

5.3 集合和字典的应用

- ◎ 集合和字典是非常有用的容器，善用它们往往可以事半功倍。

例5-4 处理学生成绩

输入：(以END结束)

- 学生的学号，姓名
- 学生的学号，课程，成绩

输出：

- 表头 学号，姓名，课程1，课程2，...，平均分
- 内容 <学号>，<姓名>，<成绩1>，<成绩2>，...，<平均分>

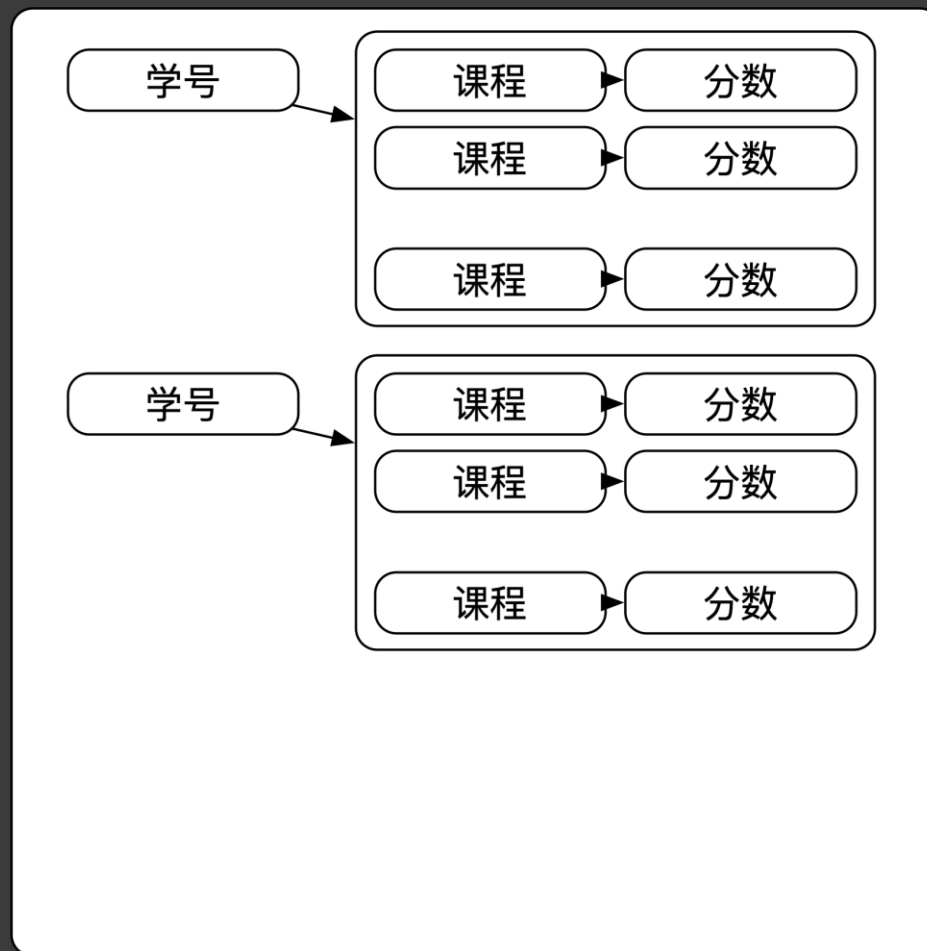
数据结构

◎ 字典：

- studentid，键是学号，值是姓名。
- score，键是课程，值是成绩
- studentscore，键是学号，值是字典score。

◎ 集合course，课程

数据结构（续）



过程

◎ 输入数据

- 不断循环地读入每一行，如果读入的是END，则循环结束。
- 如果是学号、姓名关系，那么第一个字符串应该都是数字，而如果是成绩，应该最后一个字符串是数字。

while True:

 line = input()

 if line == 'END':

 break

 words = line.split() #以空格分割输入行

 if words[-1].isnumeric(): #最后的字符串是数字

 ...

 else:

 ...

过程（续）

- 如果输入的是学号姓名关系：加入studentid字典：
`studentid[words[0]] = words[1]`
- 如果输入的是成绩：要把课程、成绩关系加入到那个学号所对应的学生的成绩字典里。
- 如果那个学生之前还没有输入过成绩，还要新建那个字典：

```
score = studentscore.get(word[0])
```

```
if score == None:
```

```
    score = {}
```

```
score[words[1]] = words[2]
```

```
studentscore[word[0]] = score
```

- 同时还需要把这门课程的名字加入课程的集合：

```
course.add(words[1])
```

过程（续2）

◎ 输出数据

- 首先要输出表头

```
coursename = list(course)
print(',', end='')
for name in coursenam:
    print(','+name, end='')
print()
```

过程（续3）

- 然后要遍历学号、姓名字典，找出所有的学号和姓名，同时找出每个学生的所有成绩。

```
for id in studentid.keys():  
    print(id + ',' + studentis[id], end="")  
    score = studentscore[id]  
    sum = 0  
    cnt = 0  
    for name in coursename:  
        print(',', end="")  
        if name in score:  
            print(score[name], end="")  
            sum += int(score[name])  
            cnt += 1  
    print(',', str(int(sum/cnt)))
```

各种数据类型汇总

数据类型	可变性	举例
整数	不可变	5
浮点数	不可变	6.7
复数	不可变	3+89j
字符串	不可变	"hello"
逻辑量	不可变	True
列表	可变	["mon",2,3,4,5]
元组	不可变	(7,8,9)
集合	可变	{1,4,6}
字典	可变	{1:"jan",2:"feb",3:"mar"}

本章小结

- ◎ 集合和字典是两种数据容器。
- ◎ 集合中没有重复的数据，数据没有位置和顺序，不能用索引来存取。集合可以做交、并、差等运算。
- ◎ 字典是用键来存取数据的，数字、字符串和元组都可以作为键。键所对应的值可以是任何类型，数字、字符串甚至列表和字典都可以作为值存放在字典中。