实验四 字典与集合

姓名：马星 学号：5418122020 班级：计算机科学与技术(卓越工程师计划)221班

一、实验目的

1. 了解散列表的实现方法以及散列表的应用；
2. 掌握字典和集合类型的使用方法；
3. 了解字典和集合的典型应用，散列类型与序列类型的异同。

二、实验内容

1. 练习字典和集合对象的创建方法。
2. 完成教材118页的课后练习第1、2题

三 实验步骤

1. 练习字典和集合对象的创建方法。

```
d1 = {}  # 创建空字典
d2 = {"a": 1, "b": 2, "c": 3}  # 创建带有初始数据的字典
d3 = dict([("a", 1), ("b", 2), ("c", 3)])  # 使用包裹元组的迭代对象进行创建
d4 = dict(zip(["a", "b", "c"], [1, 2, 3]))  # 使用zip函数关联两个列表
print(f"d1的结果为:{d1}")
print(f"d2的结果为:{d2}")
print(f"d3的结果为:{d3}")
print(f"d4的结果为:{d4}")
print("------------------")
s1 = set()  # 创建空集合
s2 = {1, 2, 3, 3, 4, 4, 4}  # 创建带有初始数据的集合,由于集合的不重复性,多个相同的值会被过滤,只保留1个
lis = [1, 2, 3, 3, 4, 4, 4]
s3 = set(lis)  # 通过列表等可迭代对象创建,同理,多个相同的值只保留一个
print(f"s1的结果为:{s1}")
print(f"s2的结果为:{s2}")
print(f"s3的结果为:{s3}")
```

```
d1的结果为:{}
d2的结果为:{'a': 1, 'b': 2, 'c': 3}
d3的结果为:{'a': 1, 'b': 2, 'c': 3}
d4的结果为:{'a': 1, 'b': 2, 'c': 3}
------------------
s1的结果为:set()
s2的结果为:{1, 2, 3, 4}
s3的结果为:{1, 2, 3, 4}
```

2. 教材118页的课后练习第1题
   (1) 创建一个字典dicTXL
   (2) 将dicOther合并进dicTXL
   (3) 增加一列"微信号"dicWX,无微信号默认为手机号

(4) 将"大王"的手机号更改五位13914000004
(5) 输入姓名查找对应通信的手机号或QQ号或微信号,如果不存在则返回"没有该同学的联系方式"

```
def show(dicTXL:dict[str:dict[str:any]]):
    aDic = list(dicTXL.values())[0]
    title = ["姓名"] + [i for i in aDic.keys()]
    for t in title:
        print(f"{t:^12}", end='')
    print()
    for name, dic in dicTXL.items():
        print(f"{name:^12}", end='')
        for value in dic.values():
            print(f"{value:^12}", end='')
        print()


# (1) 创建一个字典dicTXL
print("-------------(1) 创建一个字典dicTXL------------------")
dicTXL = {"小新": {"手机": 1391300001, "QQ": 18191220001},
          "小亮": {"手机": 1391300002, "QQ": 18191220002},
          "小刚": {"手机": 1391300003, "QQ": 18191220003}}
show(dicTXL)
# (2) 将dicOther合并进dicTXL
print("--------------(2) 将dicOther合并进dicTXL------------------")
dicOther = {"大刘": {"手机": 1391400001, "QQ": 18191230001},
            "大王": {"手机": 1391400002, "QQ": 18191230002},
            "大张": {"手机": 1391400003, "QQ": 18191230003}}
dicTXL.update(dicOther)
show(dicTXL)
# (3) 增加一列"微信号",部分微信号dicWX,无微信号默认为手机号
print("-----------(3) 增加一列'微信号',部分微信号dicWX,无微信号默认为手机号--------------
-----------")
dicWX = {"小新": 'xx9907', "小刚": 'gang1004', "大王": 'jack_w', "大刘": 'liu666'}
for name, dic in dicTXL.items():
    if name in dicWX.keys():
        dic["微信"] = dicWX.get(name)
    else:
        dic["微信"] = dic["手机"]
show(dicTXL)
# (4) 将"大王"的手机号更改为13914000004
print("-----------(4) 将'大王'的手机号更改为13914000004-----------------------")
dicTXL["大王"]["手机"]=13914000004
show(dicTXL)
# (5) 输入姓名查找对应通信的手机号或QQ号或微信号,如果不存在则返回"没有该同学的联系方式"
print("-----------(5) 输入姓名查找对应通信的手机号或QQ号或微信号,如果不存在则返回'没有该同学
的联系方式'-------------------------")
name = input("请输入要查找的人的姓名:")
print(f"查询{name}结果: ")
if name in dicTXL:
    print(dicTXL.get(name))
else:
    print("没有该同学的联系方式")
```

```
-------------(1) 创建一个字典dicTXL------------------
```

```
      姓名           手机           QQ
      小新       1391300001 18191220001
      小亮       1391300002 18191220002
      小刚       1391300003 18191220003
--------------(2) 将dicOther合并进dicTXL------------------
      姓名           手机           QQ
      小新       1391300001 18191220001
      小亮       1391300002 18191220002
      小刚       1391300003 18191220003
      大刘       1391400001 18191230001
      大王       1391400002 18191230002
      大张       1391400003 18191230003
-----------(3) 增加一列'微信号',部分微信号dicWX,无微信号默认为手机号----------------------
----
      姓名           手机           QQ           微信
      小新       1391300001 18191220001     xx9907
      小亮       1391300002 18191220002  1391300002
      小刚       1391300003 18191220003    gang1004
      大刘       1391400001 18191230001     liu666
      大王       1391400002 18191230002     jack_w
      大张       1391400003 18191230003  1391400003
-----------(4) 将'大王'的手机号更改为13914000004------------------------
      姓名           手机           QQ           微信
      小新       1391300001 18191220001     xx9907
      小亮       1391300002 18191220002  1391300002
      小刚       1391300003 18191220003    gang1004
      大刘       1391400001 18191230001     liu666
      大王      13914000004 18191230002     jack_w
      大张       1391400003 18191230003  1391400003
-----------(5) 输入姓名查找对应通信的手机号或QQ号或微信号,如果不存在则返回'没有该同学的联系方
式'-------------------------
没有该同学的联系方式
```

3. 教材118页的课后练习第2题

   对每位选手得分去掉一个最高分和一个最低分,按照平均分由高到低输出选手编号和最后得分

```python
player = {"012": [90, 94, 97, 86, 85, 89, 88, 85],
          "005": [91, 91, 92, 98, 90, 96, 90, 95],
          "108": [96, 86, 97, 96, 87, 86, 86, 96],
          "037": [95, 95, 94, 93, 97, 98, 99, 95],
          "066": [95, 87, 94, 94, 93, 99, 96, 97],
          "020": [89, 97, 91, 95, 89, 94, 97, 92]}
lis = [(num, sum(scores) - max(scores) - min(scores)) for num, scores in
player.items()]
lis.sort(key=lambda o: o[1], reverse=True)  #按照得分降序排序
print("排名\t编号\t得分")
for i, t in enumerate(lis):
    print(f"{i + 1}\t{t[0]}\t{t[1]}")
```

```
排名   编号   得分
1    037 574
2    066 569
3    020 558
4    005 555
5    108 547
6    012 532
```

四 实验总结

在这次的Python实验中，我深入学习了字典（Dictionary）和集合（Set）这两种数据结构。通过实践操作和编写代码，我对它们的用法和特点有了更深刻的理解。

字典（Dictionary）：
字典是一种可变的、无序的、键值对（key-value pair）的数据结构。在实验中，我学会了如何创建字典、添加元素、访问元素、修改元素、删除元素以及遍历字典等基本操作。字典的灵活性和高效性让我印象深刻。特别是在处理具有唯一标识符的数据时，字典能够提供非常快速的查找速度。

集合（Set）：
集合是一种可变的、无序的、不重复元素的数据结构。在实验中，我掌握了集合的创建、添加元素、删除元素、求交集、并集、差集等操作。集合的特点在于它的去重功能，这在处理数据时非常有用，比如统计一个列表中不重复的元素数量。

以下是我在实验中的一些具体感悟：

字典的键必须是不可变类型，如字符串、数字或元组，而列表由于是可变的，不能作为字典的键。
字典的方法如get()、keys()、values()等提供了便捷的数据访问方式，使得数据处理更加高效。
集合的去重特性非常适合用于去除列表中的重复项，这在数据分析中是一个常见的需求。
集合之间的操作如union()、intersection()、difference()等可以方便地进行集合间的运算，这些操作在处理集合相关问题时非常有用。
在实际编程中，字典和集合的结合使用可以解决很多复杂问题，例如统计词频、去重计数等。
理解不可变性（immutability）和可变性（mutability）对于选择合适的数据结构至关重要。

通过这次实验，我不仅掌握了字典和集合的基本操作，还学会了如何根据实际问题选择合适的数据结构来解决问题。这对于提高我的编程能力和解决实际问题具有重要意义。