

2022年10月03日(月) 5限
@研究棟A302

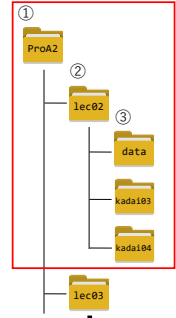
プログラミングA2 第2回

演習課題

1

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1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03、課題04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

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

【課題 03 : pokemon.py】

出現したポケモンが図鑑に登録されているかを調べるコードを実装せよ。配布した図鑑用データファイル「poke_names.txt」を読み込み、ポケモン図鑑を作成する。当該ファイルは1行が1匹のポケモンのデータであり、ポケモン番号、名前、分類、タイプがタブ区切りで書かれている。

採点の都合上、以下の要件を満たすこととする：

- 基本となる Monster クラスを定義する；
 - － インシャライザで、名前を設定する；
 - － 等価比較演算子でインスタンスの等価性を評価できるようにする。インスタンスの等価性は、名前を表すインスタンス変数の値が等しいことで定義する；
- Monster クラスを継承した ZukanMonster クラスを定義する；
 - － インシャライザで、名前、分類、タイプを設定する。名前は、親クラスのインシャライザを呼び出して設定する；
 - － インスタンスを文字列化した際に、図 1 のような出力になるように特殊メソッドをオーバーライドする；
- Zukan クラスを定義する；
 - － インシャライザを以下のように定義する；
 - * インシャライザで、図鑑用データファイルのパスを受け取り、ファイル読み込み関数を呼び出す；
 - * 読み込み関数が return するリストをインスタンス変数として受け取る；
 - － 図鑑用データファイルの読み込み関数を静的メソッドとして定義する；
 - * 引数として受け取った図鑑用データファイルを開く；
 - * 読み込んだデータに基づき、ZukanMonster クラスのインスタンスを生成する；
 - * インスタンスが並ぶリストを return する；
 - － 図鑑にポケモンが登録されているかを検索するメソッドを定義する；
 - * 引数として受け取ったインスタンスと、図鑑に登録されているインスタンス 1 つずつと等価性を比較する；
 - * 図 1 の出力になるように、適切なものを return する；
- adventure.py には変更を加えない；
- 出力の形式は、図 1 のようにする；

```
PS C:\Users\admin\Desktop\ProA2> python .\lec02\kadai03\adventure.py .\lec02\data\poke_names.txt ピカチュウ
なまえ：ピカチュウ
ぶんるい：ねずみポケモン
たいぶ：でんき
PS C:\Users\admin\Desktop\ProA2> python .\lec02\kadai03\adventure.py .\lec02\data\poke_names.txt ポケチュウ
ポケチュウは図鑑に登録されていない新種です
```

図 1 課題 03 の実行例：“ピカチュウ”は図鑑に登録されているが，“ポケチュウ”は登録されていない。

リスト 1 adventure.py

```
1 from pokemon import *
2 import sys
3
4 if __name__ == "__main__":
5     file_path = sys.argv[1] # "lec02/data/poke_names.txt"
6     zukan = Zukan(file_path)
7     poke_name = sys.argv[2]
8     mon = Monster(poke_name)
9     print(zukan.search_monster(mon))
```

【課題 04 : pokemon.py】

自分が選んだポケモンと、ランダムに選ばれたポケモンが勝負をするコードを実装せよ。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 匹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。各ポケモンインスタンスに対して、乱数によりレベルを設定する。そして、コマンドライン引数で指定した番号のポケモンを自分のポケモンとし、リストからランダムに選ばれた敵と戦う。レベルと種族値の合計が高いほうが勝ちとなる。

採点の都合上、以下の要件を満たすこととする：

- Monster クラスを定義する；
 - － イニシャライザを以下のように定義する；
 - * 名前 (title), 分類, タイプを設定する；
 - * 種族値リストを設定する；
 - * 名前マングリングしたレベル用の変数を 0 に初期化する；
 - － レベル用の変数の値を取得、設定するプロパティ（ゲッターとセッター）を定義する。セッターでは、負の値が与えられた時に、図 2 のように「不正の値のため、符号を反転して設定します」と出力するとともに、符号を反転させた正の値を設定する；
 - － インスタンスを文字列化した際に、図 2 のように名前 (title) とレベルが出力されるように特殊メソッドをオーバーライドする；
 - － インスタンスを大小比較した際に、種族値の合計とレベルの和により大小を定義するように特殊メソッドをオーバーライドする；
- battle.py には変更を加えない；
- 名前ファイル「poke_names.txt」と種族値ファイル「base_stats.txt」を読み込む関数は、リスト 3 を使用する (Moodle にある pokemon.py にすでに書かれている)；
- 出力の形式は、図 2 のようにする；

```
PS C:\Users\admin\Desktop\ProA2> python .\lec02\kadai04\battle.py .\lec02\data\poke_names.txt .\lec02\data\base_stats.txt 121
不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します

不正な値のため、符号を反転して設定します
不正な値のため、符号を反転して設定します
僕のポケモン： スターミー (Lv.88)
vs アズマオウ (Lv.4)
スターミー (Lv.88) の勝ち！
vs シェルダー (Lv.18)
スターミー (Lv.88) の勝ち！
vs アーボック (Lv.27)
スターミー (Lv.88) の勝ち！
vs バンギラス (Lv.40)
スターミー (Lv.88) の負け(>_<)
vs ヤドキング (Lv.21)
スターミー (Lv.88) の勝ち！
```

図 2 課題 04 の実行例：【上側】レベル用の乱数が負の値になった時の出力である。期待値として 25 回くらい発生する／【下側】121 番のスターミーを自分のポケモンとして指定している。ランダムに選ばれた 5 匹のポケモンと戦い、4 勝 1 敗の状態である。

リスト 2 battle.py

```
1 from pokemon import *
2 from random import randint, choice
3 import sys
4
5 if __name__ == "__main__":
6     names = read_names(sys.argv[1]) # .\lec02\data\poke_names.txt
7     stats = read_stats(sys.argv[2]) # .\lec02\data\base_stats.txt
8     monsters = [Monster(n, s) for n, s in zip(names, stats)]
9     for mon in monsters:
10         mon.level = randint(-10, 100) # 全ポケモンに対して、レベルを乱数で設定する
11     ## ここまでが準備
12
13     my_monster = monsters[int(sys.argv[3])-1] # 自分のパートナーポケモン
14     print("僕のポケモン: ", my_monster)
15
16     for _ in range(5):
17         teki = choice(monsters)
18         print("vs", teki)
19         if my_monster > teki:
20             print(my_monster, "の勝ち!")
21         else:
22             print(my_monster, "の負け(>_<)")
```

リスト 3 pokemon.py

```
1 class Monster:
2     課題
3
4
5 def read_names(file_path):
6     names = []
7     with open(file_path, "r", encoding="utf8") as rfo:
8         for row in rfo:
9             _, tit, spe, typ = row.rstrip().split("\t")
10             names.append([tit, spe, typ])
11     return names
12
13 def read_stats(file_path):
14     stats = []
15     with open(file_path, "r") as rfo:
16         for row in rfo:
17             row = row.rstrip()
18             stats.append([int(col) for col in row.split(" ")])
19     return stats
```