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
по дисциплине «Программирование на Python»

Выполнил студент Ромащенко Данил группы ИВТ-б-о-24-1

Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2025

Тема: «Работа с множествами и словарями в языке Python».

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

Ход работы

Вариант 18

Ссылка на репозиторий: https://github.com/l1N322/lab-5.git

1) Был создан общедоступный репозиторий, в котором использована лицензия MIT и язык программирование Python.

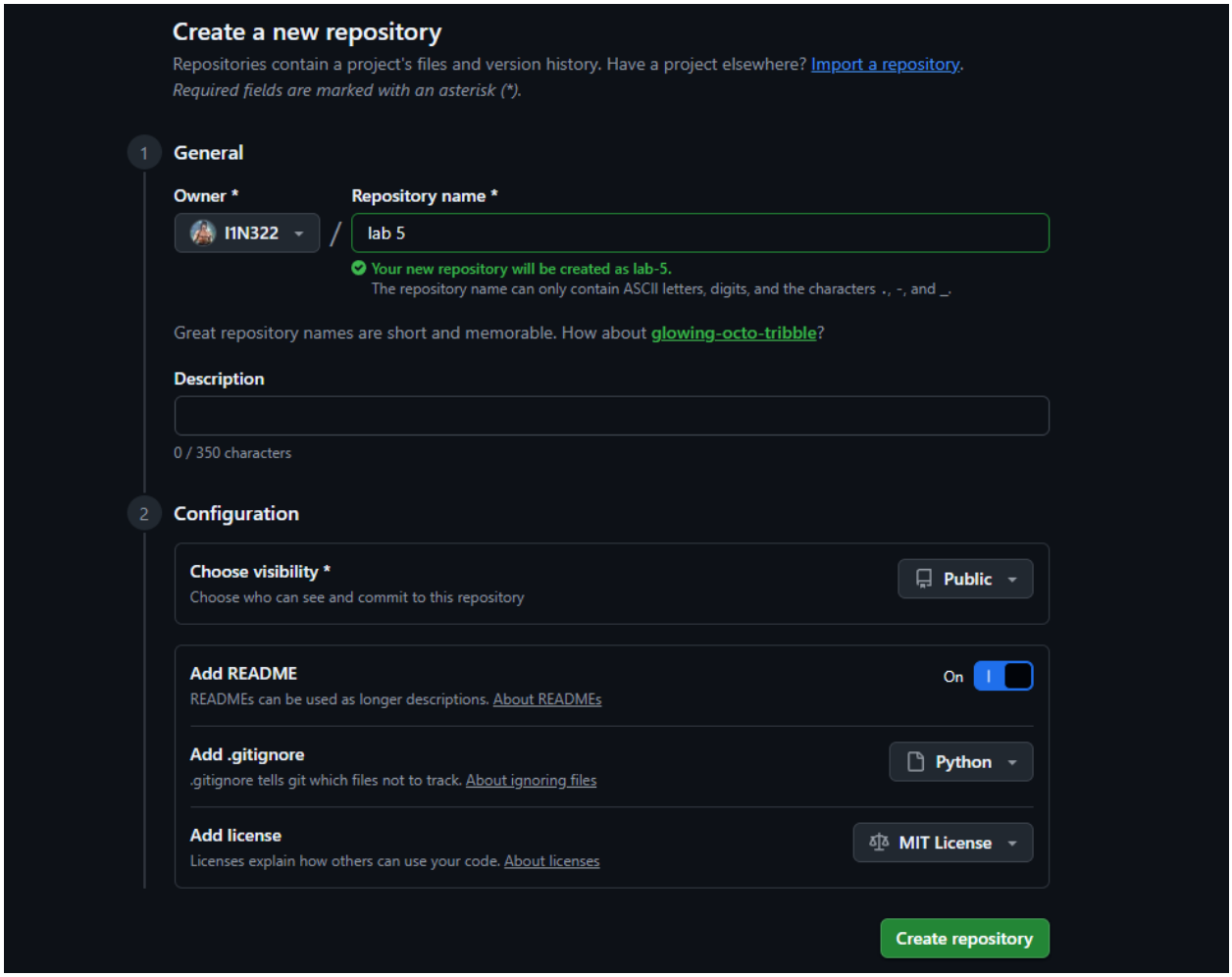

Рисунок 1. Созданный репозиторий

2) Был дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

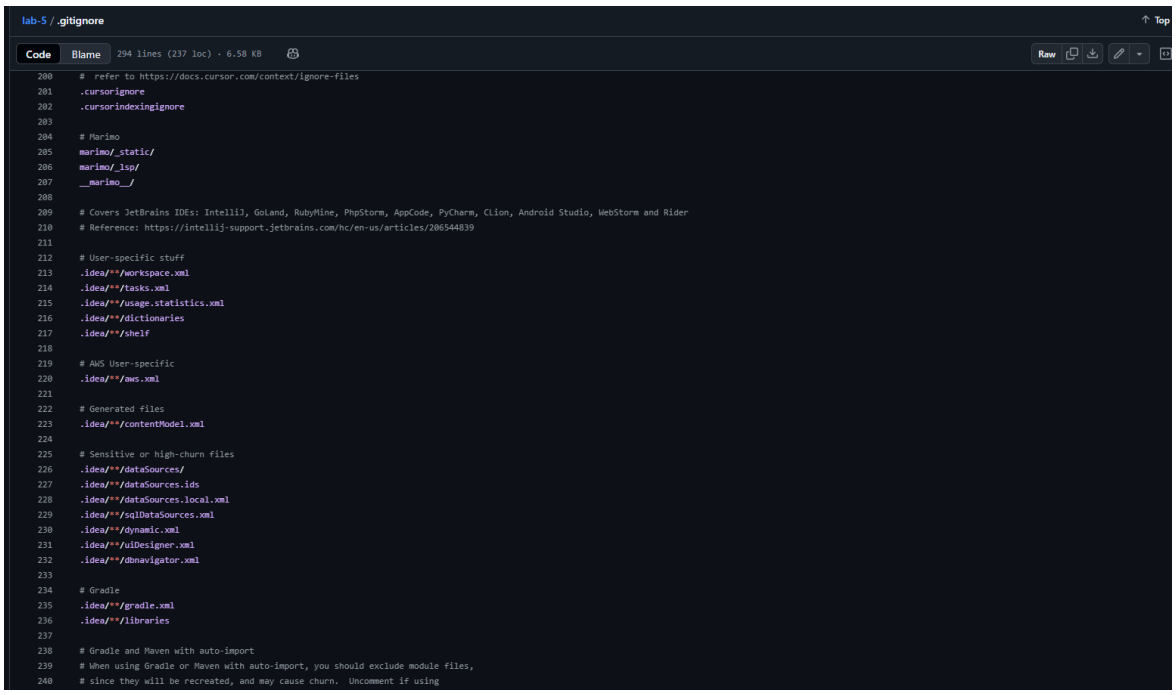

Рисунок 2. Добавленные правила в файл .gitignore

3) Было выполнено клонирование созданного репозитория на компьютер.

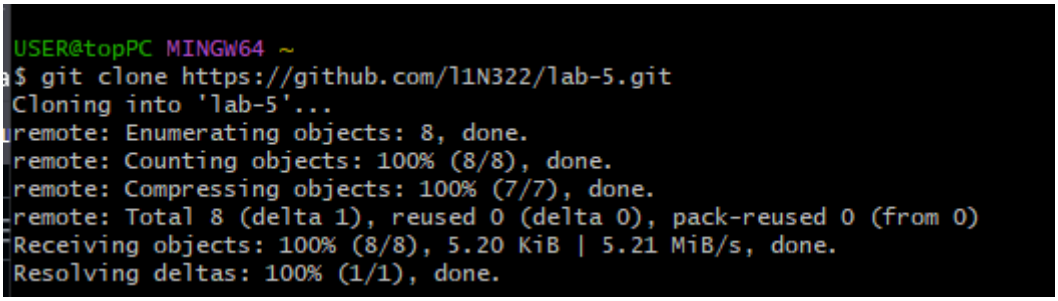

Рисунок 3. Клонирование репозитория

4) Был создан проект PyCharm в папке репозитория.

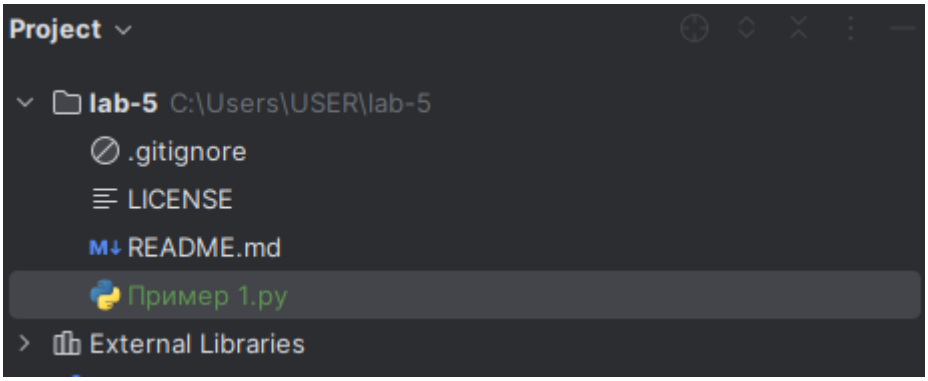

Рисунок 4. Созданный проект

5) Были проработаны примеры лабораторной работы и зафиксированы результаты выполнения каждой из программ при разных исходных данных, вводимых с клавиатуры.

Пример 1.py

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

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

Рисунок 5. Пример 1

Пример 1

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Пример 1.py"
x = {'o', 'e', 'j', 'k', 'd'}
y = {'c', 'v', 'y', 'f', 'o', 'g', 'b', 'h'}

Process finished with exit code 0
```

Рисунок 6. Результат выполнения примера 1

Пример 1.py Пример 2.py

```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,
```

Рисунок 7. Пример 2

```
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |              Ф.И.О.              |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
+------+----------------------------------+----------------------+----------+
>>> help
Список команд:

add - добавить работника;
list - вывести списко работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
```

Рисунок 8. Результат выполнения примера 2

```
>>> add
Фамилия и инициалы? Пушкин А.С.
Должность? Директор
Год поступления? 2020
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Пушкин А.С.                      | Директор             |     2020 |
+------+----------------------------------+----------------------+----------+
>>> select 4
   1: Пушкин А.С.
>>> exit
```

Рисунок 9. Результат выполнения примера 2 с другими данными

6) Были выполнены задания и индивидуальные задания.

7) Задание 1: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

Задание 1.py

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

if __name__ == "__main__":
    glas = set("аеёиоуыэюяаеiou")

    text = input("Введите строку: ").lower()

    count = sum(1 for char in text if char in glas)

    print(count)
```

Рисунок 10. Задание 1

Задание 1

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Задание 1.py"
Введите строку: Привет мир!
3

Process finished with exit code 0
```

Рисунок 11. Результат выполнения задания 1

8) Задание 2: определите общие символы в двух строках, введенных с клавиатуры.

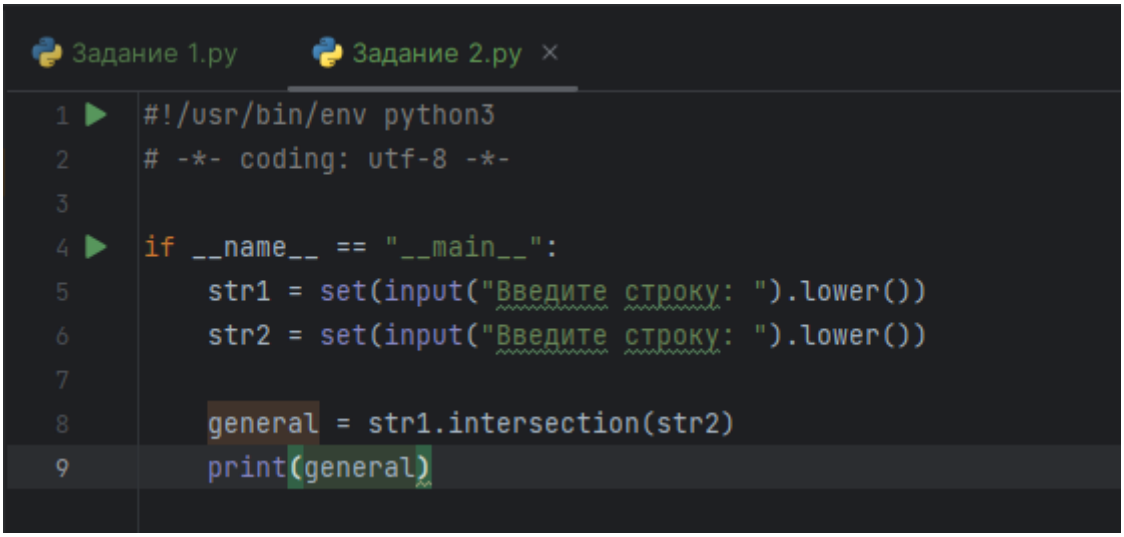

Рисунок 12. Задание 2

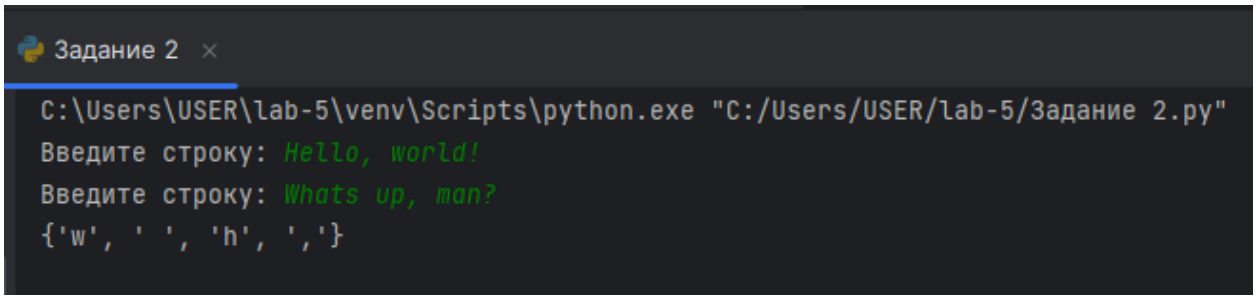

Рисунок 13. Результат выполнения задания 2

9) Задание 3: создайте словарь, связав его с переменной school , и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 16, 26, 6а, 7в и т. п.). 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 (удален) другой класс. Вычислите общее количество учащихся в школе.

Задание 1.py | Задание 2.py | Задание 3.py

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

if __name__ == "__main__":
    school = {
        '1а': 25,
        '1б': 27,
        '2а': 30,
        '2б': 28,
        '6а': 24,
        '7в': 26
    }

    print("Исходные данные:", school)
    print("Общее количество учащихся:", sum(school.values()))

    school['2а'] = 32
    print(f" После изменения в 2а классе: {school}")

    school['8г'] = 29
    print(f" После добавления 8г класса: {school}")

    del school['1б']  # Удаляем класс 1б
    print(f"в) После удаления 1б класса: {school}")

    sum_students = sum(school.values())
    print(f"Общее количество учащихся в школе: {sum_students}")
```

Рисунок 14. Задание 3

Задание 3

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Задание 3.py"
Исходные данные: {'1а': 25, '1б': 27, '2а': 30, '2б': 28, '6а': 24, '7в': 26}
Общее количество учащихся: 160
 После изменения в 2а классе: {'1а': 25, '1б': 27, '2а': 32, '2б': 28, '6а': 24, '7в': 26}
 После добавления 8г класса: {'1а': 25, '1б': 27, '2а': 32, '2б': 28, '6а': 24, '7в': 26, '8г': 29}
 После удаления 1б класса: {'1а': 25, '2а': 32, '2б': 28, '6а': 24, '7в': 26, '8г': 29}
Общее количество учащихся в школе: 164

Process finished with exit code 0
```

Рисунок 15. Результат выполнения задания 3

10) Задание 4: создайте словарь, где ключами являются числа, а значениями - строки. Примените к нему метод items(), с помощью

полученного объекта dict_items создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями - числа.

Задание 1.py | Задание 2.py | Задание 3.py | Задание 4.py

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

if __name__ == "__main__":
    dict = {
        1: "apple",
        2: "banana",
        3: "cherry",
        4: "strawberry",
        5: "pineapple",
    }

    print(f"Исходный словарь {dict}")

    dict_items = dict.items()
    print(f"Объект dict_items: {dict_items}")

    swapped = {v: k for k, v in dict_items}

    print(f"Изменённый словарь {swapped}")
```

Рисунок 16. Задание 4

Задание 4

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Задание 4.py"
Исходный словарь {1: 'apple', 2: 'banana', 3: 'cherry', 4: 'strawberry', 5: 'pineapple'}
Объект dict_items: dict_items([(1, 'apple'), (2, 'banana'), (3, 'cherry'), (4, 'strawberry'), (5, 'pineapple')])
Изменённый словарь {'apple': 1, 'banana': 2, 'cherry': 3, 'strawberry': 4, 'pineapple': 5}

Process finished with exit code 0
```

Рисунок 17. Результат выполнения задания 4

11) Индивидуальное задание 1: определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную.

$$A = \{b, c, g, I, w\};$$
$$B = \{e, g, h, q, w\};$$
$$C = \{c, d, k, l, y\};$$
$$D = \{a, g, h, u, v, z\};$$
$$X = (A \cap C) \cup B;$$
$$Y = (\bar{A} \cap D) \cup (C / B).$$

Индивидуальное задание 1.py

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "g", "i", "w"}
    b = {"e", "g", "h", "q", "w"}
    c = {"c", "d", "k", "l", "y"}
    d = {"a", "g", "h", "u", "v", "z"}

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

    an = u.difference(a)

    y = (an.intersection(d)).union(c.difference(b))
    print(f"y = {y}")
```

Рисунок 18. Индивидуальное задание 1

Индивидуальное задание 1

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Индивидуальное задание 1.py"
x = {'e', 'h', 'w', 'q', 'g', 'c'}
y = {'k', 'h', 'a', 'l', 'c', 'd', 'z', 'v', 'y', 'u'}

Process finished with exit code 0
```

Рисунок 19. Результат выполнения индивидуального задания 1

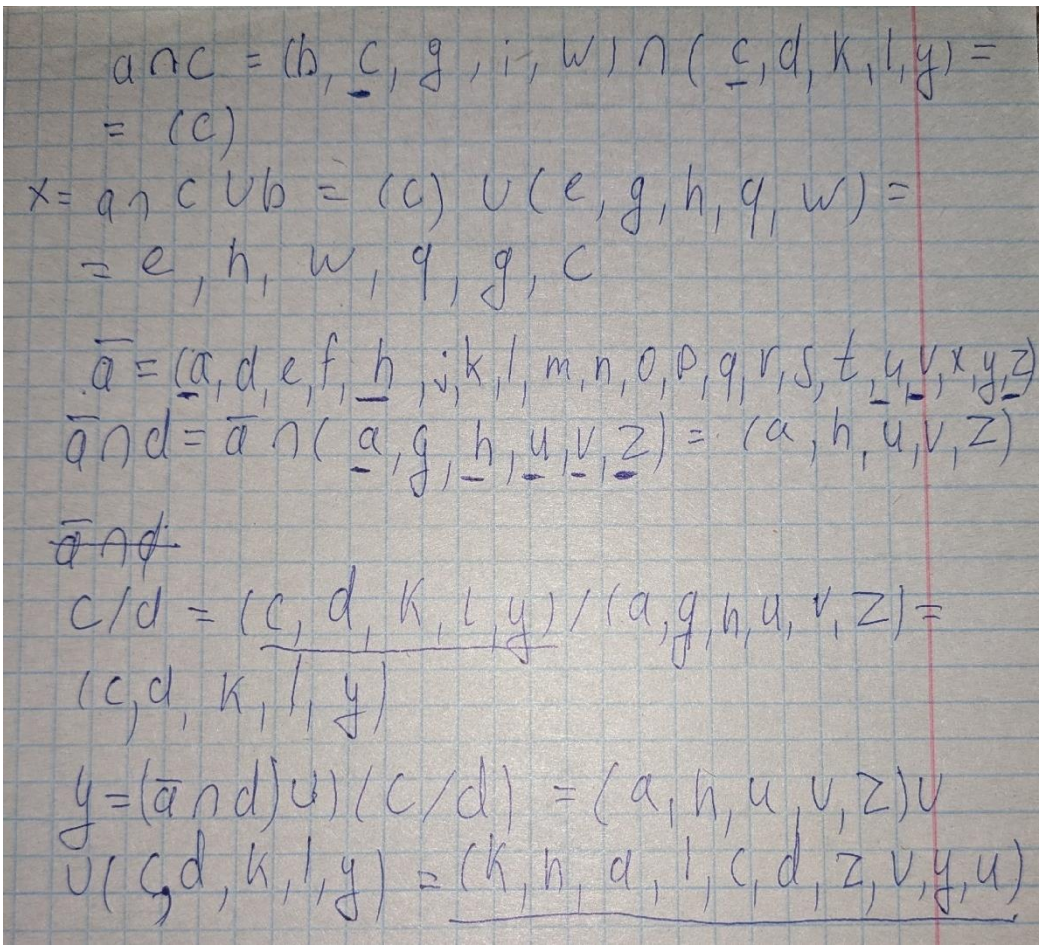

Рисунок 20. Проверка результатов

12) Индивидуальное задание 2: Составить программу с использованием списков и словарей для решения задачи: Использовать словарь, содержащий следующие ключи: название товара; название магазина, в котором продается товар; стоимость товара в руб. 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 магазинов; вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры; если такого магазина нет, выдать на дисплей соответствующее сообщение.

Индивидуальное задание 2.py

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

if __name__ == "__main__":
    products = []
    while True:
        product = input("Введите название товара (или 'стоп'): ")
        if product.lower() == 'стоп':
            break
        shop = input("Введите название магазина: ")
        price = float(input("Введите стоимость: "))
        products.append({'товар': product, 'магазин': shop, 'цена': price})

    products.sort(key=lambda item: item.get('магазин', ''))

    print("Все товары:")
    for p in products:
        print(f"{p['магазин']} - {p['товар']} - {p['цена']} руб.")

    search = input("Введите магазин для поиска: ")
    found = [p for p in products if p['магазин'].lower() == search.lower()]

    if found:
        print(f"Товары в магазине '{search}':")
        for p in found:
            print(f"{p['товар']} - {p['цена']} руб.")
    else:
        print(f"Магазин '{search}' не найден.")
```

Рисунок 21. Индивидуальное задание 2

Индивидуальное задание 2

```
C:\Users\USER\lab-5\venv\Scripts\python.exe "C:/Users/USER/lab-5/Индивидуальное задание 2.py"
Введите название товара (или 'стоп'): хлеб
Введите название магазина: Продукты
Введите стоимость: 50
Введите название товара (или 'стоп'): молоко
Введите название магазина: Гастроном
Введите стоимость: 80
Введите название товара (или 'стоп'): стоп
Все товары:
Гастроном - молоко - 80.0 руб.
Продукты - хлеб - 50.0 руб.
Введите магазин для поиска: гастроном
Товары в магазине 'гастроном':
молоко - 80.0 руб.
```

Рисунок 22. Результат выполнения индивидуального задания 2

13) Были зафиксированы и отправлены сделанные изменения на сервер GitHub.

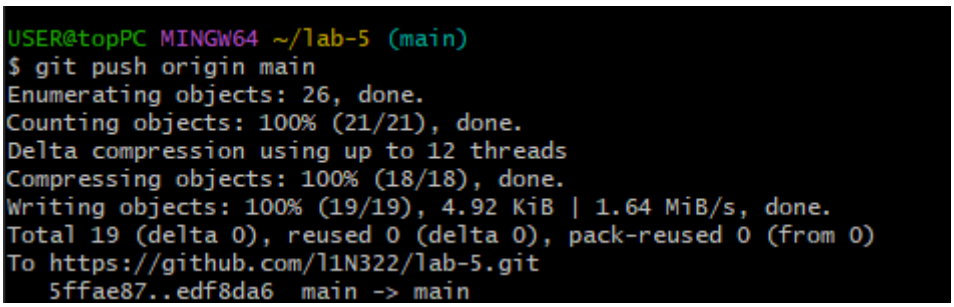

Рисунок 23. Отправка изменений

Ответы на контрольные вопросы:

1) Множества – неупорядоченная совокупность уникальных значений. В качестве элемента этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы и строки. В отличие от массивов и списков, порядок следования значений не учитывается при обработке его содержимого. Над одним, а также над несколькими множествами можно выполнять ряд операций, благодаря функциям стандартной библиотеки языка программирования Python.

2) Для создания множества можно просто присвоить переменной последовательность значений, выделив их фигурными скобками. Другой вариант создания множества подразумевает использование вызова set.

3) Для проверки присутствия/отсутствия данного значения в множестве нужно использовать in.

4) Перебор всех элементов множества можно выполнить с помощью следующего кода:

for a in {0, 1, 2, 3}

print(a)

5) Для создания множества можно воспользоваться генератором (Set comprehensions), позволяющим заполнять списки, а также другие наборы данных с учетом неких условий.

6) Чтобы внести новые значения в множество, потребуется вызвать метод add

7) Для удаления элемента из множества используются следующие функции:

remove – удаление элемента с генерацией исключения в случае, если такого элемента нет;

discard – удаление элемента без генерации исключения в случае, если такого элемента нет;

pop – удаление первого элемента, генерируется исключение при попытке удаление из пустого множества.

Для полной очистки используется метод clear не принимающий аргументов

8) Чтобы объединить все элементы двух разных множеств, стоит воспользоваться методом union на одном из объектов

Чтобы найти пересечение двух разных множества, следует применить функцию intersection, принимающую в качестве аргумента один из наборов данных

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом difference. Функция позволяет найти элементы, уникальные для второго набора данных, которых в нем нет

9) Чтобы выяснить, является ли множество a подмножеством b, нужно воспользоваться методом issubset.

Чтобы узнать, является ли множество a надмножеством b, необходимо вызвать метод issuperset

10) Множество, содержимое которого не поддаётся изменению имеет тип frozenset. Значение из этого набора нельзя удалить, как и добавить новые

11) Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция join. Её аргументов является набор данных в виде нескольких строк.

Чтобы получить из множества словарь, следует передать функции dict набор из нескольких пар значений, в каждом из которых будет находиться ключ.

Для получения списка используется вызов list, получающий в качестве аргумента множество a.

12) Словарь представляет собой структуру данных (которая еще называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу. Данные в словаре хранятся в формате ключ – значение. Словарь – это изменяемый неупорядоченный набор элементов

13) Да, функция len() может использоваться со словарями. Она возвращает количество пар ключ-значение в словаре.

14) Методы обхода словарей: цикл по ключам, цикл по значениям, цикл по парам ключ-значение, через метод keys().

15) Способы получения значения из словаря по ключу: через квадратные скобки, метод get(), с проверкой наличия ключа.

16) Способы установки значения в словаре по ключу: прямое присваивание, метод setdefault(), метод update().

17) Словарь включений — это краткий способ создания словаря с использованием цикла и условий в одной строке.

18) Zip() объединяет элементы нескольких итерируемых объектов в кортежи.

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
my_dict = dict(zip(keys, values))
# Результат: {'a': 1, 'b': 2, 'c': 3}
```

19) Модуль datetime предоставляет классы для работы с датами и временем:

- datetime.date — работа с датами (год, месяц, день).
- datetime.time — работа с временем (час, минута, секунда).
- datetime.datetime — комбинация даты и времени.
- datetime.timedelta — разница между датами/временем.
- datetime.timezone — работа с часовыми поясами.

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