プログラミングA2
第6回

練習問題解答例

練習問題：count_char.py

デフォルト値付き辞書を用いて，ポケモンの名前として使われる文字をカウントするコードを実装せよ．

すなわち，文字が**キー**，出現回数が**値**となる．

count_char.py

```
from collections import defaultdict

titles = read_names("poke_names.txt")
chr_counts = defaultdict(int)
for title in titles:
    for t in title:    ← 1文字ずつ
        chr_counts[t] += 1
pprint(chr_counts)
```

実行例

```
defaultdict(<class 'int'>,
            {'2': 1, '♀': 1, '♂': 1, 'ァ': 2, 'ア': 15, 'ィ': 13,…
```

練習問題：stat_tuple.py

base_stats.txtを読み込み，1匹の種族値を1つの名前付きタプルStatで表し，それらを格納したリストを作成せよ．

つまり，namedtupleが251個並んだリストを作成する．

stat_tuple.py

```
from collections import namedtuple
if __name__ == "__main__":
    stats = read_stats("base_stats.txt")
    field_names = ["H", "A", "B", "C", "D", "S"]
    tpls = []
    for stat in stats:
        tpl = namedtuple("Stat", field_names)(*stat)
        tpls.append(tpl)
    pprint(tpls)
```

実行例

```
[Stat(H=45, A=49, B=49, C=65, D=65, S=45),
 Stat(H=60, A=62, B=63, C=80, D=80, S=60),
 Stat(H=80, A=82, B=83, C=100, D=100, S=80),…
```

練習問題：stat_dataclass.py

base_stats.txtを読み込み，1匹の種族値を1つのデータクラスStatで表し，それらを格納したリストを作成せよ．

つまり，dataclassが251個並んだリストを作成する．

stat_dataclass.py

```
from dataclasses import dataclass
@dataclass
class Stat:
    H: int
    省略
    S: int

if __name__ == "__main__":
    stats = read_stats("base_stats.txt")
    dcls = []
    for stat in stats:
        dcl = Stat(*stat)
        dcls.append(dcl)
    pprint(dcls)
```

実行例

```
[Stat(H=45, A=49, B=49, C=65, D=65, S=45),
 Stat(H=60, A=62, B=63, C=80, D=80, S=60),
 Stat(H=80, A=82, B=83, C=100, D=100, S=80),…
```

おまけ問題：stat_dataclass.py

課題10のShuzokuchi.pyのように，__add__()と__sub__()を定義し，Statインスタンス同士を「+」と「-」で算術演算できるようにせよ．

stat_dataclass.py

```
from dataclasses import dataclass
@dataclass
class Stat:
    def __getitem__(self, key):
        return self.__dict__[key]

    def __add__(self, other):
        return __class__(**{key: self[key]+other[key]
                            for key in self.__dict__})

    def __sub__(self, other):
        return __class__(**{key: self[key]-other[key]
                            for key in self.__dict__})
```

←なぜ定義している？何に必要？

クラス自体（＝Stat）↑
インスタンスを生成している

※__getitem__()を定義しなかった場合はどうするのだろう？

練習問題：type_counter.py

poke_names.txtからタイプを読み込み，
タイプのペアをカウントするコードを実装せよ．

タイプが1つの場合もカウントすること．

そして，出現回数が多い上位5件を表示せよ．

[要件]

- ファイルを読み込み，タイプの集合を要素としたリストを返すread_types()関数を定義し，使用する
- 上記のリストに基づき，Counterインスタンスを生成する
- Counterクラスのインスタンスメソッドを用いて上位5件を表示する

実行例

```
[(frozenset({'みず'}), 25),
 (frozenset({'ノーマル'}), 23),
 (frozenset({'ほのお'}), 16),
 (frozenset({'でんき'}), 12),
 (frozenset({'ノーマル', 'ひこう'}), 10),…
```

解答例：type_counter.py

type_counter.py

```
from pprint import pprint
from collections import Counter

def read_types(file_path):
    types = list()
    with open(file_path, "r", encoding="utf8") as rfo:
        for row in rfo:
            _, _, typ, *_ = row.rstrip().split("¥t")
            types.append(frozenset(typ.split()))
    return types #タイプ文字列のfrozensetが並ぶリスト


if __name__ == "__main__":
    types = read_types("lec06/data/poke_names.txt")
    typ_cnt = Counter(types)
    pprint(typ_cnt.most_common(5))
```