Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
дисциплины «Программирование на Python»

Выполнил:
Щегольков Савва Игоревич
2 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Автоматизированные системы
обработки информации и
управления», очная форма обучения

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема: Работа с множествами и словарями в языке Python.

Цель: приобретение навыков по работе с множествами и словарями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Ссылка на репозиторий: https://github.com/xouixao/lab4

Задание 1: Подсчитать количество гласных в строке, введённой с клавиатуры, с использованием множеств

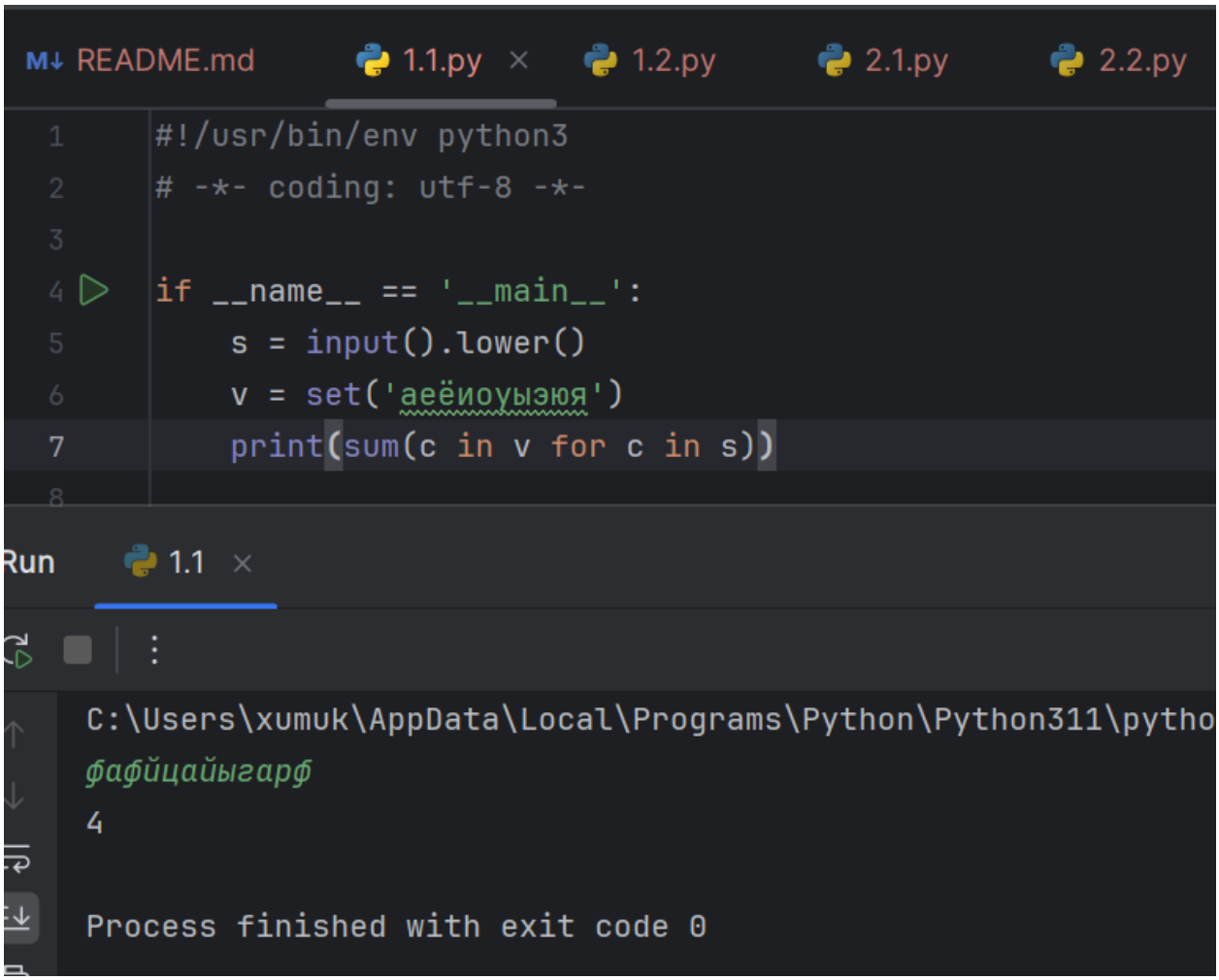

Рисунок 1. Задание 1

Задание 2: определить общие символы в двух строках, введённых с клавиатуры.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    print(set(input()) & set(input()))
```

Run 1.2

```
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\p
abcde
edceqwe
{'d', 'c', 'e'}

Process finished with exit code 0
```

Рисунок 2 - Задание 2.

Задание 3: Создать словарь school, изменить несколько значений, добавить и удалить класс, посчитать общее количество учеников.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    school = {"1A": 25, "1B": 23, "2A": 27, "2B": 26}
    school["1A"] = 26
    school["3A"] = 20
    del school["2B"]
    print(sum(school.values()))
```

Run 1.3

```
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\pyt
96

Process finished with exit code 0
```

Рисунок 3 – Задание 3

Задание 4: Создать словарь (ключи – числа, значения – строки), по items() сделать обратный словарь.

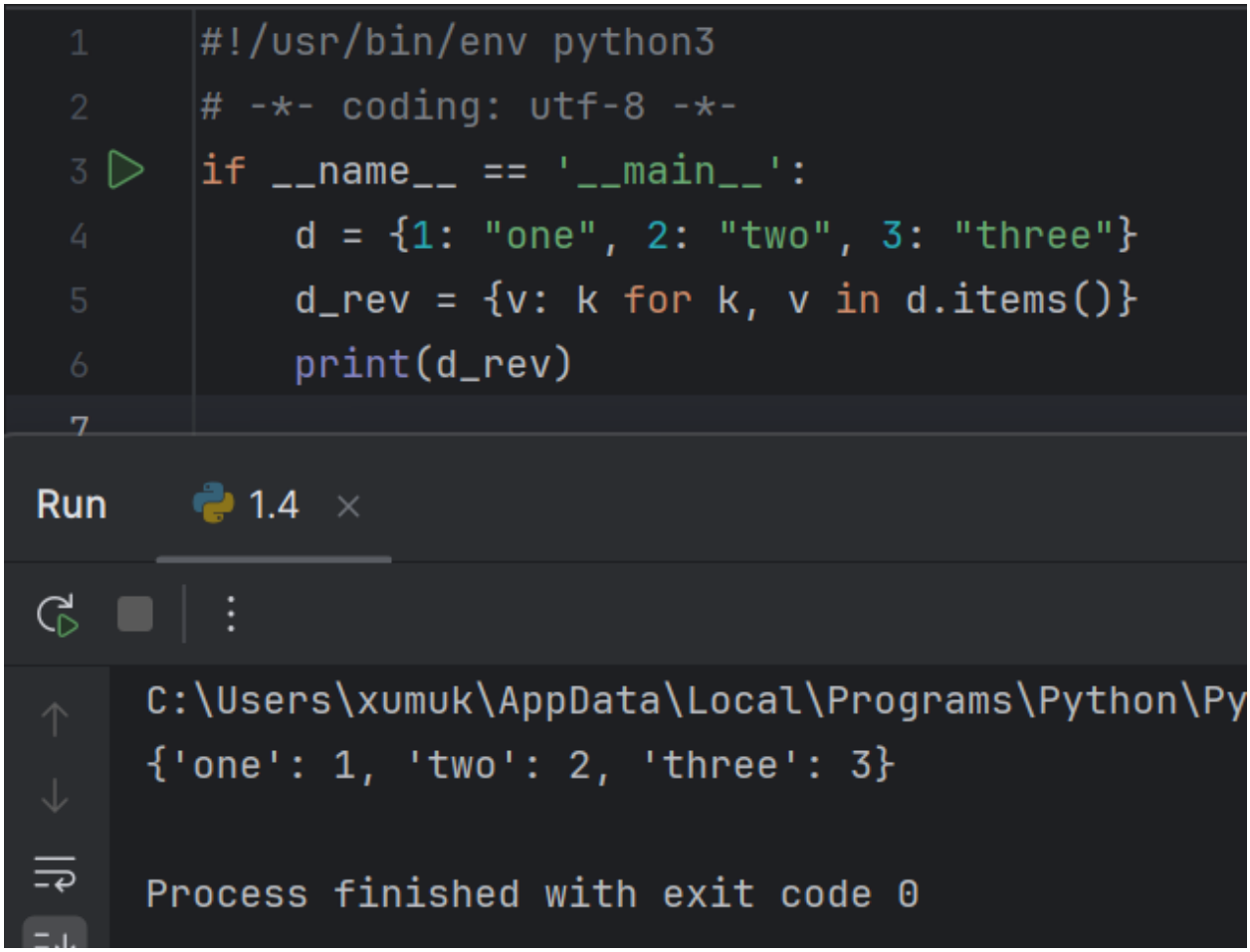

Рисунок 4 - Задание 4.

$$A = \{a, b, d, I, x\};$$
$$B = \{d, e, h, i, n, u\};$$
$$C = \{e, f, m, n\};$$
$$D = \{a, c, h, k, r, s, w, x\};$$
$$X = (A/C) \cap \bar{B};$$
$$Y = (\bar{A} \cap D) \cup (C/B).$$

Рисунок 5 – Условие задания 5.

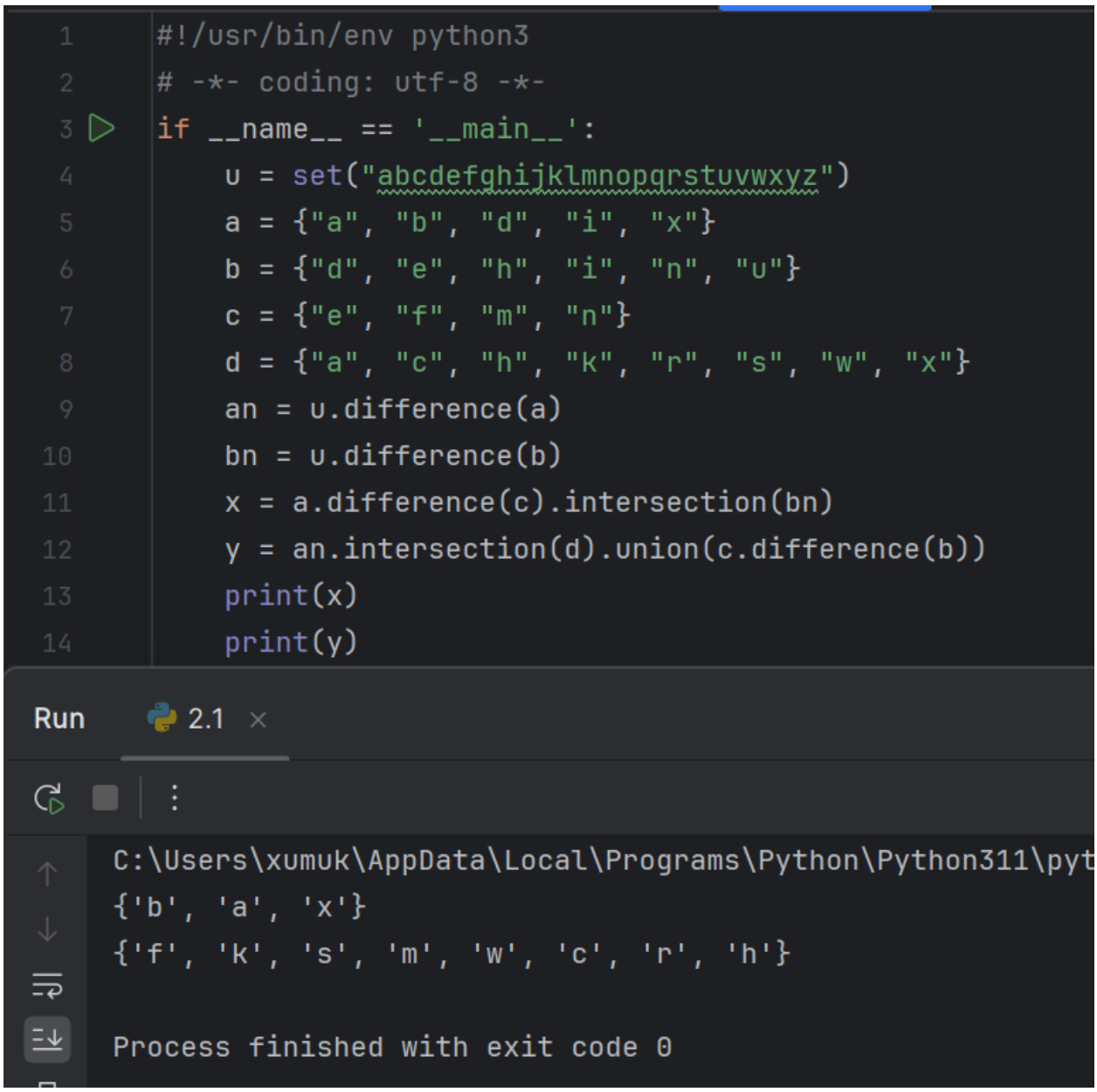

Рисунок 6 – Задание 5

5. Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Рисунок 7 – Условие задания 6

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    r = []
    while True:
        comand = input().lower()
        if comand == "exit":
            break
        if comand == "add":
            to = input("To where? ")
            n = input("Flight id? ")
            t = input("Type of plain? ")
            rn = {
                "to": to,
                "n": n,
                "t": t
            }
            r.append(rn)
            if len(r) > 0:
                r.sort(key=lambda item: item.get("to"))
        if comand == "list":
            t1 = input("Type of plain? ")
            d = 0
            for rn in r:
                if rn.get("t") == t1:
                    print("Going to " + rn.get("to") + " With flight id " + rn.get("n"))
                    d += 1
            if d == 0:
                print("Not a type of plain or that type dosn`t servicing a flight")
```

Рисунок 8 – Задание 6

Контрольные вопросы:

1. Множество (set) в Python — это неупорядоченная коллекция уникальных хешируемых элементов; оно изменяемо, не допускает дубликатов и оптимизировано под быстрые операции проверки принадлежности и множество-логические операции.

2. Создают множества литералом {1, 2, 3}, функцией set(iterable) (например, set([1,2,2])), пустое множество только через set(), а также с помощью set-включений вида {expr for x in iterable}.

3. Проверка присутствия или отсутствия элемента выполняется операторами in и not in: выражения x in s и x not in s возвращают логическое значение, при этом проверка в среднем выполняется за O(1).

4. Перебор элементов множества реализуют циклом for: for x in s: ...; порядок обхода не гарантируется, но можно дополнительно отсортировать элементы: for x in sorted(s): ....

5. Set comprehension — это конструкция вида {expr for x in iterable if cond}, которая за один проход формирует новое множество, одновременно преобразуя элементы и отфильтровывая ненужные.

6. Добавление одного элемента во множество выполняется методом s.add(x), а сразу нескольких — s.update(iterable), где iterable может быть списком, множеством, кортежем и т.п.

7. Удаление элемента выполняют методами s.remove(x) (бросает KeyError, если элемента нет) или s.discard(x) (молча игнорирует отсутствие); s.pop() удаляет и возвращает произвольный элемент, а s.clear() полностью очищает множество.

8. Основные операции над множествами: объединение s | t или s.union(t), пересечение s & t или s.intersection(t), разность s - t или s.difference(t) и симметрическая разность s ^ t или s.symmetric_difference(t); есть также присваивающие варианты |=, &=, -=, ^=.

9. Определить отношение подмножества и надмножества можно методами s.issubset(t) и s.issuperset(t) либо операторами <=, <, >=, >: например, a <= b означает, что a — подмножество b, а a >= b — надмножество.

10. Множество frozenset — это неизменяемый (иммутабельный) вариант множества: элементы в нем те же (уникальные, неупорядоченные), но после создания его нельзя модифицировать, зато такие объекты хешируемы и могут использоваться как ключи словаря или элементы других множеств.

11. Преобразование множества в строку чаще всего выполняют через str(s) или с помощью ' '.join(map(str, s)); в список и кортеж — через list(s) и tuple(s); в словарь множество можно превратить, например, используя генератор пар: d = {x: True for x in s} или dict.fromkeys(s, value).

12. Словарь (dict) — это изменяемое отображение «ключ → значение», где ключи уникальны и хешируемы, а значения могут быть любыми объектами; словарь хранит пары и позволяет очень быстро получать значение по ключу.

13. Функция len() применяется к словарям и возвращает количество пар ключ–значение, то есть размер словаря: len(d).

14. Обход словаря можно выполнять по ключам for k in d: ... или for k in d.keys(): ..., по значениям for v in d.values(): ..., а также сразу по парам for k, v in d.items(): ...; все эти варианты возвращают итерируемые объекты, которые можно использовать в циклах и генераторах.

15. Значение по ключу получают оператором индексирования d[key] (при отсутствии ключа будет KeyError), методом d.get(key, default=None) для безопасного получения с значением по умолчанию, либо d.setdefault(key, default) — он возвращает значение по ключу и при отсутствии создаёт новую пару с указанным значением.

16. Установить значение по ключу можно простым присваиванием d[key] = value (ключ создаётся или значение перезаписывается), методом d.update({key: value}) или d.update(iterable_of_pairs), а также через setdefault, если нужно задать значение только для отсутствующих ключей.

17. Словарь включений (dict comprehension) — это выражение вида {key_expr: value_expr for x in iterable if cond}, позволяющее за один проход сформировать новый словарь из другого итерируемого объекта с одновременной обработкой и фильтрацией данных.

18. Функция zip() создаёт итератор кортежей, «склеивая» несколько итерируемых объектов по позициям: zip(a, b) даёт пары (a[i], b[i]); её используют для параллельного обхода коллекций (for x, y in zip(xs, ys): ...), построения словарей dict(zip(keys, values)), а также для «транспонирования» матриц list(zip(*matrix)).

19. Модуль datetime предоставляет

классы date, time, datetime, timedelta, timezone и функции для работы с датой и временем: получение текущего момента (datetime.now(), date.today()), арифметику с интервалами (datetime + timedelta), сравнение дат, форматирование и разбор строк (strftime, strptime), управление часовыми поясами и перевод времени между ними.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с множествами и словарями при написании программ.