

おまけ

0.1 クラスの応用

凡例：class-X

ここで学生個人のデータを管理することを考えましょう。

学生は学生番号と履修科目と得点を管理しています。履修科目には必須科目（数学、英語、国語）とコース別の Acourse（絵画、音楽）、Bcourse（会計、情報）の選択必須科目があり、どちらかを選択しなければならないことを考えます。また、学生の管理するデータを入力することと、学生ごとの合計点を計算する機能が必要だとします。

まずは必須科目を管理する機能と必須科目だけを5人分入力し、それぞれの合計点を出力するプログラムを作成します。これをクラスとしてまとめるため、共通科目だけを扱う機能を次のように記載します。

一つのクラス

```
1  class Student: #共通科目を取り扱うためのクラス
2      std_subjects = ["Mathematics" ,"English","Japanese"]
3      def input_number(self):
4          self.number = input( "number = " )
5      def input_scores(self):
6          self.scores = []
7          input_data(self.std_subjects,self.scores)
8          #データを入力するための手続きを関数としてまとめておきました。
9          #エラー処理およびテキスト入力された得点を数値に変換する手続き
10
11     def sum_of_scores(self):
12         self.summary = 0
13         for score in self.scores:
14             self.summary = self.summary + score
15         return self.summary
16
17     # クラスの宣言はここまで
18     #ここでインデントを戻す
19     def input_data(subjects,scores):
20         for s in subjects:
```

```

21         x = input( s + " = " )
22         y = int ( x )
23         if y > 100 or y < 0 :
24             y = -1
25             print ( "エラー" )
26         scores.append( y )
27
28     #グローバル変数（リスト）中身は Student クラスの実体を格納する
29     student_list = []
30
31     #Student クラスの実体を 5 名分追加
32     for n in range(5):
33         student_list.append( Student() )
34
35     #学生の番号と得点の入力と合計点の出力
36     for std in student_list:
37         std.input_number()
38         std.input_scores()
39         print(std.sum_of_scores())

```

凡例：class-2

凡例 class-2

凡例 class-1 の共通機能を基にして、Acourse、Bcourse の選択必須科目の機能を追加します。

Acourse の学生 3 名と Bcourse の学生 2 名の得点を入力して（各コースの順番はバラバラでも可能）それぞれの学生の合計点と Acourse を選択している学生の平均点、Bcourse を選択している学生の平均点を計算するプログラムです。

クラスの継承

```

1
2     class Student: #共通科目を取り扱うためのクラス
3         std_subjects = ["Mathematics" ,"English","Japanese"]
4         def input_number(self):
5             self.number = input( "number = " )
6         def input_scores(self):
7             self.scores = []

```

```

8         input_data(self.std_subjects,self.scores)
9         #データを入力するための手続きを関数としてまとめておきました。
10        #エラー処理およびテキスト入力された得点を数値に変換する手続き
11
12        def sum_of_scores(self):
13            self.summary = 0
14            for score in self.scores:
15                self.summary = self.summary + score
16            return self.summary
17
18    def input_data(subjects,scores):
19        for s in subjects:
20            x = input( s + " = " )
21            y = int ( x )
22            if y > 100 or y < 0 :
23                y = -1
24                print ( "エラー" )
25            scores.append( y )
26    #ここまでは凡例「class-1」の24行目までと同一です。
27
28    class AcourseStudent(Student):
29        acourse_subjects = ["picture","music"]
30        def input_scores(self):
31            super().input_scores()
32            input_data(self.acourse_subjects,self.scores)
33    class BcourseStudent(Student):
34        bcourse_subjects = ["accounting","information"]
35        def input_scores(self):
36            super().input_scores()
37            input_data(self.bcourse_subjects,self.scores)
38
39    student_list = []
40    #凡例 class-1 の学生の追加部分を次のように書き換え
41    student_list.append( AcourseStudent() )
42    student_list.append( AcourseStudent() )

```

```
43  student_list.append( BcourseStudent() )
44  student_list.append( BcourseStudent() )
45  student_list.append( AcourseStudent() )
46
47  #学生の番号と得点の入力と合計点の出力
48  for std in student_list:
49      std.input_number()
50      std.input_scores()
51
52  for std in student_list:
53      print("number = " , std.number , " summary = " , \
54          std.sum_of_scores())
```

おまけ python による gui プログラミング (冬休みの宿題)

```
pip install pyqt5
```

凡例 : class-2

```
1  # pylint: disable=missing-docstring
2  # pylint: disable=no-name-in-module
3  # pylint: disable=unused-import
4  import sys
5  from PyQt5.QtWidgets import (QWidget, QPushButton, QApplication)
6
7  class Example(QWidget):
8      def __init__(self):
9          super().__init__()
10         self.btn = QPushButton(' こんにちは', self) #ボタンの生成
11         self.btn.clicked.connect(self.push)
12         self.btn.setGeometry(0,0,300,200)
13         self.setGeometry(300, 300, 300, 200)
14         self.show()
15
16     def push(self):
17         self.btn.setText("押されました")
18
19 app = QApplication(sys.argv)
20 ex = Example()
21 sys.exit(app.exec_())
```