

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.6

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Примаков В.Д « »_____20__г.

Подпись студента _____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____

(подпись)

ВЫПОЛНЕНИЕ

Пример

```
#!/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 соответствии 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)

```

```

>>> add
Фамилия и инициалы? Примаков В.Д.
Должность? Студент
Год поступления? 2020
>>> add
Фамилия и инициалы? Турклиев В.Н.
Должность? Студент
Год поступления? 2020
>>> add
Фамилия и инициалы? Катарин П.В.
Должность? Студент
Год поступления? 2016
>>> add
Фамилия и инициалы? Путин В.В.
Должность? Царь
Год поступления? 1999
>>> list

```

№	Ф.И.О.	Должность	Год
1	Катарин П.В.	Студент	2016
2	Примаков В.Д.	Студент	2020
3	Путин В.В.	Царь	1999
4	Турклиев В.Н.	Студент	2020

```

>>>

```

```
>>> select 1
1: Катарин П.В.
2: Примаков В.Д.
3: Путин В.В.
4: Турклиев В.Н.
>>> select 4
1: Катарин П.В.
2: Путин В.В.
>>> select 20
1: Путин В.В.
```

Задание 9.

Изначальный словарь:

```
school = {'1a': '21', '2a': '19', '36': '15', '4B': '23'}
```

А) В одном из классов изменилось количество учащихся:

```
school = {'1a': '19', '2a': '19', '36': '15', '4B': '23'}
```

Б) В школе появился новый класс:

```
school = {'1a': '19', '2a': '19', '36': '15', '4B': '23', '5a': '18'}
```

В) В школе был расформирован один класс:

```
school = {'1a': '19', '36': '15', '48': '23', '5a': '18'}
```

Общее число школьников:

```
pythonProject C:\Users\surai5a\Desktop\pythonProject\Scripts\Scripts\python.exe "C:/Users/surai5a/Desktop/pythonProject/Scripts/Scripts/python.exe"
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  school = {'1a': '19', '36': '15', '4b': '23', '5a': '18'}
5  sum = 0
6
7  for i in school:
8      sum += int(school[i])
9
10 print(f"Общее количество учащихся: {sum}")
```

Задание 11.

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.

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

if __name__ == '__main__':
    dict1 = {
        1: 'one',
        2: 'two',
        3: 'three',
        4: 'four'
    }
    dict2 = {}
    for x, y in dict1.items():
        dict2.setdefault(y, x)
    print(f"Dict1: {dict1}")
    print(f"Dict2: {dict2}")
```

```
Dict1: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
Dict2: {'one': 1, 'two': 2, 'three': 3, 'four': 4}
```

```
Process finished with exit code 0
```

Индивидуальное задание

18. 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.

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

import sys

if __name__ == '__main__':

    goods = []

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

        if command == 'exit':
            break

        elif command == 'add':
            name = input("Название товара: ")
            shop = input("Название магазина: ")
            price = input("Стоимость: ")

            good = {
                'name': name,
                'shop': shop,
                'price': price,
            }

            goods.append(good)
            # Отсортировать список в случае необходимости.
            if len(goods) > 1:
                goods.sort(key=lambda item: item.get('shop', ''))

        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 8
            )
```

```

    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
            "№",
            "Название",
            "Магазин",
            "Цена"
        )
    )
    print(line)

    for idx, good in enumerate(goods, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                idx,
                good.get('name', ''),
                good.get('shop', ''),
                good.get('price', 0)
            )
        )
    print(line)

elif command.startswith('select '):

    parts = command.split(' ', maxsplit=1)
    shopName = parts[1]

    count = 0

    for good in goods:
        if shopName == good.get('shop', shopName):
            count += 1
            print(
                '{:>4}: {}'.format(count, good.get('name', ''))
            )

    if count == 0:

```

```

        print("Такого магазина не существует либо нет товаров.")

elif command == 'help':
    print("Список команд:\n")
    print("add - добавить товар;")
    print("list - вывести список товаров;")
    print("select <имя магазина> - запросить товары магазина;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```



```
>>> help
Список команд:

add - добавить товар;
list - вывести список товаров;
select <имя магазина> - запросить товары магазина;
help - отобразить справку;
exit - завершить работу с программой.
>>> |
```

```
Название товара: Сахар
Название магазина: Пятерочка
Стоимость: 55p
>>> add
Название товара: Соль
Название магазина: Пятерочка
Стоимость: 45p
>>> add
Название товара: Печенье
Название магазина: Магнит
Стоимость: 40p
>>> add
Название товара: Торт
Название магазина: Магнит
Стоимость: 250p
>>> list
+-----+-----+-----+-----+
| № |      Название      | Магазин |      Цена |
+-----+-----+-----+-----+
|  1 | Печенье            | Магнит  |    40p |
|  2 | Торт               | Магнит  |   250p |
|  3 | Сахар              | Пятерочка |    55p |
|  4 | Соль               | Пятерочка |    45p |
+-----+-----+-----+-----+
>>>
```

```
>>> select пятерочка
пятерочка
    1: Сахар
    2: Соль
>>> select магнит
магнит
    1: Печенье
    2: Торт
>>> |
```

```
>>> select перекресток
перекресток
Такого магазина не существует либо нет товаров.
>>> |
```

Ссылки на репозитории

GitHub - https://github.com/surai5a/laba_2_6

Ответы на контрольные вопросы

1. Словарь – это структура данных (ассоциативный массив), предназначенный для хранения произвольных объектов с доступом по ключу.
2. Для поиска количества пар в словаре может быть использована функция `len()`
3. Обойти словарь можно с помощью:
 - a. Простого перебора ключей и значений с помощью `for` (`for i in dict`)
 - b. Перебора значений и ключей с помощью `for` и методов `.values()` и `.keys()`, соответственно.
 - c. Перебора пар, в виде кортежей, с использованием метода `.items()` и цикла `for`.
4. Получить значение по ключу можно с помощью метода `.get()`
5. С помощью метода `.setdefault(key, val)` можно добавить новую пару в словарь, либо изменить значение уже существующего ключа.
6. Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.
7. Функция `zip()` создает кортежи из заданных наборов чисел. С помощью этой функции можно задать, к примеру, словарь:
 - a. `names = [Am, Bm, Cm]`
 - b. `nums = [0, 4, 8]`
 - c. `dict(zip(names, nums))`
 - d. `{‘Am’: 0, ‘Bm’: 4, ‘Cm’: 8}`
8. Модуль `datetime` имеет несколько классов:
 - a. `datetime.date(year, month, day)` - стандартная дата.
 - b. `datetime.time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None)` - стандартное время, не зависит от даты.
 - c. `datetime.timedelta` - разница между двумя моментами времени, с точностью до микросекунд.
 - d. `datetime.tzinfo` - абстрактный базовый класс для информации о временной зоне.
 - e. `datetime.datetime(year, month, day, hour=0, minute=0, second=0, microsecond=0, tzinfo=None)` - комбинация даты и времени.

В общем и целом библиотека `datetime` предоставляет доступ к любым видам представления времени, даты и всему тому, что имеет отношение к этому.