

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:
Мамонтов Д.В.,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г

ХОД РАБОТЫ

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

Рисунок 1 – код программы

```
33         if len(workers) > 1:
34             workers.sort(key=lambda item: item.get('name', ''))
35
36         elif command == 'list':
37             # Заголовок таблицы.
38             line = '+-{}-+-{}-+-{}-+-{}-+'.format(
39                 '-' * 4,
40                 '-' * 30,
41                 '-' * 20,
42                 '-' * 8
43             )
44             print(line)
45             print(
46                 '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
47                     "№",
48                     "Ф.И.О.",
49                     "Должность",
50                     "Год"
51                 )
52             )
53             print(line)
54
55             # Вывести данные о всех сотрудниках.
56             for idx, worker in enumerate(workers, 1):
57                 print(
58                     '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
59                         idx,
60                         worker.get('name', ''),
61                         worker.get('post', ''),
62                         worker.get('year', 0)
63                     )
64                 )
```

Рисунок 2 – код программы (продолжение)

```

66         print(line)
67
68     elif command.startswith('select '):
69         # Получить текущую дату.
70         today = date.today()
71
72         # Разбить команду на части для выделения номера года.
73         parts = command.split(' ', maxsplit=1)
74         # Получить требуемый стаж.
75         period = int(parts[1])
76
77         # Инициализировать счетчик.
78         count = 0
79         # Проверить сведения работников из списка.
80         for worker in workers:
81             if today.year - worker.get('year', today.year) >= period:
82                 count += 1
83                 print(
84                     '{:>4}: {}'.format(count, worker.get('name', ''))
85                 )
86
87         # Если счетчик равен 0, то работники не найдены.
88         if count == 0:
89             print("Работники с заданным стажем не найдены.")
90
91     elif command == 'help':
92         # Вывести справку о работе с программой.
93         print("Список команд:\n")
94         print("add - добавить работника;")
95         print("list - вывести список работников;")
96         print("select <стаж> - запросить работников со стажем;")
97         print("help - отобразить справку;")

```

Рисунок 3 – код программы (продолжение)

```

97         print("help - отобразить справку;")
98         print("exit - завершить работу с программой.")
99
100     else:
101         print(f"Неизвестная команда {command}", file=sys.stderr)
102

```

Рисунок 4 – код программы (конец)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Задание:

Использовать словарь, содержащий следующие ключи: фамилия, имя; номер телефона; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

Код:

```

import sys

def date_key(birth):
    data = birth.split(".")
    return (data[2], data[1], data[0])

```

```

if __name__ == '__main__':
    # Список людей.
    students = []
    # 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 человеке.
            name = input("Фамилия и имя? ")
            phone = input("Номер телефона? ")
            birth = input("Дата рождения? ")
            # Создать словарь.
            student = {
                'name': name,
                'phone': phone,
                'birth': birth,
            }
            # Добавить словарь в список.
            students.append(student)
            # Отсортировать список в случае необходимости.
            if len(students) > 1:
                students.sort(key=lambda item: date_key(item.get('birth',
'')))
        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 13
            )
            print(line)
            print(
                '| {:^4} | {:^30} | {:^20} | {:^13} |'.format(
                    "No",
                    "Фамилия и имя",
                    "Номер телефона",
                    "Дата рождения"
                )
            )
            print(line)
            # Вывести данные о всех людях.
            for idx, student in enumerate(students, 1):
                print(
                    '| {:>4} | {:<30} | {:<20} | {:>13} |'.format(
                        idx,
                        student.get('name', ''),
                        student.get('phone', ''),
                        student.get('birth', '')
                    )
                )
            print(line)
        elif command.startswith('phone '):
            # Разбить команду на части для выделения номера телефона.
            parts = command.split(' ', maxsplit=1)
            # Получить требуемый номер.
            phone = parts[1]
            # Проверить сведения людей из списка.
            found = False

```

```

        for student in students:
            if student.get('phone', '') == phone:
                print(
                    student.get('name', '')
                )
                found = True
        if not found:
            print("Нет человека с таким номером.")
    elif command == 'help':
        # Вывести справку о работе с программой.
        print("Список команд:\n")
        print("add - добавить человека;")
        print("list - вывести список людей;")
        print("phone <номер> - запросить человека по номеру;")
        print("help - отобразить справку;")
        print("exit - завершить работу с программой.")
    else:
        print(f"Неизвестная команда {command}", file=sys.stderr)

```

```

>>> add
Фамилия и имя? Мамонтов Даниил
Номер телефона? +79884426394
Дата рождения? 03.06.2002
>>> add
Фамилия и имя? Иванов Иван
Номер телефона? +79621234567
Дата рождения? 01.01.2003
>>> add
Фамилия и имя? Бэббедж Чарльз
Номер телефона? +79187924321
Дата рождения? 07.09.2002
>>> list
+-----+-----+-----+-----+
| No | Фамилия и имя | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
| 1 | Мамонтов Даниил | +79884426394 | 03.06.2002 |
| 2 | Бэббедж Чарльз | +79187924321 | 07.09.2002 |
| 3 | Иванов Иван | +79621234567 | 01.01.2003 |
+-----+-----+-----+-----+
>>> phone +79187924321
Бэббедж Чарльз
>>> phone +79187925413
Нет человека с таким номером.

```

Рисунок 1 – результат работы программы

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.strptime(format)` - см. функцию `strptime` 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"