

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №9
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: работа со словарями в языке Python.

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

Порядок выполнения работы

1. Создал репозиторий Github.

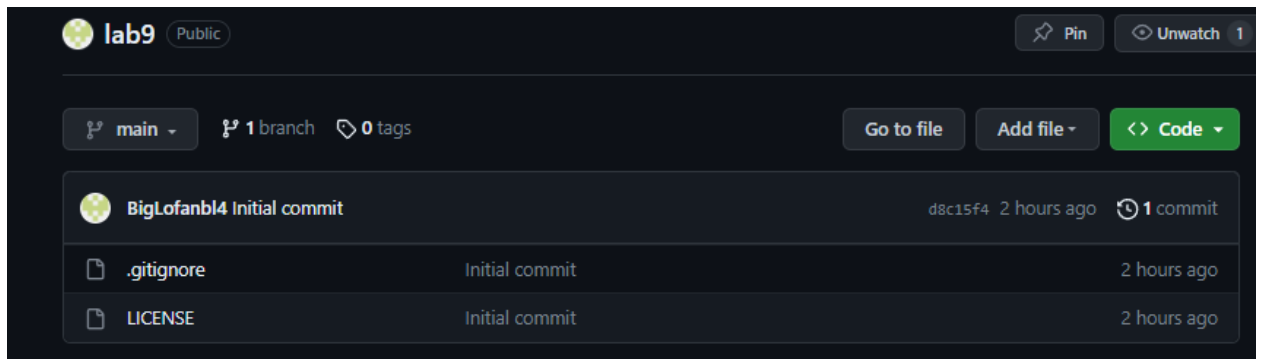


Рисунок 1 – Созданный репозиторий

2. Проработал примеры из лабораторной работы.

```
>>> add
Фамилия и инициалы? Jabroni J.M
Должность? master
Год поступления? 2023
>>> add
Фамилия и инициалы? Пушкин А.С.
Должность? Писатель
Год поступления? 1837
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          | Должность | Год |
+-----+-----+-----+-----+
| 1 | Jabroni J.M | master | 2023 |
| 2 | Пушкин А.С. | Писатель | 1837 |
+-----+-----+-----+-----+
>>> select 1
1: Пушкин А.С.
```

Рисунок 2 – Вывод программы из примера

3. Задание №1: создайте словарь, связав его с переменной `school`, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 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.

```

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

if __name__ == "__main__":
    school = {
        "1a": 30,
        "16": 34,
        "26": 29,
        "6a": 28,
        "7в": 24,
        "8г": 32,
        "9a": 31,
    }
    # а) в одном из классов изменилось количество учащихся
    school["26"] = 26

    # б) в школе появился новый класс
    school.setdefault("66", 43)

    # с) в школе был расформирован (удален) другой класс
    school.pop("8г")

    # общее количество учащихся в школе
    print(sum(school.values()))

```

Рисунок 3.1 – Код программы

```

Python 3.7.4 Shell
{ '1a': 30, '16': 34, '26': 26, '6a': 28, '7в': 24, '9a': 31, '66': 43 }
216

```

Рисунок 3.2 – Вывод программы

4. Задание №2: 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.

```

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

if __name__ == "__main__":
    a = {
        1: "one",
        2: "two",
        3: "three",
        4: "four",
        5: "bob"
    }

    new = {val: key for key, val in a.items()}
    print(new)

```

Рисунок 4.1 – Код программы

```

...мной инженерии/97/cab9/progs/ex2.py
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'bob': 5}

```

Рисунок 4.2 – Вывод программы

5. Индивидуальное задание: использовать словарь, содержащий следующие ключи: фамилия, имя; знак Зодиака; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

```

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

import sys
from datetime import datetime

if __name__ == "__main__":
    # 14 вариант
    people = []

    while True:
        command = input("Введите команду (add, info, exit): ").strip().lower()

        if command == "exit":
            break
        elif command == "add":
            person = {}
            person["surname"] = input("Введите фамилию: ")
            person["name"] = input("Введите имя: ")
            person["zodiac"] = input("Введите знак зодиака: ")
            person["birthday"] = input("Дата рождения (число.месяц.год):").split(".")

            people.append(person)

            people.sort(
                key=lambda x: datetime.strptime(".".join(x["birthday"]), "%d.%m.%Y")
            )

        elif command == "info":
            last_name = input("Введите фамилию: ")
            found = False
            for i in people:
                if i["surname"] == last_name:
                    print(f"Фамилия: {i['surname']}")
                    print(f"Имя: {i['name']}")
                    print(f"Знак зодиака: {i['zodiac']}")
                    print(f"Дата рождения: {i['birthday']}")
                    found = True

            if not found:
                print("Человка с такой фамилий нет в списке.")

        elif command == "help":
            print("add - добавление нового человека")
            print("info - данные о человеке по его фамилии")
            print("exti - завершение программы")

        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 5.1 – Код программы

```

Введите команду (add, info, exit): add
Введите фамилию: Плугатырев
Введите имя: Владислав
Введите знак зодиака: Козерог
Дата рождения (число.месяц.год):12.01.2005
Введите команду (add, info, exit): add
Введите фамилию: Чикодан
Введите имя: Алексей
Введите знак зодиака: Рак
Дата рождения (число.месяц.год):08.08.2003
Введите команду (add, info, exit): info
Введите фамилию: Плугатырев
Фамилия: Плугатырев
Имя: Владислав
Знак зодиака: Козерог
Дата рождения: ['12', '01', '2005']
Введите команду (add, info, exit): exit
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\9\lab9>

```

Рисунок 5.2 – Вывод программы

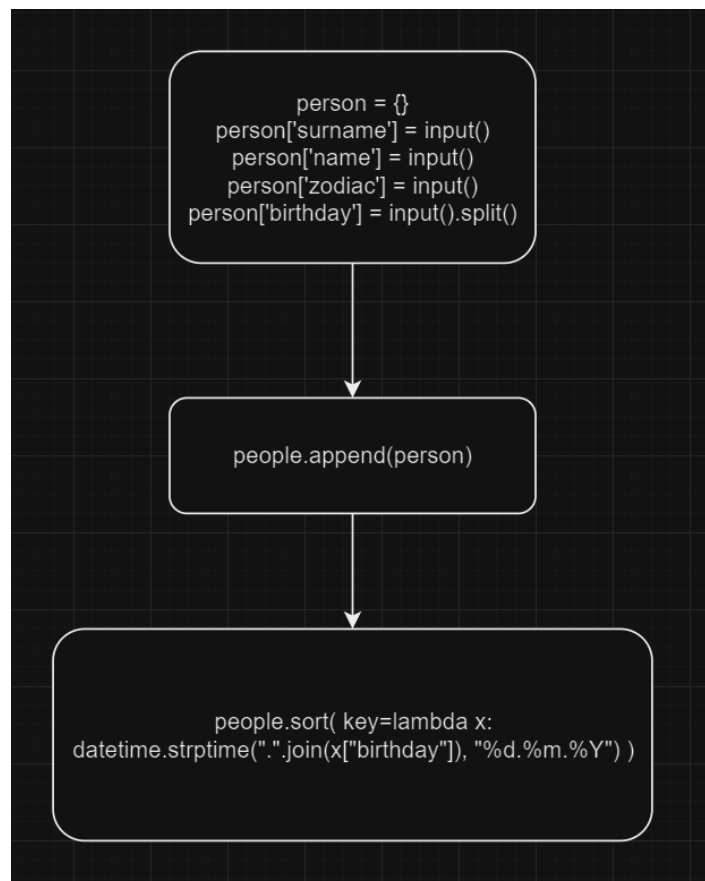


Рисунок 5.3 – UML-диаграмма команды add

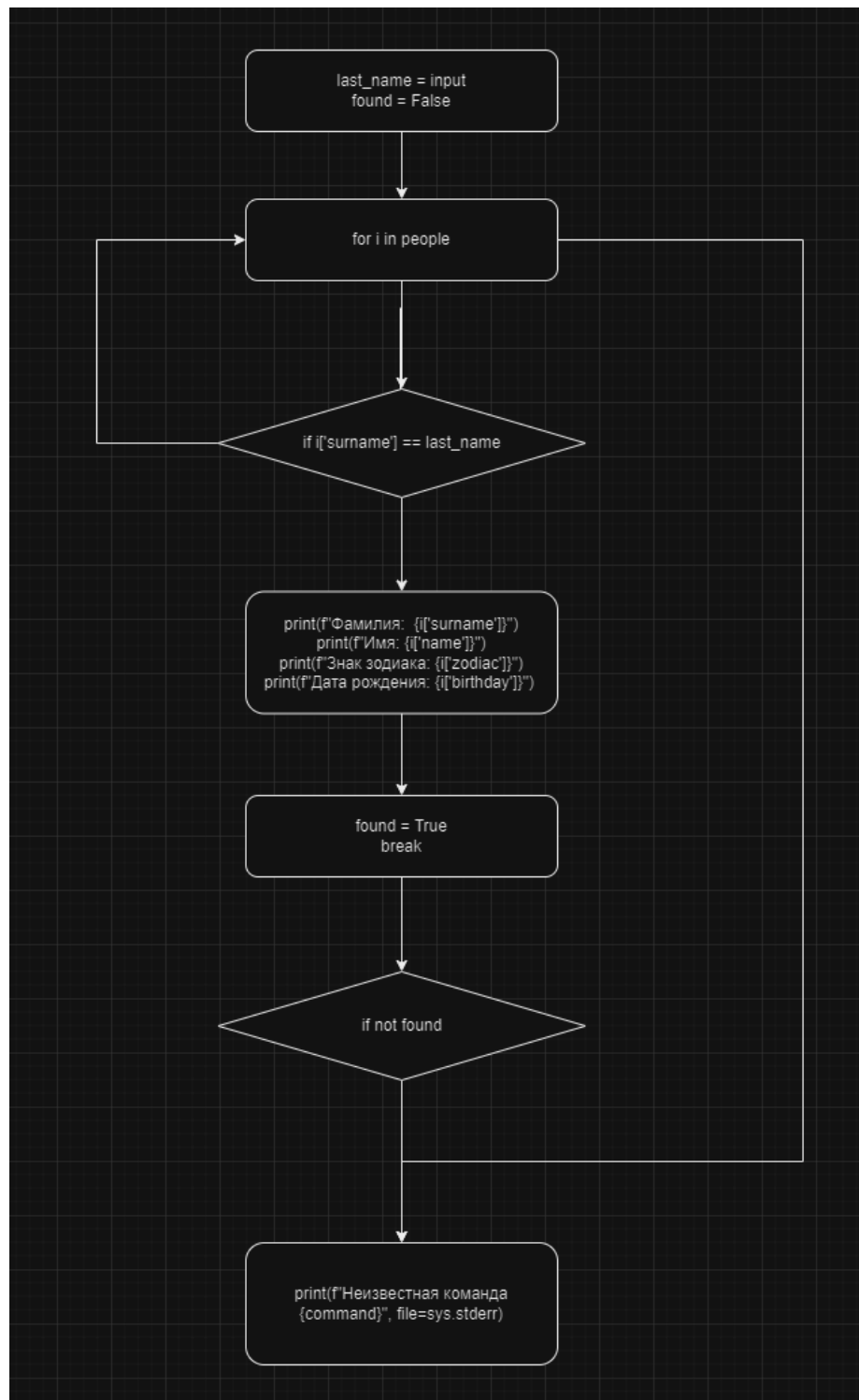


Рисунок 5.4 – UML-диаграмма команды info

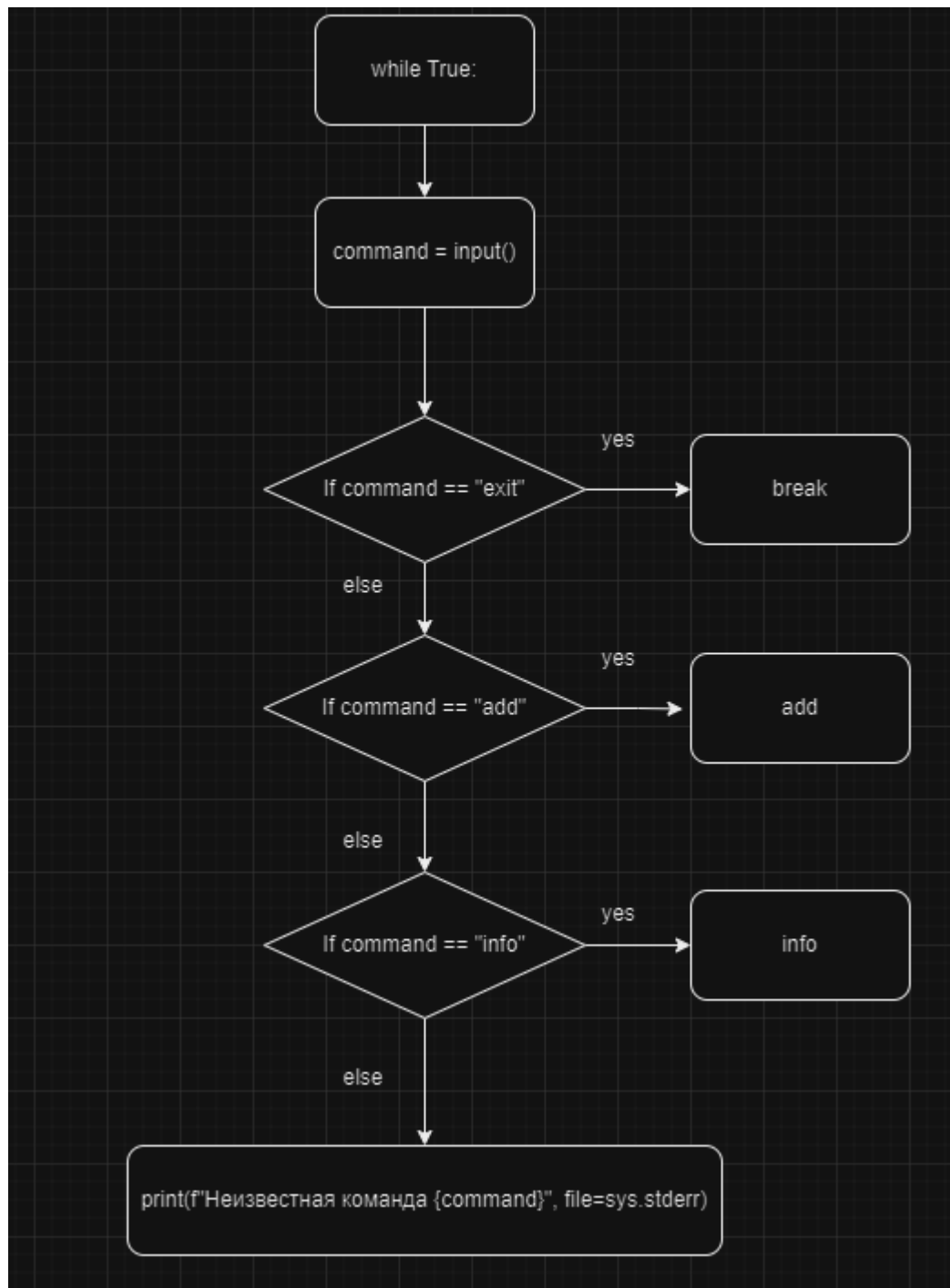


Рисунок 5.5 – UML-диаграмма логики программы

Ответы на контрольные вопросы

1. Словарь в Python - это неупорядоченная коллекция произвольных объектов с доступом по ключу. Ключи могут быть любого неизменяемого типа данных, например, строки, числа или кортежи. Значения могут быть любого типа данных, включая другие словари. Словари создаются с помощью фигурных скобок { } или функции dict().

2. Да, функция `len()` может использоваться при работе со словарями. Она возвращает количество элементов в словаре.
3. В Python есть несколько методов для обхода словарей. Например, метод `keys()` возвращает список всех ключей в словаре, а метод `values()` возвращает список всех значений. Метод `items()` возвращает список кортежей, содержащих пары ключ-значение.
4. Для получения значения из словаря по ключу можно использовать оператор `[]`. Например, если `d` - это словарь, то `d['ключ']` вернет значение, связанное с ключом 'ключ'.
5. Для установки значения в словаре по ключу можно использовать оператор `[]`. Например, если `d` - это словарь, то `d['ключ'] = 'значение'` установит значение 'значение' для ключа 'ключ'.
6. Словарь включений (dictionary comprehension) - это способ создания словаря с помощью выражения в фигурных скобках `{}`. Он позволяет создавать словари на основе других словарей или итерируемых объектов.
7. Функция `zip()` в Python используется для объединения двух или более списков в кортежи. Например, `zip([1, 2, 3], ['a', 'b', 'c'])` вернет список кортежей `[(1, 'a'), (2, 'b'), (3, 'c')]`. Эта функция может быть полезна, когда нужно объединить данные из нескольких списков.
8. Модуль `datetime` в Python предоставляет классы для работы с датой и временем. Например, класс `datetime.date` представляет дату, а класс `datetime.time` представляет время. Модуль также предоставляет функции для работы с датами и временем, такие как `datetime.datetime.now()`, которая возвращает текущую дату и время.