**ЛАБОРАТОРНАЯ РАБОТА №7.**

**Работа со словарями в Python.**

**Цель работы.**

1. Ознакомиться с особенностями словарей в Python;
2. Создать приложение, демонстрирующее различные способы работы со словарями в Python.

**Краткие теоретические сведения:**

Словари (dict) – это одна из наиболее часто используемых структур данных, позволяющая хранить объекты, для доступа к которым используется ключ. В этом уроке будут рассмотрены операции создания, удаления, работы со словарями и их методы.

**Что такое словарь (dict) в 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. Если вспомнить такую структуру как список, то доступ к его элементам осуществляется по индексу, который представляет собой целое неотрицательное число, причем мы сами, непосредственно, не участвуем в его создании (индекса). В словаре аналогом индекса является ключ, при этом ответственность за его формирование ложится на программиста.

**Создание, изменение, удаление словарей и работа с его элементами**

*Создание словаря*Пустой словарь можно создать, используя функцию *dict()*, либо просто указав пустые фигурные скобки.

>>> d1 = dict()

>>> print(type(d1))

<*class* 'dict'>

>>> d2 = {}

>>> print(type(d2))

<*class* 'dict'>

Если необходимо создать словарь с заранее подготовленным набором данных, то можно использовать один из перечисленных выше подходов, но с перечислением групп ключ-значение.

>>> d1 = dict(Ivan="manager", Mark="worker")

>>> print(d1)

{'Mark': 'worker', 'Ivan': 'manager'}

>>> d2 = {"A1":"123", "A2":"456"}

>>> print(d2)

{'A2': '456', 'A1': '123'}

*Добавление и удаление элемента*Чтобы добавить элемент в словарь нужно указать новый ключ и значение.

>>> d1 = {"Russia":"Moscow", "USA":"Washington"}

>>> d1["China"]="Beijing"

>>> print(d1)

{'Russia': 'Moscow', 'China': 'Beijing', 'USA': 'Washington'}

Для удаления элемента из словаря можно воспользоваться командой *del*.

>>> d2 = {"A1":"123", "A2":"456"}

>>> del d2["A1"]

>>> print(d2)

{'A2': '456'}

*Работа со словарем*  
Проверка наличия ключа в словаре производится с помощью оператора *in*.

>>> d2 = {"A1":"123", "A2":"456"}

>>> "A1" in d2

**True**

>>> "A3" in d2

**False**

Доступ к элементу словаря, осуществляется как же как доступ к элементу списка, только в качестве индекса указывается ключ.

>>> d1 = {"Russia":"Moscow", "USA":"Washington"}

>>> d1["Russia"]

'Moscow'

**Методы словарей**

У словарей доступен следующий набор методов.

***clear()***  
Удаляет все элементы словаря.

>>> d2 = {"A1":"123", "A2":"456"}

>>> print(d2)

{'A2': '456', 'A1': '123'}

>>> d2.clear()

>>> print(d2)

{}

***copy()***  
Создается новая копия словаря.

>>> d2 = {"A1":"123", "A2":"456"}

>>> d3 = d2.copy()

>>> print(d3)

{'A1': '123', 'A2': '456'}

>>> d3["A1"]="789"

>>> print(d2)

{'A2': '456', 'A1': '123'}

>>> print(d3)

{'A1': '789', 'A2': '456'}

***fromkeys(seq[, value])***  
Создает новый словарь с ключами из *seq* и значениями из *value*. По умолчанию *value* присваивается значение *None.*

***get(key)***  
Возвращает значение из словаря по ключу *key*.

>>> d = {"A1":"123", "A2":"456"}

>>> d.get("A1")

'123'

***items()***  
Возвращает элементы словаря (ключ, значение) в отформатированном виде.

>>> d = {"A1":"123", "A2":"456"}

>>> d.items()

dict\_items([('A2', '456'), ('A1', '123')])

***keys()***  
Возвращает ключи словаря.

>>> d = {"A1":"123", "A2":"456"}

>>> d.keys()

dict\_keys(['A2', 'A1'])

***pop(key[, default])***  
Если ключ *key* есть в словаре, то данный элемент удаляется из словаря и возвращается значение по этому ключу, иначе будет возвращено значение *default*. Если *default* не указан и запрашиваемый ключ отсутствует в словаре, то будет вызвано исключение *KeyError*.

>>> d = {"A1":"123", "A2":"456"}

>>> d.pop("A1")

'123'

>>> print(d)

{'A2': '456'}

***popitem()***  
Удаляет и возвращает пару (ключ, значение) из словаря. Если словарь пуст, то будет вызвано исключение *KeyError*.

>>> d = {"A1":"123", "A2":"456"}

>>> d.popitem()

('A2', '456')

>>> print(d)

{'A1': '123'}

***setdefault(key[, default])***  
Если ключ *key* есть в словаре, то возвращается значение по ключу. Если такого ключа нет, то в словарь вставляется элемент с ключом *key* и значением *default*, если *default* не определен, то по умолчанию присваивается *None*.

>>> d = {"A1":"123", "A2":"456"}

>>> d.setdefault("A3", "777")

'777'

>>> print(d)

{'A2': '456', 'A3': '777', 'A1': '123'}

>>> d.setdefault("A1")

'123'

>>> print(d)

{'A2': '456', 'A3': '777', 'A1': '123'}

***update([other])***  
Обновить словарь парами (*key/value*) из *other*, если ключи уже существуют, то обновить их значения.

>>> d = {"A1":"123", "A2":"456"}

>>> d.update({"A1":"333", "A3":"789"})

>>> print(d)

{'A2': '456', 'A3': '789', 'A1': '333'}

***values()***  
Возвращает значения элементов словаря.

>>> d = {"A1":"123", "A2":"456"}

>>> d.values()

dict\_values(['456', '123'])

**Порядок выполнения работы**

1. Выполнить задания, выбрав индивидуально предметную область
2. Оформить отчет
3. Подготовить ответы на контрольные вопросы

**Задания на лабораторную работу**:

**Задание 1: создание, заполнение, получение и удаление (общее задание)**

1. Выбрать (подобрать самостоятельно) тему;
2. реализовать добавление записи в словарь, удаление и обновление записи;
3. Вывод статистической информации.

***Примечания к заданию:***

1. Примеры тем: контрольно-пропускной пункт университета, обработка заказов магазина, регистрация новых пользователей форума программистов;
2. Реализовать полноценный доступ к элементам словаря (добавление, редактирование, удаление);
3. Реализовать вывод и сортировку элементов словаря.

**Задание 2: создать программный продукт для работы с шестнадцатеричными значениями цветов палитры RGB (общее задание)**

1. Реализовать ручное добавление цвета и генерацию случайного цвета в допустимых диапазонах цветов (#000000 – #FFFFFF);
2. Реализовать определение преобладающего цвета во введенном пользователем элементе и возможность добавления этого элемента в словарь, если значение прошло проверку на вхождение в диапазон;
3. Для каждого добавляемого цвета генерировать уникальное значение ключа (0:’#E2F3E6’, 1:’#00FF00’…).

**Контрольные вопросы:**

1. Какие варианты создания словаря вы знаете?
2. Каким образом происходит обращение к элементу словаря?
3. Какие способы добавления элемента в словарь вы знаете?.
4. Какая ошибка может возникнуть, если элемента к которому происходит обращение отсутствует в словаре? Как ее обработать?