

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №9
дисциплины «Программирование на Python»

Выполнил:
Горбунов Данила Евгеньевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

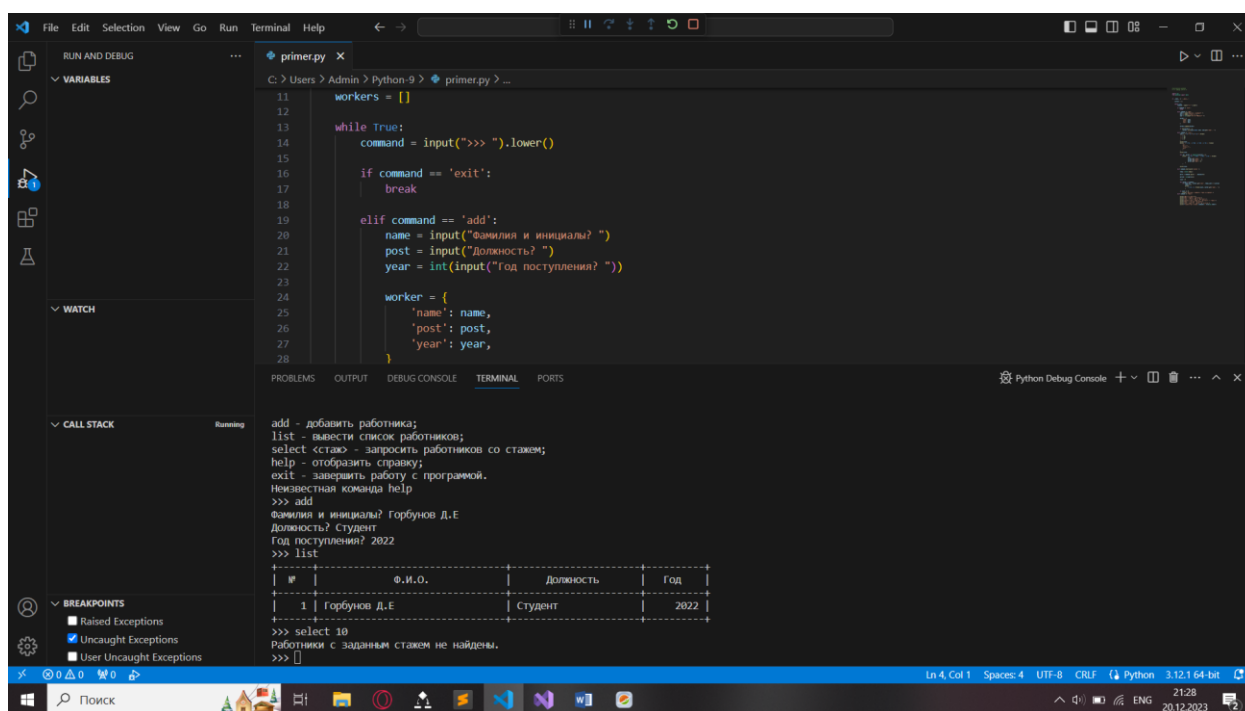
Ставрополь, 2023 г.

Тема: Работа со словарями в языке Python

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

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствии с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера при различных исходных данных, вводимых с клавиатуры.



```
11 workers = []
12
13 while True:
14     command = input(">>> ").lower()
15
16     if command == 'exit':
17         break
18
19     elif command == 'add':
20         name = input("Фамилия и инициалы? ")
21         post = input("Должность? ")
22         year = int(input("Год поступления? "))
23
24         worker = {
25             'name': name,
26             'post': post,
27             'year': year,
28         }
29
30         workers.append(worker)
31
32     elif command == 'list':
33         list_workers(workers)
34
35     elif command == 'select':
36         select_workers(workers)
37
38     elif command == 'help':
39         help_text()
40
41     else:
42         print("Неизвестная команда help")
43
44     print("\n")
45
46     if command != 'exit':
47         continue
48
49     break
```

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
Неизвестная команда help
>>> add
Фамилия и инициалы? Горбун Д.Е.
Должность? Студент
Год поступления? 2022
>>> list

#	Ф.И.О.	Должность	Год
1	Горбун Д.Е.	Студент	2022

>>> select 10
Работники с заданным стажем не найдены.
>>>

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую задачу: создайте словарь, связав его с переменной school, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т. п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в

школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе.

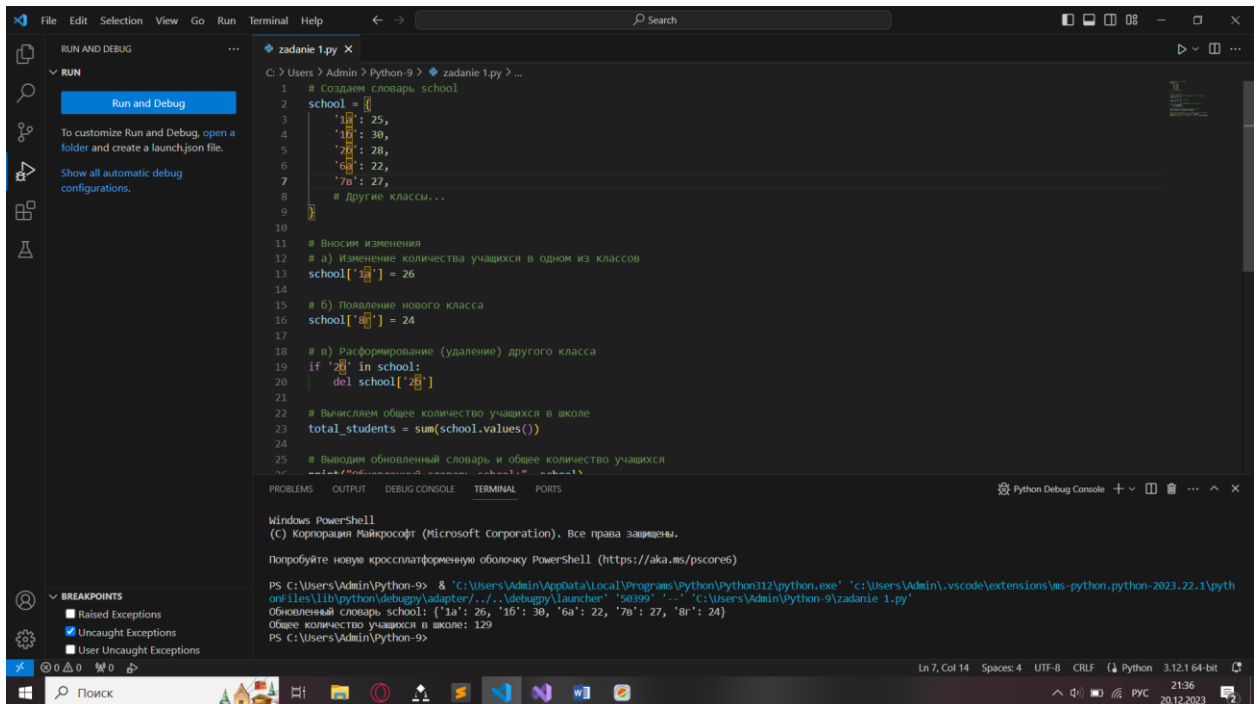


Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод `items()`, с помощью полученного объекта `dict_items` создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

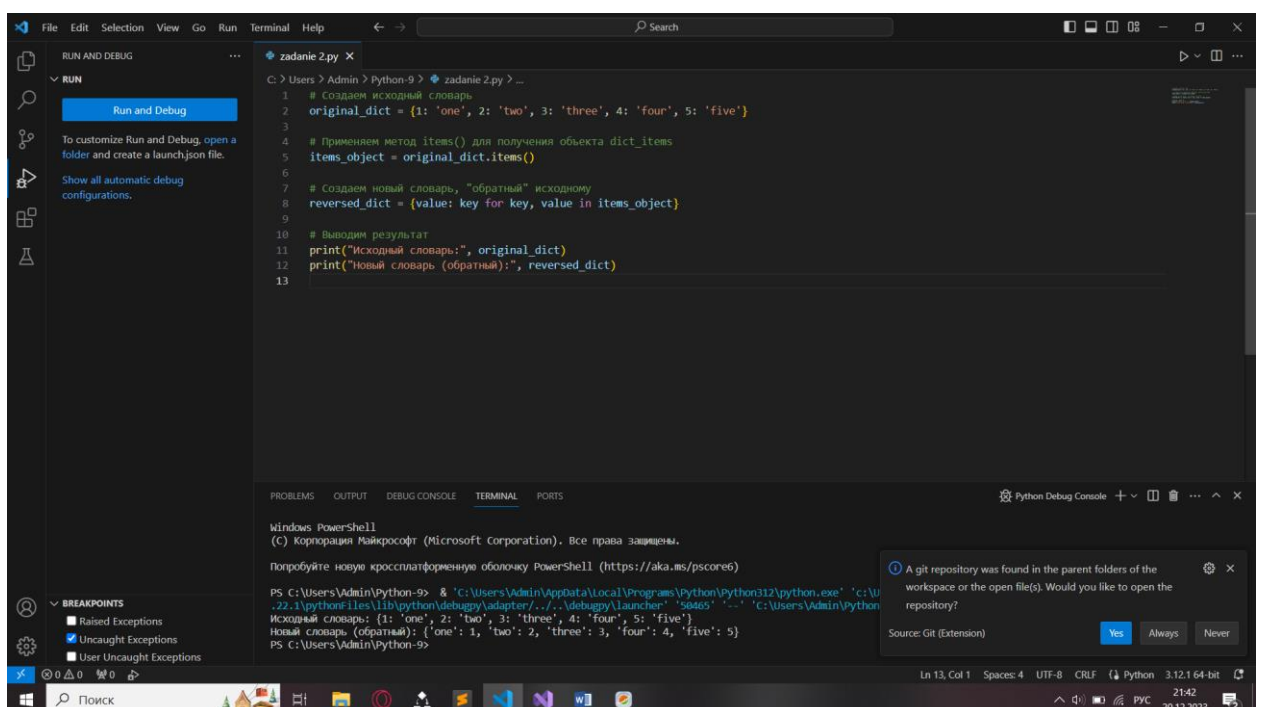
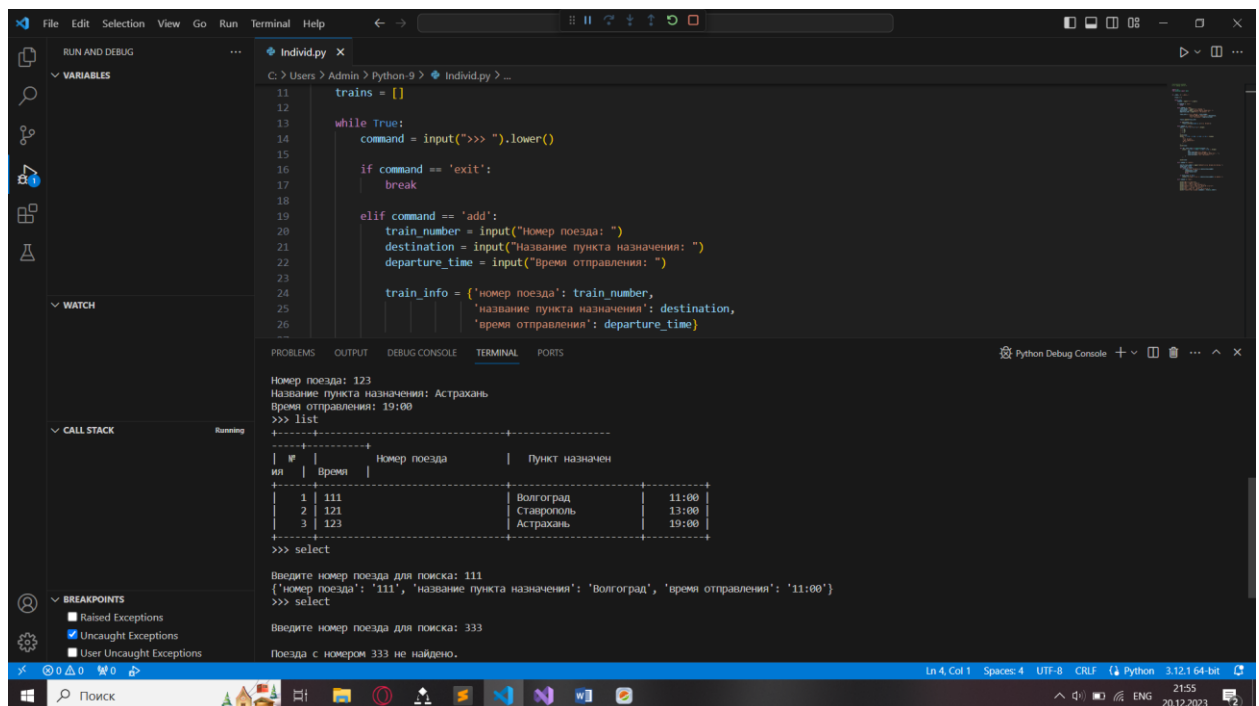


Рисунок 3. Результат работы программы из задачи 2

7. Выполнил индивидуальное задание. Привёл в отчете скриншот работы программы.

Задание. Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.



```
11 trains = []
12
13 while True:
14     command = input(">>> ").lower()
15
16     if command == 'exit':
17         break
18
19     elif command == 'add':
20         train_number = input("Номер поезда: ")
21         destination = input("Название пункта назначения: ")
22         departure_time = input("Время отправления: ")
23
24         train_info = {'номер поезда': train_number,
25                       'название пункта назначения': destination,
26                       'время отправления': departure_time}
```

Номер поезда: 123
Название пункта назначения: Астрахань
Время отправления: 19:00

```
>>> list
```

	Номер поезда	Пункт назначения
1	111	Волгоград
2	121	Ставрополь
3	123	Астрахань

```
>>> select
```

Введите номер поезда для поиска: 111
('номер поезда': '111', 'название пункта назначения': 'Волгоград', 'время отправления': '11:00')

```
>>> select
```

Введите номер поезда для поиска: 333
Поезда с номером 333 не найдено.

Рисунок 4. Результат работы программы из индивидуального задания

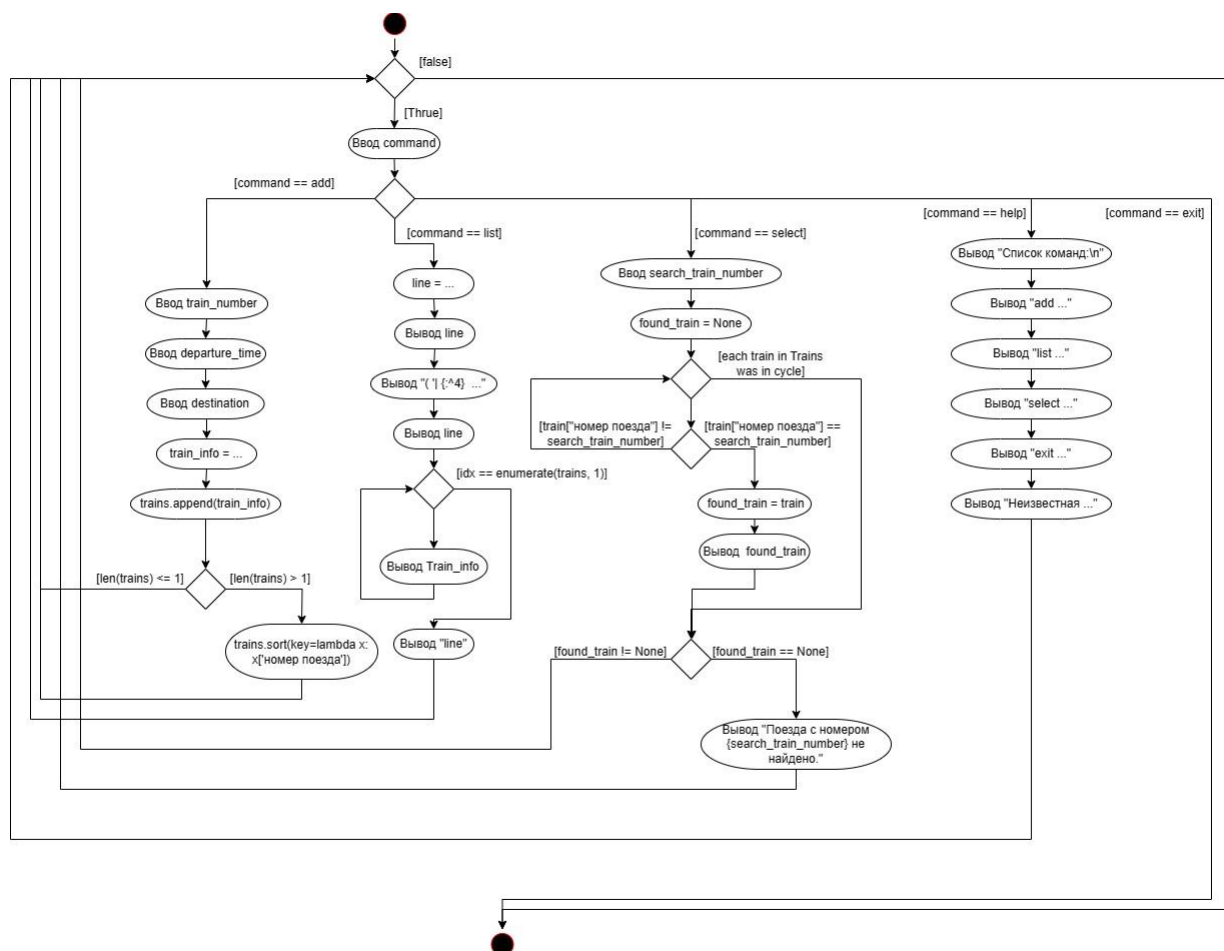


Рисунок 5. UML-диаграмма работы программы из индивидуального задания

Контрольные вопросы

1. Что такое словари в языке Python?

В языке Python словари – это изменяемые коллекции, которые хранят данные в формате ключ-значение. Каждый элемент словаря состоит из уникального ключа и соответствующего ему значения. Словари в Python являются неупорядоченными, что означает, что элементы в словаре не имеют определенного порядка. Словари в Python представлены фигурными скобками `{}` и могут содержать любые типы данных в качестве значений.

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

Да, функция `len()` может быть использована для определения количества элементов в словаре. Она возвращает количество пар ключ-значение в словаре.

3. Какие методы обхода словарей Вам известны?

Цикл for-in: Можно использовать цикл for-in для итерации по ключам словаря и получения соответствующих значений. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

Методы keys(), values() и items(): Метод keys() возвращает итерируемый объект, содержащий все ключи словаря. Метод values() возвращает итерируемый объект, содержащий все значения словаря. Метод items() возвращает итерируемый объект, содержащий кортежи (ключ, значение) для каждой пары ключ-значение в словаре. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
for key in my_dict.keys():
    print(key)
for value in my_dict.values():
    print(value)
for key, value in my_dict.items():
    print(key, value)
```

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

В Python можно получить значения из словаря по ключу с помощью следующих способов: Использование оператора доступа к элементам словаря ([]): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
print(my_dict['apple'])
```

Метод get(): Метод get() позволяет получить значение по ключу. Если ключ не существует, метод возвращает значение по умолчанию (None, если не указано другое значение). Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
print(my_dict.get('apple'))
```

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

В Python можно установить значение в словаре по ключу с помощью следующих способов:

Использование оператора доступа к элементам словаря ([]): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках и присвоив новое значение. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict['apple'] = 4
print(my_dict)
```

Метод update(): Метод update() позволяет обновить словарь, добавив новую пару ключ-значение или обновив значение существующего ключа.

Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict.update({'apple': 4, 'mango': 6})
print(my_dict)
```

6. Что такое словарь включений?

Словарь включений (dictionary comprehension) - это способ создания словаря на основе итерации и условий. Он позволяет создавать словари более компактно и эффективно. Синтаксис словаря включений выглядит следующим образом:

```
{ключ_выражение: значение_выражение for элемент in
итерируемый_объект if условие}
```

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

Функция zip() в Python используется для объединения элементов из нескольких итерируемых объектов в один итерируемый объект, состоящий из кортежей. Количество элементов в результирующем итерируемом объекте равно количеству элементов в самом коротком итерируемом объекте.

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

Модуль `datetime` в Python предоставляет функционал для работы с датой и временем. Некоторые возможности модуля `datetime` включают: Создание объектов даты и времени: Модуль `datetime` позволяет создавать объекты для представления даты, времени или комбинации даты и времени

Вывод: в ходе выполнения работы были приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.