

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

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

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №9**

**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-20-1

Ваньянц И.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2021

# ХОД РАБОТЫ

## 1. Пример 1.

```
#!/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)

```

Рисунки 1, 2, 3 – Код программы

```

>>> add
Фамилия и инициалы? Ваньянц И.М.
Должность? Студент
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Ваньянц И.М. | Студент | 2020 |
+-----+-----+-----+-----+
>>> |

```

Рисунок 4 – Пример работы программы

## 2. Индивидуальное задание

Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. 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 пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.



```

elif command == 'list':
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 15
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
            "No",
            "Пункт назначения",
            "Номер рейса",
            "Тип самолета"
        )
    )
    print(line)

    for idx, flight in enumerate(flights, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:<15} |'.format(
                idx,
                flight.get('flight_destination', ''),
                flight.get('flight_number', ''),
                flight.get('airplane_type', 0)
            )
        )
    print(line)

```

```

elif command.startswith('select '):
    parts = command.split(' ', maxsplit=1)
    airplane_type = (parts[1].capitalize())
    print(f"Для типа самолета {airplane_type}:")
    count = 0
    for flight in flights:
        if flight.get('airplane_type') == airplane_type:
            count += 1
            print(
                '{:>4}: Пункт назначения: {}; Номер рейса: {}'.format(
                    count,
                    flight.get('flight_destination',
                                ''),
                    flight.get('flight_number', ''))
            )
    if count == 0:
        print("рейсы не найдены")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить рейс;")
    print("list - вывести список всех рейсов;")
    print("select <тип самолета> - запросить рейсы указанного типа "
          "самолета;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунки 5, 6, 7 – код программы

```

>>> add
Введите название пункта назначения Сочи
Введите номер рейса 13
Введите тип самолета Грузовой
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Сочи | 13 | Грузовой |
+-----+-----+-----+-----+
>>> |

```

Рисунок 8 – пример выполнения программы



## ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Что такое словари в языке Python?  
Словарь ( dict ) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу. Данные в словаре хранятся в формате ключ – значение.
- 2) Может ли функция len() быть использована при работе со словарями?  
Да
- 3) Какие методы обхода словарей Вам известны?  
for i in nums: print(nums[i]) for key, value in nums.items(): print(key, 'is', value)  
Методы словаря keys() и values() позволяют получить отдельно перечни ключей и значений.
- 4) Какими способами можно получить значения из словаря по ключу?  
dict["key"] dict.get("key")
- 5) Какими способами можно установить значение в словаре по ключу?  
dict.setdefault("key", "value") dict["key"] = "value"
- 6) Что такое словарь включений?  
Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.
- 7) Самостоятельно изучите возможности функции zip() приведите примеры ее использования.  
Функция zip() в Python создает итератор, который объединяет элементы из нескольких источников данных. У функции zip() множество сценариев применения.
- 8) Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с датой и временем обладает этот модуль?  
datetime.fromordinal(ordinal) - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.  
datetime.now(tz=None) - объект datetime из текущей даты и времени.  
datetime.combine(date, time) - объект datetime из комбинации объектов date и time.  
datetime.strptime(date\_string, format) - преобразует строку в datetime (так же, как и функция strptime из модуля time).  
datetime.strftime(format) - см. функцию strftime из модуля time.  
datetime.date() - объект даты (с отсечением времени).  
datetime.time() - объект времени (с отсечением даты).  
datetime.toordinal() - количество дней, прошедших с 01.01.1970.  
datetime.timestamp() - возвращает время в секундах с начала эпохи.  
datetime.weekday() - день недели в виде числа, понедельник - 0, воскресенье - 6.  
datetime.isocalendar() - кортеж (год в формате ISO, ISO номер недели, ISO день недели).  
datetime.isoformat(sep='T') - красивая строка вида "YYYYMMDDTHH:MM:SS.mmmmmmm" или, если microsecond == 0, "YYYYMMDDTHH:MM:SS"