プログラミング演習 I（平成 29 年度）
– レポート課題 –【10/26 修正版】

担当: 時井，松村

提出期限 平成 29 年 11 月 13 日 (月) 15 時

プログラム作成上の注意

これまでプログラムの 1 行目に書いていた `#encoding: Shift_JIS` を`#encoding: Windows-31J` に変更してください．日本語入力を伴うプログラムの際に必要になります．

課題 1

次の 3 つの機能からなる面積を比較するプログラム

1. 名前とその面積をキーボードから入力させ，ハッシュに格納する．
2. 1. で入力された全ての名前とその面積，および筑波キャンパスに対する面積比を表示する．
3. 2 つの名前を入力すると，それらの面積を比較する．

プログラムの条件

1. 名前とその面積のデータをハッシュとして蓄積する部分
   - 名前をキーとし面積を値とするハッシュ（ハッシュ名 `place`）を作成する．ただし，このハッシュには`"筑波キャンパス"`をキーとして，その面積 2577286 をあらかじめ登録しておく．
   - 面積の単位は m^2 とする．
   - 名前，面積をキーボードから入力できるようにし，ハッシュ`place` に蓄積する．
   - 入力は `while` を使った繰り返し文で記述し，名前と面積は最大 100 組入力できるようにする．
   - 名前として.（ピリオド）を入力すると，ハッシュに蓄積するループを抜ける．
   - 既にハッシュに存在する名前を入力したときは，その名前と対応する面積を表示し，上書きするかそのままにするかを`"y"`（上書きする）か`"n"`（上書きしない）で選択できるようにする．
   - それぞれの操作に対して，適切なメッセージを表示する．
2. 名前と面積の表示部分
   - 以下の 2 つのメソッドを定義して使う．
     (a) 2 つの名前 (`place1`，`place2`) と名前と面積の対応を格納したハッシュ（`place`）を引数とし，その 2 つの名前についての面積の比率（`place2` の面積に対する `place1` の面積の割合）を**返す**メソッド `get_ratio`

(b) ハッシュ(place) を引数とし，ハッシュに蓄積された全ての名前と対応する面積，筑波キャンパスの面積を 1 とした場合の割合を**表示する**メソッド display_place_list．ただし，割合を求める際にメソッド get_ratio を利用すること．なお，筑波キャンパスの面積に対する割合の単位は ITF. とする．

- 表示はメソッド display_place_list を適切に使って行う．

3. ２つの名前を入力し，面積を比較する部分

- ２つの名前を入力できるようにする．
- ２つの名前を入力する部分は while を使った繰り返し文で記述し，最大 100 回比較できるようにする．
- 名前のどちらかに.（ピリオド）が入力されると比較するループを抜けてプログラムを終了する．
- ２つの面積を比較して，大きい方が小さい方の何個分かを表示する．ただし，比較する際に，get_ratio を使って返された値を利用して条件分岐をすること．

実行結果　以下の入力をキーボードで行うことを想定したファイル rep1_data.txt を提供するのでそれを使って実行する．

1. 以下の表の名前と面積の対応をハッシュに格納する．ただし，あとで上書きするために，皇居のデータは間違った値を入れておく．

名前	面積（m^2）	
銀河系の円盤部	7×10^{41}	
山手線内	63,000,000	
鳥取砂丘	5,500,000	
牛久沼	3,490,000	
筑波キャンパス	2,577,286	（＜＝このデータだけはプログラム中で登録しておく）
皇居	1,420,000	（＜＝このデータは後で上書きするため間違った値を入れる）
東京ディズニーランド	520,000	
バチカン市国	440,000	
東京ドーム	46,755	
iPad	0.0448	
10 円玉	0.00043	

2. 既にハッシュに存在する名前を入力した場合の機能が正しく動いていることを示すため，もう一度，"皇居"を入力し，"y"を選択して上書きする．
3. "."を入力して，ハッシュへの格納を終了する．
4. 面積比較では，以下の 4 通りを試す．

- 筑波キャンパスと東京ドーム
- 東京ディズニーランドと鳥取砂丘
- iPad と 10 円玉
- 筑波キャンパスと iPad

5. "."を入力して，比較を終了する．

実行結果（例）

```
Z:\progI> ruby s1799999-1.rb < rep1_data.txt
● データをハッシュに入れます.
名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 面積（m^2）を入力してください> 名前を入力してください> 皇居:14200.0
面積を変更しますか（y/n）>面積（m^2）を入力してください> 名前を入力してください>

● ハッシュに蓄積された全ての名前と面積を表示します.
名前　面積（割合）
筑波キャンパス　2577286 m^2（1 ITF.）
銀河系の円盤部　7.0e+41 m^2（2.7160353953732725e+35 ITF.）
山手線内　63000000.0 m^2（24.444318558359452 ITF.）
鳥取砂丘　5500000.0 m^2（2.1340278106504282 ITF.）
牛久沼　3490000.0 m^2（1.3541376471218174 ITF.）
皇居　1420000.0 m^2（0.5509671802042925 ITF.）
東京ディズニーランド　520000.0 m^2（0.20176262937058595 ITF.）
バチカン市国　440000.0 m^2（0.17072222485203428 ITF.）
東京ドーム　46755.0 m^2（0.01814117641581105 ITF.）
iPad　0.0448 m^2（1.7382626530388944e-08 ITF.）
10 円玉　0.00043 m^2（1.668421742872153e-10 ITF.）

● 面積の比較をします.
名前を２つ入力してください.
名前 l > 名前２> 筑波キャンパスは東京ドーム 55.12321676826008 個分です.
名前を２つ入力してください.
名前 l > 名前２> 鳥取砂丘は東京ディズニーランド 10.576923076923077 個分です.
名前を２つ入力してください.
名前 l > 名前２> iPad は 10 円玉 104.18604651162791 個分です.
名前を２つ入力してください.
名前 l > 名前２> 筑波キャンパスは iPad 57528705.35714286 個分です.
名前を２つ入力してください.
名前 l >
Z:\progI>
```

課題２

山手線の降車駅を入力すると，下記の情報を表示するプログラムを作成する．

[**表示する情報**]

1. 東京駅から降車駅までの駅名一覧 (外回り）
2. 所要時間（外回り）

[**プログラムの条件**]

- 駅名を保存する配列に関して
  - 山手線外回りの駅名を保存する配列を準備する．
  - 外回りの停車駅を順に保存する配列名は，`eki_soto` とする．
  - 東京, 有楽町, .., 秋葉原, 神田の順に保存する．
- 所用時間を保存する配列に関して
  - 山手線外回りの各駅間の所用時間（分）を保存する配列を準備する．
  - 外回りの停車駅を順に保存する配列名は， `jikan_soto` とする．
  - 東京と有楽町，有楽町と新橋， .... 秋葉原と神田，神田と東京の所要時間の順に保存する．
- 降車駅名をキーボードから入力し，変数に保存する．
- 入力された降車駅が駅名配列に存在するか判定し，true または false を返すメソッドを作成する．
  - 引数は，降車駅名と駅名配列とする
  - 降車駅名が駅名配列に存在しない場合は，false を返す．
  - 降車駅名が駅名配列に存在する場合は，true を返す．
- 降車駅名が配列に存在するか判定するメソッドを呼び，true または false によって下記の処理を行う
  - true がかえってきた場合 (降車駅名が存在する場合) は，東京駅から降車駅までの駅名一覧表示と所要時間を計算し表示するメソッドを呼ぶ．
  - false がかえってきた場合は，駅名が存在しない旨を表示する
- 東京駅から降車駅までの駅名一覧表示メソッド
  - 引数は，降車駅名，駅名配列，所要時間配列とする．
  - 降車駅が何番目にあるか探索し，変数にその数を保存する．
  - 降車駅が何番目にあるかを表示する．
  - 山手線外回りの東京駅から降車駅間の駅名と所要時間を計算し表示する．
- 駅名と所用時間は下記のデータを使うこと (授業ページに掲載予定)
  - 外回り:駅名

```
["東京", "有楽町", "新橋", "浜松町", "田町",
"品川", "大崎", "五反田", "目黒", "恵比寿",
"渋谷", "原宿", "代々木", "新宿", "新大久保",
"高田馬場", "目白", "池袋", "大塚", "巣鴨",
"駒込", "田端", "西日暮里", "日暮里", "鶯谷",
"上野", "御徒町", "秋葉原", "神田"]
```

– 外回り所要時間 (分) 東京 ↔ 有楽町, 有楽町 ↔ 新橋, ... 秋葉原 ↔ 神田, 神田 ↔ 東京の順

```
[2, 2, 2, 2, 3,
2, 2, 2, 3, 2,
3, 2, 2, 2, 3,
1, 3, 3, 1, 2,
3, 1, 2, 1, 2,
2, 2, 2, 2]
```

- 実行結果としては，東京，田町，うえの，神田を示すこと．(実行結果表示例参照)

[実行結果表示例]

```
Z:\progI>ruby s1199999-2.rb
降車駅名を入力してください：東京
東京は，0 個めの駅です
======================
[外回り]
東京
乗車時間は，0 分です
======================

Z:\progI>ruby  s1199999-2.rb
降車駅名を入力してください：田町
田町は、4 個めの駅です
======================
[外回り]
東京==>有楽町==>新橋==>浜松町==>田町
乗車時間は、8 分です
======================

======================
Z:\progI>ruby  s1199999-2.rb
降車駅名を入力してください：うえの
駅名がありません

Z:\progI>ruby  s1199999-2.rb
降車駅名を入力してください：神田
神田は，28 個めの駅です
======================
[外回り]
```

東京==>有楽町==>新橋==>浜松町==>田町==>品川==>大崎==>五反田==>目黒==>恵比寿==>渋谷==>原宿==>代々木==>新宿==>新大久
保==>高田馬場==>目白==>池袋==>大塚==>巣鴨==>駒込==>田端==>西日暮里==>日暮里==>鶯谷==>上野==>御徒町==>秋葉原==>神田
乗車時間は，59 分です
======================

Z:\progI>

プログラムの構成

```
#encoding: Windows-31J
# 駅名配列
# 乗車時間配列
# 入力された降車駅名があるかどうか判定するメソッド，true または false を返す
# 降車駅名までの駅名と乗車時間を表示するメソッド

  # 降車駅が何番目にあるかを探索し，変数にその数を保存

  # 東京駅から降車駅まで表示（何番目にあるか保存した変数を使う）

# 降車駅名を入力，駅名判定メソッドを呼ぶ.
# 駅名があった場合は表示メソッドを呼ぶ
```

[**下記のように段階をおって，課題の要件を全て満たすようにするとよい**]

1. 東京から神田までの駅一覧と東京から神田までの所用時間（分）を計算する.
2. 降車駅を入力し，駅名配列に含まれるかを探索する.
3. 降車駅が何番目にあるかを変数に保存する.
4. 降車駅までの駅名と所要時間を計算し表示する.
5. 要件を満たすメソッドを作成する.

[**さらに上を目指す方**]

- 乗車駅を入力できるようにする方は，メソッドの引数に乗車駅名も加えてよい.
- 内回りも表示する方は，外回りと内回りはひとつのメソッドで行う.

提出における注意

- 締切：締切を厳守すること．締切に遅れたレポートは受理しない．
- 提出物：紙，プログラム
- 提出場所：(紙）学務課レポート提出用ポスト,(プログラム）progI-ml@klis.tsukuba.ac.jp
  自分の受講クラスのポストに入れること．他のクラスのポストに入れた場合は受理しない．
- プログラムについて
  - 作成したプログラム ***.rb を電子メールに添付して送付する．
  - ファイル名は，ユーザ名-課題番号.rb とする．すなわち，sXXXXXXX-1.rb，sXXXXXXX-2.rb の二つである．
  - 宛先は，progI-ml@klis.tsukuba.ac.jp
  - 件名は，rep1
- 紙について以下を遵守しないレポートは減点の対象となる．
  - 設問毎に A4 用紙にまとめ左上をステープラ（ホチキス）でとめる．(今回は設問が 2 個あるので，合計 2 部提出)
  - 設問毎に表紙，本文の順でとめる．
    * 表紙： 科目名，曜日クラス，第 1 回レポート　設問番号，提出日，学籍番号，氏名を記入する．見本を Web ページに掲載するので，その形式にしたがうこと．
    * 本文： 行番号付きプログラムリスト，実行結果とプログラムの説明を この順で 載せる．
      プログラムリストと実行結果は続けて記述しても良いが，プログラムの説明は別ページとすること．
      いずれも機械出力とする．その際，プログラムリストと実行結果は，等幅フォント とする．
      出力したプログラムや実行結果に手書きしないこと．
      なお，付録に形式の一例を載せたので参考にしてほしい．
  - 実行結果に関する注意事項
    * 実行は，全て全学計算機システムの Windows 上で行うこと．
    * 実行 (例 ruby ***.rb)，出力される実行結果の順となるように，一連の操作の出力を記述する．(注意：余分な操作を途中で入れないこと）
    * 実行結果は必要に応じて複数示すこと．
  - 可読性が良くなるよう努めること.「プログラム」は適切な字下げと空行の挿入をし,「説明」は適切な見出しを付けるなど，構造がわかるように注意すること．
  - 片面印刷にすること．

その他の注意

- 同一/類似レポートは「両成敗」．すなわち，見た方も見せた方も不正行為とみなし，単位を出さない．また，他の科目にも影響することもある．他人が不正にプログラムにアクセスできないよう，各自の責任において対処すること．
- レポートの差し替えや再提出は認めないので，内容をよく確認してから提出すること．
- レポートを書く上での一般常識として，参考文献がある場合には書誌情報を載せること．

- フォーマットが細かく設定されているのは決して意地悪からではない．100 名を越える受講者のレポートを効率的かつ公平に採点するために必要なことだからである．

  言い替えれば，フォーマットを無視したレポートは演習担当者の作業を著しく妨げることになる点を理解してほしい．

(付録) 行番号付きプログラム等の形式の例

以下は一例であるので，各自で考えてより読みやすい形式に整えること．

設問 1：球の体積計算プログラムを作成する．

```
---------------------------------------
プログラムリスト
---------------------------------------
 1  print("球の半径を入力してください\n")
 2  r = gets.chomp.to_f
 3  v =
 4
 .......................

---------------------------------------
実行結果
---------------------------------------
Z:\> ruby s9911999-1.rb
 .......................
 .......................

Z:\> ruby s9911999-1.rb
 .......................
 .......................
```

```
---------------------------------------
プログラムの説明
---------------------------------------
 .......................
.......................
.......................
```