

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 студент
2 курса, группы ПИЖ-б-о-20-1
Тотубалина С.С.

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

Ход работы

```
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,
```

Рис. 1 – код программы lab.9_ex.1.py

```

30         'year': year,
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.

```

Рис. 2 – код программы lab.9_ex.1.py

```

59     for idx, worker in enumerate(workers, 1):
60         print(
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', ''))

```

Рис. 3 – код программы lab.9_ex.1.py

```

88         )
89
90         # Если счетчик равен 0, то работники не найдены.
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)

```

Рис. 4 – код программы lab.9_ex.1.py

```

>>> 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? Тотубалина С. С.
Должность? Программист
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |   Год   |
+-----+-----+-----+-----+
|  1 | Тотубалина С. С.          | Программист         |  2020   |
+-----+-----+-----+-----+

```

Рис. 5 – результат работы программы lab.5_ex.2.py при правильном вводе

```

>>> help
>>> Неизвестная команда help

```

Рис. 6 – результат работы программы lab.5_ex.2.py при неправильном вводе

```

1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  ▶  if __name__ == '__main__':
8      flights = []
9      while True:
10         command = input(">>> ").lower()
11         if command == 'exit':
12             break
13
14         elif command == 'add':
15             flight_destination = input("Введите название пункта назначения ")
16             flight_number = input("Введите номер рейса ")
17             airplane_type = input("Введите тип самолета ")
18             flight = {
19                 'flight_destination': flight_destination,
20                 'flight_number': flight_number,
21                 'airplane_type': airplane_type,
22             }
23             flights.append(flight)
24             if len(flights) > 1:
25                 flights.sort(
26                     key=lambda item:
27                         item.get('flight_destination', ''))
28
29         elif command == 'list':
30             line = '+-{}-+-{}-+-{}-+-{}-+'.format(
31                 '-' * 4,
32                 '-' * 30,
33                 '-' * 20,
34                 '-' * 15
35             )
36             print(line)
37             print(
38                 '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
39                     "No",
40                     "Пункт назначения",

```

Рис. 7 – код программы individual_9.py (Вариант №22)

```

41         "Номер рейса",
42         "Тип самолета"
43     )
44 )
45 print(line)
46
47 for idx, flight in enumerate(flights, 1):
48     print(
49         '| {:>4} | {:<30} | {:<20} | {:<15} |'.format(
50             idx,
51             flight.get('flight_destination', ''),
52             flight.get('flight_number', ''),
53             flight.get('airplane_type', 0)
54         )
55     )
56     print(line)
57
58 elif command.startswith('select '):
59     parts = command.split(' ', maxsplit=1)
60     airplane_type = (parts[1].capitalize())
61     print(f"Для типа самолета {airplane_type}:")
62     count = 0
63     for flight in flights:
64         if flight.get('airplane_type') == airplane_type:
65             count += 1
66             print(
67                 '{:>4}: Пункт назначения: {}; Номер рейса: {}'.format(
68                     count,
69                     flight.get('flight_destination',
70                         ''),
71                     flight.get('flight_number', ''))
72             )
73     if count == 0:
74         print("рейсы не найдены")
75
76 elif command == 'help':
77     # Вывести справку о работе с программой.
78     print("Список команд:\n")
79     print("add - добавить рейс;")
80     print("list - вывести список всех рейсов;")

```

Рис. 8 – код программы individual_9.py (Вариант №22)

```

81     print("select <тип самолета> - запросить рейсы указанного типа "
82           "самолета;")
83     print("help - отобразить справку;")
84     print("exit - завершить работу с программой.")
85 else:
86     print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рис. 9 – код программы individual_9.py (Вариант №22)

```
>>> add
Введите название пункта назначения Армавир
Введите номер рейса 5
Введите тип самолета Пассажирский
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Армавир | 5 | Пассажирский |
+-----+-----+-----+-----+
>>> exit

Process finished with exit code 0
```

Рис. 10 – результат работы программы individual_9.py (Вариант №22)

Ответы на 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"