

**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 №2.6 по дисциплине основы программной  
инженерии**

Выполнила:

Лобанова Елизавета

Евгеньевна,

2 курс, группа ПИЖ-б-о-20-1

Проверил:

Доцент кафедры прикладной

математики и

компьютерной безопасности,

Воронкин Р.А.

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2021 г.

## Ход работы

```
#!/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(
        "No",
        "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 '):
```

```

elif command.startswith('select '):
    today = date.today()
    parts = command.split(' ', maxsplit=1)
    period = int(parts[1])
    count = 0
    for worker in workers:
        if today.year - worker.get('year', today.year) >= period:
            count += 1
            print(
                '{:>4}: {}'.format(count, worker.get('name', ''))
            )

    if count == 0:
        print("Работники с заданным стажем не найдены.")

elif command == 'help':
    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)

```

Пример 1

```

>>> 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? Первый А.А.
Должность? Разработчик
Год поступления? 2011
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Первый А.А.              | Разработчик         |      2011     |
+-----+-----+-----+-----+

```

Результат 1 программы (1)

```
>>> add
Фамилия и инициалы? Тихий О.О.
Должность? Программист
Год поступления? 2015
>>> select 1
1: Первый А.А.
2: Тихий О.О.
>>> select 2
1: Первый А.А.
2: Тихий О.О.
>>> select 10
1: Первый А.А.
>>> select 15
Работники с заданным стажем не найдены.
```

Результат 1 программы (2)

```
school = {
    "1A": 32,
    "1B": 25,
    "2A": 22,
    "2B": 31,
    "3A": 32,
    "3B": 18,
}

for key, value in school.items():
    print(f" В {key} классе количество детей = {value}.")

# Часть а)
print("\nВ одном из классов поменялось количество детей, теперь:\n")
school['2B'] = 16
for key, value in school.items():
    print(f" В {key} классе количество детей = {value}.")

# Часть б)
print("\nПоявился новый класс, теперь:\n")
school.setdefault("3B", 31)
for key, value in school.items():
    print(f" В {key} классе количество детей = {value}.")
```

```
# Часть б)
print("\nПоявился новый класс, теперь:\n")
school.setdefault("3B", 31)
for key, value in school.items():
    print(f" В {key} классе количество детей = {value}.")

# Часть с)
print("\nРасформировали один класс, теперь:\n")
school.pop("1A")
for key, value in school.items():
    print(f" В {key} классе количество детей = {value}.")

count = 0
for value in school.values():
    count += value
print(f"\nВсего учеников в школе - {count}")
```

### Задание 1

```
В 1А классе количество детей = 30.  
В 1Б классе количество детей = 26.  
В 2А классе количество детей = 21.  
В 2Б классе количество детей = 32.  
В 3А классе количество детей = 28.  
В 3Б классе количество детей = 19.
```

### Результат 1 программы (1)

В одном из классов поменялось количество детей, теперь:

```
В 1А классе количество детей = 30.  
В 1Б классе количество детей = 26.  
В 2А классе количество детей = 21.  
В 2Б классе количество детей = 16.  
В 3А классе количество детей = 28.  
В 3Б классе количество детей = 19.
```

### Результат 1 программы (2)

Появился новый класс, теперь:

```
В 1А классе количество детей = 30.  
В 1Б классе количество детей = 26.  
В 2А классе количество детей = 21.  
В 2Б классе количество детей = 16.  
В 3А классе количество детей = 28.  
В 3Б классе количество детей = 19.  
В 3В классе количество детей = 31.
```

### Результат 1 программы (3)



Расформировали один класс, теперь:

В 1Б классе количество детей = 26.  
В 2А классе количество детей = 21.  
В 2Б классе количество детей = 16.  
В 3А классе количество детей = 28.  
В 3Б классе количество детей = 19.  
В 3В классе количество детей = 31.

Результат 1 программы (4)

Всего учеников в школе - 141

Результат 1 программы (5)

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

if __name__ == '__main__':
    some_dict = {
        1: "abc",
        2: "home",
        3: "test",
        4: "task"
    }
    print(f"Словарь до изменений:\n{some_dict}")
    dict_items = some_dict.items()
    changed_dict = {i: j for j, i in dict_items}
    print(f"Словарь после изменений:\n{changed_dict}")
```

Задание 2

Словарь до изменений:

```
{1: 'abc', 2: 'home', 3: 'test', 4: 'task'}
```

Словарь после изменений:

```
{'abc': 1, 'home': 2, 'test': 3, 'task': 4}
```

Process finished with exit code 0

Результат 2 программы (1)

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

import sys
from datetime import datetime

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 имя: ")
            tel = input("Номер телефона: ")
            dateString = input("День рождения: ")
            # Создать словарь.
```

```
worker = {
    'name': name,
    'tel': tel,
    'date': datetime.strptime(dateString, "%Y-%m-%d")
}

# 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,
        '-' * 16
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^16} |'.format(
            "No",
            "Ф.И.",
```

```
        "Телефон",
        "Дата рождения"
    )
)
print(line)
# Вывести данные о всех сотрудниках.
for idx, worker in enumerate(workers, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>16} |'.format(
            idx,
            worker.get('name', ''),
            worker.get('tel', ''),
            worker.get('date', 0)
        )
    )
print(line)

elif command.startswith('select '):
    # 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 worker['date'].month == period:
        count += 1
        print(
            '{:>4}: {}'.format(count, worker.get('name', ''))
        )

# Если счетчик равен 0, то работники не найдены.
if count == 0:
    print("В этом месяце ни у одного из работников нет дня рождения.")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <месяц> - месяц рождения работника;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

#### Индивидуальное задание 1

```

>>> add
Фамилия и имя: Провин Денис
Номер телефона: 12-45-66
День рождения: 1999-03-08
>>> add
Фамилия и имя: Втеркин Павел
Номер телефона: 34-08-21
День рождения: 1990-02-09
>>> add
Фамилия и имя: Трикин Иван
Номер телефона: 33-67-39
День рождения: 2000-03-04

```

#### Результат индивидуального задания 1 (1)

```
>>> select 03
1: Провин Денис
2: Трикин Иван
```

Результат индивидуального задания 1 (2)

#### 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.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"