

تمرین دوازدهم

با سلام خدمت دانشجویان عزیز.

هدف از این تمرین آشنایی شما با الگوهای برنامه سازی شیء گرا است. بطور خاص الگوی وضعیت یا State Pattern. علاوه بر این مفاهیم زیر نیز مرور میشود.

- Inheritance
- Abstract Class
- Virtual Method
- Interface
- Polymorphism
- Lambda Expressions
- Writing Testable Code

هدف دیگری که از این تمرین داریم این است که از کد پیچیده وحشت نکنید. هر کلاس را به تنهایی متوجه بشوید که چکار میکند، روابط بین کلاس ها را هم متوجه بشوید. ممکنه که نتوانید تصور کنید کل مجموعه تمام کلاس ها با هم چکار میکنند. ولی هر کلاس را اگر متوجه بشوید و ارتباطش با کلاس های دیگر را هم متوجه بشوید، میتوانید هر کلاس را درست پیاده سازی کنید. علاوه بر این به اهمیت و فایده یونیت تست پی ببرید که چقدر کمک میکند به دیباگ کردن و درست کردن کد.

قبل از هر چیز، برای انجام این پروژه Feature Branch جدید از روی master به نام Assignment12 بسازید. تا حالا دیگه باید یاد گرفته باشید ولی جهت احتیاط... دستورات زیر را باید در نسخه git کامپیوتر شخصی خود بزنید:

```
git checkout master
git pull
git checkout -b Assignment12
```

چنانچه هر گونه خطایی از این دستورات گرفتید، پیام خطا را با دقت مطالعه کنید و رفع اشکال کنید یا کمک بگیرید برای رفع اشکال.

یک پروژه جدید (Console App (.NET Framework) در ویژوال استودیو به نام Assignment12 درست کنید و تمام فایلها بجز ProgramTests.cs را به آن اضافه کنید. بعد یک پروژه تست درست کنید و فایل ProgramTests.cs را به آن اضافه کنید. اگر مراحل را درست انجام داده باشید هر دو پروژه شما باید بدون خطا کامپایل بشود و از دوازده تا تست یازده تاش خطا بدهد.

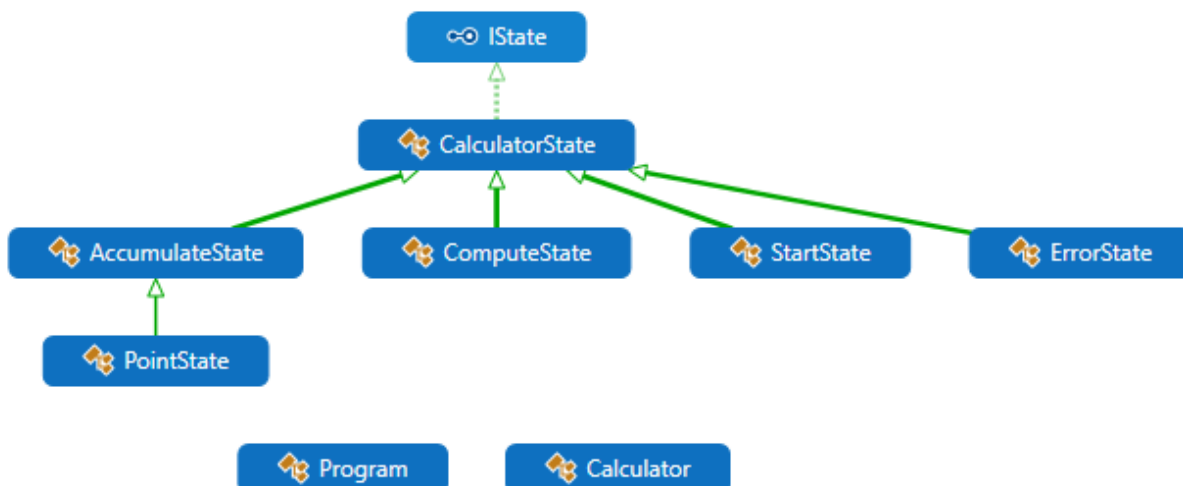
موضوع تمرین

موضوع این تمرین نوشتن یک ماشین حساب بسیار ساده است که چهار عملگر اصلی را انجام میدهد. برای نحوه کار ماشین حساب به ابتدای ویدیوی جلسه دوشنبه سی و یکم اردیبهشت (جلسه بیست و یکم) مراجعه کنید.

یه خورده این طرف و اونطرف کد را نگاه کنید. وحشت نکنید! این کد را من کامل زدم با یه تعدادی یونیت تست. بعد در ده جا پیاده سازی متد یا کلاس رو حذف کردم و گذاشتم به عهده شما که با علامت های #1، #2، ... علامت گذاری شده.

این ماشین حساب باید مثل یک ماشین حساب معمولی کار بکند. اگر مطمئن نیستید در یک حالتی باید چکار کند، یک ماشین حساب ساده بردارید و امتحان کنید.

نمودار روابط کلاس ها در پایین موجود است.



مثل تمرین قبل دقت، کنید که تعداد خط های کد که شما باید اضافه کنید کم است. لطفا با دقت و حوصله انجام دهید و اگر چیزی را متوجه نشدید حتما بپرسید. از دیباگ کردن برنامه برای فهم اینکه چطوری کار میکند استفاده کنید. لطفا در کامنت ها توضیح اضافه در مورد روابط کلاس ها و اینکه هر کلاسی چرا عضو کلاس دیگر است یا از آن ارث میرد یا ... بدهید. اگر فقط روی جاهایی که کد پاک شده تمرکز کنید کارتون سخت میشه. باید همه کلاس ها و روابط شون رو متوجه بشین تا بتونید جاهای خالی را پر کنید.

بعد از اینکه از درست کار کردن برنامه اطمینان پیدا کردید، عملگر توان را به ماشین حساب اضافه کنید. نشانه عملگر توان کاراکتر ^ است.

برای نمره کامل لازم است:

1. کد شما کامل و درست باشد.
2. بعد از تحویل تمرین UnitTest های جدید توسط حل تمرین ها به تمرین اضافه میشود و اگر آنها خطا بدهند از نمره تمرین کم میشود.
3. بیلد مربوط به Pull Request شما موفقیت آمیز بوده باشد.
4. تست ها همگی موفقیت آمیز باشند.
5. پوشش کد TestCoverage صد در صد باشد.
6. XML کامنت کامل باشد و کامنت ها نشانگر فهم شما از مفاهیم شیء گرا باشد.

اختیاری / مثبتی

پروژه بالا را بدون مفاهیم شیء گرا پیاده سازی کنید. فکر کنید پروژه درس مبانی است، ولی با C#.

این قسمت میتواند تا بیست درصد به نمره تمرین شما اضافه کند.

برای شما آرزوی موفقیت دارم!