

班级：2022211301

姓名：卢安来

学号：2022212720

- 1、编写函数，实现“对输入英文中单词出现次数统计”的功能。
 - (1) 输入为一个不含标点符号的英文文本。
 - (2) 使用列表解析中所学的方法，将输入的英文文本按单词存储形成一个列表。
 - (3) 统计每个英文单词出现的次数，并构建一个字典，其中key为文中每个出现的单词，value为该单词出现的次数。
 - (4) 返回值：上面构建的字典。

解答：

编写代码如下。

```
def count_word_occurrences(text: str) -> dict[str, int]:  
    """  
    Counts the occurrences of each word in the provided text.  
  
    Args:  
        text (str): The text to count word occurrences in.  
  
    Returns:  
        dict[str, int]: A dictionary with words as keys and their counts as values.  
    """  
    words: list[str] = text.split()  
    return {word: words.count(word) for word in words}
```

2、编写函数，实现“字典合并”的功能。

- (1) 输入：两个字典。
- (2) 对于两个字典中都存在的 key，把它们的 value 算数相加作为结果。
- (3) 对于只有一个字典中存在的 key，直接把该键值对追加到结果字典中。
- (4) 返回值：合并后得到的新字典。

解答：

编写代码如下。

```
def merge_word_counts(dict1: dict[str, int], dict2: dict[str, int]) -> dict[str, int]:  
    """  
    Merges two word count dictionaries by summing the counts of common words.  
  
    Args:  
        dict1 (dict[str, int]): The first word count dictionary.  
        dict2 (dict[str, int]): The second word count dictionary.  
  
    Returns:  
        dict[str, int]: A merged dictionary with summed word counts.  
    """  
    return {  
        key: dict1.get(key, 0) + dict2.get(key, 0) for key in set(dict1) | set(dict2)  
    }
```

3、编写函数，完成“按首字母统计”的功能。

(1) 输入：一个字典，key 为单词，value 为该单词出现的次数。

(2) 按首字母对上述字典中 key 的单词进行分类和统计，建立一个新的字典。其中，key 为 26 个字母（a 到 z），value 为以这个首字母开头的词汇出现的次数。

(3) 返回值：上述按首字母统计次数的字典。

解答：

编写代码如下。

```
def count_words_by_initial(word_count_dict: dict[str, int]) -> dict[str, int]:  
    """  
    Counts the total occurrences of words starting with each letter of the alphabet.  
  
    Args:  
        word_count_dict (dict[str, int]):  
            A dictionary with words as keys and their counts as values.  
  
    Returns:  
        dict[str, int]: A dictionary with initial letters as keys and total counts as values.  
    """  
    return {  
        initial: sum(  
            count  
            for word, count in word_count_dict.items()  
            if word[0].lower() == initial  
        )  
        for initial in {chr(code) for code in range(ord("a"), ord("z") + 1)}  
    }
```

4、编写主调程序，完成如下函数调用和屏幕输出：

(1) 自己任选两篇 500 词左右的英文文章，去除其中的标点符号，单词之间以空格分隔，形成两个独立的字符串 s1 和 s2。

(2) 以 s1 作为输入调用函数 1，得到字典 d1，并打印出 s1 中的单词数量，和 d1 中的 key 的数量。

(3) 以 s2 作为输入调用函数 1，得到字典 d2，并打印出 s2 中的单词数量，和 d2 中的 key 的数量。

(4) 以 d1 和 d2 作为输入，调用函数 2，得到 d3，并打印出 d3 中 key 的数量，以及字典中每个键值对的内容（每个单词及其出现的次数）。

(5) 以 d3 为输入，调用函数 3，得到 d4，并打印出 d4 中的每个键值对的内容。

解答：

编写代码如下。

```
def main(article_path1, article_path2) -> None:
    """
    The main function that reads two text files, cleans the text, counts word occurrences,
    merges the counts, and counts words by initial letters.

    Args:
        article_path1 (str): The path to the first text file.
        article_path2 (str): The path to the second text file.

    Returns:
        None
    """
    try:
        with open(file=article_path1, mode="r", encoding="utf8") as file1:
            s1: str = file1.read()

        with open(file=article_path2, mode="r", encoding="utf8") as file2:
            s2: str = file2.read()

        s1: str = "".join(
            ch if ch.isalnum() or ch.isspace() else " " for ch in s1
        ).lower()
        s2: str = "".join(
            ch if ch.isalnum() or ch.isspace() else " " for ch in s2
        ).lower()

        # Function calls and print statements as per the assignment
        d1: dict[str, int] = count_word_occurrences(text=s1)
        print(f"Total words in s1: {len(s1.split())}")
        print(f"Unique words in d1: {len(d1)}")

        d2: dict[str, int] = count_word_occurrences(text=s2)
        print(f"Total words in s2: {len(s2.split())}")
        print(f"Unique words in d2: {len(d2)}")
        print("")
```

```
d3: dict[str, int] = merge_word_counts(dict1=d1, dict2=d2)
print(f"Unique words in d3: {len(d3)}")
print("Word counts in d3:")
for word, count in d3.items():
    print(f"\t{word}: {count}")
print("")

d4: dict[str, int] = count_words_by_initial(word_count_dict=d3)
print("Initial letter counts in d4:")
for initial, count in d4.items():
    print(f"\t{initial}: {count}")
print("")

except FileNotFoundError:
    print("One of the provided file paths does not exist.")

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Please provide two file paths as command line arguments.")
    else:
        main(article_path1=sys.argv[1], article_path2=sys.argv[2])
```

输入命令 `python main.py text1.txt text2.txt` 即可运行，`text1.txt` 和 `text2.txt` 分别节选自 A Vindication of the Rights of Woman 的 Chapter 1 和 Chapter 2。经检验，运行结果符合预期。