

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №9 по дисциплине «Основы  
программной инженерии»

Выполнил:  
Чернова Софья Андреевна,  
2 курс, группа ПИЖ-б-о-20-1,  
Проверил:  
Доцент кафедры инфокоммуникаций,  
Воронкин Р.А.

Ставрополь, 2021 г

1. Ход работы:
- 1.1 Пример (рис. 1, 2, 3, 4, 5, 6):

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5      from datetime import date
6
7  ▶  if __name__ == '__main__':
8      # Список работников.
9      workers = []
10
11     # Организовать бесконечный цикл запроса команд.
12     while True:
13         # Запросить команду из терминала.
14         command = input(">>> ").lower()
15
16         # Выполнить действие в соответствии с командой.
17         if command == 'exit':
18             break
19
20         elif command == 'add':
21             # Запросить данные о работнике.
22             name = input("Фамилия и инициалы? ")
23             post = input("Должность? ")
24             year = int(input("Год поступления? "))
25
26             # Создать словарь.
27             worker = {
28                 'name': name,
29                 'post': post,
30                 'year': year,
31             }
```

Рисунок 1 – код программы

```

31     }
32
33     # Добавить словарь в список.
34     workers.append(worker)
35     # Отсортировать список в случае необходимости.
36     if len(workers) > 1:
37         workers.sort(key=lambda item: item.get('name', ''))
38
39     elif command == 'list':
40         # Заголовок таблицы.
41         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
42             '-' * 4,
43             '-' * 30,
44             '-' * 20,
45             '-' * 8
46         )
47         print(line)
48         print(
49             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
50                 "№",
51                 "Ф.И.О.",
52                 "Должность",
53                 "Год"
54             )
55         )
56         print(line)
57
58     # Вывести данные о всех сотрудниках.
59     for idx, worker in enumerate(workers, 1):
60         print(
61             '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(

```

Рисунок 2 – код программы (продолжение)

```

61         '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
62             idx,
63             worker.get('name', ''),
64             worker.get('post', ''),
65             worker.get('year', 0)
66         )
67     )
68
69     print(line)
70
71 elif command.startswith('select '):
72     # Получить текущую дату.
73     today = date.today()
74
75     # Разбить команду на части для выделения номера года.
76     parts = command.split(' ', maxsplit=1)
77     # Получить требуемый стаж.
78     period = int(parts[1])
79
80     # Инициализировать счетчик.
81     count = 0
82     # Проверить сведения работников из списка.
83     for worker in workers:
84         if today.year - worker.get('year', today.year) >= period:
85             count += 1
86             print(
87                 '{:>4}: {}'.format(count, worker.get('name', ''))
88             )
89
90     # Если счетчик равен 0, то работники не найдены.
91     if count == 0:

```

Рисунок 3 – код программы (продолжение)

```

91         if count == 0:
92             print("Работники с заданным стажем не найдены.")
93
94         elif command == 'help':
95             # Вывести справку о работе с программой.
96             print("Список команд:\n")
97             print("add - добавить работника;")
98             print("list - вывести список работников;")
99             print("select <стаж> - запросить работников со стажем;")
100            print("help - отобразить справку;")
101            print("exit - завершить работу с программой.")
102
103        else:
104            print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 4 – код программы (конец)

```

>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? Чернова С.А.
Должность? Программист
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Чернова С.А. | Программист | 2020 |
+-----+-----+-----+-----+

```

Рисунок 5 – результат работы программы при правильном вводе

```

>>> ads
>>> Неизвестная команда ads

```

Рисунок 6 – результат работы программы при неверном вводе

1.2 Индивидуальное задание №5, вариант 24 (рис. 7, 8, 9, 10):

```

1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  ▶ if __name__ == '__main__':
7      flights = []
8      while True:
9          command = input(">>> ").lower()
10         if command == 'exit':
11             break
12
13         elif command == 'add':
14             flight_destination = input("Введите название пункта назначения ")
15             flight_number = input("Введите номер рейса ")
16             airplane_type = input("Введите тип самолета ")
17             flight = {
18                 'flight_destination': flight_destination,
19                 'flight_number': flight_number,
20                 'airplane_type': airplane_type,
21             }
22             flights.append(flight)
23             if len(flights) > 1:
24                 flights.sort(
25                     key=lambda item:
26                         item.get('flight_destination', ''))
27
28         elif command == 'list':
29             line = '+-{}-+-{}-+-{}-+-{}-+'.format(
30                 '-' * 4,
31                 '-' * 30,

```

Рисунок 7 – код программы

```

31         '-' * 30,
32         '-' * 20,
33         '-' * 15
34     )
35     print(line)
36     print(
37         '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
38             "No",
39             "Пункт назначения",
40             "Номер рейса",
41             "Тип самолета"
42         )
43     )
44     print(line)
45
46     for idx, flight in enumerate(flights, 1):
47         print(
48             '| {:>4} | {:<30} | {:<20} | {:<15} |'.format(
49                 idx,
50                 flight.get('flight_destination', ''),
51                 flight.get('flight_number', ''),
52                 flight.get('airplane_type', 0)
53             )
54         )
55     print(line)
56
57     elif command.startswith('select '):
58         parts = command.split(' ', maxsplit=1)
59         airplane_type = (parts[1].capitalize())
60         print(f"Для типа самолета {airplane_type}:")
61         count = 0

```

Рисунок 8 – код программы (продолжение)

```

61         count = 0
62         for flight in flights:
63             if flight.get('airplane_type') == airplane_type:
64                 count += 1
65                 print(
66                     '{:>4}: Пункт назначения: {}; Номер рейса: {}'.format(
67                         count,
68                         flight.get('flight_destination',
69                             ''),
70                         flight.get('flight_number', ''))
71                 )
72         if count == 0:
73             print("рейсы не найдены")
74
75     elif command == 'help':
76         print("Список команд:\n")
77         print("add - добавить рейс;")
78         print("list - вывести список всех рейсов;")
79         print("select <тип самолета> - запросить рейсы указанного типа "
80             "самолета;")
81         print("help - отобразить справку;")
82         print("exit - завершить работу с программой.")
83     else:
84         print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 9 – код программы (конец)

```

>>> help
Список команд:

add - добавить рейс;
list - вывести список всех рейсов;
select <тип самолета> - запросить рейсы указанного типа самолета;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Введите название пункта назначения Moscow
Введите номер рейса M134
Введите тип самолета Passenger
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Moscow | M134 | Passenger |
+-----+-----+-----+-----+
>>> select Passenger
Для типа самолета Passenger:
1: Пункт назначения: Moscow; Номер рейса: M134

```

Рисунок 10 – результат работы программы



## 2. 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. Так что если, например, надо перебрать только значения или только ключи, лучше воспользоваться одним из этих методов.

### 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. Например, она пригодится, если нужно создать набор словарей из двух массивов, каждый из которых содержит имя и номер сотрудника.

### 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 "YYYY-MMDDTHH:MM:SS.mmmmmm" или, если `microsecond == 0`, "YYYY-MMDDTHH:MM:SS"