【課題 25：counte_char.py】

配布した poke_names.txt に書かれたポケモンの名前に使用されている文字をカウントするコードをマルチスレッドにより実装せよ．すなわち，名前として使用される文字に対して，その出現回数を要素とする辞書を作成する．

採点の都合上，以下の要件を満たすこととする：

(1) 読み込みファイル（poke_names.txt）のパスをコマンドライン引数で指定する；
(2) threading モジュールの Thread オブジェクトを使用する；
(3) セマフォを 10 に設定する；
(4) 各文字とその出現回数を集計した辞書をソートし，出力結果が図 1 になるようにする；

```
集計結果：(80種) [('ー', 102), ('ン', 76), ('ラ', 46), ('リ', 43), ('ド', 35), ('イ', 34), ('ル', 33), ('ス', 31), ('ッ', 29), ('ク', 27), ('マ', 27), ('ト', 25), ('ウ', 24), ('タ', 23), ('ニ', 21), ('キ', 21), ('コ', 19), ('オ', 18), ('シ', 17), ('カ', 16), ('ア', 15), ('グ', 15), ('フ', 14), ('ュ', 14), ('ゴ', 14), ('ガ', 13), ('サ', 13), ('ロ', 13), ('ィ', 13), ('ブ', 13), ('ダ', 12), ('バ', 12), ('ビ', 12), ('ギ', 11), ('ナ', 11), ('チ', 11), ('ム', 11), ('ノ', 10), ('プ', 10), ('デ', 10), ('ツ', 10), ('メ', 9), ('ボ', 9), ('パ', 9), ('レ', 9), ('ワ', 9), ('ゲ', 8), ('ハ', 8), ('ヤ', 8), ('エ', 8), ('ャ', 7), ('ピ', 7), ('ョ', 7), ('ポ', 7), ('ミ', 7), ('ネ', 6), ('モ', 6), ('テ', 6), ('ペ', 6), ('ヒ', 5), ('ジ', 5), ('ズ', 5), ('ザ', 4), ('セ', 4), ('ホ', 4), ('ソ', 3), ('ゾ', 3), ('ォ', 2), ('ケ', 2), ('ェ', 2), ('ァ', 2), ('ヨ', 2), ('ヌ', 2), ('ヘ', 2), ('ゼ', 1), ('♀', 1), ('♂', 1), ('ペ', 1), ('ユ', 1), ('2', 1)]
```

図 1　課題 25 の実行結果

リスト 1　count_char.py

```python
import collections
import sys

def read_names(file_path):
    with open(file_path, encoding="utf8") as rfo:
        names = list()
        for row in rfo:
            col = row.rstrip().split("\t")
            names.append(col[1])
    return names


def counter(cnt_dct, name):
    for n in name:
        cnt_dct[n] += 1


if __name__ == "__main__":
    names = read_names(sys.argv[1]) # "lec11/data/poke_names.txt"
    cnt_dct = collections.defaultdict(int)
    for name in names:
        counter(cnt_dct, name)

    sorted_dct = sorted(cnt_dct.items(), key=lambda item:item[1], reverse=True)
    print(f"集計結果：({len(sorted_dct)}種)", sorted_dct)
```

【課題 26：monte_sample.py】

モンテカルロサンプリングにより，円周率を求めるコードをマルチプロセスにより実装せよ．今回のモンテカルロサンプリングでは，0 以上 1 未満の乱数ペアを多数発生させ，半径 1 の円の内側の点を表す乱数ペア（乱数ペアの 2 乗和が 1 未満のもの）をカウントする．そして，円の内側の点が全乱数ペアのうち何割あったかにより円周率を求める．わからない場合は，`https://manabitimes.jp/math/1182` を参照すること．

採点の都合上，以下の要件を満たすこととする：

(1) プログラム全体の所要時間を計測し，最後に所要時間を出力する：
(2) トータルで，100,000,000 個の乱数によるモンテカルロサンプリングを行う；
(3) これを 1000 個のタスクに分割する；
(4) multiprocessing モジュールの Pool オブジェクトを使用し，同時実行するプロセス数を制御する；
(5) プロセス数は，コマンドライン引数により指定する；
(6) map メソッドにより以下の sampling 関数を実行し，return された結果リストを受け取る；
(7) 結果リストを集計し，円周率を算出する；
(8) 出力結果が図 2 になるようにする；
(9) sampling 関数を定義する；
- 生成する乱数ペアの個数を表すパラメータ num を持つ；
- 横座標 x と縦座標 y のそれぞれを表す乱数を num 回生成する；
- 条件（$x^2 + y^2 < 1$）を満たす乱数ペアをカウントし，集計結果を return する；

```
PS C:\Users\admin\Desktop\ProA2> python .\lec11\kadai26\monte_sample.py 1
314171992.0/100000000 = 3.14171992
所要時間：33.01
PS C:\Users\admin\Desktop\ProA2> python .\lec11\kadai26\monte_sample.py 10
314130736.0/100000000 = 3.14130736
所要時間：10.01
```

図 2　課題 26 の実行結果

リスト 2 monte_sample.py

```
1  import multiprocessing
2  import random
3  import sys
4  import time
5
6  def sampling(num):
7      return cnt
8
9  if __name__ == "__main__":
10     total = 100000000
11     num = int(total/1000)
12     nums = [num for _ in range(1000)]
```