



دانشکده‌ی مهندسی کامپیوتر

برنامه‌سازی پیشرفته (سی شارپ)
تمرین‌های سری هفتم (واسط‌ها)

علی حیدری، محمدمهدی عبدالله‌پور، مبینا کاشانیان
استاد: سید صالح اعتمادی

مهلت ارسال: ۱۳ اردیبهشت ۱۳۹۸

فهرست مطالب

۲	۱	مقدمه و آماده‌سازی
۲	۱.۱	نکات مورد توجه
۲	۲.۱	آماده‌سازی‌های اولیه
۲	۱.۲.۱	آماده‌سازی‌های مربوط به git
۳	۲.۲.۱	آماده‌سازی‌های مربوط به visual studio
۴	۲	پیاده‌سازی تمرین
۴	۱.۲	enum Degree
۴	۲.۲	ICitizen واسط
۴	۳.۲	ITeacher واسط
۴	۴.۲	کلاس Khalle
۴	۵.۲	کلاس Dabir
۴	۶.۲	کلاس Professor
۵	۷.۲	کلاس PoliceSite
۵	۸.۲	کلاس EduInstitute
۵	۳	ارسال تمرین
۵	۱.۳	مشاهده‌ی وضعیت اولیه‌ی فایل‌ها
۵	۲.۳	اضافه کردن فایل‌های تغییر یافته به stage
۶	۳.۳	commit کردن تغییرات انجام شده
۶	۴.۳	ارسال تغییرات انجام شده به Remote repository
۷	۵.۳	ساخت Pull Request
۷	۶.۳	ارسال Pull Request به بازبیننده

۱ مقدمه و آماده‌سازی

۱.۱ نکات 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 پاسخ‌های ارسال‌شده هر کس حتماً باید توسط خود او نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ‌های ارسال‌شده 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.
- برای طرح سوال و پرسش و پاسخ از صفحه درس در [Quera](#) استفاده کنید.

۲.۱ آماده‌سازی‌های اولیه

قواعد نام‌گذاری تمرین را از جدول ۱ مطالعه کنید.

جدول ۱: قراردادهای نام‌گذاری تمرین

Naming conventions					
Branch	Directory	Solution	Project	Test Project	Pull Request
fb_A7	A7	A7	A7	A7Tests	HW7

۱.۲.۱ آماده‌سازی‌های مربوط به git

اگر چه در کارگاه git مفاهیم و روش کار با آن آموزش داده شد اما بار دیگر در اینجا کارهایی را که باید در ابتدای تمرین انجام دهید را مرور می‌کنیم.

✓ ابتدا به شاخه‌ی `master` بروید.

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A6)
2 $ git checkout master
3 Switched to branch 'master'
4 Your branch is up to date with 'origin/master'.
```

✓ تغییرات انجام‌شده در Remote Repository را دریافت کنید.

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (master)
2 $ git pull
3 remote: Azure Repos
4 remote: Found 8 objects to send. (90 ms)
5 Unpacking objects: 100% (8/8), done.
6 From https://9752XXXX.visualstudio.com/AP97982/_git/AP97982
7 e7fd3b5..2cc74de master -> origin/master
8 Checking out files: 100% (266/266), done.
9 Updating e7fd3b5..2cc74de
10 Fast-forward
11 .gitattributes | 63 +
12 A7/A7.sln | 37 +
13 A7/A7/A7.csproj | 61 +
14 A7/A7/App.config | 6 +
15 A7/A7/Program.cs | 15 +
```

```

16 A7/A7/Properties/AssemblyInfo.cs | 36 +
17 .
18 .
19 .

```

✓ یک شاخه‌ی جدید با نام `fb_A7` بسازید و تغییر شاخه دهید.

```

1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (master)
2 $ git checkout -b fb_A7
3 Switched to a new branch 'fb_A7'
4 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
5 $

```

توصیه می‌شود پس از پیاده‌سازی هر کلاس تغییرات انجام شده را `commit` و `push` کنید.

۲.۲.۱ آماده‌سازی‌های مربوط به `visual studio`

ساختار فایل پایه‌ای که در اختیار شما قرار می‌گیرد به صورت زیر است:

```

1 A7
2 Project
3 Dabir.cs
4 Degree.cs
5 Eduinstitute.cs
6 ICitizen.cs
7 ITeacher.cs
8 Khalle.cs
9 PoliceStation.cs
10 Professor.cs
11 Program.cs
12 ProjectTests
13 ProgramTests.cs
14
15 2 directories, 11 files

```

در فایل پایه دو پوشه وجود دارد شما باید فایل‌های (های) موجود در پوشه‌ی `Project` را به پروژه‌ی اصلی (`A7`) و فایل‌های (های) موجود در پوشه‌ی `ProjectTests` را به پروژه‌ی تست (`A7Tests`) اضافه کنید.

(دقت کنید که در این تمرین تست‌ها در فایل پایه به صورت `comment` شده است و شما باید آن‌ها و چهار متد کمکی در انتهای فایل تست را از این حالت خارج کنید.)

۲ پیاده‌سازی تمرین

۱.۲ `enum Degree`

سطح تحصیلات هر شخص می‌تواند در ۴ وضعیت دیپلم، لیسانس، فوق لیسانس و دکترا باشد. نوع داده‌ی شمارشی با نام `Degree` پیاده‌سازی کنید که شامل: `PhD`، `Master`، `Bachelor`، `Diploma` باشد.

پس از پیاده‌سازی این نوع داده‌ی شمارشی تست `Q1DegreeEnumTest` پاس خواهد شد. ^{۱۰/۱}

۲.۲ واسط `ICitizen`

از آنجایی که تمام شهروندان دارای کدملی و نام هستند یک واسط تعریف می‌کنیم تا کلاس‌های دیگر با پیاده‌سازی آن این ویژگی‌ها را دارا باشند این کار باعث کم‌شدن کد تکراری خواهد شد. برای این واسط دو ویژگی `Name` و `NationalId` را به همراه `getter` و `setter` مناسب پیاده‌سازی کنید.

۳.۲ واسط `ITeacher`

تمام کسانی که تدریس می‌کنند دارای نام و مدرک تحصیلی هستند از آنجایی که موسسه‌ی آموزشی می‌خواهد برای تمام این افراد لینک آواتار آن‌ها را نگهداری کند شما باید این ویژگی را هم پیاده‌سازی کنید. دقت کنید که همان طور که گفته شد تمام کسانی که تدریس می‌کنند دارای مدرک

تحصیلی هستند اما ما اطلاعات آخرین مدرک تحصیلی آن‌ها را نگه‌داری می‌کنیم. علاوه بر ویژگی‌هایی که گفته شد بدیهی است که تمام کسانی که تدریس می‌کنند رفتار «درس دادن» هم دارند.

گام اول: برای این واسط سه ویژگی `Name`، `TopDegree` و `ImageUrl` را پیاده‌سازی کنید و برای آن‌ها `getter` و `setter` مناسب بنویسید.

گام دوم: برای این واسط یک متد با نام `Teach` بنویسید که مقدار بازگشتی آن یک رشته است.

۴.۲ کلاس Khalle

هر مربی مهد کودک علاوه بر آن که درس می‌دهد، یک شهروند هم هست.

گام اول: واسط‌های `ICitizen` و `ITeacher` را برای این کلاس پیاده‌سازی کنید تا این کلاس هم ویژگی‌های یک شهروند را دارا باشد و هم ویژگی‌های یک آموزش‌دهنده را. در پیاده‌سازی متد `Teach` یک رشته را به عنوان مقدار بازگشتی این متد برگردانید به طوری که حاوی نام کلاس در ابتدا سپس نام نمونه * با یک فاصله و در ادامه رشته‌ی `" is teaching"` باشد. مثلاً مقدار بازگشتی متد `Teach` برای یک نمونه با نام `Fateme` از این کلاس `"Khallem Fateme is teaching"` است.

پس از پیاده‌سازی این واسط تست `Q4KhallemTest1_ITeacher` و `Q4KhallemTest2_ICitizen` پاس خواهد شد. ^{۸/۳}

گام دوم: باید سازنده‌ی این کلاس را تکمیل کنید به طوری که بتوان با پاس دادن مقدار معتبر از تمام ویژگی‌هایی که برای این کلاس پیاده‌سازی کردید شی‌ای از نوع این کلاس ساخته شود که لزوماً دارای مقادیر پاس شده‌ی متناظر با هر ویژگی باشد.

پس از پیاده‌سازی این قسمت تست `Q4KhallemTest3_Constructor` پاس خواهد شد. ^{۷/۴}

۵.۲ کلاس Dabir

هر دبیر علاوه بر آن که درس می‌دهد، یک شهروند هم هست. علاوه بر آن موسسه‌ی آموزشی می‌خواهد تعداد رتبه‌های زیر ۱۰۰ی که شاگرد هر دبیر بوده‌اند را نگه‌داری کند.

گام اول: واسط‌های `ICitizen` و `ITeacher` را برای این کلاس پیاده‌سازی کنید تا این کلاس هم ویژگی‌های یک شهروند را دارا باشد و هم ویژگی‌های یک آموزش‌دهنده را. در پیاده‌سازی متد `Teach` یک رشته را به عنوان مقدار بازگشتی این متد برگردانید به طوری که حاوی نام کلاس در ابتدا سپس نام نمونه با یک فاصله و در ادامه رشته‌ی `" is teaching"` باشد.

پس از پیاده‌سازی این واسط تست `Q5DabirTest1_ITeacher` و `Q5DabirTest2_ICitizen` پاس خواهد شد. ^{۵/۶}

گام دوم: یک ویژگی اضافه با نام `Under100StudentCount` برای این کلاس بنویسید و برای آن `getter` و `setter` مناسب را پیاده‌سازی کنید.

گام سوم: باید سازنده‌ی این کلاس را تکمیل کنید به طوری که بتوان با پاس دادن مقدار معتبر از تمام ویژگی‌هایی که برای این کلاس پیاده‌سازی کردید شی‌ای از نوع این کلاس ساخته شود که لزوماً دارای مقادیر پاس شده‌ی متناظر با هر ویژگی باشد.

پس از پیاده‌سازی این قسمت تست `Q5DabirTest3_Constructor` پاس خواهد شد. ^{۴/۷}

۶.۲ کلاس Professor

هر استاد دانشگاه علاوه بر آن که درس می‌دهد، یک شهروند هم هست. علاوه بر آن موسسه‌ی آموزشی می‌خواهد تعداد مقاله‌های علمی هر استاد دانشگاه را نگه‌داری کند.

گام اول: پس از ایجاد یک کلاس با نام `Professor` واسط‌های `ICitizen` و `ITeacher` را برای آن پیاده‌سازی کنید تا این کلاس هم ویژگی‌های یک شهروند را دارا باشد و هم ویژگی‌های یک آموزش‌دهنده را. در پیاده‌سازی متد `Teach` یک رشته را به عنوان مقدار بازگشتی این متد برگردانید به طوری که حاوی نام کلاس در ابتدا سپس نام نمونه با یک فاصله و در ادامه رشته‌ی `" is teaching"` باشد.

پس از پیاده‌سازی این واسط تست `Q6ProfessorTest1_ITeacher` و `Q6ProfessorTest2_ICitizen` پاس خواهد شد. ^{۲/۹}

گام دوم: یک ویژگی اضافه با نام `ResearchCount` برای این کلاس بنویسید و برای آن `getter` و `setter` مناسب را پیاده‌سازی کنید.

گام سوم: باید سازنده‌ی این کلاس را تکمیل کنید به طوری که بتوان با پاس دادن مقدار معتبر از تمام ویژگی‌هایی که برای این کلاس پیاده‌سازی کردید شی‌ای از نوع این کلاس ساخته شود که لزوماً دارای مقادیر پاس شده‌ی متناظر با هر ویژگی باشد.

پس از پیاده‌سازی این قسمت تست `Q6ProfessorTest3_Constructor` پاس خواهد شد. ^{۱/۱۰}

*Instance

۷.۲ کلاس PoliceSite

سایت پلیس دارای لیستی از شهروندانی است که سابقه‌ی ارتکاب جرم دارند و این قابلیت را داراست که با گرفتن یک شی که دارای ویژگی‌ها و رفتارهای شهروند باشد وضعیت سوء پیشینه‌ی آن را اعلام کند.

گام اول: یک ویژگی با نام `BlackList` برای آن تعریف کنید به طوری که شهروندان مجرم در آن نگاه‌داری شود. سپس برای آن `getter` و `setter` مناسب بنویسید.

گام دوم: یک متد با نام `BackgroundCheck` برای این کلاس تعریف کنید به طوری که با پاس دادن یک شی که رابط `ICitizen` را پیاده‌سازی کرده باشد وضعیت سوء پیشینه‌ی آن را اعلام کند. (سوء پیشینه دارد یا ندارد)

۸.۲ کلاس EduInstitute

هر موسسه‌ی آموزشی بنا بر نوع آن (مهد کودک، مدرسه، دانشگاه) باید آموزش‌دهنده‌ی متناسب با خودش را دارا باشد. هم‌چنین هر موسسه‌ی آموزشی دارای سه ویژگی عنوان، کم‌ترین مدرک تحصیلی مورد پذیرش و لیستی از آموزش‌دهندگان است.

گام اول: ویژگی با نام `Title` برای این کلاس تعریف کنید به طوری نام موسسه در آن نگاه‌داری شود.

گام دوم: ویژگی با نام `MinimumDegree` برای این کلاس تعریف کنید به طوری که حداقل مدرک تحصیلی قابل پذیرش در آن نگاه‌داری شود. گام سوم: ویژگی با نام `Teachers` برای این کلاس تعریف کنید به طوری که لیستی از آموزش‌دهندگان در آن نگاه‌داری شود. دقت کنید که این کلاس باید با انواع داده‌ی عام[†] پیاده‌سازی شود تا اطلاعات مربوط به نوع هر آموزش‌دهنده‌ای (`Dabir`، `Khalle` و `Professor`) در آن بتوان نگاه‌داری کرد.

گام چهارم: برای ویژگی‌ها `getter` و `setter` مناسب بنویسید.

گام پنجم: باید سازنده‌ی این کلاس را تکمیل کنید به طوری که بتوان با پاس دادن مقدار معتبر از تمام ویژگی‌هایی که برای این کلاس پیاده‌سازی کردید شی‌ای از نوع این کلاس ساخته شود که لزوماً دارای مقادیر پاس شده‌ی متناظر با هر ویژگی باشد.

گام ششم: یک متد با نام `IsEligible` برای این کلاس تعریف کنید به طوری که با پاس دادن یک شی از نوع داده‌ی عام این کلاس صلاحیت آن را از لحاظ مدرک تحصیلی بررسی کند.

گام هفتم: یک متد با نام `Register` برای این کلاس تعریف کنید به طوری که با پاس دادن یک شی از نوع داده‌ی عام این کلاس در صورت دارا بود صلاحیت آن را اضافه کند و در صورت موفقیت‌آمیز بودن مقدار `true` و در صورتی که فاقد صلاحیت باشد اضافه نشود و مقدار `false` را برگرداند.

پس از پیاده‌سازی صحیح این کلاس و کلاس قبل تست `FinalTest` پاس خواهد شد. ^{۱۱/۰}

۳ ارسال تمرین

در اینجا یک‌بار دیگر ارسال تمرینات را با هم مرور می‌کنیم:

۱.۳ مشاهده‌ی وضعیت اولیه‌ی فایل‌ها

ابتدا وضعیت فعلی فایل‌ها را مشاهده کنید:

```

1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
2 $ git status
3 On branch fb_A7
4 Untracked files:
5   (use "git add <file>..." to include in what will be committed)
6
7   A7/
8
9 nothing added to commit but untracked files present (use "git add" to track)
```

همان‌طور که مشاهده می‌کنید فولدر `A7` و تمام فایل‌ها و فولدرهای درون آن در وضعیت `Untracked` قرار دارند و همان‌طور که در خط آخر خروجی توضیح داده شده برای `commit` کردن آن‌ها ابتدا باید آن‌ها را با دستور `git add` وارد `stage` کنیم.

۲.۳ اضافه کردن فایل‌های تغییر یافته به stage

حال باید فایل‌ها و فولدرهایی را که در `stage` قرار ندارند را وارد `stage` کنیم. برای این کار از دستور `git add` استفاده می‌کنیم.

[†]Generics

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
2 $ git add .
```

حال دوباره وضعیت فایل‌ها و فولدرها را مشاهده می‌کنیم:

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
2 On branch fb_A7
3 Changes to be committed:
4   (use "git reset HEAD <file>..." to unstage)
5
6   new file:   A7/A7.sln
7   new file:   A7/A7/A7.csproj
8   new file:   A7/A7/App.config
9   new file:   A7/A7/Program.cs
10  new file:   A7/A7/Properties/AssemblyInfo.cs
11  new file:   A7/A7Tests/A7Tests.csproj
12  new file:   A7/A7Tests/Properties/AssemblyInfo.cs
13  new file:   A7/A7Tests/packages.config
14  .
15  .
16  .
```

همانطور که مشاهده می‌کنید فولدر A7 و تمام فولدرها و فایل‌های درون آن (به جز فایل‌هایی که در gitignore معین کرده‌ایم) وارد stage شده‌اند.

۳.۳ commit کردن تغییرات انجام شده

درگام بعدی باید تغییرات انجام شده را commit کنیم. فراموش نکنید که فقط فایل‌هایی را می‌توان commit کرد که در stage قرار داشته باشند. با انتخاب یک پیام مناسب تغییرات صورت گرفته را commit می‌کنیم:

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
2 $ git commit -m "Implement HW7"
3 [fb_A7 c1f21df] Implement HW7
4 15 files changed, 595 insertions(+)
5 create mode 100644 A7/A7.sln
6 create mode 100644 A7/A7/A7.csproj
7 create mode 100644 A7/A7/App.config
8 create mode 100644 A7/A7/Program.cs
9 create mode 100644 A7/A7/Properties/AssemblyInfo.cs
10 create mode 100644 A7/A7Tests/A7Tests.csproj
11 create mode 100644 A7/A7Tests/Properties/AssemblyInfo.cs
12 create mode 100644 A7/A7Tests/packages.config
13 .
14 .
15 .
```

۴.۳ ارسال تغییرات انجام شده به Remote repository

گام بعدی ارسال تغییرات انجام شده به Remote Repository است.

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
2 $ git push origin fb_A7
3 Enumerating objects: 25, done.
4 Counting objects: 100% (25/25), done.
5 Delta compression using up to 8 threads
6 Compressing objects: 100% (22/22), done.
7 Writing objects: 100% (25/25), 9.56 KiB | 890.00 KiB/s, done.
8 Total 25 (delta 4), reused 0 (delta 0)
9 remote: Analyzing objects... (25/25) (5 ms)
10 remote: Storing packfile... done (197 ms)
```

```
11 remote: Storing index... done (84 ms)
12 To https://9752XXXX.visualstudio.com/AP97982/_git/AP97982
13 * [new branch]      fb_A7 -> fb_A7
```

۵.۳ ساخت Pull Request

با مراجعه به سایت [Azure DevOps](#) یک Pull Request جدید با نام `HW7` بسازید به طوری که امکان `merge` کردن شاخه‌ی `fb_A7` را بر روی شاخه‌ی `master` را بررسی کند. (این کار در صورتی انجام می‌شود که کد شما کامپایل شود و همچنین تست‌های آن پاس شوند) در نهایت با انتخاب گزینه‌ی `set auto complete` در صفحه‌ی Pull Request مربوطه تعیین کنید که در صورت وجود شرایط `merge` این کار انجام شود. دقت کنید که گزینه‌ی `Delete source branch` نباید انتخاب شود.

۶.۳ ارسال Pull Request به بازبیننده

در نهایت Pull Request ساخته شده را برای بازبینی، با بازبیننده‌ی خود به اشتراک بگذارید.