

**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 Работа со словарями в языке Python

Выполнил:

Шальнев Владимир Сергеевич,
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.

ВЫПОЛНЕНИЕ:

```
C:\Users\Serj\PycharmProjects>git clone https://github.com/HAXF13D/laboratory-9
Cloning into 'laboratory-9'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (2/2), done.
```

Клонирование репозитория

```
C:\Users\Serj\PycharmProjects>cd laboratory-9

C:\Users\Serj\PycharmProjects\laboratory-9>git checkout develop
Switched to a new branch 'develop'
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
```

Переход на ветку develop

```
#!/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 соответствии 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 '):
    # 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)

```

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? Проверка З.А
Должность? Программист
Год поступления? 2018
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Проверка З.А             | Программист         |      2018     |
+-----+-----+-----+-----+

```

Значение №1

```

>>> add
Фамилия и инициалы? Тест С.И.
Должность? Тестировщик
Год поступления? 2014
>>> select 3
      1: Проверка З.А
      2: Тест С.И.
>>> select 5
      1: Тест С.И.
>>> select 8
Работники с заданным стажем не найдены.
>>> dsf;j
>>> Неизвестная команда dsf;j

```

Значение №2

```

C:\Users\Serj\PycharmProjects\laboratory-9>git status
On branch develop
Your branch is up to date with 'origin/develop'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  example.py

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\Serj\PycharmProjects\laboratory-9>git add .
C:\Users\Serj\PycharmProjects\laboratory-9>git commit -m "Add example"
[develop 4e19ba3] Add example
1 file changed, 95 insertions(+)
create mode 100644 example.py

```

Коммит изменений

```

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

if __name__ == '__main__':

    school = {
        "1А": 32,
        "1Б": 25,
        "2А": 22,
        "2Б": 31,
        "3А": 32,
        "3Б": 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['2Б'] = 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("3В", 31)
    for key, value in school.items():
        print(f" В {key} классе количество детей = {value}.")

    # Часть б)
    print("\nРасформировали один класс, теперь:\n")
    school.pop("1А")
    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}")

```

Задача 1

В 1А классе количество детей = 32.
В 1Б классе количество детей = 25.
В 2А классе количество детей = 22.
В 2Б классе количество детей = 31.
В 3А классе количество детей = 32.
В 3Б классе количество детей = 18.

Значение №1

В одном из классов поменялось количество детей, теперь:

В 1А классе количество детей = 32.
В 1Б классе количество детей = 25.
В 2А классе количество детей = 22.
В 2Б классе количество детей = 16.
В 3А классе количество детей = 32.
В 3Б классе количество детей = 18.

Значение №2

Появился новый класс, теперь:

В 1А классе количество детей = 32.
В 1Б классе количество детей = 25.
В 2А классе количество детей = 22.
В 2Б классе количество детей = 16.
В 3А классе количество детей = 32.
В 3Б классе количество детей = 18.
В 3В классе количество детей = 31.

Значение №3

Расформировали один класс, теперь:

В 1Б классе количество детей = 25.
В 2А классе количество детей = 22.
В 2Б классе количество детей = 16.
В 3А классе количество детей = 32.
В 3Б классе количество детей = 18.
В 3В классе количество детей = 31.

Значение №4

Всего учеников в школе - 144

Значение №5

```
C:\Users\Serj\PycharmProjects\laboratory-9>git status
On branch develop
Your branch is ahead of 'origin/develop' by 1 commit.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    task1.py

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Serj\PycharmProjects\laboratory-9>git add .

C:\Users\Serj\PycharmProjects\laboratory-9>git commit -m "Add first task"
[develop 367d358] Add first task
 1 file changed, 41 insertions(+)
 create mode 100644 task1.py
```

Коммит изменений

```
#!/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}")
```

Задача 2

```
Словарь до изменений:
{1: 'abc', 2: 'home', 3: 'test', 4: 'task'}
Словарь после изменений:
{'abc': 1, 'home': 2, 'test': 3, 'task': 4}
```

Значение №1

```
C:\Users\Serj\PycharmProjects\laboratory-9>git status
On branch develop
Your branch is ahead of 'origin/develop' by 2 commits.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  task2.py

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Serj\PycharmProjects\laboratory-9>git add .

C:\Users\Serj\PycharmProjects\laboratory-9>git commit -m "Add task 2"
[develop 95a15bd] Add task 2
1 file changed, 15 insertions(+)
create mode 100644 task2.py
```

Коммит изменений

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

import sys
from datetime import datetime

if __name__ == '__main__':
    trains = []
    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break

        elif command == 'add':
            destination = input("Название пункта назначения? ")
            number = int(input("Номер поезда? "))
            time = input("Время отправления ЧЧ:ММ? ")
            time = datetime.strptime(time, '%H:%M')
            train = {
                'destination': destination,
                'number': number,
                'time': time,
            }
```



```

trains.append(train)
if len(trains) > 1:
    trains.sort(key=lambda item: item.get('destination', ''))

elif command == 'list':
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 28,
        '-' * 14,
        '-' * 19
    )
    print(line)
    print(
        '| {:^4} | {:^28} | {:^14} | {:^19} |'.format(
            "No",
            "Название пункта назначения",
            "Номер поезда",
            "Время отправления"
        )
    )
    print(line)
    for idx, train in enumerate(trains, 1):
        print(
            '| {:>4} | {:<28} | {:<14} | {:>19} |'.format(
                idx,
                train.get('destination', ''),
                train.get('number', ''),
                train.get('time', 0).strftime("%H:%M")
            )
        )
    print(line)

elif command.startswith('select '):
    count = 0
    parts = command.split(' ', maxsplit=1)
    time = datetime.strptime(parts[1], '%H:%M')
    for train in trains:
        if train.get("time") > time:
            count += 1
            print(
                '{:>4}: {} {}'.format(count, train.get('destination',
                ''), train.get("number"))
            )
    if count == 0:
        print("Отправлений позже этого времени нет.")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить отправление;")
    print("list - вывести список отправлений;")
    print("select <ЧЧ:ММ> - вывод на экран информации опоездах,
отправляющихся после этого времени;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

Индивидуальное задание №1

```

>>> add
Название пункта назначения? Butovo
Номер поезда? 1247
Время отправления ЧЧ:ММ? 12:50
>>> add
Название пункта назначения? Afonkino
Номер поезда? 125
Время отправления ЧЧ:ММ? 14:30
>>> select 12:20
    1: Afonkino 125
    2: Butovo 1247
>>> select 13:00
    1: Afonkino 125
>>> list
+-----+-----+-----+
| No  | Название пункта назначения | Номер поезда | Время отправления |
+-----+-----+-----+
|  1  | Afonkino                  | 125          | 14:30             |
|  2  | Butovo                    | 1247         | 12:50             |
+-----+-----+-----+
>>> |

```

Значение №1

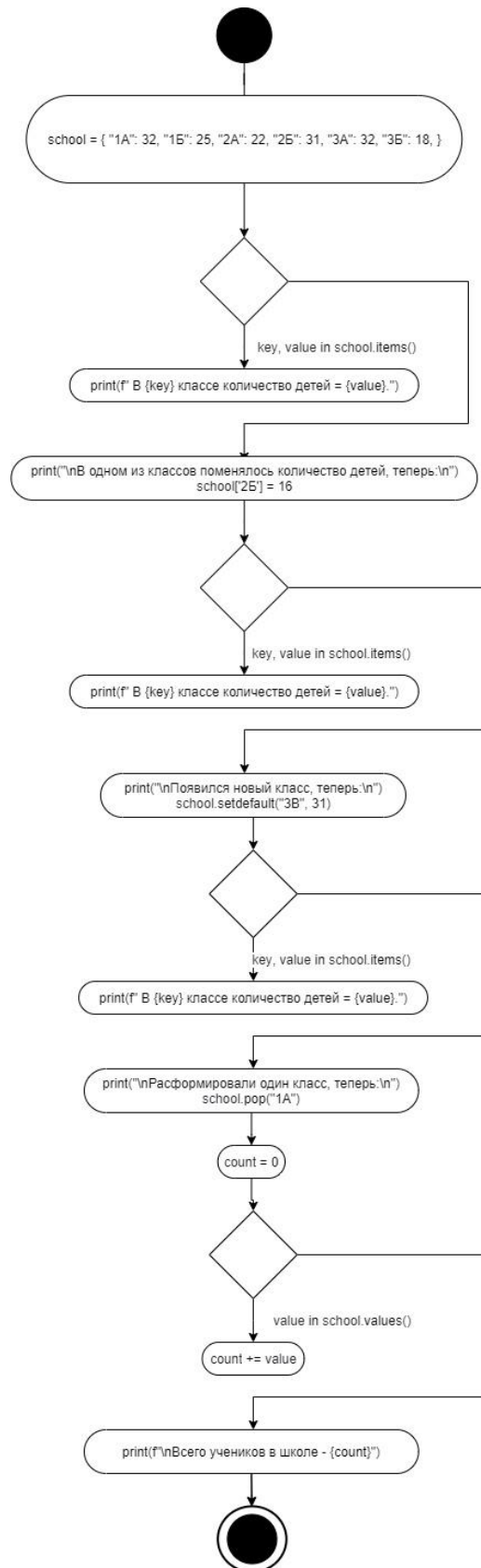
```

C:\Users\Serj\PycharmProjects\laboratory-9>git add .

C:\Users\Serj\PycharmProjects\laboratory-9>git commit -m "Individual task"
[develop e61c096] Individual task
 1 file changed, 79 insertions(+)
 create mode 100644 indiv_task.py

```

Коммит изменений



UML диаграмма 1 примера

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