

2 занятие

Школа::Кода

«Основы программирования на языке
Python»

Проверка присутствия



ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ

1. Реальная задача (идея)
2. Математическая постановка (методы, формулы)
3. Алгоритмизация (алгоритм)
4. Программирование (текст программы)
5. Отладка и применение (исполняемый код, результаты)

ПРИМЕР

- Сравнить 2 числа. Если первое число больше второго, то результатом считать единицу, иначе – ноль.

```
a=int(input())
b=int(input())
if a>b:
    print(1)
else:
    print(0)
```

$$f(x,y) = \begin{cases} 1, & \text{если } x>y; \\ 0, & \text{если } x\leq y \end{cases}$$

```
целое x, y, z;
ввести (x, y)
если x>y то z=1;
иначе z=0;
вывести (z);
```

ФОРМЫ ПРЕДСТАВЛЕНИЯ

- Словесная – ориентирована на человека
- Псевдокод
- Схематическая (например, графическая блок-схема)
- Язык программирования – ориентирована на машину

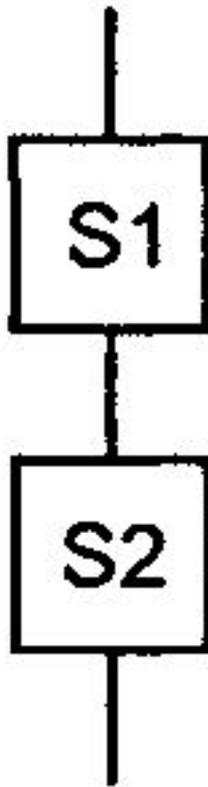
ЗА И ПРОТИВ

Давайте подумаем над плюсами
и минусами каждого подхода

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ

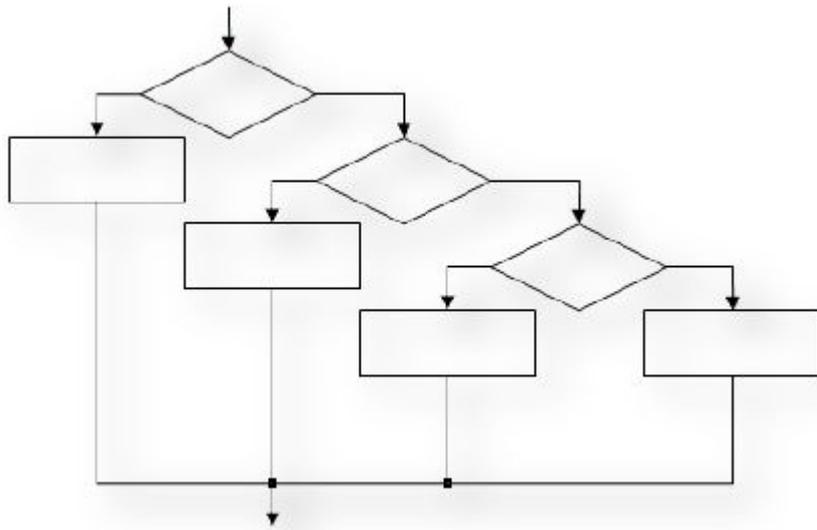
- последовательное выполнение (линейная);
- ветвление;
- цикл (или: цикл-пока, цикл-до).

ЛИНЕЙНАЯ



- Действие 1
- Действие 2
- ...
- Действие n

ВЕТВЛЕНИЕ 1



если (условие1)

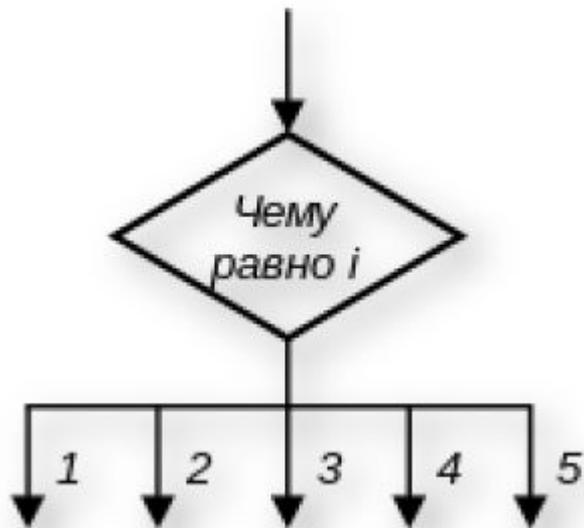
ТО

действие 1

иначе

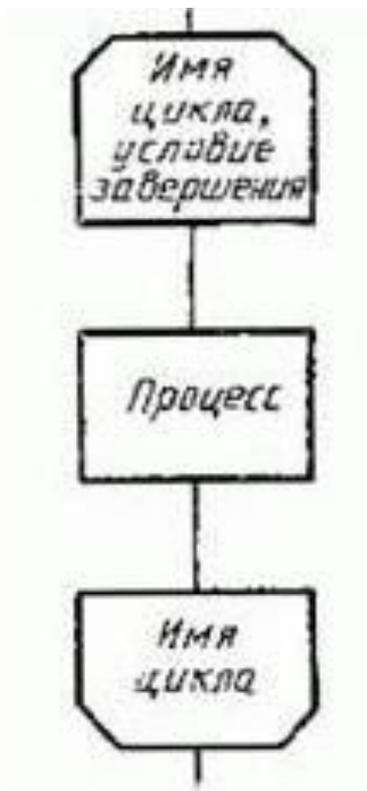
действие 2

ВЕТВЛЕНИЕ 2



- выбор (значение):
- константа 1:
действие1
- константа 2:
действие2
- иначе: действие N

ЦИКЛ С ПРЕДУСЛОВИЕМ

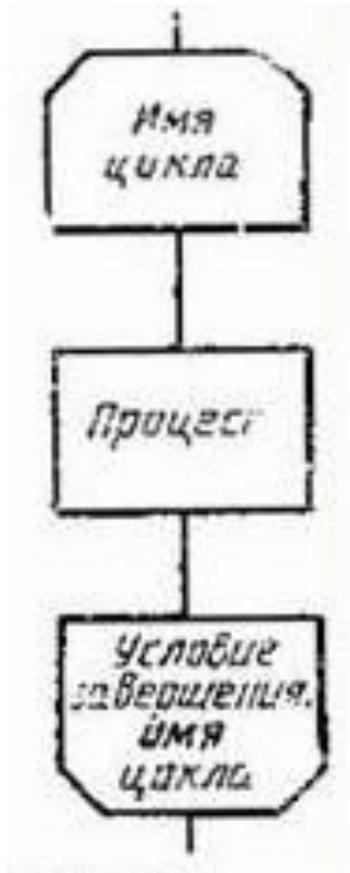


пока (условие)

цикл

действия

ЦИКЛ С ПОСТУСЛОВИЕМ



ЦИКЛ
действия
до (условие)

ЦИКЛ СО СЧЕТЧИКОМ



Написать в тетрадке алгоритмы.

1. Как заварить чайник?
2. Алгоритм действий перед школой
3. Поход в кино