

Bài tập số: 1

Tên dự án: Tính toán và phân tích điểm thi (Test Grade Calculator)

Tổng quan dự án

Trong bài tập này, ta cần viết một chương trình để tính toán điểm thi cho nhiều lớp với số lượng hàng nghìn học sinh. Mục đích của chương trình này là giúp giảm thời gian chấm bài. Ta sẽ áp dụng các functions (hàm) khác nhau trong Python để viết chương trình với các tác vụ sau:

1. Mở các file văn bản bên ngoài được yêu cầu với exception-handling
2. Quét từng dòng của câu trả lời bài thi để tìm dữ liệu hợp lệ và cung cấp báo cáo tương ứng
3. Chấm điểm từng bài thi dựa trên tiêu chí đánh giá (rubric) được cung cấp và báo cáo
4. Tạo file kết quả được đặt tên phù hợp

Tại sao cần làm dự án này?

Bài tập này cung cấp cho học viên một nguồn dữ liệu thực tế. Chúng ta sẽ thực hành sử dụng các hàm khác nhau trong Python để giải quyết một bài toán cụ thể.

Chúng ta học được gì?

Ta sẽ học cách viết một chương trình Python tập trung vào lập trình cơ bản, sử dụng list, function và xử lý với chuỗi. Hơn nữa, ta sẽ biết cách truy cập file để đọc dữ liệu và ghi kết quả thống kê được.

Dự án này giúp ích gì cho nghề nghiệp?

Bài tập này sẽ giúp ta phát triển các kỹ năng trong lập trình Python, từ đó có thể áp dụng trong các bộ dữ liệu khác nhau.

CHI TIẾT DỰ ÁN

Làm thế nào để hoàn thành bài tập?

1. Tải [File dữ liệu](#) và lưu trong 1 folder trên máy tính của bạn
2. Thực hiện theo các bước trong mục Hướng dẫn dự án để hoàn thành các yêu cầu (tasks) của dự án

(Lưu ý: Đọc kỹ phần miêu tả dữ liệu tương ứng với từng yêu cầu trong mục hướng dẫn dự án)

Tài nguyên tham khảo

- Version Control

Chúng ta nên sử dụng **Git** ngay từ đầu. Hãy đảm bảo tuân theo đúng các yêu cầu về format trong

Git Style Guide

- **Hướng dẫn Python Style**

Ta nên viết code và markup đáp ứng các thông số kỹ thuật được cung cấp trong các hướng dẫn này: [Python Style Guide](#).

Yêu cầu dự án

Hoàn thành các tasks dưới đây:

Task 1:

Tạo một chương trình Python mới có tên “`lastname_firstname_grade_the_exams.py`.” (Đảm bảo tệp mã nguồn của bạn nằm trong cùng thư mục với tệp dữ liệu bạn vừa tải xuống.)

Tiếp theo, viết một chương trình cho phép người dùng nhập tên của một tệp. Cố gắng mở tệp được cung cấp để truy cập đọc. Nếu tệp tồn tại, bạn có thể in ra một thông báo xác nhận. Nếu tệp không tồn tại, bạn nên cho người dùng biết rằng không thể tìm thấy tệp và nhắc lại họ.

Sử dụng try/except để thực hiện việc này (đừng chỉ sử dụng một loạt câu lệnh “if” — chương trình này càng “chung chung” càng tốt). **Bạn sẽ không nhận được điểm cho phần này của chương trình nếu bạn viết một cái gì đó như sau để xác định các tệp dữ liệu hợp lệ:**

```
filename = input("Enter a filename: ")
if filename == "class1":
    # open class1.txt
elif filename == "class2":
    # open class2.txt
else:
    print ("Sorry, I can't find this filename")
```

Đây là một mẫu kết quả sau khi chạy chương trình:

Enter a class file to grade (i.e. class1 for class1.txt): foobar

File cannot be found.

Enter a class file to grade (i.e. class1 for class1.txt): class1

Successfully opened class1.txt

Task 2:

Tiếp theo, bạn sẽ cần phân tích dữ liệu có trong tệp bạn vừa mở để đảm bảo rằng nó ở đúng định dạng. Mỗi tệp dữ liệu chứa một loạt câu trả lời của học sinh ở định dạng sau:

N12345678,B,A,D,D,C,B,D,A,C,C,D,B,A,B,A,C,B,D,A,C,A,A,B,D,D

hoặc

N12345678,B,,D,,C,B,,A,C,C,,B,A,B,A,,,A,C,A,A,B,D,

Giá trị đầu tiên là số ID của sinh viên. 25 chữ cái sau là câu trả lời của học sinh cho kỳ thi. Tất cả các giá trị được phân tách bằng dấu phẩy. Nếu không có chữ cái nào sau dấu phẩy, điều này có nghĩa là học sinh đã bỏ qua việc trả lời câu hỏi.

Lưu ý rằng một số dòng dữ liệu có thể bị hỏng! Ví dụ: dòng dữ liệu này không có đủ câu trả lời:

N12345678,B,A,D,D,C,B

Và dòng dữ liệu này có quá nhiều câu trả lời:

N12345678,B,A,D,D,C,B,D,A,C,C,D,B,A,B,A,C,B,D,A,C,A,A,B,D,D,A,B,C,D,E

Nhiệm vụ của bạn cho phần này của chương trình là thực hiện như sau:

1. Báo cáo tổng số dòng dữ liệu được lưu trữ trong tệp.
2. Phân tích từng dòng và đảm bảo rằng nó là "hợp lệ".
 - Một dòng hợp lệ chứa danh sách 26 giá trị được phân tách bằng dấu phẩy
 - N# cho một học sinh là mục đầu tiên trên dòng. Nó phải chứa ký tự "N" theo sau là 8 ký tự số.
3. Nếu một dòng dữ liệu không hợp lệ, bạn nên báo cáo cho người dùng bằng cách in ra một thông báo lỗi. Bạn cũng nên đếm tổng số dòng dữ liệu hợp lệ trong tệp.

Gợi ý: Sử dụng phương pháp `split` để tách dữ liệu ra khỏi tệp. Bạn có thể cần sử dụng phương pháp này một vài lần cùng với một hoặc hai vòng lặp. Hãy suy nghĩ về thứ tự mà bạn cần chia các mục của mình. Ví dụ: tệp của bạn được sắp xếp sao cho hồ sơ của một học sinh chiếm toàn bộ dòng trong tệp. Việc tách trước khi ngắt dòng sẽ tách biệt dữ liệu của từng học sinh. Sau đó, bạn sẽ cần phải chia nhỏ từng mục dựa trên ký tự phân tách để rút ra câu trả lời cho từng học sinh.

Đây là một mẫu chạy chương trình của bạn cho hai tệp dữ liệu đầu tiên. Bạn có thể tìm thấy danh sách đầy đủ đầu ra dự kiến cho tất cả các tệp dữ liệu trong gói có thể tải xuống cho bài tập này.

```
Enter a class to grade (i.e. class1 for class1.txt): class1
```

```
Successfully opened class1.txt
```

```
**** ANALYZING ****
```

```
No errors found!
```

```
**** REPORT ****
```

```
Total valid lines of data: 20
```

```
Total invalid lines of data: 0
```

```
Enter a class to grade (i.e. class1 for class1.txt): class2
```

```
Successfully opened class2.txt
```

```
**** ANALYZING ****
```

```
Invalid line of data: does not contain exactly 26 values:
```

```
N00000023,,A,D,D,C,B,D,A,C,C,,C,,B,A,C,B,D,A,C,A,A
```

```
Invalid line of data: N# is invalid
```

```
N0000002,B,A,D,D,C,B,D,A,C,D,D,D,A,,A,C,D,,A,C,A,A,B,D,D
```

```
Invalid line of data: N# is invalid
```

```
NA0000027,B,A,D,D,,B,,A,C,B,D,B,A,,A,C,B,D,A,,A,A,B,D,D
```

```
Invalid line of data: does not contain exactly 26 values:
```

```
N00000035,B,A,D,D,B,B,,A,C,,D,B,A,B,A,A,B,D,A,C,A,C,B,D,D,A,A
```

```
**** REPORT ****
```

```
Total valid lines of data: 21
```

```
Total invalid lines of data: 4
```

Task 3:

Tiếp theo, bạn sẽ viết một chương trình để chấm điểm các bài thi cho một phần nhất định. Kỳ thi gồm 25 câu hỏi, trắc nghiệm. Đây là một chuỗi đại diện cho các câu trả lời:

```
answer_key = "B,A,D,D,C,B,D,A,C,C,D,B,A,B,A,C,B,D,A,C,A,A,B,D,D"
```

Chương trình của bạn nên sử dụng những câu trả lời này để tính điểm cho mỗi dòng dữ liệu hợp lệ. Điểm có thể được tính như sau:

- +4 điểm cho mỗi câu trả lời đúng
- 0 điểm cho mỗi câu trả lời bị bỏ qua
- -1 điểm cho mỗi câu trả lời sai

Bạn cũng sẽ muốn tính toán các thống kê sau cho toàn bộ lớp:

- Điểm trung bình
- Điểm cao nhất
- Điểm thấp nhất
- Miền giá trị của điểm (cao nhất trừ thấp nhất)
- Giá trị trung vị (Sắp xếp các điểm theo thứ tự tăng dần. Nếu # học sinh là số lẻ, bạn có thể lấy giá trị nằm ở giữa của tất cả các điểm (tức là [0, 50, 100] — trung vị là 50). Nếu # học sinh là chẵn bạn có thể tính giá trị trung vị bằng cách lấy giá trị trung bình của hai giá trị giữa (tức là [0, 50, 60, 100] — giá trị trung vị là 55)).

Gợi ý: Khi đã cho điểm các học sinh, bạn nên sử dụng một list để lưu trữ điểm số của từng học sinh; sau đó bạn có thể tính toán số liệu thống kê sau khi đã kiểm tra mọi học sinh trong tệp.

Đây là một mẫu chạy chương trình của bạn cho hai tệp dữ liệu đầu tiên. Bạn có thể tìm thấy danh sách đầy đủ đầu ra dự kiến cho tất cả các tệp dữ liệu trong gói có thể tải xuống cho bài tập này.

```
Enter a class to grade (i.e. class1 for class1.txt): class1
```

```
Successfully opened class1.txt
```

```
**** ANALYZING ****
```

```
No errors found!
```

```
**** REPORT ****
```

```
Total valid lines of data: 20
```

```
Total invalid lines of data: 0
```

```
Mean (average) score: 75.60
```

```
Highest score: 91
```

```
Lowest score: 59
```

```
Range of scores: 32
```

```
Median score: 73
```

```
Enter a class to grade (i.e. class1 for class1.txt): class2
```

```
Successfully opened class2.txt
```

```
**** ANALYZING ****
```

```
Invalid line of data: does not contain exactly 26 values:
```

```
N00000023,,A,D,D,C,B,D,A,C,C,,C,,B,A,C,B,D,A,C,A,A
```

```
Invalid line of data: N# is invalid
```

```
N0000002,B,A,D,D,C,B,D,A,C,D,D,D,A,,A,C,D,,A,C,A,A,B,D,D
```

```
Invalid line of data: N# is invalid
```

```
NA00000027,B,A,D,D,,B,,A,C,B,D,B,A,,A,C,B,D,A,,A,A,B,D,D
```

```
Invalid line of data: does not contain exactly 26 values:
```

```
N00000035,B,A,D,D,B,B,,A,C,,D,B,A,B,A,A,B,D,A,C,A,C,B,D,D,A,A
```

```
**** REPORT ****
```

```
Total valid lines of data: 21
```

```
Total invalid lines of data: 4
```

```
Mean (average) score: 78.00
```

```
Highest score: 100
```

```
Lowest score: 66
```

```
Range of scores: 34
```

```
Median score: 76
```

Task 4:

Cuối cùng, yêu cầu chương trình của bạn tạo một tệp “kết quả” chứa các kết quả chi tiết cho từng học sinh trong lớp của bạn. Mỗi dòng của tệp này phải chứa số ID của học sinh, dấu phẩy và sau đó là điểm của họ. Bạn nên đặt tên tệp này dựa trên tên tệp gốc được cung cấp — ví dụ: nếu người dùng muốn phân tích “class1.txt”, bạn nên lưu trữ kết quả trong tệp có tên “class1_grades.txt”.

Đây là một mẫu chạy chương trình của bạn cho hai tệp dữ liệu đầu tiên. Bạn có thể tìm thấy danh sách đầy đủ đầu ra dự kiến cho tất cả các tệp dữ liệu trong gói có thể tải xuống cho bài tập này.

```
# this is what class1_grades.txt should look like
N00000001,59
N00000002,70
N00000003,84
N00000004,73
N00000005,83
N00000006,66
N00000007,88
N00000008,67
N00000009,86
N00000010,73
N00000011,86
N00000012,73
N00000013,73
N00000014,78
N00000015,72
N00000016,91
N00000017,66
N00000018,78
N00000019,78
N00000020,68
# this is what class2_grades.txt should look like
N00000021,68
N00000022,76
N00000024,73
N00000026,72
N00000028,73
N00000029,87
N00000030,82
N00000031,76
N00000032,87
N00000033,77
N00000034,69
N00000036,77
N00000037,75
N00000038,73
N00000039,66
N00000040,73
N00000041,91
N00000042,100
N00000043,86
N00000044,90
N00000045,67
```

Task 5: Chỉ sử dụng pandas và numpy khi bạn triển khai task 1 đến task 4.

Lưu ý về nộp bài

1. Lưu chương trình Python của bạn với tên `lastname_firstname_grade_the_exams.py`
2. Viết phần hướng dẫn cho dự án và đưa vào README.md. Bạn có thể tham khảo thêm tại khóa [**Writing READMEs**](#)
3. Nhánh chính (master branch) là nhánh lưu trữ Github mặc định. Nếu bạn muốn nộp một nhánh khác, bạn cần phải đặt nó làm nhánh mặc định mới ([**new default branch**](#)) bên trong kho lưu trữ Github của bạn

Yêu cầu chức năng

#	Tiêu chí (Criteria)	Đặc tả (Specification)	Trọng số (Weight)	Điểm	Bắt buộc (Mandatory)	Grading Type
1	Mở các tệp dữ liệu bên ngoài được yêu cầu	Có thể đọc tất cả các file được tải về Sử dụng exception nếu tên của tệp input không tồn tại trong folder dữ liệu (đừng chỉ dùng một loạt các câu lệnh "if")	2	1,5	Yes	Scale (0-5)
2	Scan file và phân tích	Scan từng dòng trong tệp và đảm bảo format hợp lệ Báo cáo dữ liệu "không hợp lệ" kèm theo lý do Báo cáo tổng số dòng hợp lệ và số dòng không hợp lệ của dữ liệu lưu trong tệp	2	1,5	Yes	Scale (0-5)
3	Tính toán điểm thi	Chấm điểm bài thi dựa vào tiêu chí đánh giá (rubrics) đã cho Báo điểm trung bình, điểm cao nhất, thấp nhất, miền giá trị của điểm và giá trị trung vị	2	1,5	Yes	Scale (0-5)
4	Tạo tệp kết quả	Viết kết quả vào tệp (số lượng file xuất ra phải bằng số lượng file nhập vào) Mỗi dòng chứa số ID của học viên, dấu phẩy, và điểm của họ	2	1,5	Yes	Scale (0-5)

Yêu cầu phi chức năng

#	Tiêu chí (Criteria)	Đặc tả (Specification)	Trọng số (Weight)	Điểm	Bắt buộc (Mandatory)	Grading Type
5	README	File README cần có đủ các bước hướng dẫn chi tiết để chạy ứng dụng thành công	1	0,8	No	Scale (0-5)
6	Chú thích	Yêu cầu có chú thích (Comment line) cho các đoạn code quan trọng hoặc để giải thích các đoạn code dài	1	0,8	No	Scale (0-5)

7	Chất lượng Code	Code cần tuân thủ theo các yêu cầu về format (nhất quán, logic và dễ hiểu) theo chuẩn Python Style Guide	1	0,8	No	Scale (0-5)
----------	------------------------	---	----------	------------	-----------	--------------------

Yêu cầu nâng cao

#	Tiêu chí (Criteria)	Đặc tả (Specification)	Trọng số (Weight)	Điểm	Bắt buộc (Mandatory)	Grading Type
1	Sử dụng Pandas	Sử dụng pandas khi thực thi các task	2	1,6	No	Scale (0-5)