

## فایل نمرات

می‌خواهیم برنامه‌ای بنویسیم تا بتوانیم با فایل نمرات دانشجویان کار کرده و کارهایی که پیش از این توسط مسئولین آموزش به صورت دستی بر روی این فایل‌ها انجام می‌شد را خودکار کنیم. هر سطر از این فایل دارای سه بخش است که با کاراکتر فاصله ( " " ) از هم جدا شده‌اند. این سه بخش به ترتیب عبارتند از:

۱. شماره دانشجویی (عدد صحیح)

۲. کد درس (عدد صحیح)

۳. نمره (عدد اعشاری)

یعنی هر سطر از این فایل مشخص می‌کند که یک دانشجو در یک درس چه نمره‌ای را گرفته است.

چند نیاز اساسی در رابطه با این فایل‌ها وجود دارد:

- می‌خواهیم بتوانیم یک سطر دلخواه از این فایل را در قالب یک شیء از جنس `Grade` بخوانیم.
- می‌خواهیم بتوانیم یک شیء از جنس `Grade` را به صورت یک سطر از فایل نمرات و در انتهای فایل ذخیره کنیم.
- می‌خواهیم بتوانیم معدل دانشجویان در یک درس خاص را محاسبه کنیم.
- می‌خواهیم بتوانیم معدل یک دانشجو را محاسبه کنیم.

### جزئیات

یک کلاس به نام `Grade` شامل سه خصوصیت با اسامی زیر بنویسید.

- شماره دانشجویی: `student_id`
- کد درس: `course_code`
- نمره: `score`

کلاس شما باید دارای متد سازنده‌ای باشد که این سه خصوصیت را به همین ترتیب دریافت و مقداردهی کند.

**این کلاس در فایل پروژه اولیه تکمیل شده است و شما باید تنها کلاس `CourseUtil` را تکمیل کنید.**

با توجه به نیازمندی‌هایی که در بالا گفته شد، یک کلاس با نام `CourseUtil` تعریف کنید که همه متدهای جدول زیر را طبق رفتار توضیح داده شده پیاده‌سازی کند.

 source.py

```
1 class Grade:
2     def __init__(self, stu_id, crc_code, score):
3         self.student_id = stu_id
4         self.course_code = crc_code
5         self.score = score
6
7
8 class CourseUtil:
9     def set_file(self, address):
10         pass
```

```
11
12 def load(self, line_number):
13     pass
14
15 def calc_student_average(self, student_id):
16     pass
17
18 def calc_course_average(self, course_code):
19     pass
20
21 def count(self):
22     pass
23
24 def save(self, grade):
25     pass
```

رفتار	خروجی	متد
آدرس فایل نمرات به ورودی این متد داده می‌شود.	-	set_file(address)
یک شماره خط از فایل را در ورودی می‌گیرد و اطلاعات موجود در آن خط از فایل را در قالب یک شیء Grade برمی‌گرداند.	Grade	load(line_number)
یک شیء Grade در ورودی می‌گیرد و طبق فرمت مورد نظر، آن را در انتهای فایل اضافه می‌کند.	-	save(grade)
میانگین نمرات دانشجویان در درس با کد course_code را برمی‌گرداند.	float	calc_course_average(course_code)
میانگین نمرات دانشجو با شماره دانشجویی student_id در درس‌های مختلف را برمی‌گرداند.	float	calc_student_average(student_id)
تعداد کل نمرات موجود در فایل نمرات را برمی‌گرداند.	int	count()

توضیحات بیشتر

- آدرس فایل با کمک متد set\_file در اختیار کلاس قرار داده می‌شود. شما باید خودتان فایل را به نحوه دلخواه باز کنید.
- در متد load اگر مقدار line\_number بزرگتر از تعداد خط‌های فایل بود، باید None برگردانید. شماره‌ی خطوط از ۱ شروع می‌شود.
- متد set\_file ممکن است چند بار فراخوانی شود که پس از هر فراخوانی، اطلاعات فایل جدید باید خوانده شود.
- در فایل نمرات برای هر دانشجو در هر درس فقط یک نمره وجود دارد. بنابراین اگر برای دانشجو و درسی که نمره اش در فایل وجود دارد نمره جدیدی با متد save ثبت کنیم، نباید در فایل تغییری ایجاد شود.
- در انتهای فایل \n وجود ندارد. شما نیز باید طوری فایل را تغییر دهید که هیچ گاه در انتهای فایل \n وجود نداشته باشد.
- تضمین می‌شود که در هنگام محاسبه میانگین‌ها شماره دانشجویی یا شماره درس داده شده در فایل موجود است.

نمونه

در کد زیر یک فایل شامل دو سطر به عنوان ورودی داده شده است.

```
1 util = CourseUtil()
2 util.set_file(file)
3 print(util.count())    # 2
4
5 grade = Grade(445612,1234,12)
6 util.save(grade)
7 print(util.count())    # 3
8
9 util.set_file(file)
10 print(util.count())    # 3
```

نکات

- می‌توانید فایل اولیه این سوال را از این لینک دریافت کنید.
- در توابعی که از شما خواسته شده مقداری را بازگردانید، باید مقدار مورد نظر را return کنید.

نحوه ارسال

شما باید کلاس CourseUtil موجود در فایل source.py را تکمیل و سپس این فایل را ارسال کنید.

منبع سؤال: مسابقات جاواکاپ