



دانشکده‌ی مهندسی کامپیوتر

برنامه‌سازی پیشرفته (سی‌شارپ)
آزمون عملی دوم
(مجموعه سوالات سری دوم)

علی حیدری
استاد: سید صالح اعتمادی

۹ اردیبهشت ۱۳۹۸

فهرست مطالب

۲	۱	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
۴	۱.۲	مقدمه و شرح سوال
۴	۲.۲	مهارت و تسلط به مفاهیم پایه برنامه‌نویسی
۴	۳.۲	تسلط به مفهوم Generics
۴	۴.۲	تسلط به مفهوم کلاس و واسط
۴	۳	ارسال
۴	۱.۳	مشاهده‌ی وضعیت اولیه‌ی فایل‌ها
۵	۲.۳	اضافه کردن فایل‌های تغییر یافته به stage
۵	۳.۳	commit کردن تغییرات انجام شده
۵	۴.۳	ارسال تغییرات انجام شده به Remote repository
۶	۵.۳	ساخت Pull Request
۶	۶.۳	ارسال Pull Request به بازیکننده

۱ آمادہ سازی

۱.۱ نکات مورد توجه

- در این آزمون شما می‌توانید به یکی از دو سری سوال پاسخ دهید در صورت پاسخ به مجموعه سوالات سری اول نمره‌ی شما از ۱۰۰ و در صورتی که به مجموعه سوالات سری دوم پاسخ دهید نمره‌ی شما از ۵۰ محاسبه خواهد شد.
- نمره‌ی مجموعه سوالات قابل جمع‌بندی نیست (در صورتی که هر دو مجموعه سوالات را حل کنید فقط یکی از آن‌ها بررسی می‌شود).
- استفاده از ویدیوهای آموزشی حین امتحان مجاز نیست.
- هرگونه استفاده از تلفن همراه حین امتحان مجاز نیست.
- می‌توانید از هرگونه متن کاغذی یا دیجیتالی (کتاب، جزوه، PDF) استفاده کنید.
- پاسخ هر سوال را در ریپازیتوری Git خودتان بارگذاری نمایید.
- توجه داشته باشید که برای کسب نمره‌ی قبولی درس کسب حداقل نصف نمره‌ی هر سری تمرین و امتحان الزامی می‌باشد.
- کپی کردن هرگونه کد از روی اینترنت یا غیراینترنت مجاز نیست. پاسخ ارسالی هر کس حتماً باید توسط خود او نوشته شده باشد. کمک گرفتن از دیگران در طول مدت امتحان مجاز نیست و منجر به درج نمره‌ی مردود برای این درس می‌شود.
- حین امتحان تنها اجازه ارتباط با استاد درس را دارید. هرگونه ارتباط با هر فرد دیگری در جلسه امتحان یا خارج از جلسه امتحان به صورت حضوری یا مجازی مجاز نمی‌باشد.
- در صورت نیاز به خروج از محل امتحان قبل از اتمام امتحان، امکان خروج بعد از هماهنگی با استاد به صورت یک نفر، یک نفر هست.
- صدا و صفحه نمایش شما باید از طریق نرم‌افزار [Flashback recorder](#) به طور کامل از ابتدا تا انتهای امتحان ضبط و ذخیره شود.

۲.۱ آمادہ سازی‌های اولیه

قواعد نام‌گذاری آزمون را از جدول ۱ مطالعه کنید.

جدول ۱: قراردادهای نام‌گذاری آزمون

Naming conventions					
Branch	Directory	Solution	Project	Test Project	Pull Request
fb_E1B	E1B	E1B	E1B	E1BTests	Exam1Basic

۱.۲.۱ آمادہ سازی‌های مربوط به git

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

✓ ابتدا به شاخه‌ی master بروید.

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_A7)
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 E1B/E1B.sln | 37 +
13 E1B/E1B/E1B.csproj | 61 +
14 E1B/E1B/App.config | 6 +
15 E1B/E1B/Program.cs | 15 +
16 E1B/E1B/Properties/AssemblyInfo.cs | 36 +
17 .
18 .
19 .

```

✓ یک شاخه‌ی جدید با نام fb_E1B بسازید و تغییر شاخه دهید.

```

1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (master)
2 $ git checkout -b fb_E1B
3 Switched to a new branch 'fb_E1B'
4 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_E1B)
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 E1B
2 +---Project
3 |   BasicQuestions.cs
4 |
5 \---ProjectTests
6     BasicQuestionsTests.cs

```

در فایل پایه دو پوشه وجود دارد شما باید فایل(های) موجود در پوشه‌ی Project را به پروژه‌ی اصلی (E1B) و فایل(های) موجود در پوشه‌ی ProjectTests را به پروژه‌ی تست (E1BTests) اضافه کنید.

۲ پیاده‌سازی

۱.۲ مقدمه و شرح سوال

این بخش از امتحان به منظور سنجش میزان تسلط به مفاهیم و مهارت برنامه‌نویسی دانشجویانی طراحی شده که این ترم تحصیلی را با میزان تسلط کم‌تری به برنامه‌نویسی شروع کرده‌اند و با جدیت تمرین‌ها و مطالب درس را پی‌گیری می‌کنند ولی هنوز تسلط و مهارت کافی را کسب نکرده‌اند. انتظار می‌رود این دانشجویان این بخش از سوال‌ها را بطور کامل و بدون نقص انجام دهند.

۲.۲ مهارت و تسلط به مفاهیم پایه برنامه‌نویسی

متد OddSum را به شکلی پیاده‌سازی کنید که تست OddSumTest پاس شود. وظیفه این متد جمع عناصری از آرایه ورودی است که مقدار آن‌ها فرد است. برای مثال برای آرایه‌ای با مقادیر ۲, ۳, ۵, ۱ مقدار بازگشتی باید عدد $9 = 3 + 5 + 1$ باشد.

۳.۲ تسلط به مفهوم Generics

گام اول: سه متد خالی به نام Swap در کلاس BasicQuestions موجود هستند. وظیفه این متدها جابه‌جایی مقدار دو پارامتر ورودی با هم است. به تفاوت بین این متدها دقت کنید. این متدها را به گونه‌ای پیاده‌سازی کنید که تست‌های SwaplongTest، SwapdoubleTest و SwapIntTest پاس شوند.

گام دوم: این سه متد را پاک کرده (بله پاک کنید بطور کامل) و با یک متد Generic جای‌گزین کنید به شکلی که تست‌ها عنوان شده در گام اول پاس شوند.

۴.۲ تسلط به مفهوم کلاس و واسط

گام اول: کلاس `Human` را به گونه ای پیاده سازی کنید که سازنده آن دو پارامتر `string name` و `int age` را به عنوان ورودی از سازنده دریافت کرده و در یک `field` با نام مناسب ذخیره کند. سپس یک `property` به نام `Name` متناظر با پارامتر ورودی `Name` تعریف و پیاده سازی کنید. بعد از پیاده سازی این کلاس تست `HumanTest` را از حالت کامنت در آورید. لازم است که پروژه اصلی و تست بدون خطا کامپایل/بیلد شوند و این تست پاس شود.

گام دوم: واسط (`interface`) به نام `IHasAge` را به گونه تعریف کنید که یک متد `GetAge()` بدون پارامتر ورودی و با مقدار برگشتی از نوع `int` داشته باشد. گام سوم تغییرات لازم در تعریف و پیاده سازی کلاس `Human` به گونه ای بدهید که واسط `IHasAge` را پیاده سازی کند. بعد از انجام تغییرات لازم، تست `HumanAgeTest` را از حالت کامنت خارج کنید. لازم است پروژه اصلی و تست بدون خطا کامپایل/بیلد شود و این تست پاس شود.

۳ ارسال

در اینجا یک بار دیگر ارسال آزمون را با هم مرور می کنیم:

۱.۳ مشاهده وضعیت اولیه ی فایل ها

ابتدا وضعیت فعلی فایل ها را مشاهده کنید:

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_E1B)
2 $ git status
3 On branch fb_E1B
4 Untracked files:
5   (use "git add <file>..." to include in what will be committed)
6
7   E1B/
8
9 nothing added to commit but untracked files present (use "git add" to track)
```

همان طور که مشاهده می کنید فولدر `E1B` و تمام فایل ها و فولدرهای درون آن در وضعیت `Untracked` قرار دارند و همان طور که در خط آخر خروجی توضیح داده شده برای `commit` کردن آن ها ابتدا باید آن ها را با دستور `git add` وارد `stage` کنیم.

۲.۳ اضافه کردن فایل های تغییر یافته به stage

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

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_E1B)
2 $ git add .
```

حال دوباره وضعیت فایل ها و فولدرها را مشاهده می کنیم:

```
1 Ali@DESKTOP-GS7PR56 MINGW64 /c/git/AP97982 (fb_E1B)
2 On branch fb_E1B
3 Changes to be committed:
4   (use "git reset HEAD <file>..." to unstage)
5
6   new file:   E1B/E1B.sln
7   new file:   E1B/E1B/E1B.csproj
8   new file:   E1B/E1B/App.config
9   new file:   E1B/E1B/Program.cs
10  new file:   E1B/E1B/Properties/AssemblyInfo.cs
11  new file:   E1B/E1BTests/E1BTests.csproj
12  new file:   E1B/E1BTests/Properties/AssemblyInfo.cs
13  new file:   E1B/E1BTests/packages.config
14  .
15  .
16  .
```

همانطور که مشاهده می‌کنید فولدر E1B 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

در گام 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_E1B)
2 $ git commit -m "Implement Exam1"
3 [fb_E1B c1f21df] Implement Exam1
4 15 files changed, 595 insertions(+)
5 create mode 100644 E1B/E1B.sln
6 create mode 100644 E1B/E1B/E1B.csproj
7 create mode 100644 E1B/E1B/App.config
8 create mode 100644 E1B/E1B/Program.cs
9 create mode 100644 E1B/E1B/Properties/AssemblyInfo.cs
10 create mode 100644 E1B/E1BTests/E1BTests.csproj
11 create mode 100644 E1B/E1BTests/Properties/AssemblyInfo.cs
12 create mode 100644 E1B/E1BTests/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_E1B)
2 $ git push origin fb_E1B
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_E1B -> fb_E1B
```

۵.۳ ساخت Pull Request

با مراجعه به سایت Azure DevOps یک Pull Request جدید با نام Exam1Basic بسازید به طوری که امکان merge کردن شاخه‌ی fb_E1B 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.