Лабораторная работа №7. МНОЖЕСТВА И СЛОВАРИ. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

7.1. Множества в Python

Множество в Python – это структура данных, аналогичная множествам в математике. Для них определены те же операции.

Множество может состоять из различных элементов неизменяемых типов данных: строковых и числовых, но не могут иметь изменяемый тип, например, элементом множества не может быть другое множество или список.

Примечание. Порядок элементов в множестве неопределен.

Множество, кроме пустого, задается перечислением всех его элементов в фигурных скобках.

Примечание. Пустыми фигурными скобками определяется словарь (см. ниже), а не пустое множество!!!

Функция `set(value)` преобразовывает список или строку (value) в множество, состоящее из их элементов. С ее помощью можно задать пустое множество: `set()`

```
# Пример 7.1.1) Перечисление и получение множества
из строки
N = {1, 2, 3}
M = set('xyz')
L = set()
print(N)
print(M)
print(L)
```

Каждый элемент может входить в множество только один раз, порядок задания элементов не важен.

```
#Пример7.1.2)
A={1,2,3}
B={3,3,2,3,1}
print('A={1,2,3}')
print('B={3,3,2,3,1}')
print('A==B:',A==B)#выведет True, так как множества
A и B равны.
```

```
print()
s=set('текст')
print('Множество букв слова "текст":',s)
```

Примечание. Элементы одного типа упорядочиваются в порядке возрастания.

Функция `len()` служит для определения числа элементов в множестве

Перебор элементов множества (в неопределенном порядке!) можно осуществить в цикле `for`.

```
# Пример 7.1.3) Нахождение длины и перебор элементов множества
set_of_something = {0, 5, 15, 'лес', 'день', 10}
print('Длина множества равна:', len(set_of_something))
print()
for i in set_of_something:
    print(i)
```

Оператор `in` выдает значение True, если элемент принадлежит заданному множеству и – False, если иначе. Возможна и противоположная операция: `not in`.

`.add(value)` – добавление элемента `value` в множество.

```
# Пример 7.1.4) Проверка принадлежности элемента множеству и добавление элемента к множеству.
A = {'аспирин', 'анальгин', 'аскофен'}
print('аспирин' in A, 'парацетомол' in A, 'парацетомол' not in A)
A.add('парацетомол')
print(A)
```

`.discard(value)` –удаление элемента из множества. Отсутствующий элемент игнорируется.

`.remove(value)` – удаление элемента из множества. Если элемент
`value` отсутствует в множестве – генерируется ошибка.

Другие операции над множествами:

A | B A.union(B)	Объединение множеств A и B (элементы входят или в A, или в B или в оба множества вместе).
A |= B A.update(B)	Дополнение A элементами множества B.
A & B A.intersection(B)	Пересечение множеств A и B (элементы входят одновременно A, и в B).
A &= B A.intersection_update(B)	Удаление из A элементов, которы нет в множестве B.
A - B A.difference(B)	Разность A и B (элементы, входят множество A, но не входят в B).
A -= B A.difference_update(B)	Удаление из A всех элементов, которые входят в B.
A ^ B A.symmetric_difference(B)	Симметрическая разность множеств A и B (элементы, входят A или в B, но не в оба из них одновременно).
A ^= B A.symmetric_difference_update(B)	Запись в A симметрической разности множеств A и B (см. выш
A <= B A.issubset(B)	Проверка того, что множество A является подмножеством B (True, если верно).
A < B	То же, что A<= B and A != B
A >= B A.issuperset(B)	Проверка того, что множество B является подмножеством A (True, если верно).
A > B	То же, что A>= B and A != B

```
# Пример 7.1.5) Объединение множеств A и B и
дополнение B до A.
A = {0, 1, 2, 3}
B = {4, 3, 2, 1}
C = A.union(B)
print(C)
B.update(A)
print(B)
```

Задача 7.1. Дана строка, содержащая Вашу фамилию. Преобразуйте ее в множество и выведите результат на экран.

Задача 7.2. Имеется множество букв алфавита. Удалите из него все буквы, составляющие Вашу фамилию.

7.2. Работа со словарями в Python

Словарь в Python – это структура, содержащая неупорядоченные данные с идентификацией элементов не по числовому индексу, а по произвольному ключу.

Создать словарь можно, например, при помощи литерала или с помощью функции dict.

```
# Пример 7.2.1) Создание (2 способами) и работа со
списком

UDK = {'001.6': 'Научные знания', '001.83':
'Научное  сотрудничество',
      '001.86': 'Конкорданции', '004.63': 'Файлы',
'004.633.4': 'Обновление',
      '51': 'Математика', '611.01': 'Общая
анатомия', '611.013': 'Эмбриология',
      '611.018.74': 'Эндотелий (endothelium)',
```

```
'615.099.06': 'Осложнения',
      '616.8-009.11-031.47': 'Моноплегия',
'617.751.98': 'Слепота'}

UDK = dict([('001.6', 'Научные знания'), ('001.83',
'Научное сотрудничество'),
('001.86', 'Конкорданции'),  ('004.63',  'Файлы'),
 ('004.633.4', 'Обновление'), ('51', 'Математика'),
  ('611.01', 'Общая  анатомия'),
('611.013','Эмбриология'),
   ('611.018.74',  'Эндотелий (endothelium)'),
('615.099.06',  'Осложнения'),
    ('616.8-009.11-031.47',  'Моноплегия'),
('617.751.98', 'Слепота')])

UDK = dict(zip(['001.6', '001.83', '001.86',
'004.63', '004.633.4', '51', '611.01', '611.013',
'611.018.74', '615.099.06', '616.8-009.11-031.47',
'617.751.98'],
['Научные знания', 'Научное сотрудничество',
'Конкорданции', 'Файлы', 'Обновление',
 'Математика', 'Общая анатомия', 'Эмбриология',
'Эндотелий (endothelium)', 'Осложнения',
'Моноплегия', 'Слепота']))

display(UDK)

print()
print('Сравните:')
print(UDK)
```

```
print()
print('Еще один пример, где в качестве ключа может использоваться только строка:')

colors = dict(Красный = 'Red', Оранжевый = 'Orange', Желтый = 'Yellow', Зеленый = 'Green', Голубой = 'Blue', Синий = 'Darkblue', Фиолетовый = 'Violet')
display(colors)
```

Работа с элементами словаря может включать, например, следующие операции.

Получение значения элемента словаря по ключу: Dictionary[key] (аналогично обращению к элементу списка). Если указанного ключа нет в словаре, то возникает ошибка KeyError.

Для определения значения по ключу также может быть использован метод get: Dictionary.get(key). Если элемента с ключом get нет в словаре, то возвращается значение None. Если нужно вывести другое значение, то можно использовать формат записи метода с двумя аргументами: Dictionary.get(key, value) – возвращает значение value, если элемент с ключом key в словаре не найден.

Проверить принадлежность элемента словарю можно при помощи операторов in и not in.

Для добавления нового элемента в словарь нужно присвоить ему какое-то значение: Dictionary[key] = value.

Для удаления элемента из словаря можно использовать операцию del Dictionary [key] (если такого ключа в словаре нет – возникает ошибка: KeyError.

Следующие методы возвращают *представления* элементов словаря. Представления во многом похожи на множества, но они

изменяются, если менять значения элементов словаря.

`.keys()` возвращает представление ключей всех элементов;
`.values()` возвращает представление всех значений,
`.items()` возвращает представление всех пар (кортежей) из ключей и значений.

```
'''
Пример. 7.2.2) Имеется словарь офтальмологических терминов. Программа определяет, содержится ли ключ (в примере - гифема) в словаре, после чего выводит найденное по ключу значение на печать и потом удаляет его из словаря, если такого ключа нет, то добавляет его с соответствующим значением (кровоизлияние в переднюю камеру глаза). Затем пробегает получившийся словарь и выводит только элементы, в значении которых содержится заданная строка (кровоизлияние), добавляя ' - ' между ключом и значением.
'''
MED_OFTALM = dict(амблиопия = 'понижение зрения, обусловленное функциональными расстройствами зрительного анализатора',
   блефарит = 'воспаление краев век',
   васкулит = 'воспаление стенок кровеносных сосудов',
   гемофтальм = 'кровоизлияние в стекловидное тело',
   дальтонизм = 'врожденное расстройство цветового зрения',
   ирит = 'воспаление радужной оболочки глаза различной этиологии',
   катаракта = 'заболевание глаза, характеризующееся помутнением хрусталика',
   лагофтальм = 'неполное смыкание век, при котором глазная щель остается постоянно открытой',
   миопия = 'близорукость',
   нистагм = 'быстро повторяющееся движение глазных яблок (дрожание глаз)',
   офтальмия = 'общее название воспалительных
```

```
заболеваний глаз')
key_1 = 'гифема'
if key_1 in MED_OFTALM: # если есть такой ключ в
словаре
    print(MED_OFTALM[key_1])       #печатает
соответствующее значение
    del MED_OFTALM[key_1] # удаляет значение по
ключу key_1
else:
    MED_OFTALM[key_1] = 'кровоизлияние в переднюю
камеру глаза' # создает в словаре новую запись

for key_2,val in MED_OFTALM.items(): # пробегает
все ключи в словаре
    if 'кровоизлияние' in val: # если строка
'кровоизлияние' содержится в текущем значении
    print(key_2, ' - ', MED_OFTALM[key_2])
```

Задача 7.3. Создайте словарь, содержащий не менее 5 значений: Код_болезни (ключ) – Наименование_болезни (значение) по МКБ-10, где все значения «Наименование_болезни» начинаются на ту же букву, что и Ваше имя (если таковых нет, то фамилия или отчество). Выведите на экран (каждое с новой строки) все Наименования_болезней, содержащие в коде заболевания текущее число (например, если сегодня 01.02.2022, то в коде болезни нужно искать «1»).

7.3. Функции в Python

В Python есть множество встроенных функций, например, print(), input(), abs(), len() и др., а также дополнительных библиотек с готовыми решениями. Кроме того, имеется возможность создания собственных функций для упрощения структуры программы, например, там, где есть повторение кода для разных, в том числе, неупорядоченных значений.

Формат записи:

```
def [ваше_название_функции]([аргументы функции
```

```
через запятую, при наличии]):
    [блок кода]
    return [выражения через запятую]
```

Вызов функции происходит из любого места программы (но после ее определения) при обращении к ней по имени с перечислением в скобках требуемых значений для всех ее аргументов, если они есть. Обычно функции определяют в начале программы.

Примечание. Вместо `return` может использоваться `print()`, если нужно просто напечатать результат выполнения функции. Если не требуется, чтобы функция что-либо выводила или возвращала какое- то значение, то можно указать вместо него: `pass`.

```
#Пример 7.3.1)Вычисление площади прямоугольника и
#квадрата вынесено в функции. С результатом
#выполнения второй функции производить какие-то
#дальнейшие манипуляции не получится. Третья
#функция используется для обмена значениями между
#глобальными переменными a и c.
def square_1(x,y):
  S=x*y
  return S

def square_2(z):
  S=z**2
  print(S)


def replacing():
  global a
  global c
  if a>c:
      a,c=c,a
  pass

a,b,c,d=float(input('a:')),float(input('b:')),float
```

```
(input('c<a:')),float(input('d:'))
print(square_1(a,b)+square_1(c,d)+square_1(1,5))
replacing()
print(square_1(a,b)+square_1(c,d)+square_1(1,5))
square_2(a)
```

Примечание1. Служебное слово global позволяет изменять внутри функции значение глобальной переменной (определенной в теле программы вне функций), в отличие остальных переменных, определенных в теле функции (локальных), которые работают только внутри функции.

Примечание2. Обращение к локальной переменной из программы ведет к ошибке!!!

Примечание3. Принята следующая терминология.

Аргумент – это данные, которые передается в функцию при ее вызове.

Параметр (параметрическая переменная)– это локальная переменная, которой присваивается значение аргумента, переданное в функцию.

Любые изменения параметра в теле функции никак не влияют на ее аргумент.

Иногда говорят о **фактических** и **формальных параметрах** соответственно.

Задача 7.4. Напишите программу, которая печатает значение z для введенных с клавиатуры x и y столько раз, каков номер Вашего варианта с точностью, равной месяцу Вашего рождения. Для вычисления z напишите функцию, сразу выводящую результат на экран.

В-1. $z=3x+7,6y^2+14$;

В-2. $z=11x^3+2y-1,2$;

В-3. $z=13x-y^9+8,4$;

В-4. $z=-8x^5+11,8y^2+4$;

В-5. $z=x+17,1y^4-11$;

В-6. $z=7x^{11}+9,9y+2$;

В-7. $z=x^7+7,6y-7$;

В-8. $z=6,6x^3+5y^2+4$;

В-9. $z=9x^2-1,1y^2+33$;

B-10. $z=3,5x^4+3y^2+11$;
B-11. $z=3x+7,2y^2+14$;
B-12. $z=-1,7x^3+7y^2-15,6$;
B-13. $z=3,1x-4y^2+1,4$;
B-14. $z=x^4+12y^2-4,8$;
B-15. $z=2,7x+13y^5+6,9$;
B-16. $z=-2,3x^7+2,7y^3+2,2$;
B-17. $z=4,4x^4-4y+44$;
B-18. $z=6,3x^2-y^3+21$;
B-19. $z=8,9x+1,7y^2-19$;
B-20. $z=3,3x+7,3y^9+6$.

Задача 7.5. Вычислите z столько раз, каков номер Вашего варианта (см. предыдущую задачу). На экран выведите только их сумму с точностью, равной номеру Вашего варианта. Для вычисления z напишите функцию.

Задача 7.6. Напишите функцию, которая определяет, содержится ли во введенной строке Ваше имя, и, если содержится, то заменяет его на текст «самый замечательный человек».

Задача 7.7. Напишите программу, которая определяет, содержится ли в введенной строке Ваше имя, и, если содержится, то заменяет его на текст «самый замечательный человек», если имя стоит не в начале строки, и на то же, но с заглавной буквы, если в начале.

Задача 7.8. Напишите программу, которая создает список делителей данного числа.

Задача 7.9. На вход программе подается натуральное число n, большее номера Вашего варианта, а затем n целых чисел. Напишите программу, которая создает из указанных чисел список и выводит его на экран. После этого удаляет из полученного списка все элементы, стоящие под номерами, кратными Вашему

варианту, а затем выводит полученный список.

Задача 7.10. На вход программе подается натуральное число n, а затем n строк. Напишите программу, которая создает список из символов всех строк, а затем выводит его.

Задача 7.11. На вход программе подается натуральное число n, затем n строк, затем еще одна строка — поисковый запрос. Напишите программу, которая выводит все введенные строки, в которых встречается поисковый запрос.

Задача 7.12. На вход программе подается строка, содержащая Ваши фамилию, имя и отчество, разделенные пробелами. Напишите программу, которая выводит их в столбик.

Задача 7.13. Имеется 3 списка. В первом хранятся имена Ваших друзей, во втором – их фамилии, в третьем – номера телефонов в числовом формате (всего не более 5 человек). Получите из них список, в котором каждый элемент содержит имя, фамилию и, после двоеточия, номер одного друга.
Например: ['Иван Иванов: 89202952437,'Петр Петров: 89993227788',…]

Задача 7.14. Даны два множества чисел. Найдите все числа, которые входят как в первое, так и во второе множество и выведите их в порядке возрастания

Задача 7.15. Дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для слова из словаря, записанного в последней строке, определите его синоним.

Литература.

1. "Поколение Python": курс для начинающих // https://stepik.org/course/58852/promo (дата обращения 22.11.22).
2. Академия искусственного интеллекта для школьников https://ai-academy.ru/ (дата обращения 22.03.22).
3. 5 элементов культуры данных // https://analytikaplus.ru/5-elementov-kultury-dannyh/ (дата обращения 22.11.22).
4. Все о Python. Программирование на Python 3 // https://all- python.ru/ (дата обращения 22.03.22)
5. Интерактивный учебник языка Python / Павленко В., Соломатин В., Д. П. Кириенко, команда Pythontutor // https://pythontutor.ru/lessons/sets/(дата обращения 22.11.22).
6. Питонтьютор // https://pythontutor.ru/ (дата обращения 22.11.22).
7. Питончик // https://pythonchik.ru/ (дата обращения 22.11.22).
8. Руководство по языку Python / Лукашевский С. // https://pyprog.pro/python/py/py_guide.html (дата обращения 22.11.22).