

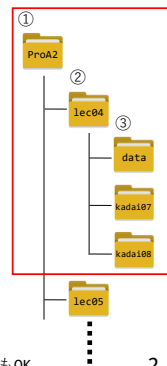
2022年10月24日(月)5限
@研究棟A302

プログラミングA2 第4回

演習課題

準備

1. デスクトップなどの任意の場所にプログラミングA2用のフォルダを作成する
2. 1.の下に第4回授業用のフォルダを作成する
3. 2.の下にデータファイル用のフォルダ、課題07、課題08用のフォルダを作成する



※必ずしも、このようなディレクトリ構造でなくてもOK

2

1

2

注意点

- 採点の都合上、指示された要件を満たすコードを実装すること
- 授業時間中に提出された場合は2点満点
- 授業時間外に提出された場合は1点満点
- 授業時間中に1問も提出できなかった場合は、学修支援センターで指導・チェックを受けること
※チェックだけでもOK
※1問だけ終わっていない状況でも、LSCに行くことはもちろんOK
- 授業時間外の質問は、Slackにて伏見が受け付ける(24時間365日営業)

3

3

5限課題チェックの流れ

1. 受講生：課題ができれば、#授業中_課題チェック依頼 に依頼文を投稿する
- 例：「課題03のチェックをお願いします」
2. TASA：スレッドに返信する で応答する
- 例：「チェックするので、DM送ってください」
3. 受講生：担当TASAにDMで課題プログラムを送信する
4. TASA：
 1. DMでフィードバック（正解／不正解などこが悪いのかを簡単に）
 2. 済を付ける
5. 受講生：Moodleに提出する
6. TASA：Moodleに点数を入力する（あとで一気に入力する場合もある）

4

4

時間外課題チェックの流れ

時間内に1問も提出できなかった場合は、学修支援センター(LSC)で指導・チェックを受けること
・意図：理解できていない可能性が高い→理解度を高める
・利点：加点が保証される(独自で提出して間違っていたら0点)

1. 受講生：課題ができれば、Moodleに提出する

※時間外提出の場合は、Slackでのチェック依頼不要

- LSCでチェック済の場合：
- 2. 教員：その日のうちにMoodleに加点する
- LSCで未チェックの場合：
- 2. TASA：次回講義時間中に採点、Moodleに点数を入力する

5

5

授業中質疑応答の流れ

- 緊急案件の場合
 - 挙手によりTASAを呼ぶ
- 上記以外の場合
 - Slackの #授業中_302_質問ボックス or #授業中_303_質問ボックスに質問内容を書き込む。
 - 質問内容を書くときは、エラー内容、スクショ、コードのコピペなどを投稿すること。
 - 込み入った内容でテキスト化するのが面倒なときは、「対面希望」とおおよその座席を書き込む。
 - 順番に呼び出すなどして教員が対応する。

基本的にSlackに投稿された順に対応する

6

6

【課題 07 : shiritori.py】

しりとりの方領でポケモンの名前を順に抽出する iterable なクラスとそのイテレータクラスを実装せよ。

採点の都合上、以下の要件を満たすこととする：

- (1) iterable な Zukan クラスを定義する；
 - (a) Zukan クラスが iterable になるように、適切な特殊メソッドを追加する；
 - (b) ただし、Zukan クラス自体はイテレータではないことに注意する；
- (2) iterable な Zukan クラスに対応するイテレータクラスを定義する；
 - (a) イニシャライザで、Zukan クラスのインスタンスを受け取り、インスタンス変数に設定する；
 - (b) イニシャライザで、Zukan クラスのインスタンスから名前文字列をランダムに 1 つ選び、インスタンス変数に設定する（これが、しりとり最初の単語となる）；
 - (c) Zukan クラスのインスタンス（iterable）から、現在選ばれている単語（名前文字列）の最後の文字と等しい頭文字を持つ単語をすべて抽出する（これが、次の単語の候補となる）；
 - (d) 候補単語がなかった場合は、適切な例外を raise する；
 - (e) 候補単語がある場合は、その中からランダムに選択した単語を次の単語とする；
- (3) 読み込みファイルのパスは、コマンドライン引数により指定する；
- (4) 配布した shiritori.py にあらかじめ書かれている部分は、変更しないこと；
- (5) 出力の形式は、図 1 のようにする；

```
init: ヤミカラス
001   ストライク
002   クラブ
003   ブルー

PS C:\Users\admin\Desktop\ProA2>
xe' 'c:\Users\admin\.vscode\extens
auncher' '58441' '--' 'c:\Users\ad
init: クサイハナ
001   ナゾノクサ
002   サナギラス
003   スイクン
```

図 1 課題 07 の実行例：

リスト 1 shiritori.py

```
1 import sys
2 from random import choice
3
4 class Zukan:
5     def __init__(self, file_path):
6         self.titles = __class__.read_file(file_path)
7
8     @staticmethod
9     def read_file(file_path):
10         titles = []
11         with open(file_path, "r", encoding="utf8") as rfo:
12             for row in rfo:
13                 _, tit, _, *_ = row.rstrip().split("\t")
14                 titles.append(tit)
15         return titles
16
17
18 if __name__ == "__main__":
19     zukan = Zukan(sys.argv[1]) # lec04/data/poke_names.txt
20     for i, z in enumerate(zukan, 1):
21         print(f"{i:03d}\t{z}")
```

【課題 08 : party.py】

ポケモン 3 体からなるパーティーを構築し、スコアが最も高いパーティーを探すコードを実装せよ。配布した名前ファイル「poke_names.txt」と種族値ファイル「base_stats.txt」を読み込み、全 251 種のポケモンインスタンスを要素とするリストを作成する。そして、251 種から 3 体選ぶ全組合せに対して、スコアを計算する。ポケモンパーティーのスコアは、3 体のポケモンの種族値の合計値とする。

251 種のポケモンから 3 体を選ぶ組合せの総数は ${}_{251}C_3 = 2,604,125$ であり、膨大となるため、全組合せを要素とするコンテナ（リストなど）を構築することなく、itertools モジュールの combinations 関数を利用すること。

採点の都合上、以下の要件を満たすこととする：

- (1) ポケモン 1 体を表す Monster クラスはすでに定義されたものを使用する；
- (2) 全ポケモンのリストを表す iterable なクラス MonsterList を定義する；
 - (a) 【済】静的メソッド read_files() は、名前ファイル「poke_names.txt」と種族値ファイル「base_stats.txt」を読み込み、名前文字列と種族値リストから Monster クラスのインスタンスを生成し、それらを要素とするリストを返すように定義してある；
 - (b) 【済】イニシャライザは、read_files() の戻り値を、インスタンス変数に設定している；
 - (c) MonsterList クラスが iterable になるように、適切な特殊メソッドを定義する。※ただし、今回はインデックスによって要素にアクセスできるようにする（つまり、__iter__() ではない）；
- (3) ポケモンパーティーに対して、スコア（種族値の合計）を計算する関数を定義する；
- (4) 3 ポケモンからなるパーティーを生成して、最大値を求めるジェネレータ関数を定義する；
 - (a) itertools の combinations 関数によるイテレータを用いて、全パーティーを 1 つずつ生成する；
 - (b) ※総数が非常に大きいため、全パーティーを要素とするコンテナは作らないこと；
 - (c) 生成したパーティーに対して、上で定義した関数によりスコアを計算する；
 - (d) 現在の（仮の）最大値と比較して、スコアが高いパーティーが出現したら、ジェネレータ関数の呼び出し元に、パーティーとスコアを返す；
 - (e) 呼び出し元では、戻された値（パーティーとスコア）を print する；
- (5) 名前ファイルのパス、種族値ファイルのパスは、コマンドライン引数により、この順番で指定する；
- (6) 出力の形式は、図 2 のようにする；

```

PS C:\Users\admin\Desktop\ProA2> c:; cd 'c:\Use
xe' 'c:\Users\admin\.vscode\extensions\ms-python
auncher' '58465' '--' 'c:\Users\admin\Desktop\Pr
(1248, (フシギダネ, フシギソウ, フシギバナ))
(1257, (フシギダネ, フシギソウ, リザードン))
(1278, (フシギダネ, フシギソウ, ウインディ))
(1303, (フシギダネ, フシギソウ, フリーザー))
(1323, (フシギダネ, フシギソウ, カイリュー))
(1403, (フシギダネ, フシギソウ, ミュウツー))
(1423, (フシギダネ, フシギバナ, フリーザー))
(1443, (フシギダネ, フシギバナ, カイリュー))
(1523, (フシギダネ, フシギバナ, ミュウツー))
(1532, (フシギダネ, リザードン, ミュウツー))
(1553, (フシギダネ, ウインディ, ミュウツー))
(1578, (フシギダネ, フリーザー, ミュウツー))
(1598, (フシギダネ, カイリュー, ミュウツー))
(1678, (フシギダネ, ミュウツー, ルギア))
(1685, (フシギソウ, カイリュー, ミュウツー))
(1765, (フシギソウ, ミュウツー, ルギア))
(1785, (フシギバナ, フリーザー, ミュウツー))
(1805, (フシギバナ, カイリュー, ミュウツー))
(1885, (フシギバナ, ミュウツー, ルギア))
(1894, (リザードン, ミュウツー, ルギア))
(1915, (ウインディ, ミュウツー, ルギア))
(1940, (フリーザー, ミュウツー, ルギア))
(1960, (カイリュー, ミュウツー, ルギア))
(2040, (ミュウツー, ルギア, ホウオウ))

```

図 2 課題 08 の実行例 :

リスト 2 party.py

```

1 from itertools import combinations
2 import sys
3
4 class Monster:
5     def __init__(self, title, stats):
6         self.title = title
7         self.stats = stats
8
9     def __repr__(self):
10         return self.title
11

```

```
12
13 class MonsterList:
14     def __init__(self, file_path1, file_path2):
15         self.monsters = __class__.read_files(file_path1, file_path2)
16
17     @staticmethod
18     def read_files(file_path1, file_path2):
19         titles = []
20         with open(file_path1, "r", encoding="utf8") as rfo:
21             for row in rfo:
22                 _, tit, _, *_ = row.rstrip().split("\t")
23                 titles.append(tit)
24
25         stats = []
26         with open(file_path2, "r") as rfo:
27             for row in rfo:
28                 row = row.rstrip()
29                 stats.append([int(col) for col in row.split(" ")])
30
31         monsters = [Monster(title, stat) for title, stat in zip(titles, stats)]
32         return monsters
33
34
35 if __name__ == "__main__":
36     monster_lst = MonsterList(sys.argv[1], sys.argv[2]) # lec04/data/poke_names.txt, lec04/data/
37     base_stats.txt
38     ジェネレータイテレータ = ジェネレータ関数 ()
39     for res in ジェネレータイテレータ:
40         print(res)
```
