МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Департамент цифровых, робототехнических систем и электроники

«Работа с множествами и словарями в языке Python»
Отчет по лабораторной работе № 5
по дисциплине «Программирование на Python»
Вариант 9

Выполнил студент группы ИВТ-б-о-24-1
Каиров Вадим Сосланович
«__» ноября 2025г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2025

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

Ход работы:

Ссылка на GitHub: https://github.com/ozetin/labochka5

Создаём удалённый репозиторий в Github и клонируем его к себе на компьютер.

Пишем в нём все программы из лабораторной работы и делаем коммиты.

Задание 1. Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    schet = 0
    glasnie = set('уеыаоэяиюё')
    a = input("Введите строку: ").lower()
    for i in a:
        if i in glasnie:
            schet += 1
    print(schet)
```

Рисунок 1. Код решения задания 1

```
C:\Users\vadim\AppData\Local\Progra
Введите строку: Программирование
7
```

Рисунок 2. Результат выполнения программы

Задание 2. Решите задачу: определите общие символы в двух строках, введенных с клавиатуры.

```python
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    str1 = set(input("Введите строку: "))
    str2 = set(input("Введите строку: "))
    obsh = set(str1.intersection(str2))
    print(obsh)
```

Рисунок 3. Код решения задания 2.

```
C:\Users\vadim\AppData\Local\Programs
Введите строку: программирование
Введите строку: пайтон
{'а', 'н', 'о', 'п'}
```

Рисунок 4. Результат выполнения программы

Задание 3. Решите задачу: Создайте словарь, связав его с переменной school, и наполните данными, которые бы отражали количество учащихся в разных классах. Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован класс. Вычислите общее количество учащихся в школе.

```python
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    school = {}
    print("add - добавить класс")
    print("stop - достаточно классов")

    while True:
        a = input()
        if a == 'stop':
            break
        elif a == 'add':
            name = input("Введите название класса: ")
            kol = int(input("Введите количество учеников: "))
            school[name] = kol
        else:
            print("такой команды нет", file=sys.stderr)

    print("change - изменить количество учеников в классе")
    print("new - добавить класс")
    print("destroy - расформировать класс")
    print("stop - остановить изменения")

    while True:
        a = input()

        if a == 'change':
            klass = input("Класс, который изменяем: ")
            kol = int(input("Новое количество учеников: "))
            school[klass] = kol

        elif a == 'new':
            name = input("Введите название класса: ")
            kol = int(input("Введите количество учеников: "))
            school.setdefault(name, kol)

        elif a == 'destroy':
            name = input("Введите название класса: ")
            del school[name]
        elif a == 'stop':
            break
        else:
            print("Такой команды нет", file=sys.stderr)

    schet = 0
    for i in school:
        schet += school[i]
    print(school)
    print(f"В школе {schet} учеников")
```

Рисунок 5. Код решения задания 3

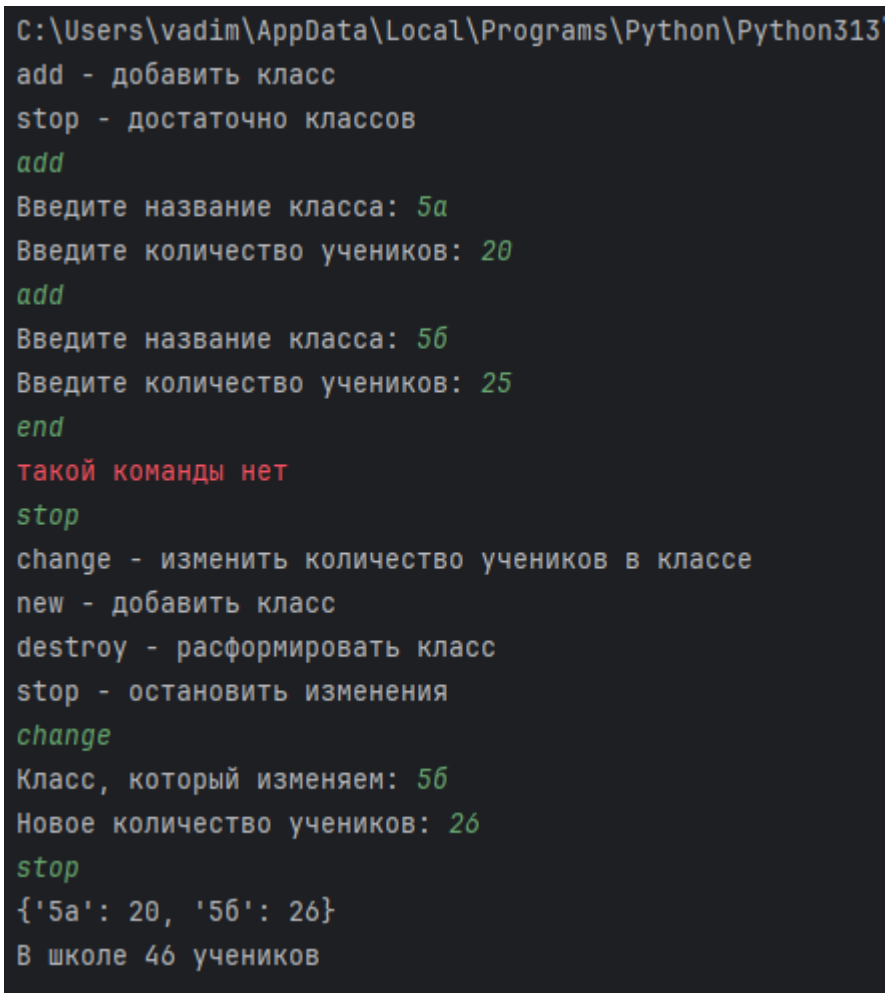

Рисунок 6. Результат выполнения программы

Задание 4. Создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод items(), с помощью полученного объекта создайте словарь, обратный исходному, т.е. ключами являются строки, а значениями – числа.

```python
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    slov = {}
    print("add - добавить пару")
    print("stop - достаточно пар")

    while True:
        a = input()

        if a == "add":
            kluch = int(input("Введите число-ключ: "))
            znach = input("Введите значение-строку: ")
            slov[kluch] = znach
        elif a == "stop":
            break
        else:
            print("Неизвестная команда", file=sys.stderr)
    new_slov = {v: k for k, v in slov.items()}
    print(slov)
    print(new_slov)
```

Рисунок 7. Код решения задания 4

```
C:\Users\vadim\AppData\Local\Programs\
add - добавить пару
stop - достаточно пар
add
Введите число-ключ: 5
Введите значение-строку: привет
add
Введите число-ключ: 6
Введите значение-строку: пока
stop
{5: 'привет', 6: 'пока'}
{'привет': 5, 'пока': 6}
```

Рисунок 8. Результат выполнения программы

Индивидуальное задание 1. Определите результат выполнения операций над множествами. Считать элементы множества строками. Проверить

результаты вручную. Номер варианта задания необходимо получить у преподавателя.

A = {"a", "e", "f", "i"}

B = {"a","b", "k", "n"}

C = {"e","f","n","o","w","x"}

D = {"a","d","e","o","p","t","u"}

X = (A ∪ B) ∩ D

Y = (not(A) ∩ not(B)) / (C ∪ D)

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    u = set("qwertyuiopasdfghjklzxcvbnm")
    A = {"a", "e", "f", "i"}
    B = {"a","b", "k", "n"}
    C = {"e","f","n","o","w","x"}
    D = {"a","d","e","o","p","t","u"}

    An = u.difference(A)
    Bn = u.difference(B)

    X = (A.union(B)).intersection(D)
    Y = (An.intersection(Bn)).difference(C.union(D))

    print(f"Y = {Y}")
    print(f"X = {X}")
```

Рисунок 9. Код решения индивидуального задания 1

```
C:\Users\vadim\AppData\Local\Programs\Python\Python313\python.exe
Y = {'r', 'l', 'z', 'y', 'v', 'h', 'g', 'c', 'j', 'q', 's', 'm'}
X = {'a', 'e'}
```

Рисунок 10. Результат выполнения программы

Индивидуальное задание 2. Использовать словарь, содержащий следующие ключи: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута. Написать программу,

выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам маршрутов; вывод на экран информации о маршруте, номер которого введен с клавиатуры; если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

```python
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':

    marshruti = []
    print("add - добавить маршрут")
    print("end - закончить ввод маршрутов")
    while True:
        a = input().lower()

        if a == "add":
            start = input("Введите название начального пункта: ")
            end = input("Введите название конечного пункта: ")
            number = int(input("Введите номер маршрута: "))

            marshrut = {
                "start": start,
                "end": end,
                "number": number
            }
            marshruti.append(marshrut)

            if len(marshruti) > 1:
                marshruti.sort(key=lambda x: x.get("number",""))
        elif a == "end":
            break
        else:
            print("неизвестная команда")

    isk_numb = int(input("Введите номер искомого маршрута: "))
    schet = 0

    for i in marshruti:
        if i["number"] == isk_numb:
            schet += 1
            print(f"Начальный пункт: {i['start']}")
            print(f"Конечный пункт: {i['end']}")
            print(f"Номер маршрута: {i['number']}")
    if schet == 0:
        print("Таких маршрутов нет", file=sys.stderr)
```

Рисунок 11. Код решения индивидуального задания 2

```
C:\Users\vadim\AppData\Local\Programs\Python\Python313
add - добавить маршрут
end - закончить ввод маршрутов
add
Введите название начального пункта: ставрополь
Введите название конечного пункта: москва
Введите номер маршрута: 2
add
Введите название начального пункта: таганрог
Введите название конечного пункта: пермь
Введите номер маршрута: 1
end
Введите номер искомого маршрута: 2
Начальный пункт: ставрополь
Конечный пункт: москва
Номер маршрута: 2
```

Рисунок 12. Результат выполнения программы

Также в ходе данной работы были проработаны примеры из методического материала.

```python
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    u = set("qwertyuiopasdfghjklzxcvbnm")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Рисунок 13. Пример 1

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
from datetime import date

if __name__ == '__main__':
    workers = []

    while True:
        command = input(">>> ").lower()

        if command == "exit":
            break

        elif command == "add":
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))

            worker = {"name": name,
                      "post": post,
                      "year": year
            }
            workers.append(worker)

            if len(workers) > 1:
                workers.sort(key=lambda item: item.get("name", ""))

        elif command == "list":
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                *args: "-" * 4,
                "-" * 30,
                "-" * 20,
                "-" * 8
            )
            print(line)
            print(
                "| {:^4} | {:^30} | {:^20} | {:^8} | ".format(
                    *args: "№",
                    "Ф.И.О.",
                    "Должность",
                    "Год"
                )
            )
            print(line)

            for idx, worker in enumerate(workers, 1):
                print(
                    "| {:>4} | {:<30} | {:<20} | {:>8} | ".format(
                        *args: idx,
                        worker.get("name", ""),
                        worker.get("post", ""),
                        worker.get("year", 0)
                    )
                )
            print(line)
```

Рисунок 14. Пример 2

```
            print(line)

        elif command.startswith("select"):
            today = date.today()

            parts = command.split( sep: " ", maxsplit=1)
            period = int(parts[1])

            count = 0
            for worker in workers:
                if today.year - worker.get('year', today.year) >= period:
                    count += 1
                    print(
                        "{:>4}: {} ".format( *args: count, worker.get("name", ""))
                    )

            if count == 0:
                print("Работники с заданным стажем не найдены.")
        elif command == "help":
            print("Список команд:\n")
            print("add - добавить работника;")
            print("list - вывести список работников;")
            print("select - запросить работников со стажем;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой;")
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунок 15. Пример 2

Контрольные вопросы:

1) Что такое множества в языке Python?

Множество в Python – неупорядоченная совокупность уникальных значений. В качестве элементов могут выступать любые неизменяемые объекты, такие как числа, символы, строки.

2) Как осуществляется создание множеств в Python?

A = {1, 2, 3, 0, 1, 2, 3}

ИЛИ

A = set()

3) Как проверить присутствие/отсутствие элемента в множестве?

С помощью in.

4) Как выполнить перебор элементов множества?

Цикл for

5) Что такое set comprehension?

Set comprehension – это генератор, позволяющий заполнять списки и другие наборы данных с учетом неких условий.

A = {I for I in [1, 2, 0, 1, 3, 2]}

6) Как выполнить добавление элемента во множество?

Чтобы внести новое значение потребуется метод add.

a.add(4)

7) Как выполнить удаление одного или всех элементов множества?

Remove() – метод, который удаляет указанный элемент с генерацией исключения если элемент отсутствует.

Discard - метод, который удаляет указанный элемент без генерации исключения если элемент отсутствует.

Pop – метод, удаляет и возвращает элемент по индексу. Генерирует исключение при попытке удаления из пустого множества.

8) Как выполняются основные операции над множествами: объединение, пересечение, разность?

Объединение осуществляется методом union(b)

A = {0, 1, 2}

B = {1, 2, 3}

C = a.union(b)

Чтобы найти пересечение используется функция intersection.

A = {0, 1, 2}

B = {1, 2, 3}

C = a.intersection(b)

Чтобы найти разность используется метод difference

A = {0, 1, 2}

B = {1, 2, 3}

C = a.difference(b)

9) Как определить, что некоторое множество является надмножеством или подмножеством другого множества?

Является ли подмножеством используется метод issubset

A = {0, 1, 2}

B = {1, 2, 3}

Print(A.issubset(B))

Является ли подмножеством используется метод issuperset

10) Каково назначение множеств frozenset?

Это множества, которые нельзя изменить

A = frozenset({“hello”, “world”})

11) Как осуществляется преобразование множеств в строку, список, словарь?

Строка

A = {“set”,”str”,”dict”}

B = ‘,’.join(A)

Словарь

A = {(‘a’, 2), (‘b’, 4)}

B = dict(A)

Список

A = {1, 2, 3, 0, 1, 2, 3}

B = list(a)

Print(B)

[0, 1, 2, 3]

12) Что такое словари в языке Python?

Словарь представляет собой структуру данных, предназначенную для хранения произвольных объектов с доступом по ключу. Данные хранятся в формате ключ – значение.

13) Может ли функция len() быть использована при работе со словарями?

Да, она может использоваться со словарями и возвращает количество пар “ключ-значение” в нём

14) Какие методы обхода словарей вам известны?

Прямой итератор (по ключам), .keys(), .values(), .items()

15) Какими способами можно получить значение из словаря по ключу?

1. Прямое обращение по ключу
2. Метод .get(). Можно задать значение по умолчанию.
3. Метод .setdefault()

16) Какими способами можно установить значение в словаре по ключу?

1. Прямое присваивание по ключу
2. Метод .update(). Позволяет массово установить или обновить несколько значений сразу.
3. Метод .setdefault() . Если ключа нет — создаёт его с указанным значением. Если ключ уже есть — ничего не меняет.

17) Что такое словарь включений?

Словарь включений (dictionary comprehension) — это короткий, удобный способ создания словаря с помощью одной строки.

{ключ: выражение for элемент in итерируемый_объект [if условие]}

18) Самостоятельно изучите возможности функции zip() приведите примеры ее использования.

zip() — это встроенная функция Python, которая объединяет несколько итерируемых объектов (например, списки, кортежи, строки) в итератор кортежей, где каждый кортеж содержит элементы с одинаковыми индексами из всех последовательностей.

```
names = ["Ann", "Bob", "Kate"]
ages = [25, 30, 28]

people = dict(zip(names, ages))
result = zip(names, ages)
print(people)
print(list(result))
```

{'Ann': 25, 'Bob': 30, 'Kate': 28}
[('Ann', 25), ('Bob', 30), ('Kate', 28)]

19) Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с датой и временем обладает этот модуль?

Модуль datetime — один из самых полезных и часто используемых модулей Python. Он предоставляет всё необходимое для работы с датами, временем, интервалами, форматированием и вычислениями.

datetime.date - хранит только дату (год, месяц, день)

datetime.time - хранит только время (часы, минуты, секунды, микросекунды)

datetime.datetime - хранит дату и время вместе

datetime.timedelta - представляет разницу (интервал) между датами или временем

Вывод: в ходе данной работы были приобретены навыки по работе с множествами и словарями при написании программ с помощью языка программирования Python версии 3.x.