Федеральное государственное бюджетное образовательное учреждение высшего образования

"Калининградский государственный технический университет"

Кафедра систем управления и вычислительной техники

Работа защищена с оценкой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и Ф.И.О. преподавателя)

ПРОГРАММИРОВАНИЕ

Пояснительная записка по курсовой работе

|  |  |
| --- | --- |
| Работа допущена | Работу выполнил студент |
| к защите | учебной группы 19-ИЭ-1 |
| доц. Калинина С. А. | Богданов М. Д. |
| (подпись и Ф.И.О.  руководителя работы) | (подпись и Ф.И.О студента) |
| \_\_\_\_\_\_\_\_ | 24.04.2020 |
| (дата) | (дата) |

Калининград, 2020

1. Тема “Структуры данных. Словари в Python. Операции со словарями и методы словарей”……………………………………………………………3
   1. Теоритическое описание темы………………………………....3
   2. Примеры задач и программ по теме…………………………...8
   3. Список использованных источников………………..………...11
2. Проектирование и программирование задачи……..………..……11
   1. Постановка задачи……………………………………......…….11
   2. Описание используемых в программе классов………….……11
   3. Макет GUI с описанием использованных виджетов…………12
   4. Описание событий и методов с ними связанных………….…12
   5. Текст программы……………………………………………….12
   6. Текст исходного файла………………………………………...15
   7. Скриншоты выполнения программы…………………………18
3. Тема: “Структуры данных. Словари в Python. Операции со словарями и методы словарей.”
   1. Теоретическое описание темы.

В Python 3 имеется три встроенных типа данных: списки, кортежи и словари

Списки (list)

Список – это структура данных, которая содержит упорядоченный набор элементов, т.е. хранит последовательность элементов. Это легко представить, если вспомнить список покупок, в котором перечисляется, что нужно купить, с тем лишь исключением, что в списке покупок каждый элемент обычно размещается на отдельной строке, тогда как в Python они разделяются запятыми.

Список элементов должен быть заключён в квадратные скобки, чтобы Python понял, что это список. Как только список создан, можно добавлять, удалять или искать элементы в нём. Поскольку элементы можно добавлять и удалять, мы говорим, что список – это изменяемый тип данных, т.е. его можно модифицировать.

Кортежи (tuple)

Кортежи служат для хранения нескольких объектов вместе. Их можно рассматривать как аналог списков, но без такой обширной функциональности, которую предоставляет класс списка. Одна из важнейших особенностей кортежей заключается в том, что они неизменяемы, так же, как и строки. Т.е. модифицировать кортежи невозможно.

Кортежи обозначаются указанием элементов, разделённых запятыми; по желанию их можно ещё заключить в круглые скобки.

Кортежи обычно используются в тех случаях, когда оператор или пользовательская функция должны наверняка знать, что набор значений, т.е. кортеж значений, не изменится.

Словари (dictionary)

Словарь – это некий аналог адресной книги, в которой можно найти адрес или контактную информацию о человеке, зная лишь его имя; т.е. некоторые ключи (имена) связаны со значениями (информацией). Ключ должен быть уникальным – вы ведь не сможете получить корректную информацию, если у вас записаны два человека с полностью одинаковыми именами.

В словарях в качестве ключей могут использоваться только неизменяемые объекты (как строки), а в качестве значений можно использовать как неизменяемые, так и изменяемые объекты.

Пары ключ-значение указываются в словаре следующим образом: “dict = {key1 : value1, key2 : value2 }”. Ключ и значение разделяются двоеточием, а пары друг от друга отделяются запятыми, а затем всё это заключается в фигурные скобки.

Пары ключ-значение никоим образом не упорядочены в словаре. Если нам необходим некоторый порядок, нам придётся отдельно отсортировать словарь перед обращением к нему.

Словари являются экземплярами/объектами класса dict.

dictionary = {'персона': 'человек', 'марафон': 'гонка бегунов длиной около 26 миль', 'противостоять': 'оставаться сильным, несмотря на давление', 'бежать': 'двигаться со скоростью'}

Данный словарь использует строки в качестве ключей, однако ключом может являться в принципе любой неизменяемый тип данных. Значением же конкретного ключа может быть что угодно. Вот ещё один пример словаря, где ключами являются числа, а значениями — строки:

gender\_dict = {0: 'муж', 1: 'жен'}

Важное уточнение: если мы попробуем использовать изменяемый тип данных в качестве ключа, то получим ошибку:

dictionary = {(1, 2.0): 'кортежи могут быть ключами', 1: 'целые числа могут быть ключами', 'бежать': 'строки тоже', ['носок', 1, 2.0]: 'а списки не могут'}

### **Получение данных из словаря**

Для получения значения конкретного ключа используются квадратные скобки []. Предположим, что в нашем словаре есть пара 'марафон': 26.

# берём значение с ключом "марафон"

dictionary['марафон']

Опять же, мы получим ошибку, если попытаемся получить значение по несуществующему ключу. Для избежания подобных ошибок существуют методы, о которых написано далее.

### **Добавление и обновление ключей**

Добавление новых пар в словарь происходит достаточно просто:

# Добавляем ключ "туфля" со значением "род обуви, закрывающей ногу не выше щиколотки"

dictionary['туфля'] = 'род обуви, закрывающей ногу не выше щиколотки'

Обновление существующих значений происходит абсолютно также:

# Обновляем ключ "туфля" и присваиваем ему значение "хорошая туфля"

dictionary['туфля'] = 'хорошая туфля'

### **Удаление ключей**

Для удаления ключа и соответствующего значения из словаря можно использовать del

# Удаляем значение с ключом "противостоять" из словаря

del dictionary['противостоять']

## **Методы**

Словари в Python имеют множество различных полезных методов, которые помогут нам в работе с ними. Вот некоторые из них:

### **Update**

Метод update() пригодится, если нужно обновить несколько пар сразу. Метод принимает другой словарь в качестве аргумента.

# Добавляем две пары в словарь dictionary, используя метод update

dictionary.update({'бежал': 'бежать в прошедшем времени', 'туфли': 'туфля во множественном числе'})

>>> dictionary

{'марафон': 'гонка бегунов длиной около 26 миль', 'персона': 'человек', 'бежал': 'бежать в прошедшем времени', 'бежать': 'двигаться со скоростью', 'туфля': 'род обуви, закрывающей ногу не выше щиколотки', 'туфли': 'туфля во множественном числе'}

### **Get**

# Допустим, у нас есть словарь story\_count

story\_count = {'сто': 100, 'девяносто': 90, 'двенадцать': 12, 'пять': 5}

Метод get() возвращает значение по указанному ключу. Если указанного ключа не существует, метод вернёт None.

# Ключ "двенадцать" существует и метод get в данном случае вернёт 12

story\_count.get('двенадцать')

Метод можно использовать для проверки наличия ключей в словаре:

>>> story\_count.get('два')

None

Также можно указать значение по умолчанию, которое будет возвращено вместо None, если ключа в словаре не окажется:

# Метод вернёт 0 в случае, если данного ключа не существует

story\_count.get('два', 0)

### **Pop**

Метод pop() удаляет ключ и возвращает соответствующее ему значение.

>>> story\_count.pop('девяносто')

90

>>> story\_count

{'двенадцать': 12, 'сто': 100, 'пять': 5}

### **Keys**

Метод keys() возвращает коллекцию ключей в словаре.

>>> story\_count.keys()

['сто', 'пять', 'двенадцать']

### **Values**

Метод values() возвращает коллекцию значений в словаре.

>>> story\_count.values()

[100, 12, 5]

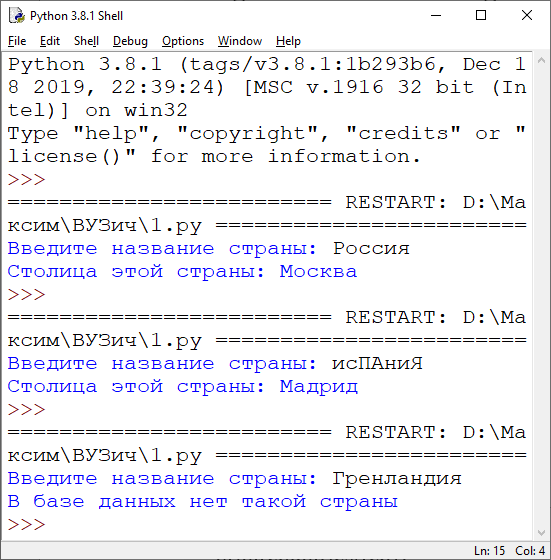
### **Items**

Метод items() возвращает пары «ключ — значение».

>>> dictionary.items()

[('персона', 'человек'), ('бежать', 'двигаться со скоростью'), ('туфля', 'род обуви, закрывающей ногу не выше щиколотки'), ('бежал', 'бежать в прошедшем времени'), ('марафон', 'гонка бегунов длиной около 26 миль'), ('туфли', 'туфля во множественном числе')]

* 1. Примеры задач и программ по теме

**1. Задание:** Пользователь вводит название страны. Программа должна выдать столицу этой страны. Учесть неправильный регистр и отсутствие данных по запросу.

countries={"Россия":"Москва",\

"США":"Вашингтон",\

"Япония":"Токио",\

"Испания":"Мадрид"}

req=input("Введите название страны: ")

req=req.capitalize()

if req in countries:

print("Столица этой страны: ",end="")

print(countries[req])

else:

print("В базе данных нет такой страны")

**2. Задание:** Выведите информацию о студентах, отсортировав их по возрасту.

arr=[[1,"Иванов Иван Иванович", 20, "19-ИЭ"],

[2,"Петров Пётр Петрович", 18, "19-ВТ"],

[3,"Соболев Иван Петрович", 19, "19-КС"]]

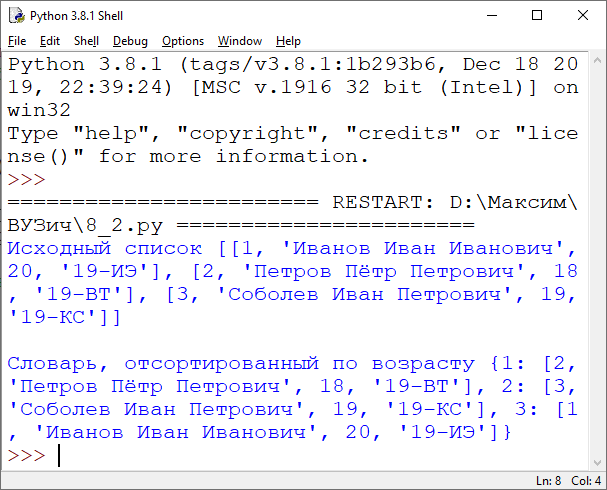
slov={}

sorting={}

for i in range(len(arr)):

slov[arr[i][0]]=arr[i][0:4:1]

ray=[]

h=1

for i in range(len(slov)):

try:

ray.append(slov[h][2])

except KeyError:

print("Ошибка ключа")

h+=1

h=1

for i in range(100):

for j in range(len(ray)):

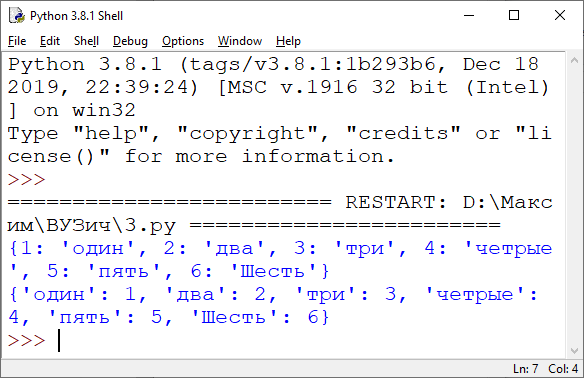
if i==ray[j]:

sorting[h]=slov[j+1]

h+=1

print(sorting)

**3. Задание:** В словаре поменять местами ключ и значение. В исходном словаре ключами являются строки, а значениями цифры.

def switcher(s):

o={}

for key, value in slov.items():

o[value] = key

print(o)

slov={1:"один", 2:"два", 3:"три", 4:"четрые", 5:"пять", 6:"Шесть"}

print(slov)

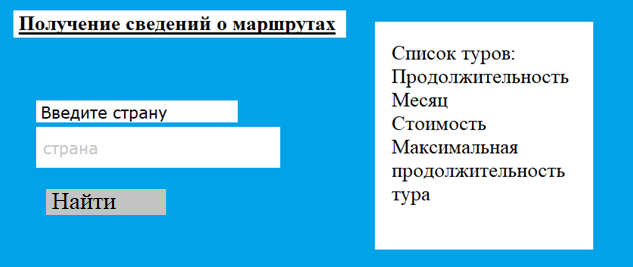
switcher(slov.items())

* 1. Список использованных источников  
     1) <https://wombat.org.ua/AByteOfPython/data_structures.html#id13>  
     2) <https://rtfm.co.ua/python-struktury-dannyx/>  
     3) <https://tproger.ru/explain/python-dictionaries/>  
     4) Практическая работа №8. Словари.

1. Проектирование и программирование задачи: “Получение сведений о маршрутах”
   1. Постановка задачи: в исходном файле представлены данные: страна, продолжительность тура, стоимость тура, месяц. По запросу месяца выдать: список туров, продолжительность, месяц, стоимость и максимальную продолжительность тура
   2. Описание используемых в программе классов

|  |  |
| --- | --- |
| Имя класса | Маршруты |
| Свойства | * Страна * Продолжительность тура * Стоимость тура * Месяц |
| Методы |  |

* 1. Макет GUI c описанием использованных виджетов



Видежет Entry – для получения запроса  
Виджет Button – для активизации выдачи результатов

Виджет Listbox – для визуализации выдачи результатов

* 1. Описание событий и методов с ними связанных  
     1) Создание экземпляров класса с использованием данных из файла  
     2) Создание GUI по макету  
     3) Добавление функционала элементам макета
  2. Текст программы:

from tkinter import \*

class Tours:

country=''

time=0

price=''

month=''

def die(): #закрытие окна графического интерфейса

root.destroy()

def grafic(data):

canv=Canvas(root, height=242, width=496, bg="white")

canv.place(x=6,y=260) #поле для графика

canv.create\_line(40, 230, 40, 10, width=2, arrow=LAST)

canv.create\_line(10, 210, 480, 210, width=2, arrow=LAST)

r=1

for c in range(60,490,36): #цифры от 1 до 12 на графике

canv.create\_text(c, 220, text=r)

r+=1

data.sort()

coords=[]

for c in range(len(data)):

coords.append([]) #вычисление координат

coords[c].append(44+data[c][0]\*30)

coords[c].append(330-(data[c][1]/200))

canv.create\_line(coords)

def find():

f=open(entr0.get(),'r')

arr=f.readlines()

ray=[]

data=[]

h=0

for i in range(0,len(arr),5):

ray.append([]) #

for j in range(4):

ray[h].append(arr[i+j]) #создание двумерного массива

if i!=len(arr)-4 or j!=3:

ray[h][j]=ray[h][j][:-1]

h=h+1

arr.clear() #очистка списка с неотсортированными данными

def calc(): #вставка в виджет Label максимальной продожительности тура

but2.config(state="disabled")

lb2.config(text=("Максимальная продолжительность тура (дни):"+str(maxT)))

but2.config(command=calc)

strana=entr1.get().capitalize()

maxT=0

txt.delete(0,1000) #очистка виджета Textbox

for i in range(len(ray)):

if ray[i][0]==strana:

req=Tours #создание экземпляра класса

req.country=ray[i][0]

req.time=int(ray[i][1])

req.price=ray[i][2]

req.month=ray[i][3]

arr.append(ray[i])

if req.time>maxT: #определение максимальной продожительности тура

maxT=req.time

arr=sorted(arr, key=lambda stra: int(stra[1])) #сортировка по количеству дней

txt.insert(END,"Список туров:")

txt.insert(END,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

if arr:

v=0

but2.config(state=NORMAL)

for i in arr:

txt.insert(END,"Дней:"+i[1])

txt.insert(END,i[2])

txt.insert(END,i[3])

txt.insert(END,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

data.append([]) #заполние двумерного списка

data[v].append(int(i[1])) #данными для построения графика

data[v].append(int(i[2][:-6]))

v+=1

canv.destroy

grafic(data)

else:

txt.insert(END,"Туры в данную")

txt.insert(END,"страну отсутсвуют")

but2.config(state="disabled")

lb2.config(text="") #очистка виджета Label

arr.clear()

root=Tk()

root.title ("Получение сведений о маршрутах")

root.geometry("512x512")

root.configure(background="#1fa5ed")

entr0=Entry(font=("Arial",14))

entr0.place(x=173, y=24, width=122, height = 24)

lb3=Label(text="Введите имя файла:",font=("Arial",10), bg="#1fa5ed",fg="white")

lb3.place(x=174, y=3, width=122, height = 24)

but3=Button(text="Ввод", bg="lightgrey", command=find)

but3.place(x=259,y=50)

lb0=Label(text="Богданов М.Д. 19-ИЭ-1",font=("Arial",11),bg="white")

lb0.place(x=0, y=0)

lb1=Label(text="Введите страну",font=("Arial",12), bg="#1fa5ed",fg="white")

lb1.place(x=42, y=90, width=120, height = 24)

entr1=Entry(font=("Arial",14))

entr1.place(x=48, y=112, width=150, height = 24)

but0=Button(text="Завершить", bg="lightgrey",command=die)

but0.place(x=4,y=226)

but1=Button(text="Найти", bg="lightgrey",command=find)

but1.place(x=50,y=140)

but2=Button(text="Максимум", bg="lightgrey",state="disabled")

but2.place(x=100,y=140)

txt=Listbox(width=24, height=10, font=("Arial", 10))

txt.place(x=323,y=20)

lb2=Label(font=("Arial",10), wraplength=190, justify=LEFT, height=3, width=18)

lb2.place(x=323, y=194)

lb4=Label(text="График зависимости цены от времени:",font=("Arial",10),bg="#1fa5ed",fg="white")

lb4.place(x=85, y=240)

scroll=Scrollbar(command=txt.yview)

scroll.place(x=480, y=20, height=174)

txt.config(yscrollcommand=scroll.set)

canv=Canvas(root, height=242, width=496, bg="white")

canv.place(x=6,y=260) #поле для графика

root.mainloop()

* 1. Текст исходного файла:

Уругвай

7

30000 рублей

август

Америка

10

45000 рублей

сентябрь

Америка

3

26000 рублей

сентябрь

Гренландия

11

58000 рублей

январь

Египет

10

46000 рублей

июнь

Ямайка

3

33000 рублей

июнь

Гренландия

2

28000 рублей

декабрь

Гренландия

3

39000 рублей

апрель

Ямайка

6

54000 рублей

июнь

Турция

4

40000 рублей

май

Турция

3

35000 рублей

апрель

Турция

8

54000 рублей

август

Турция

6

50000 рублей

октябрь

* 1. Скриншоты выполнения программы 