



مبانی برنامه سازی کامپیوتر
آزمون عملی پایتون

سید صالح اعتمادی *

۱۳ آذر ۱۳۹۸

*تشکر ویژه از آقای علی حیدری که نسخه اولیه این قالب را در ترم دوم سال تحصیلی ۹۷-۹۸ برای درس برنامه سازی پیشرفته تهیه کردند.

فهرست مطالب

۳	۱ آماده سازی
۳	۱.۱ نکات مورد توجه
۳	۲.۱ آماده سازی های اولیه
۳	۱.۲.۱ آماده سازی های مربوط به git
۴	۲.۲.۱ آماده سازی های مربوط به VSCode
۴	۲ پیاده سازی
۴	۱.۲ آماده سازی تست
۴	۱.۱.۲ فعال سازی تست ها
۴	۲.۱.۲ فرستادن تست در Azure DevOps
۴	۲.۲ پیاده سازی توابع
۵	۱.۲.۲ تست test_q0_distance
۵	۲.۲.۲ تست test_q1_poly1
۵	۳.۲.۲ تست test_q2_poly1_str
۵	۴.۲.۲ تست test_q3_digit_sum
۵	۵.۲.۲ تست test_q4_filter_dividable
۵	۶.۲.۲ تست test_q5_poly_n
۵	۷.۲.۲ تست test_q6_poly_n_str
۵	۸.۲.۲ تست test_q7_poly_n_str2
۵	۳ ارسال
۵	۱.۳ ساخت Pull Request

۱ آماده‌سازی

۱.۱ نکات مورد توجه

- صدا و صفحه نمایش شما باید از طریق نرم‌افزار **Flashback recorder** به طور کامل از ابتدا تا انتهای امتحان ضبط و ذخیره شود. دقت کنید که پس از نصب نرم‌افزار، در قسمت تنظیمات کیفیت ضبط را ۱ فریم بر ثانیه قرار دهید. ویدیوی امتحان بعد از امتحان جمع‌آوری خواهد شد.
- استفاده از هرگونه منبع کاغذی، مجازی، کتابی، نوشتاری، ... در امتحان مجاز نمی‌باشد.
- دیدن هرگونه کد از روی اینترنت یا غیراینترنت مجاز نیست. پاسخ ارسالی هر کس حتما باید توسط خود او و بدون دیدن هیچ کد دیگری نوشته شده باشد. حتی اگر کد دیگر را خود فرد قبلاً نوشته باشد. کمک گرفتن از دیگران در طول مدت امتحان مجاز نیست و منجر به درج نمره‌ی مردود برای این درس می‌شود.
- معیار ارزیابی امتحان فقط کدی است که در AzureDevOps با روشی که در ادامه آمده بارگزاری شده است.
- 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 و بدون بردن تلفن همراه و به صورت یک نفر، یک نفر هست.
- خوردن و آشامیدن در طول امتحان بدون برهم زدن نظم اشکال ندارد.

۲.۱ آماده‌سازی‌های اولیه

قواعد نام‌گذاری آزمون را از جدول ۱ مطالعه کنید.

جدول ۱: قراردادهای نام‌گذاری آزمون

Naming conventions			
Feature Branch	Directory	Pull Request Title	Target Branch
fb_Exam1	Exam1	Exam1	holymaster

۱.۲.۱ آماده‌سازی‌های مربوط به git

اگر سرکلاس و کارگاه چند بار مفاهیم و روش کار با git 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 و از یکسان بودن این شاخه با سرور اطمینان حاصل کنید.

```

1 C:\git\FC98991>git checkout master
2 Already on 'master'
3 Your branch is up to date with 'origin/master'.
4
5 C:\git\FC98991>git status
6 On branch master
7 Your branch is up to date with 'origin/master'.
8
9 nothing to commit, working tree clean
10
11 C:\git\FC98991>git pull
12 Already up to date.
13
14 C:\git\FC98991>
```

✓ سپس این کار را برای شاخه **holymaster** تکرار کنید.

```

1 C:\git\FC98991>git checkout holymaster
2 Switched to branch 'holymaster'
3 Your branch is up to date with 'origin/holymaster'.
```

```

4 C:\git\FC98991>git status
5 On branch holymaster
6 Your branch is up to date with 'origin/holymaster'.
7
8 nothing to commit, working tree clean
9
10 C:\git\FC98991>git pull
11 Already up to date.
12
13 C:\git\FC98991>
14

```

✓ یک شاخه‌ی جدید با نام `fb_Exam1` بسازید و تغییر شاخه دهید.

```

1 C:\git\FC98991>git branch fb_Exam1
2
3 C:\git\FC98991>git checkout fb_Exam1
4 Switched to branch 'fb_Exam1'
5
6 C:\git\FC98991>git status
7 On branch fb_Exam1
8 nothing to commit, working tree clean
9
10 C:\git\FC98991>

```

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

۲.۲.۱ آماده‌سازی‌های مربوط به VSCode

پوشه‌ای با نام `Exam1` درست کرده و فایل تست `exam_test.py` را در آن قرار دهید. سپس پوشه `Exam1` را با VSCode باز کنید.

۲ پیاده‌سازی

۱.۲ آماده سازی تست

سوال‌های امتحان بصورت تعدادی تست طراحی شده‌اند که لازم است تابع برای پاس شدن تست را پیاده‌سازی کنید. همه تست‌ها `comment` شده و `pytest` برای رد کردن و عدم اجرای تست تنظیم شده.

۱.۱.۲ فعال‌سازی تست‌ها

در قدم اول کامنت‌های مربوط به تست‌ها را یکی-یکی برداشته و از آن شناخته شدن تست توسط `pytest` در VSCode اطمینان حاصل کنید. برای این منظور ابتدا کامنت‌های مربوط به یک تست را بردارید. سپس تست را مطالعه کنید. نام تابع مورد تست و پارامترهای ورودی و نوع مقدار برگشتی تابع مورد تست را تشخیص دهید. سپس در فایل `exam.py` تابع را پیاده‌سازی کنید. اگر این اولین تست است توسط دستور `Python: Configure Tests` در VSCode بستر تست `pytest` را فعال کنید. سپس از شناخته شدن تستی که کامنت آن را برداشتید اطمینان حاصل کنید. این فرایند را تا برداشته شدن کلیه کامنت‌های تست ادامه دهید.

۲.۱.۲ فرستادن تست در Azure DevOps

پس از شناخته شدن کلیه تست‌ها در VSCode کد خود را `add/commit/push` کرده و سپس در `Azure DevOps` یک `Pull Request` برای بردن این تغییرات از شاخه `fb_Exam1` به شاخه `holymaster` درست کنید. چنانچه شاخه `holymaster` و `branch policy` مربوط به آنرا بدرستی تنظیم کرده باشید، بیلد مرتبط با این `Pull Request` باید موفقیت آمیز باشد.

۲.۲ پیاده‌سازی توابع

از تست شماره یک شروع کرده و دستور `@pytest.mark.skip` قبل از تست را برداشته تا تست فعال شود. سپس تست را اجرا کرده و از عدم اجرای موفقیت‌آمیز آن اطمینان حاصل کنید. سپس تابع مورد استفاده تست را بگونه‌ای پیاده‌سازی کنید که تست با موفقیت پاس شود.

۱.۲.۲ تست `test_q0_distance`تابع `q0_distance`چهار عدد صحیح (x_1, y_1, x_2, y_2) به عنوان مختصات دو نقطه را به عنوان پارامتر ورودی دریافت کرده و فاصله اقلیدسی آن دو را برمی‌گرداند.۲.۲.۲ تست `test_q1_poly1`تابع `q1_poly1` سه عدد a, b, x را به عنوان پارامتر دریافت کرده و $ax + b$ را برمی‌گرداند.۳.۲.۲ تست `test_q2_poly1_str`تابع `q2_poly1_str` دو عدد a, b را به عنوان پارامتر دریافت کرده و $ax + b$ را در قالب یک رشته حرفی (string) برمی‌گرداند. به عنوان مثال: $2x + 3$ ۴.۲.۲ تست `test_q3_digit_sum`تابع `q1_poly1` یک عدد از به عنوان پارامتر دریافت کرده و جمع ارقام آن را برمی‌گرداند.۵.۲.۲ تست `test_q4_filter_dividable`تابع `q1_filter_dividable` یک لیست و یک عدد از ورودی دریافت کرده و اعدادی از لیست که بر عدد ورودی بخش پذیر هستند را به صورت یک لیست برمی‌گرداند.۶.۲.۲ تست `est_q5_poly_n`تابع `q5_poly_n` یک لیست از ضرایب یک چند جمله‌ای درجه n و مقدار عدد x را به عنوان پارامتر دریافت کرده و مقدار چند جمله‌ای را برای عدد x حساب می‌کند.۷.۲.۲ تست `est_q6_poly_n_str`تابع `q6_poly_n_str` یک لیست از ضرایب یک چند جمله‌ای درجه n را از به عنوان پارامتر دریافت کرده و رشته حرفی (string) متناظر با آنرا برمی‌گرداند. به عنوان مثال:

```
[1, 3, 2, 1]
1+3x+2x^2+1x^3
```

۸.۲.۲ تست `est_q7_poly_n_str2`تابع `q7_poly_n_str2` یک لیست از ضرایب یک چند جمله‌ای درجه n را از به عنوان پارامتر دریافت کرده و رشته حرفی (string) متناظر با آنرا برمی‌گرداند. تفاوت این سوال با سوال قبل در عدم چاپ ۱ در ضریب و توان، بلاوه حذف عناصر چند جمله‌ای که ضریب صفر دارند. به عنوان مثال:

```
[1, 1, 4, 2, 1]
x^4+2x^3+4x^2+x+1
```

۳ ارسال

اگر موفق به پاس شدن تستی نشدید دستور مربوط به عدم اجرای تست را قبل از تست باقی بگذارید. پس از پیاده‌سازی توابع و پاس شدن تست‌هایی که فرصت کردین، نوبت به ارسال آنها می‌رسد. مثل قبل تغییرات را در شاخه `fb_Exam1` add/commit/push کنید.

۱.۳ ساخت Pull Request

با مراجعه به سایت [Azure DevOps](#) لز موفقیت بیلد برای Pull Request که در مرحله اول درست کردید اطمینان حاصل کنید و آنرا کامل کنید. دقت کنید که گزینه `Delete source branch` نباید انتخاب شود.