

ИНФОРМАТИКА

9

класс

ЗАПИСЬ ВСПОМОГАТЕЛЬНЫХ АЛГОРИТМОВ НА ЯЗЫКЕ PYTHON

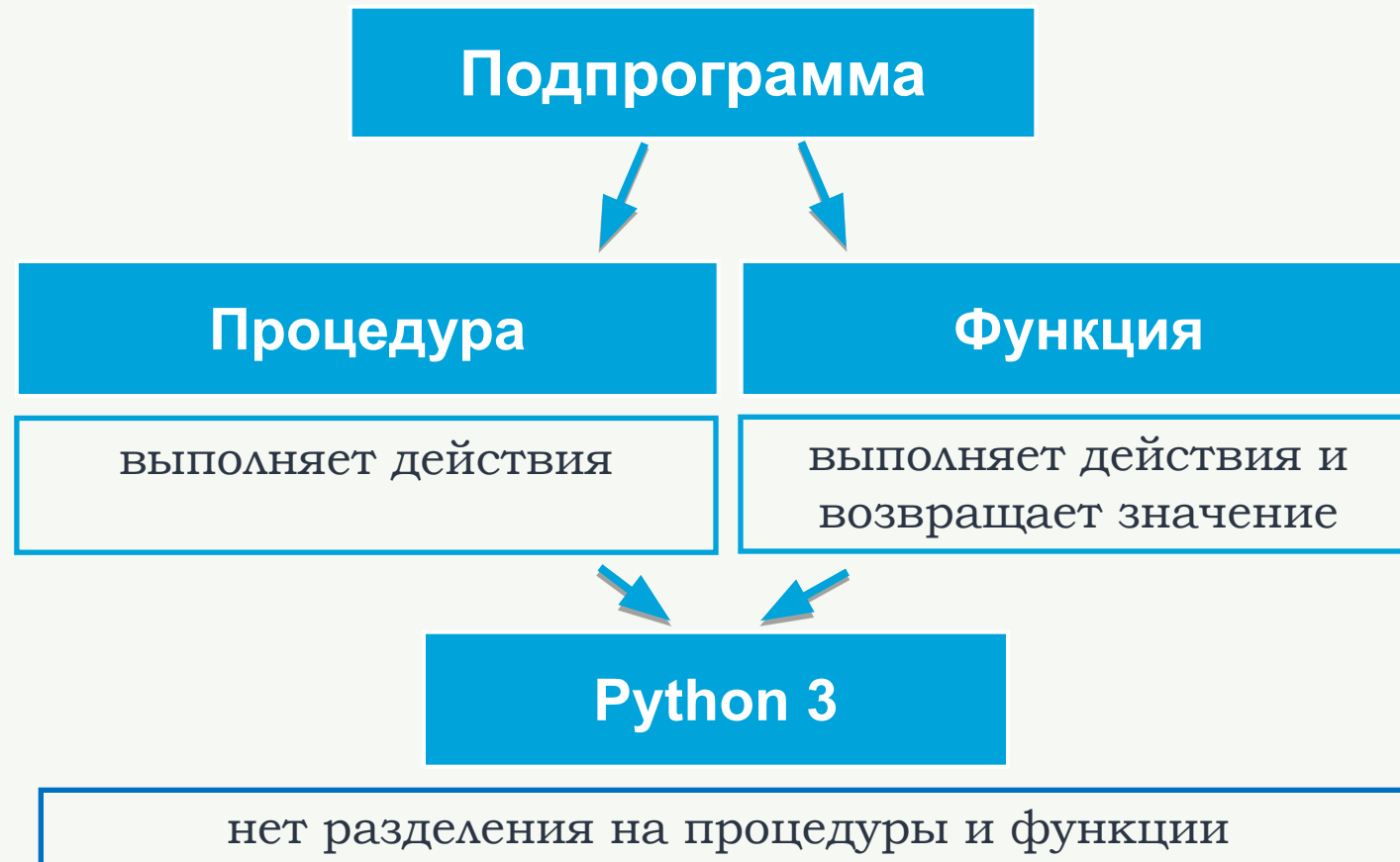
АЛГОРИТМЫ И ПРОГРАММИРОВАНИЕ

КЛЮЧЕВЫЕ СЛОВА

- ◆ основной алгоритм
- ◆ вспомогательный алгоритм
- ◆ главная программа
- ◆ подпрограмма
- ◆ процедура
- ◆ функция
- ◆ рекурсивная функция

ПОДПРОГРАММЫ

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью **подпрограмм**.



ПРОЦЕДУРА

Процедура — это подпрограмма, выполняющая некоторые действия; она может иметь произвольное количество входных параметров.

```
def <имя_процедуры> (<список параметров>) :  
    <операторы>
```

Для того чтобы процедура заработала, её необходимо вызвать по имени; причём таких вызовов может быть сколько угодно.

Для вызова функции достаточно указать её имя со списком фактических параметров в любом выражении, в условиях (после слов **if, while**) или в операторе **print** главной программы.



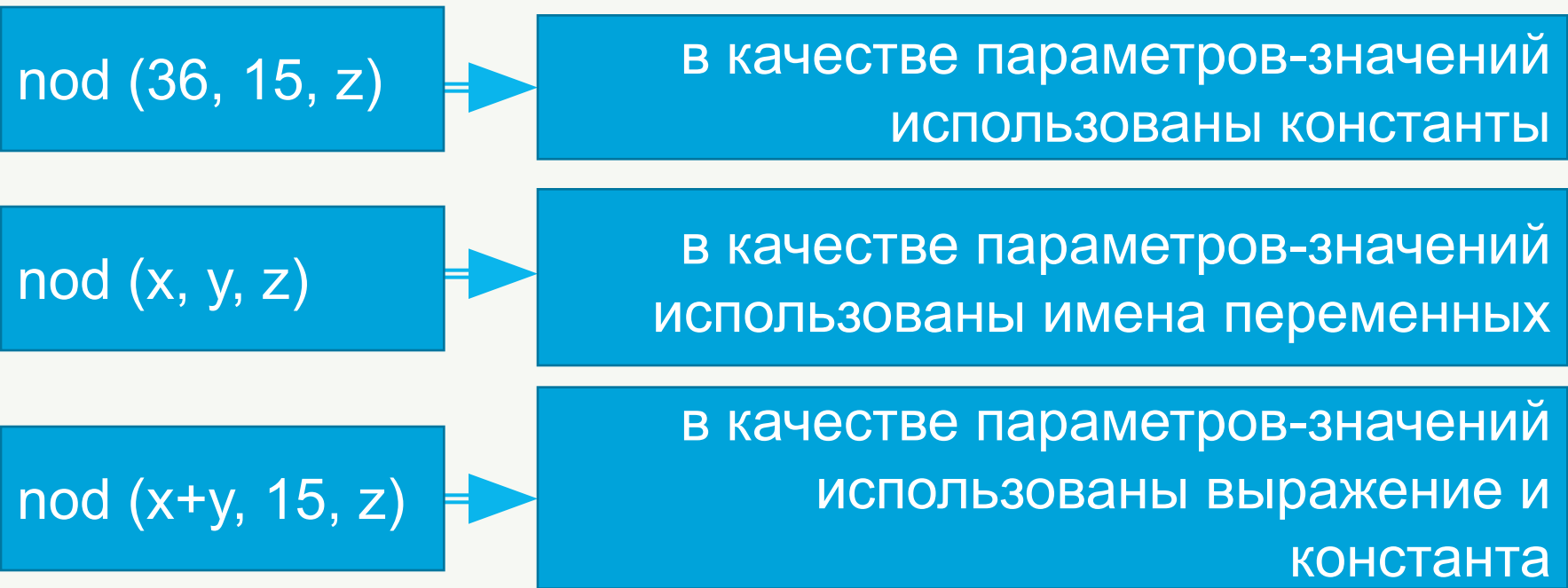


ПРОЦЕДУРА ДЛЯ НАХОЖДЕНИЯ НОД

```
def nod(a, b):  
    global x  
    while a != b:  
        if a > b:  
            a = a - b  
        else:  
            b = b - a  
    x = a
```



ВАРИАНТЫ ВЫЗОВА ПРОЦЕДУРЫ



Между фактическими и формальными параметрами должно быть полное соответствие по количеству, порядку следования и типу.

Программа вычисления НОД

```
def nod (a, b):  
    global x  
    while a != b:  
        if a > b:  
            a = a - b  
        else:  
            b = b - a  
    x = a
```

подпрограмма

```
k = int(input('k='))  
x = int(input())  
for i in range(1, k):  
    y = int(input())  
    nod(x, y)  
print('НОД=', x)
```

основная программа

ФУНКЦИЯ

Функция - это подпрограмма, имеющая единственный результат, записываемый в ячейку памяти.

Подпрограмма-функция возвращает результат (число, символьную строку и т.д.), который можно использовать в основном программе.

```
def <имя_функции> (<список параметров>) :  
    <операторы>  
    return < результат >
```

Для того чтобы функция заработала, её необходимо вызвать по имени; причём таких вызовов может быть сколько угодно.



ФУНКЦИИ БЕЗ ВОЗВРАЩАЕМОГО ЗНАЧЕНИЯ



Список формальных параметров

```
def <имя_функции> (<список параметров>):  
    <операторы>
```

Для вызова достаточно указать имя функции со списком фактических параметров.

ПРОГРАММА С ФУНКЦИЕЙ

```
def max(a, b):  
    if a > b:  
        m = a  
    else:  
        m = b  
    return m  
a, b, c, d = map(int,  
input().split())  
f = max(max(a, b), max(c,  
d))  
print('f=', f)
```

Измените программу так, чтобы с её помощью можно было найти:

- a) минимальное из четырёх чисел;
- б) разность максимального и минимального из четырёх чисел.

ПРОГРАММА С ПРОЦЕДУРОЙ

```
def nod (a, b):  
    global x  
    while a != b:  
        if a > b:  
            a = a - b  
        else:  
            b = b - a  
  
    x = a
```

```
m = [16, 32, 40, 64, 80, 128]
```

```
x = m[0]
```

```
for i in range(1, 6):
```

```
    y = m[i]
```

```
    nod(x, y)
```

```
print('НОД=', x)
```

Процедура

Основная программа

Вызов процедуры

ПРОГРАММА С ПРОЦЕДУРОЙ

```
def nod (a, b):  
    global x  
    while a != b:  
        if a > b:  
            a = a - b  
        else:  
            b = b - a  
  
    x = a  
m = [16, 32, 40, 64, 80, 128]  
x = m[0]  
for i in range(1, 6):  
    y = m[i]  
    nod(x, y)  
print('НОД=', x)
```

Измените программу так, чтобы с её помощью можно было найти:

- a) наибольший общий делитель следующих пяти чисел: 12, 24, 30, 48 и 51;
- b) наибольший общий делитель произвольных десяти целых двузначных чисел.

ФУНКЦИЯ ПОИСКА МАКСИМАЛЬНОГО ИЗ 2-Х ЧИСЕЛ

```
def max (a, b) :  
    if a > b:  
        m = a  
    else :  
        m = b  
    return m
```

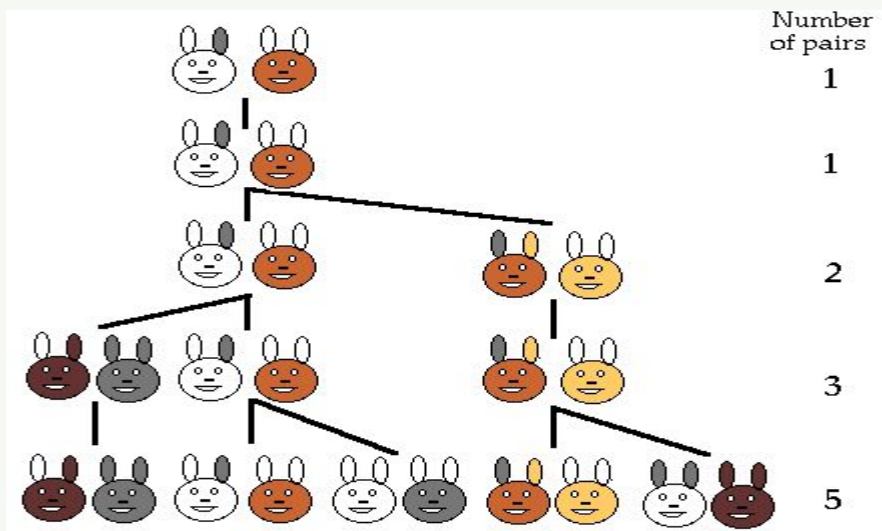
```
print (max (6, 8))
```

```
x = max (6, 8)
```

ЗАДАЧА ПРО КРОЛИКОВ

В январе Саше подарили пару новорождённых кроликов. Через два месяца они дали первый приплод - новую пару кроликов, а затем давали приплод по паре кроликов каждый месяц.

Каждая новая пара также даёт первый приплод (пару кроликов) через два месяца, а затем - по паре кроликов каждый месяц. Сколько пар кроликов будет у Саши в декабре?



Числа 1, 1, 2, 3, 5, 8, ... образуют так называемую **последовательность Фибоначчи**, названную в честь итальянского математика, впервые решившего соответствующую задачу ещё в начале XIII века.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть $f(n)$ количество пар кроликов в месяце с номером n .

По условию задачи:

$$f(1) = 1,$$

$$f(2) = 1,$$

$$f(3) = 2.$$

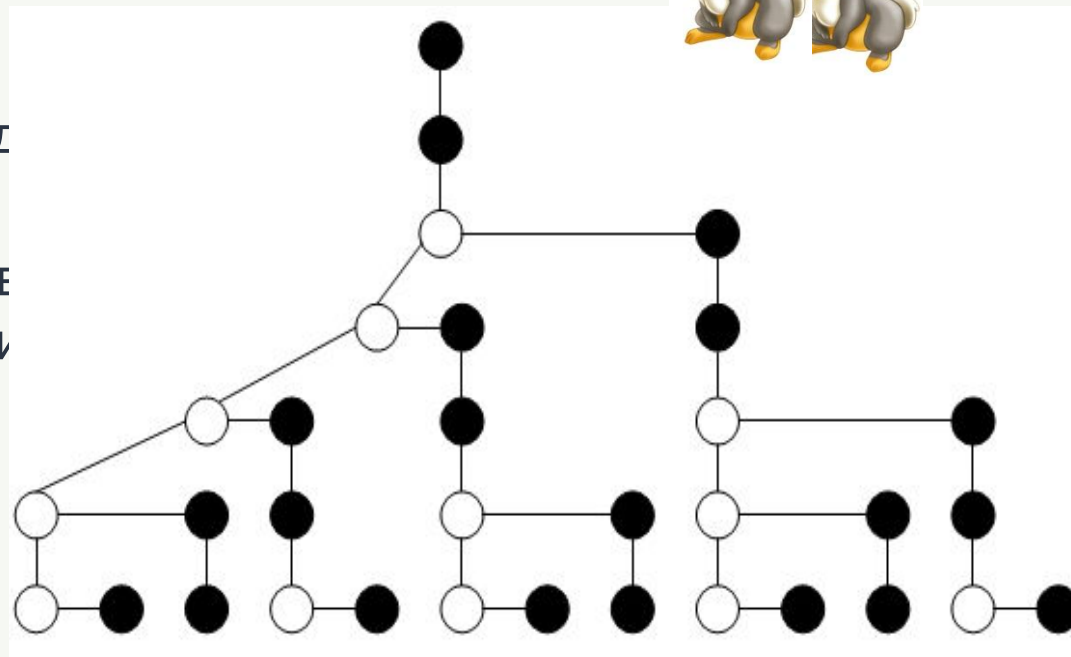
Из двух пар, имеющихя в марте, дать приплод в апреле сможет только одна: $f(4) = 3$.

Из пар, имеющихя в апреле, дать приплод в мае смогут только пары, родившиеся в марте и ранее:

$$f(5) = f(4) + f(3) = 3 + 2 = 5.$$

В общем случае:

$$f(n) = f(n-1) + f(n-2), \quad n \geq 3.$$



ФУНКЦИЯ ДЛЯ ЧИСЕЛ ФИБОНАЧЧИ

```
def f(n):  
    if n == 1 or n == 2:  
        rez = 1  
    else:  
        rez = f(n - 1) + f(n - 2)  
    return rez
```



Полученная функция **рекурсивная** - в ней реализован способ вычисления очередного значения функции через вычисление её предшествующих значений.

Напишите программу для нахождения $f(12)$.

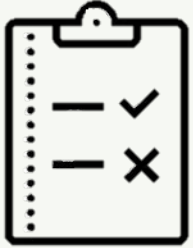
Измените программу так, чтобы она вычисляла и выводила 10 первых членов последовательности Фибоначчи.

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью подпрограмм. В Python различают два вида подпрограмм: процедуры и функции.

Процедура – подпрограмма, выполняющая некоторые действия, она может иметь произвольное количество входных параметров. Процедура не возвращает результат, который можно присвоить переменной, а только обрабатывает входные данные. В процедуру могут быть переданы глобальные переменные, которые она обрабатывает, изменяет.

В отличие от процедуры, **функция** не только выполняет какие-то команды, но и возвращает результат в виде числа, символьной строки или др. Вызов функции можно использовать в арифметических выражениях и условиях так же, как и переменную такого же типа, который возвращает функция. В теле функции можно вызывать другие функции и процедуры.

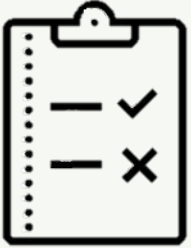
Рекурсивная функция – это функция, которая вызывает сама себя, напрямую или через другие процедуры и функции.



ВОПРОСЫ И ЗАДАНИЯ

Для чего используются подпрограммы?

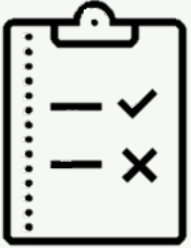




ВОПРОСЫ И ЗАДАНИЯ

В чём основное различие процедур и функций?





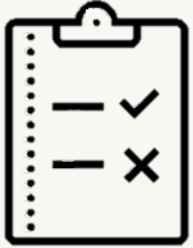
ВОПРОСЫ И ЗАДАНИЯ

Напишите программу вычисления наименьшего общего кратного следующих четырёх чисел: 36, 54, 18 и 15.

Используйте процедуру вычисления наибольшего общего делителя двух чисел.

Подсказка: $\text{НОК}(a, b) = \frac{a \cdot b}{\text{НОД}(a, b)}$





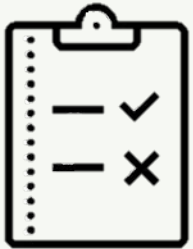
ВОПРОСЫ И ЗАДАНИЯ

Напишите программу вычисления выражения:

$$s = 1! + 2! + 3! + \dots + n!$$

Используйте функцию вычисления факториала.





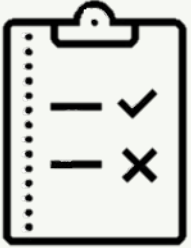
ВОПРОСЫ И ЗАДАНИЯ

Напишите программу перестановки значений переменных a , b , c в порядке возрастания, т. е. так, чтобы $a < b < c$. Используйте функцию **swap**.

Исходные данные вводятся с клавиатуры.

Пример входных данных	Пример выходных данных
1 2 3	1 2 3
2 1 3	1 2 3
3 1 2	1 2 3
2 3 1	1 2 3





ВОПРОСЫ И ЗАДАНИЯ

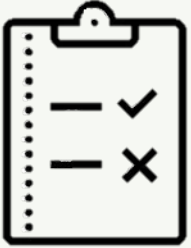
Напишите программу вычисления выражения:

$$s = x^3 + x^5 + x^n,$$

где x и n вводятся с клавиатуры.

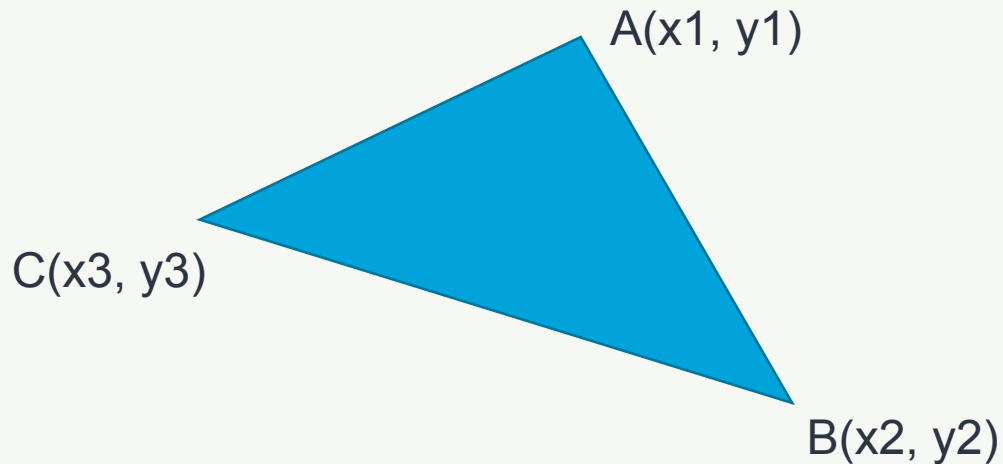
Используйте функцию вычисления степени.

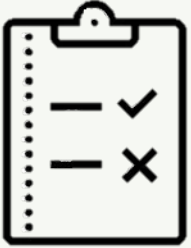




ВОПРОСЫ И ЗАДАНИЯ

Напишите функцию, вычисляющую длину отрезка по координатам его концов. С помощью этой функции напишите программу, вычисляющую периметр треугольника по координатам его вершин.

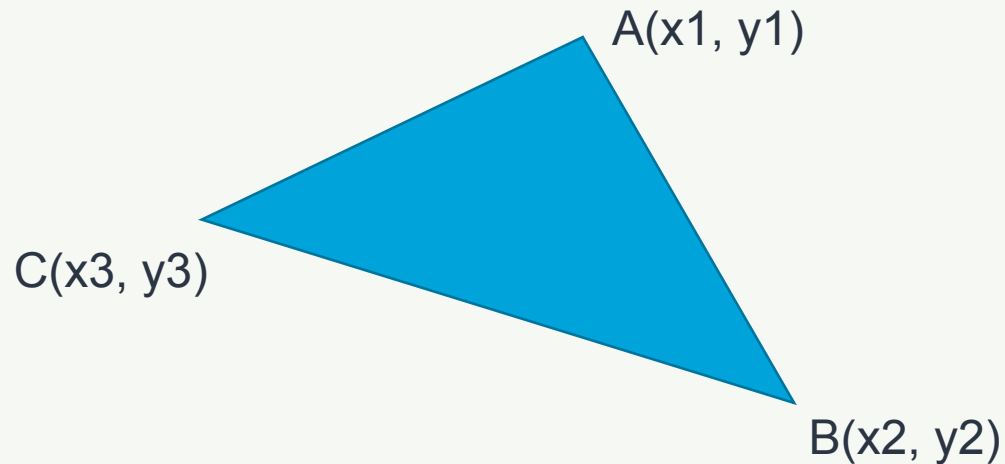




ВОПРОСЫ И ЗАДАНИЯ

Напишите функцию, вычисляющую площадь треугольника по целочисленным координатам его вершин.

С помощью этой функции вычислите площадь четырёхугольника по координатам его вершин.



Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью **подпрограмм**.

