Изучение алгоритмизации и основ программирования на языке Python в курсе Информатика и ИКТ

Множества

Преподаватель: Гупалова А.В.
Цветкова И.В.

I. Определение

- **Множество** — это структура данных, эквивалентная множествам в математике.
- Множество может состоять из различных элементов, порядок элементов в множестве неопределен.
- В множество можно:

1. добавлять и удалять элементы;
2. перебирать элементы множества;
3. выполнять операции над множествами (объединение, пересечение, разность);
4. проверять принадлежность элементу множества.

Определение

- Элементами множества может быть любой неизменяемый тип данных: числа, строки, кортежи.
- Требование неизменяемости элементов множества накладывается особенностями представления множества в памяти компьютера.

II. Задание множеств

Множество задается перечислением всех его элементов в фигурных скобках.

Например: *A = {1, 2, 3}*

Функция **set()** задает пустое множество.

Если функции set передать в качестве параметра список, строку или кортеж, то она вернет множество, составленное из элементов списка, строки, кортежа.

Например:

A = set('qwerty')

print(A) выведет

{'e', 'q', 'r', 't', 'w', 'y'}.

Задание множеств

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

Например, программа:

A = {1, 2, 3}

B = {3, 2, 3, 1}

print(A == B)

выведет True, так как A и B — равные множества.

set('Hello')

вернет множество из четырех элементов:

{'H', 'e', 'l', 'o'}.

III. Работа с элементами множеств

Функция ***len*** помогает узнать число элементов в множестве.

C = {1, 2, 3, 4, 5}
for elem in C:
    print(elem)

перебор элементов множества в неопределенном порядке

Проверить, принадлежит ли элемент множеству можно при помощи операции in, возвращающей значение типа bool:
i in A

i in not in A – обратная операция «*in*»

A.add(x) - добавляет элемент в множество A
discard и *remove* – методы удаления элемента из мн-ва
(в случае, если элемент отсутствует в мн-ве
метод discard не делает ничего, а
метод remove генерирует исключение KeyError.

pop - метод удаления из множетсва одного случайного элемента и возвращение его значения (при пустом мн-ве генерируется исключение KeyError).

Функция list – делает из множества список.

IV. Перебор элементов множества

При помощи цикла for можно перебрать все элементы множества:

```
Primes = {2, 3, 5, 7, 11}
for num im Primes:
    print(num)
```

V. Операции с множествами

A | B A.union(B)	Возвращает множество, являющееся объединением множеств A и B.
A |= B A.update(B)	Добавляет в множество A все элементы из множества B.
A & B A.intersection(B)	Возвращает множество, являющееся пересечением множеств A и B.
A &= B A.intersection_update(B)	Оставляет в множестве A только те элементы, которые есть в множестве B.
A - B A.difference(B)	Возвращает разность множеств A и B (элементы, входящие в A, но не входящие в B).
A -= B A.difference_update(B)	Удаляет из множества A все элементы, входящие в B.

A ^ B A.symmetric_difference(B)	Возвращает симметрическую разность множеств A и B (элементы, входящие в A или в B, но не в оба из них одновременно).
A ^= B A.symmetric_difference_update(B)	Записывает в A симметрическую разность множеств A и B.
A <= B A.issubset(B)	Возвращает true, если A является подмножеством B.
A >= B A.issuperset(B)	Возвращает true, если B является подмножеством A.
A < B	Эквивалентно A <= B and A != B
A > B	Эквивалентно A >= B and A != B

Задача

Во входном файле записан текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.
Определите, сколько различных слов содержится в этом тексте.

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
#Читаем весь файл в одну строку, которую поделим по пробелам
s = set( input.read().split( ))
output.write(str(len(s)))
input.close()
output.close()
```