimo

سپس متد `ChangeValue` را بگونه ای پیاده سازی کنید که مقدار `Value` را با ورودی برابر قرار دهد پس از پیاده سازی صحیح سازنده و این متد، تست `Config_Test` پاس خواهد شد.

حال متد `ChooseBest` را در کلاس `App` بگونه ای پیاده سازی کنید که ورودی آن یک `Config` باشد و خروجی مناسب داشته باشد. پس از پیاده سازی صحیح، تست `ChooseBest_Test` پاس خواهد شد.

۴ جابجایی لپتاپ!

دیمو که از خرید لپتاپ جدیدش خیلی راضی نشده، تصمیم گرفته بعضی از اجزای لپتاپ جدیدش را تعویض کند و از اجزای لپتاپ قبلی خود استفاده کند.

۱.۴ کلاس `Graphic` C#

هر کارت گرافیک دارای ویژگی های

- `Size` از نوع `int`
- `Coprocessor` از نوع `string`
- `Type` از نوع `int`

این کلاس را به همراه سازنده (Constructor) اش به گونه ای پیاده سازی کنید که همه `Property` ها، مقدار `null` را نیز بپذیرد. پس از پیاده سازی صحیح این کلاس، تست `GraphicConstructor_Tests` پاس خواهد شد.

۲.۴ کلاس `Memory` C#

هر کارت حافظه دارای ویژگی های

- `Capacity` از نوع `int`
- `Pins` از نوع `int`
- `Type` از نوع `int`

این کلاس را به همراه سازنده (Constructor) اش به گونه ای پیاده سازی کنید که همه `Property` ها، مقدار `null` را نیز بپذیرد. پس از پیاده سازی صحیح این کلاس، تست `MemoryConstructor_Tests` پاس خواهد شد.

۳.۴ کلاس `Cpu` C#

هر پردازنده دارای ویژگی های

- `Model` از نوع `string`
- `Weight` از نوع `double`
- `Speed` از نوع `string`

این کلاس را به همراه سازنده (Constructor) اش به گونه ای پیاده سازی کنید که همه `Property` ها، مقدار `null` را نیز بپذیرد. پس از پیاده سازی صحیح این کلاس، تست `CpuConstructor_Tests` پاس خواهد شد.

۴.۴ جابجایی C#

حال که کلاس های `Graphic`، `Memory` و `Cpu` را به درستی پیاده سازی کردید، به دیمو کمک کنید که اجزای لپتاپ جدیدش را با قبلی عوض کند.

متد `SwapConfigs` را در کلاس `Program` به گونه ای پیاده سازی کنید که دو `object` بگیرد و این دو را با یکدیگر عوض کند برای مفهوم تر شدن سوال به تست ها مراجعه کنید. پس از پیاده سازی صحیح این متد، تست `SwapConfigs_Tests` پاس خواهد شد.

۵ ساختار حافظه!

دیمو به خاطر علاقه زیادش به حافظه، تصمیم گرفته به عنوان اولین تمرین برنامه نویسی خود، با مفاهیم آن کار کند. او می خواهد بداند که داده های ارجاعی (Reference Type) و مقداری (Value Type) چگونه حافظه را اشغال میکنند.

۱.۵ آشنایی با Stack C#

نوع داده ای `Struct_Size5` را به گونه ای پیاده سازی کنید که اندازه متغیر این نوع داده ای، ۵ باشد. سپس به طور مشابه نوع های داده ای `Struct_Size10`، `Struct_Size12` و `Struct_Size105` را نیز پیاده سازی کنید. پس از پیاده سازی صحیح، تست `StructSize_Tests` پاس خواهد شد.

۲.۵ آشنایی با Heap C#

کلاس `MemoryHeap` را به همراه متدهایش پیاده سازی کنید. متد `Allocate` یک عدد میگیرد و به اندازه آن (با واحد byte) حافظه را اشغال میکند. متد `DeAllocate` که با فراخوانی آن حافظه که اشغال شده را خالی میکند. برای مفهوم شدن سوال به تست ها مراجعه کنید. پس از پیاده سازی صحیح، تست `HeapSize_Tests` پاس خواهد شد.

موفق و سلامت باشید.