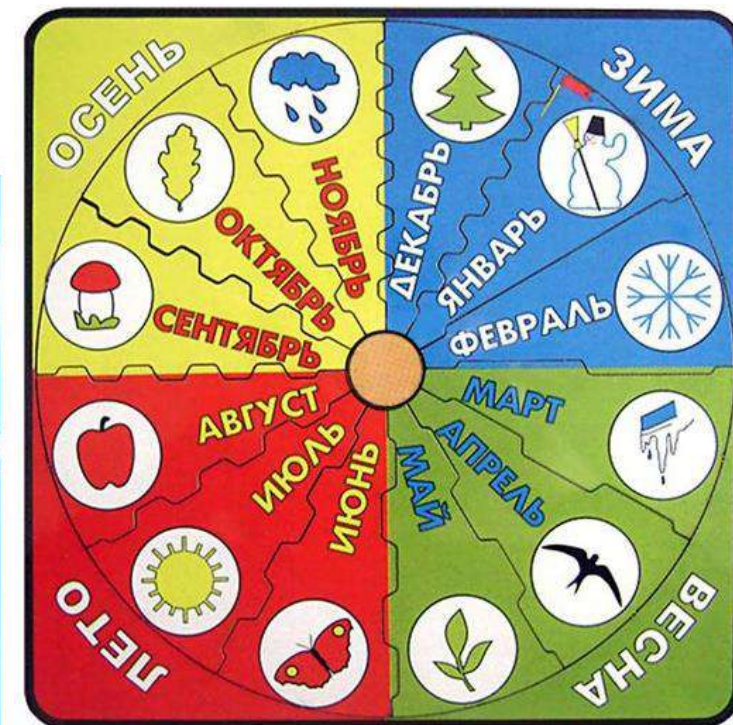



Загадка.

Что общего между днями недели, временами года и круговоротом воды в природе?



Цикличность



Программирование циклических алгоритмов

**Основные
алгоритмические
структуры в Python**

Вопросы к изучению:



Понятие цикла



Виды циклов



Программирование циклов



Практическая работа (решение задач)



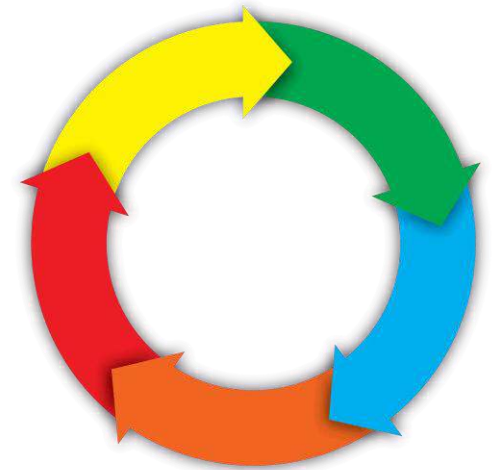
Понятие цикла

Цикл — это управляющая конструкция, предназначенная для организации многократного исполнения набора команд (блока кода).

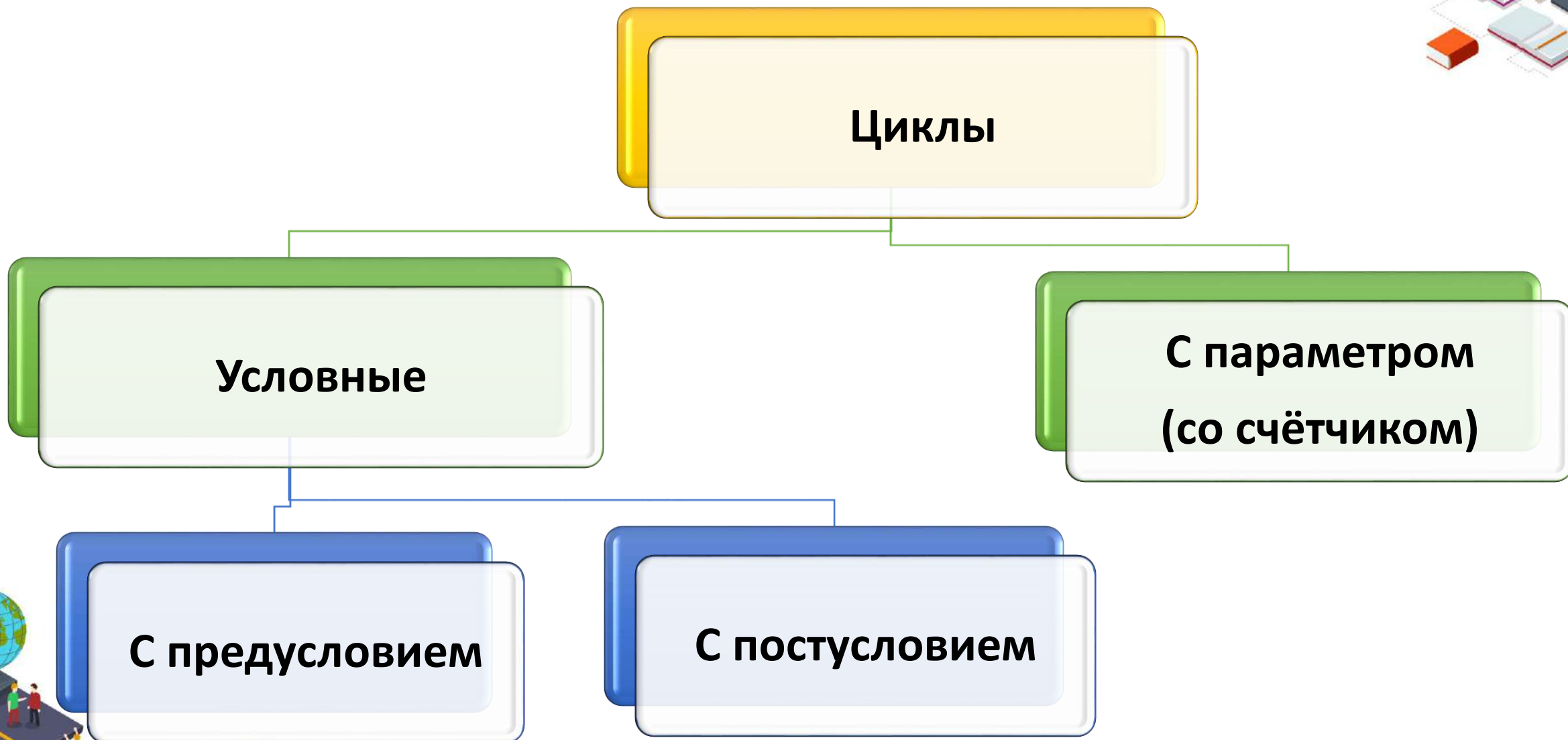


Тело цикла — последовательность инструкций, предназначенная для многократного исполнения.

Итерация — единичное выполнение тела цикла.



Виды циклов

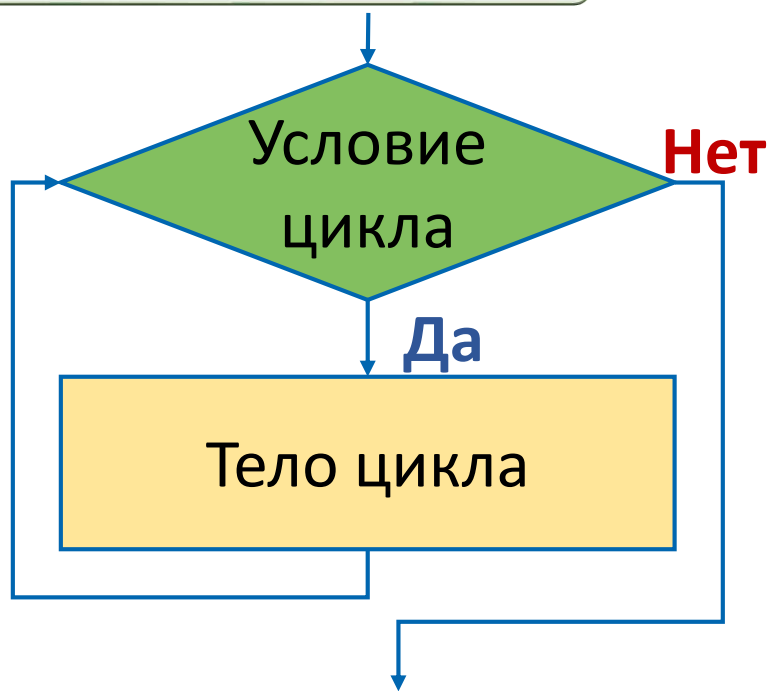


Блок-схемы условных циклов

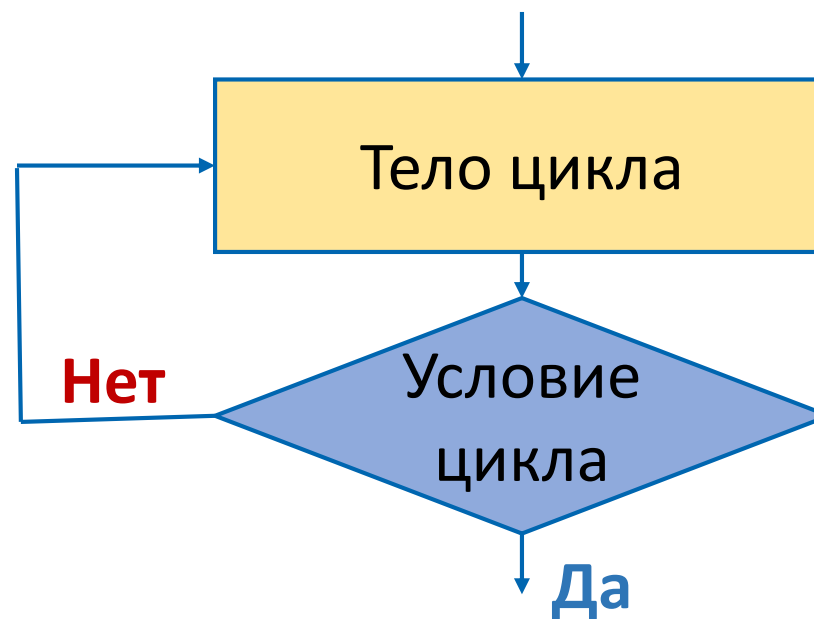


Условные

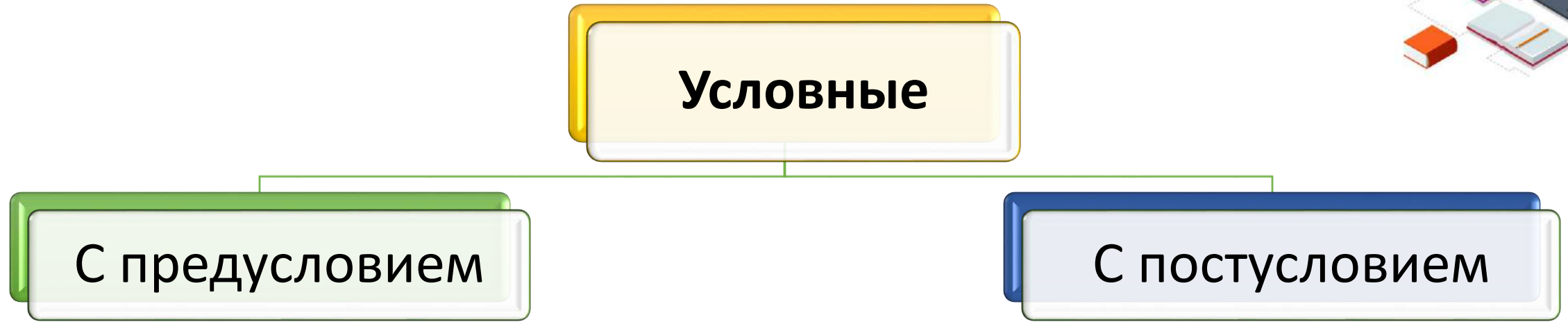
С предусловием



С постусловием



Блок-схемы условных циклов



while <условие цикла> :
 <инструкция 1>
 <инструкция 2>
 ...
 <инструкция n>
4 пробела

тело
цикла

while True:
 <инструкция 1>
 <инструкция 2>
 ...
 <инструкция n>
 if <условие цикла>:
 break

тело
цикла



Инструкции управления условным циклом



➤ завершение работы цикла – **break**

(в переводе с англ. – «прервать»,
*останавливает исполнение текущего
цикла и переходит к следующим за
ним командам*);

➤ прерывание текущего исполнения
цикла – **continue**

*(останавливает текущее
исполнение тела цикла и
переходит к следующему,
начиная с проверки условия).*

В цикле может присутствовать блок **«else»**, который сработает, если выход из цикла произошел без помощи **break**.



Цикл с предусловием

Выполняется до тех пор, пока условие **ИСТИННО**

Если условие никогда не станет ложным, то программа **зацикливается**

Если условие цикла изначально ложно, то тело цикла не выполнится ни разу



Цикл с постусловием

Проверка условия происходит после выполнения очередного шага цикла

Содержит условие выхода из цикла

Всегда выполняется хотя бы один раз



Программирование циклов с параметром (с заданным числом повторений)

Общий вид оператора:

```
for <параметр> in range (k, n, m):  
    <оператор>
```

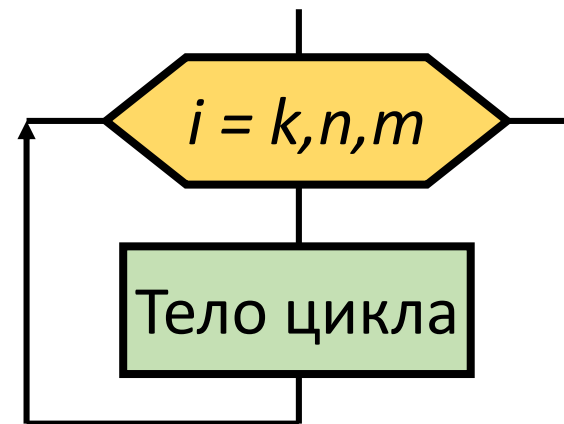
<параметр> - переменная целого типа

range() – диапазон значений:

m – **начальное значение параметра** (по умолчанию равен 0)

n – **конечное значение параметра**, не включая правую границу, число m

k – **шаг изменения параметра**, по умолчанию равен 1.



После каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условие выхода из цикла - достижение параметром конечного значения.



Цикл с параметром:

for <имя параметра> **in range** (m, n, k):

<инструкция 1>

<инструкция 2>

...

<инструкция n>

тело
цикла

4 пробела

Цикл может перебирать значения переменной **в обратном направлении**, если $m > n$, а шаг k отрицательный, например -1.

Роль множества могут выполнять:

- ✓ значения, перечисленные в скобках;
- ✓ диапазоны значений с известным шагом
- ✓ списки, словари

range (n) **range** (m, n) **range** (m, n, k)



Цикл с параметром

Количество шагов цикла **заранее известно** или **может быть вычислено** до начала цикла

В заголовке цикла указывается начальное, конечное значение и шаг цикла

В теле цикла с параметром нельзя изменять значение параметра



Вопросы:

сколько раз будет выполнено тело цикла?

а) **for** i in range(15): s = s + 1

б) **for** i in range(10, 15): s = s + 1

в) n = 10

while n != 5:

n = n - 1

г) **for** i in range(1, 1): s = s + 1

д) m = 5

for i in range(m - 1, m + 1):

s = s + 1



Решение задач

Задача №1

Напишите программу, которая считывает целое неотрицательное число n и выводит его факториал.

Факториал числа n обозначается $n!$ и равен произведению всех натуральных чисел от 1 до n включительно: $n! = 1 \times 2 \times \dots \times n$

Факториал нуля равен 1.

Ввод

5

Вывод

5 ! = 120

Подсказка

Обозначим факториал числа переменной F и присвоим ей значение 1
 $F = 1$

В теле цикла **for** будем менять значение F по формуле $F = F * i$, где i - переменная цикла, пробегающая значения от 0 до n включительно.



Решить самостоятельно

Задача №2

Напишите программу, которая в последовательности целых чисел определяет количество **двузначных** чисел, кратных **3 или 7**.

Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000. Программа должна вывести одно число: количество двузначных чисел, кратных 3 или 7.

Ввод

123

14

9

70

0

Вывод

2

Подсказка

Воспользуемся циклом **while**, условием продолжения которого будет $n \neq 0$, а в теле цикла поставим сложное условие для подсчёта количества двузначных чисел, кратных 3 или 7.



Решить самостоятельно:

Задача №3

Напишите программу, которая ведёт обратный отсчёт.

Пример:

Ввод	Вывод
5	<i>Осталось секунд: 5 Осталось секунд: 4 Осталось секунд: 3 Осталось секунд: 2 Осталось секунд: 1 Осталось секунд: 0 Пуск</i>





Решить самостоятельно:

Задача №4

Напишите программу, выводящую на экран изображение ёлочки из символов «*» заданной высоты.



Подсказка

Пусть n – количество уровней у нашей ёлочки, тогда $(2 * n - 1)$ – количество «*» на n -м уровне. Как найти количество пробелов до «*» на n -м уровне? Воспользуемся циклом **for** для написания кода.

Решить самостоятельно:

Задача №5

На контрольной работе по алгебре ученикам 9 класса было предложено 10 примеров. Неудовлетворительная оценка выставляется, если правильно решено **менее половины** примеров. Сколько неудовлетворительных оценок было получено учениками? Если **хотя бы один** из учеников **правильно решил все задачи**, выведите **YES**, иначе выведите **NO**.

Программа получает на вход количество учеников в классе **N** ($1 \leq N \leq 30$), затем для каждого ученика вводится количество правильно решённых примеров.

Ввод

Вывод

4

1

3

YES

10

5

7

Ввод

Вывод

4

2

3

NO

9

2

8



Решить самостоятельно:

Задача №6*

Напишите программу, которая отгадывает загаданное вами целое число от 1 до 1000 (пользователь загадывает число и не сообщает программе). Угадать его нужно не более чем за 10 попыток.

На каждую попытку пользователь отвечает, что загаданное число больше названного (вводит символ ">"), меньше названного ("<") или угадано правильно ("=").



Самое главное:

В языке Python имеются **два вида операторов цикла**: **while** (цикл-ПОКА), **for** (цикл с параметром), которые помогают реализовывать 3 вида циклов (2 условных и цикл с параметром).

Если число повторений тела цикла известно, то лучше воспользоваться оператором **for**;
в остальных случаях используется оператор **while**.

