

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №8

Работа со строками в языке Python.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

Цель работы: приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.

Ход работы: Пример

<https://github.com/Mirror-Shard/L2.6>

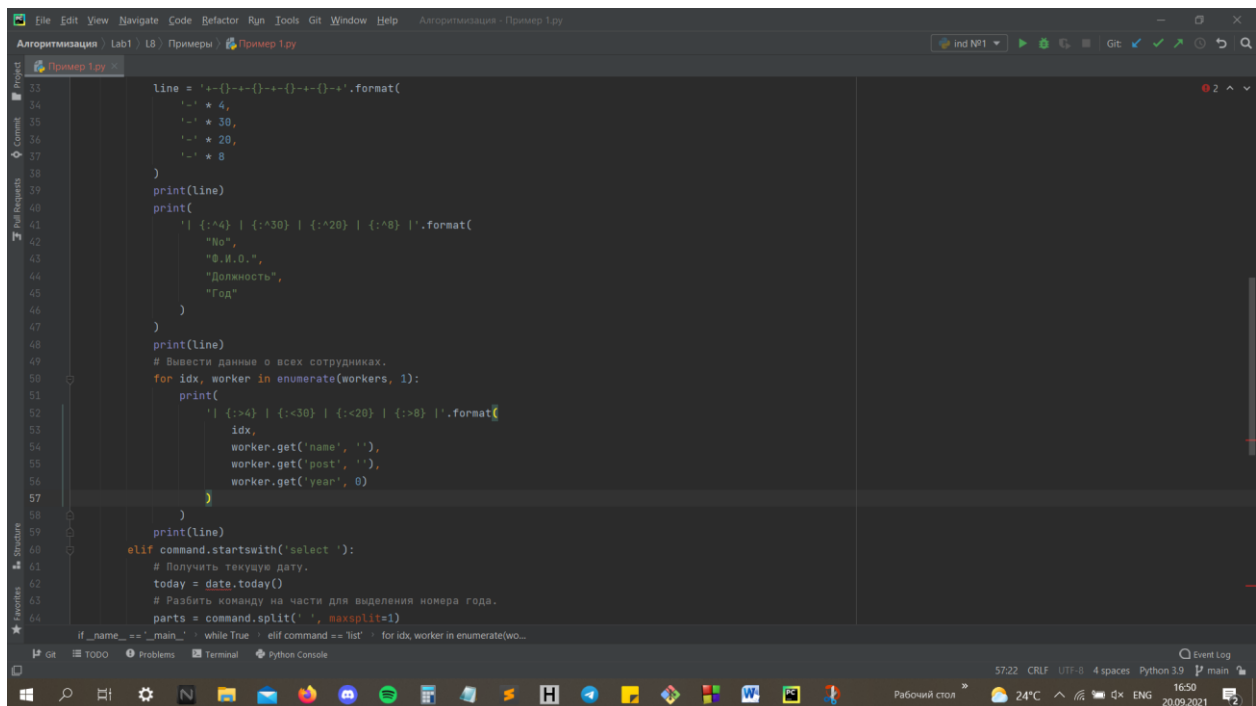
1. Создал репозиторий на github с лицензией MIT, добавил .gitignore и выбрал язык Python.
2. Проработал пример из учебника:

```

# 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,

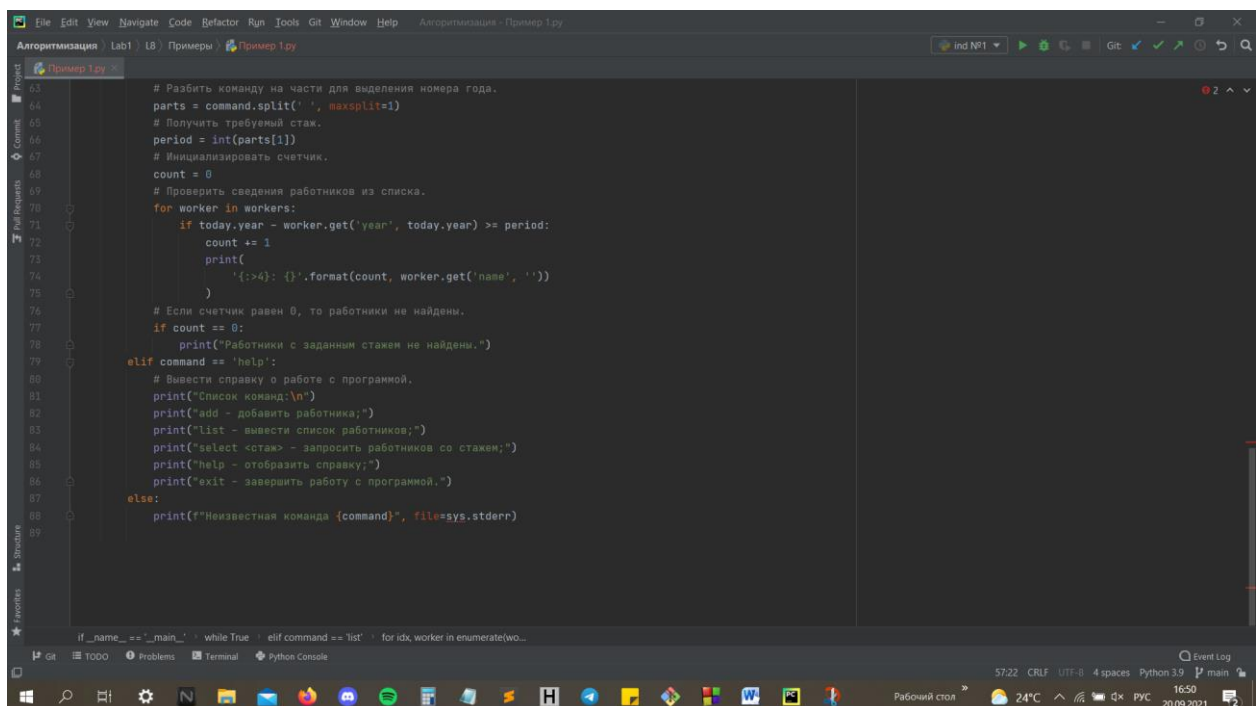
```

Рисунок 1 – Пример, часть 1



```
33 line = '{:4}--{:4}--{:4}--{:4}--{:4}'.format(
34     ' ' * 4,
35     ' ' * 30,
36     ' ' * 20,
37     ' ' * 8
38 )
39 print(line)
40 print(
41     '{:4} | {:30} | {:20} | {:8}'.format(
42         "No",
43         "0.0.0.",
44         "Должность",
45         "Год"
46     )
47 )
48 print(line)
49 # Вывести данные о всех сотрудниках.
50 for idx, worker in enumerate(workers, 1):
51     print(
52         '{:4} | {:30} | {:20} | {:8}'.format(
53             idx,
54             worker.get('name', ''),
55             worker.get('post', ''),
56             worker.get('year', 0)
57         )
58     )
59     print(line)
60 elif command.startswith('select '):
61     # Получить текущую дату.
62     today = date.today()
63     # Разбить команду на части для выделения номера года.
64     parts = command.split(' ', maxsplit=1)
```

Рисунок 2 – Пример, часть 2



```
63 # Разбить команду на части для выделения номера года.
64 parts = command.split(' ', maxsplit=1)
65 # Получить требуемый стаж.
66 period = int(parts[1])
67 # Инициализировать счетчик.
68 count = 0
69 # Проверить сведения работников из списка.
70 for worker in workers:
71     if today.year - worker.get('year', today.year) >= period:
72         count += 1
73         print(
74             '{:4}: {}'.format(count, worker.get('name', ''))
75         )
76 # Если счетчик равен 0, то работники не найдены.
77 if count == 0:
78     print("Работники с заданным стажем не найдены.")
79 elif command == 'help':
80     # Вывести справку о работе с программой.
81     print("Список команд:\n")
82     print("add - добавить работника;")
83     print("list - вывести список работников;")
84     print("select <стаж> - запросить работников со стажем;")
85     print("help - отобразить справку;")
86     print("exit - завершить работу с программой.")
87 else:
88     print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунок 3 – Пример, часть 3

Задание 1

1. Код задания 1

```
# Создание словаря классов
school = {'1a': 18, '1b': 19, '2b': 20, '6a': 21, '7c': 22}
# Всего учеников
q = 0

# Изменение количества учеников в классе
school['2b'] = 23
# Появление нового класса
school['8b'] = 24
# Расформирование класса
del school['6a']

# Вычисление общего количества участников
for i in school.values():
    q += i

# Вывод общего количества учеников
print(q)
```

Рисунок 4 – Задание 1

2. Результат работы

106

Рисунок 5 – результат работы

Задание 2

1. Код задания 2

```
# Создание словаря
a = {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
# Второй словарь (будет обратным исходному)
b = {}
# Разделение на ключи и значения
n = a.items()

# Меняет местами ключи и значения
for key, value in a.items():
    b[value] = key

print(b)
```

Рисунок 6 – Код задания 2

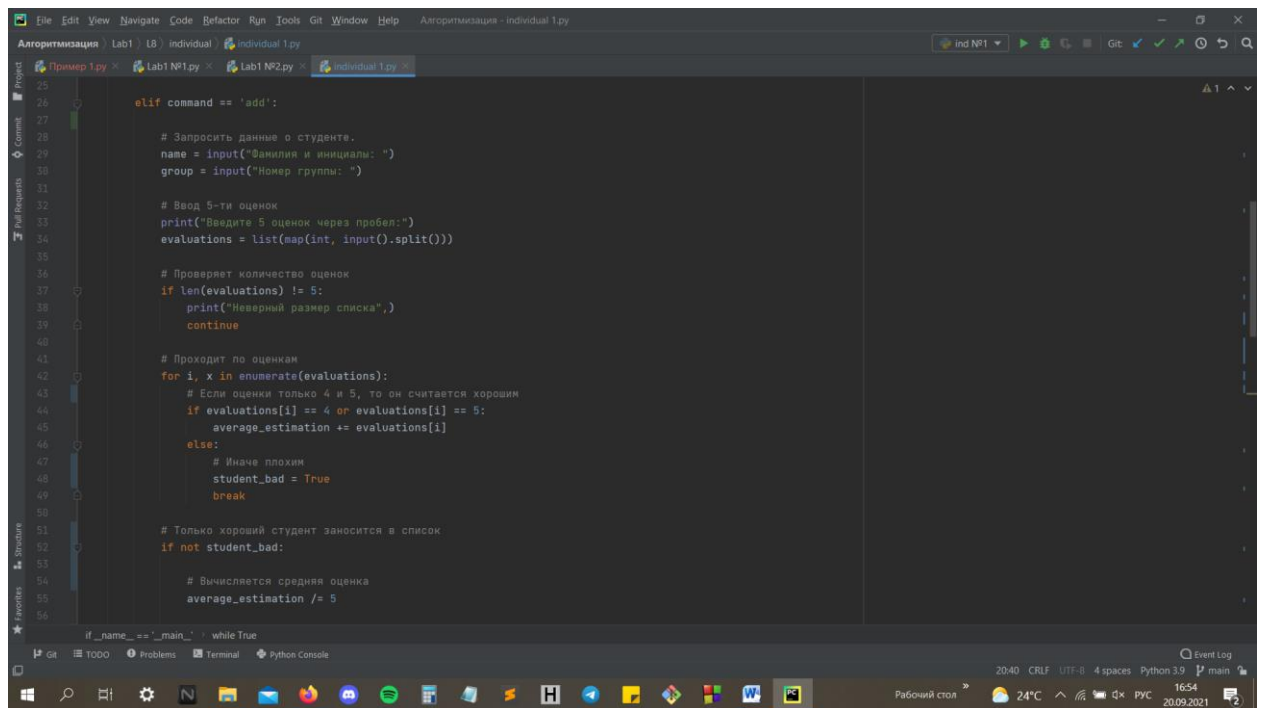
2. Результат работы

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4}
```

Рисунок 7 – Результат работы задания 2

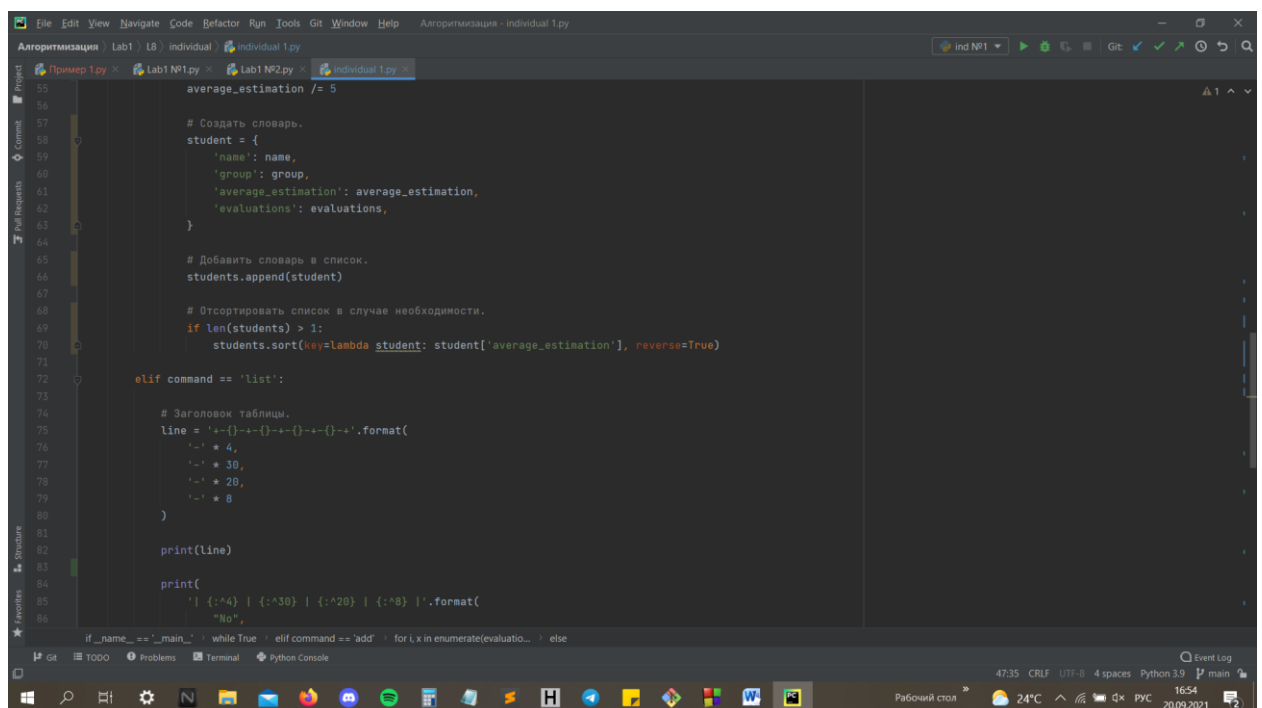
Индивидуальное задание

1. Код индивидуального задания:



```
25
26 elif command == 'add':
27
28     # Запросить данные о студенте.
29     name = input("Фамилия и инициалы: ")
30     group = input("Номер группы: ")
31
32     # Ввод 5-ти оценок
33     print("Введите 5 оценок через пробел:")
34     evaluations = list(map(int, input().split()))
35
36     # Проверять количество оценок
37     if len(evaluations) != 5:
38         print("Неверный размер списка",)
39         continue
40
41     # Проходит по оценкам
42     for i, x in enumerate(evaluations):
43         # Если оценки только 4 и 5, то он считается хорошим
44         if evaluations[i] == 4 or evaluations[i] == 5:
45             average_estimation += evaluations[i]
46         else:
47             # Иначе плохим
48             student_bad = True
49             break
50
51     # Только хороший студент заносится в список.
52     if not student_bad:
53
54         # Вычисляется средняя оценка
55         average_estimation /= 5
56
57 if __name__ == '__main__': while True
```

Рисунок 8 – Код индивидуального задания, часть 1



```
55 average_estimation /= 5
56
57 # Создать словарь.
58 student = {
59     'name': name,
60     'group': group,
61     'average_estimation': average_estimation,
62     'evaluations': evaluations,
63 }
64
65 # Добавить словарь в список.
66 students.append(student)
67
68 # Отсортировать список в случае необходимости.
69 if len(students) > 1:
70     students.sort(key=lambda student: student['average_estimation'], reverse=True)
71
72 elif command == 'list':
73
74     # Заголовок таблицы.
75     line = '+-{}-+-{}-+-{}-+-{}-+'.format(
76         '-' * 4,
77         '-' * 30,
78         '-' * 20,
79         '-' * 8
80     )
81
82     print(line)
83
84     print(
85         '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
86             "No",
87
88 if __name__ == '__main__': while True elif command == 'add' for i, x in enumerate(evaluation... else
```

Рисунок 9 – Код индивидуального задания, часть 2

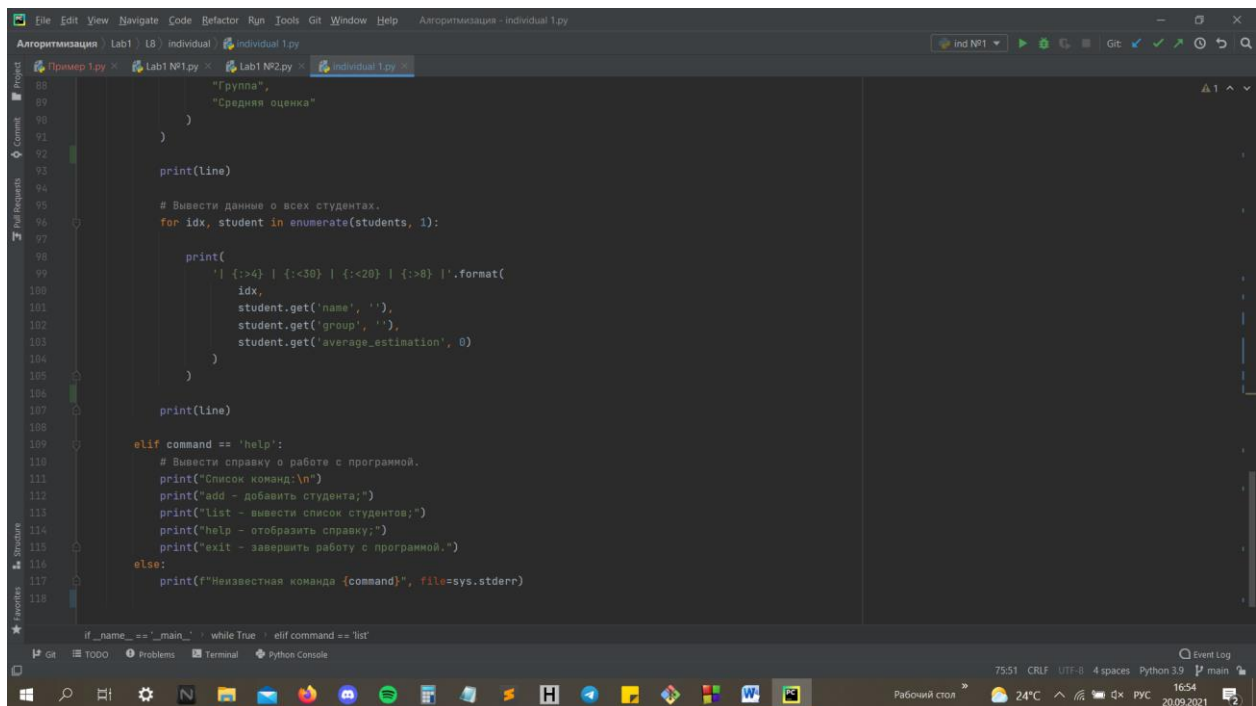


Рисунок 10 – Код индивидуального задания, часть 3

2. Результат работы

```

>>> add
Фамилия и инициалы: Хашиев Х.М.
Номер группы: 20
Введите 5 оценок через пробел:
5 5 5 5 5
>>> list
+-----+-----+-----+-----+
| No |           Ф.И.О.           |      Группа      | Средняя оценка |
+-----+-----+-----+-----+
|  1 | Хашиев Х.М.                |      20          |      5.0       |
+-----+-----+-----+-----+

```

Рисунок 11 – Результат работы индивидуального задания

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.

2. Может ли функция `len()` быть использована при работе со словарями?

Да

3. Какие методы обхода словарей Вам известны?

Перебор элементов словаря в цикле `for` и его модификации, например: `for key, value in nums.items()`

4. Какими способами можно получить значения из словаря по ключу?

Элементы словаря перебираются в цикле `for` также, как элементы других сложных объектов. С другой стороны у словаря как класса есть метод `items()`, который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение. В цикле `for` можно распаковывать кортежи, таким образом сразу извлекая как ключ, так и его значение.

5. Какими способами можно установить значение в словаре по ключу?

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

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка:

```
>>> {x: x * x for x in (1, 2, 3, 4)}  
{1: 1, 2: 4, 3: 9, 4: 16}
```

7. Самостоятельно изучите возможности функции `zip()` приведите примеры ее использования.

Функция `zip()` в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со

списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные.

8. Самостоятельно изучите возможности модуля `datetime`. Каким функционалом по работе с датой и временем обладает этот модуль?

Класс `datetime.date(year, month, day)` - стандартная дата. Атрибуты: `year, month, day`. Неизменяемый объект. Класс `datetime.time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None)` - стандартное время, не зависит от даты. Атрибуты: `hour, minute, second, microsecond, tzinfo`.

Вывод: приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.