

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Работа со словарями в языке Python»**

**Отчет по лабораторной работе № 2.6
по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1
Коновалова В.Н. « » 2022г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Ход работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте пример лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

Пример 1. Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в список, состоящий из заданных словарей;
- записи должны быть размещены по алфавиту;
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение.

Код:

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

import sys
from datetime import date

if __name__ == '__main__':
```

```

# Список работников.
workers = []

# Организовать бесконечный цикл запроса команд.
while True:
    # Запросить команду из терминала.
    command = input(">>> ").lower()

    # Выполнить действие в соответствие с командой.
    if command == 'exit':
        break

    elif command == 'add':
        # Запросить данные о работнике.
        name = input("Фамилия и инициалы? ")
        post = input("Должность? ")
        year = int(input("Год поступления? "))

        # Создать словарь.
        worker = {
            'name': name,
            'post': post,
            'year': year,
        }

        # Добавить словарь в список.
        workers.append(worker)
        # Отсортировать список в случае необходимости.
        if len(workers) > 1:
            workers.sort(key=lambda item: item.get('name', ''))

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

        # Вывести данные о всех сотрудниках.
        for idx, worker in enumerate(workers, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                    idx,
                    worker.get('name', ''),
                    worker.get('post', ''),
                    worker.get('year', 0)
                )
            )

        print(line)

    elif command.startswith('select '):

```

```

# Получить текущую дату.
today = date.today()

# Разбить команду на части для выделения номера года.
parts = command.split(' ', maxsplit=1)
# Получить требуемый стаж.
period = int(parts[1])

# Инициализировать счетчик.
count = 0
# Проверить сведения работников из списка.
for worker in workers:
    if today.year - worker.get('year', today.year) >= period:
        count += 1
        print(
            '{:>4}: {}'.format(count, worker.get('name', ''))
        )

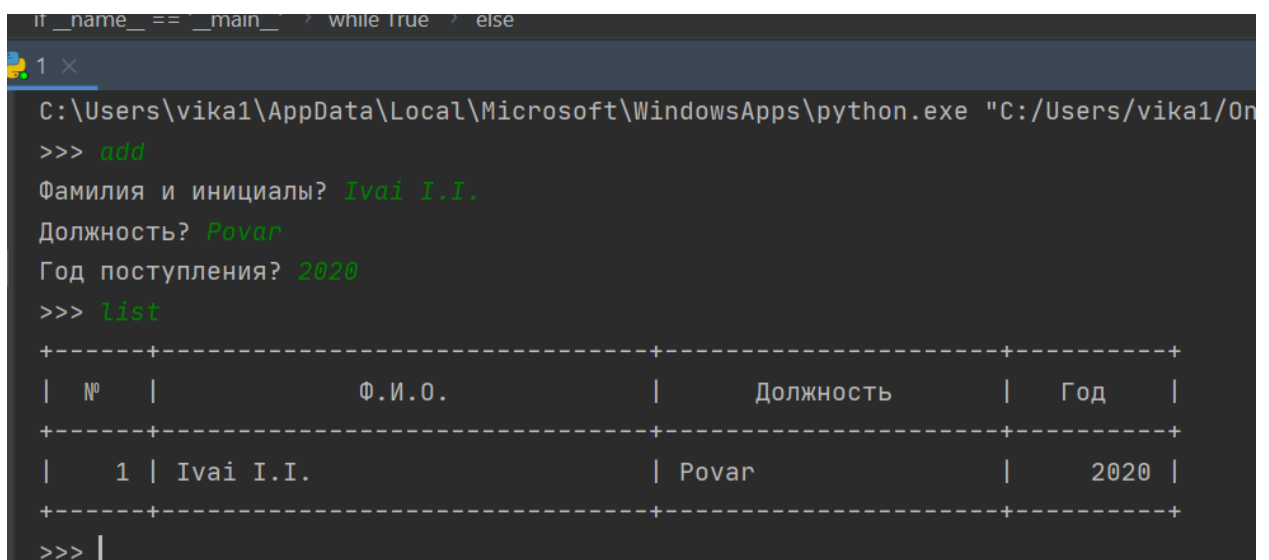
# Если счетчик равен 0, то работники не найдены.
if count == 0:
    print("Работники с заданным стажем не найдены.")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <стаж> - запросить работников со стажем;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

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

```

8. Приведите в отчете скриншоты результатов выполнения примера при различных исходных данных, вводимых с клавиатуры.



```

if __name__ == '__main__': while True: else
1 x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\python.exe "C:/Users/vika1/On
>>> add
Фамилия и инициалы? Ivai I.I.
Должность? Povar
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Ivai I.I. | Povar | 2020 |
+-----+-----+-----+-----+
>>>

```

Рисунок 6 – Результат работы программы

9. Решите задачу: 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.

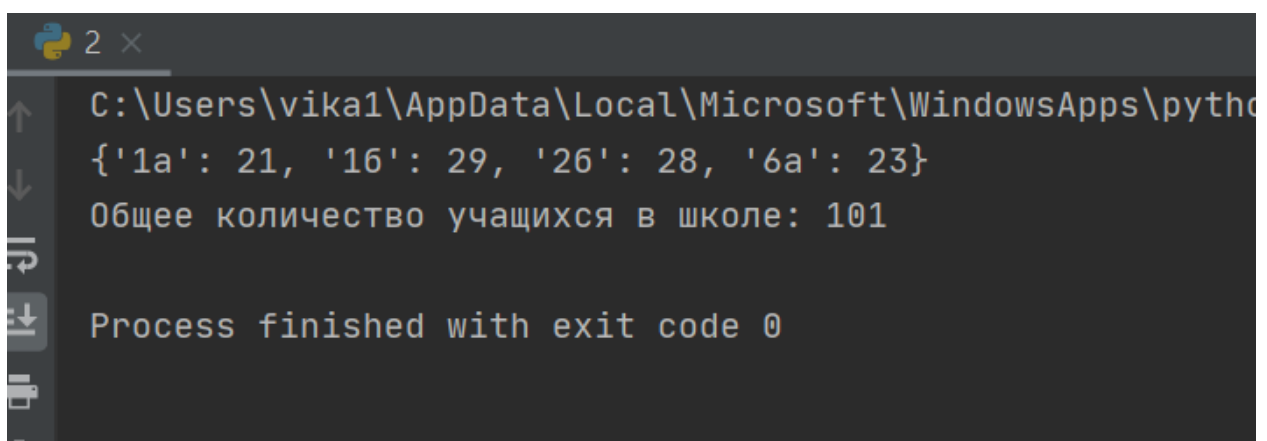
Код:

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

if __name__ == '__main__':
    school = {'1а': 24, '1б': 29, '2б': 28, '6а': 19, '7в': 30}

    school['1а'] = 21
    school['6а'] = 23
    del school['7в']
    print(school)

    summ = sum(klass for klass in school.values())
    print(f"Общее количество учащихся в школе: {summ}")
```



```
2 x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\python3
{'1а': 21, '1б': 29, '2б': 28, '6а': 23}
Общее количество учащихся в школе: 101
Process finished with exit code 0
```

Рисунок 7 – Результат работы программы

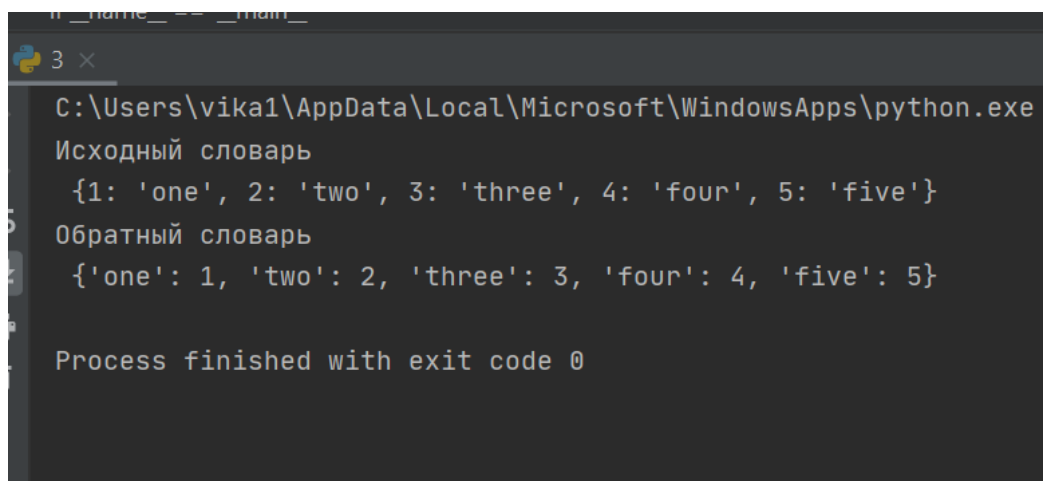
10. Зафиксируйте сделанные изменения в репозитории.

11. Решите задачу: 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.

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
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.
"""
if __name__ == '__main__':
    my_dict = {
        1: 'one',
        2: 'two',
        3: 'three',
        4: 'four',
        5: 'five'
    }
    new_dict = {}
    for key, value in my_dict.items():
        new_dict[value] = key
    print(f"Исходный словарь\n {my_dict}")
    print(f"Обратный словарь\n {new_dict}")
```



```
3 x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\python.exe
Исходный словарь
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
Обратный словарь
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}

Process finished with exit code 0
```

Рисунок 9 – Результат работы программы

12. Зафиксируйте сделанные изменения в репозитории.

- Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Использовать словарь, содержащий следующие ключи: название пункта назначения;
номер
поезда; время отправления. Написать программу, выполняющую следующие
действия:
ввод с клавиатуры данных в список, состоящий из словарей заданной структуры;
записи должны
быть упорядочены по номерам поездов;
вывод на экран информации о поезде, номер которого введен с клавиатуры; если
таких поездов нет,
выдать на дисплей соответствующее сообщение.
"""
import sys

if __name__ == '__main__':

    trains = []

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

        if command == 'exit':
            break
        elif command == 'add':
            name = input("Название пункта назначения: ")
            num = int(input("Номер поезда: "))
            time = input("Время отправления: ")

            train = {
                'name': name,
                'num': num,
                'time': time,
            }

            trains.append(train)
            if len(trains) > 1:
                trains.sort(key=lambda item: item.get('num', ''))

        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 17,
            )
            print(line)
            print(
                '| {:^4} | {:^30} | {:^20} | {:^17} |'.format(
                    "№",
                    "Пункт назначения",
                    "Номер поезда",
                    "Время отправления"
                )
            )
            print(line)

            for idx, train in enumerate(trains, 1):
                print(
                    '| {:>4} | {:<30} | {:<20} | {:>17} |'.format(
                        idx,
                        train.get('name', ''),
                        train.get('num', ''),

```

```

        train.get('time', 0)
    )

    print(line)

elif command.startswith('select '):

    parts = command.split(' ', maxsplit=2)

    number = int(parts[1])

    count = 0
    for train in trains:
        if train.get('num') == number:
            count += 1
            print('Номер поезда:', train.get('num', ''))
            print('Пункт назначения:', train.get('name', ''))
            print('Время отправления:', train.get('time', ''))

    if count == 0:
        print("Таких поездов нет!")

elif command == 'help':
    print("Список команд:\n")
    print("add - добавить поезд;")
    print("list - вывести список поездов;")
    print("select <номер поезда> - запросить информацию о выбранном
поезде;")

    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

```

+-----+-----+-----+-----+
| № | Пункт назначения | Номер поезда | Время отправления |
+-----+-----+-----+-----+
| 1 | Nevinnomissk | 1 | 15:00 |
| 2 | Stavopol | 2 | 12:50 |
+-----+-----+-----+-----+
>>> select 2
Номер поезда: 2
Пункт назначения: Stavopol
Время отправления: 12:50
>>> |

```

Рисунок 11 – Результат работы программы


```
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\python
>>> help
Список команд:

add - добавить поезд;
list - вывести список поездов;
select <номер поезда> - запросить информацию о выбранном поезде;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Название пункта назначения: Nevinnomissk
Номер поезда: 1
Время отправления: 15:00
>>> add
Название пункта назначения: Stavopol
Номер поезда: 2
```

Рисунок 11 – Результат работы программы

Вопросы для защиты работы

1. Что такое словари в языке Python?

Словари представляют собой структуры данных, в которых уникальные ключи отображают значения. Ключ и значение разделяются двоеточием, пары ключ-значение отделяются запятыми, а словарь целиком ограничивается фигурными скобками {}.

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

Да, она возвращает количество пар {key:value}

3. Какие методы обхода словарей Вам известны?

С помощью цикла:

```
num = {1: "one", 2: "two", ...}
```

```
for i in num:
```

```
    print(i) ///Выведет ключи
```

4. Какими способами можно получить значения из словаря по ключу?

```
for i in num:
```

```
    print(num [i])
```

или

```
for key, value in nums.items():
```

```
    print(key, 'is', value) // выведет пары ключ-значение
```

5. Какими способами можно установить значение в словаре по ключу?

x['one'] = 1, где one – ключ, а 1 – значение

6. Что такое словарь включений?

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

Функция zip() в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные.

Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию zip().

```
employee_numbers = [2, 9, 18, 28]
```

```
employee_names = ["Дима", "Марина", "Андрей", "Никита"]
```

```
zipped_values = zip(employee_names, employee_numbers)
```

```
zipped_list = list(zipped_values)
```

```
print(zipped_list)
```

Вывод: [('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]

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

Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать.

`datetime` включает различные компоненты. Так, он состоит из объектов следующих типов: `date` — хранит дату, `time` — хранит время, `datetime` — хранит дату и время.

Метод `now()` возвращает текущие дату и время с учетом локальных настроек.

`today()` - объект `datetime` из текущей даты и времени. Работает также, как и `datetime.now()` со значением `tz=None`.

`fromtimestamp(timestamp)` - дата из стандартного представления времени.

`toordinal()` - количество дней, прошедших с 01.01.1970

`replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]])` - возвращает новый объект `datetime` с изменёнными атрибутами

`fromordinal(ordinal)` - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970