*Лекция 4*

**Нелинейные алгоритмы**

Нелинейные алгоритмы – это алгоритмы, в которых не выполняются один за другим последовательно, как в линейных. Вместо этого они могут применяться одновременно или в неопределенном порядке.

Циклы – это частный случай нелинейных алгоритмов, они используются для автоматизации повторяющихся задач. В Python существуют два основных типа циклов: цикл for и цикл while.

## Цикл for

Цикл for используется для итерации по элементам списка, кортежа, словаря и других объектов. В частности мы рассмотрим на примерах итерации по **списку** (пример 1), **словарю** (пример 2) и стандартную итерацию по диапазону значений **range** (пример 3).

Для итерации по спискам есть два способа:

1. По индексам, с использованием диапазона значений range от 0 до len(list), где len() от объекта возвращает количество элементов, содержащиеся
2. По элементам с использованием конструкции **for** *item* **in** *object*

Рассмотрим оба варианта вывода элементов списка на экран, на примере.

## Пример 1

numbers = [5, 2, 8, 10]

for i in range(0, len(numbers)):

print(numbers[i])

for number in numbers:

print(number)

В обоих случая будет выведены все элементы списка numbers, каждый с новой строки. Но в первый раз изменялся **индекс** от 0 до длины списка не включительно, а во второй непосредственно сами элементы из списка присваивались в переменную **number**. Второй способ можно использовать для объектов элементы которых не поддаются индексации и доступны только по ключу, например словарь, как в следующем примере.

Напоминаю – словарь это объект данные в котором хранятся по схеме **ключ : значение**.

Для прохода по элементам словаря так же есть несколько способов:

1. Картежное присваивание ключа и значения при проходе по элементам объекта словарь.items()
2. Проходится по ключам словаря с помощью конструкции **for** *item* **in** *object* и получать значение с помощью словарь[ключ].

## Пример 2

person = {'name': 'John', 'age': 30, 'country': 'USA'}

for key in person:  
 print(key, ' - ', person[key])

for key, value in person.items():

print(key, ' - ', value)

При использовании любого варианта мы получим одинаковый вывод:

name - John

age - 30

country - USA

Где name, age, country – ключи словаря, а John, 30, USA – значения, соответствующие этим ключам.

Цикл while.

Цикл while используется для выполнения повторяющихся задач, пока условие верно. Пример использования цикла while:

counter = 0

while counter < 10:

print(counter)

counter += 1

Этот код выведет числа от 0 до 9 в консоль.

Важно помнить, что циклы могут создавать бесконечные петли, если не будет достигнуто условие выхода. Поэтому перед использованием цикла необходимо тщательно продумать условия выхода из него.

Давайте рассмотрим еще один пример нелинейного алгоритма с циклами.

Задача: найти сумму квадратов целых чисел от 1 до n.

Алгоритм:

Запрашиваем у пользователя число n.

Создаем переменную sum и присваиваем ей значение 0.

Запускаем цикл от 1 до n включительно.

В каждой итерации цикла мы складываем квадрат числа с текущей суммой.

Печатаем результат.

Код на Python:

n = int(input("Введите число: "))

sum = 0

for i in range(1, n+1):

sum += i \*\* 2

print("Сумма квадратов чисел от 1 до", n, "равна", sum)

Спасибо за внимание, надеюсь, это было полезно и помогло Вам лучше понять нелинейные алгоритмы и циклы в Python.