

# Фидбек на язык Али

6 мая

## Программы

①

Корректная программа на языке Али, вычисляющая  $n$ -ое число Фибоначчи:

```
WithFun
  Fun Fibonacci With n ThatsAll
    Seq
      Assign a 0
      Seq
        Assign b 1
        If == n 0
          Assign res a
          Seq
            While >= n 2
              Seq
                Assign c + a b
                Seq
                  Assign a b
                  Seq
                    Assign b c
                    Assign n - n 1
              Assign res b
          Return res
        ThatsAllFun
  Seq
    Read n
    Write Call Fibonacci With n ThatsAll
```

②

Программа, проверяющая, является ли число  $n$  точным квадратом:

---

```

WithFun
  Fun isSqr With n ThatsAll
    Seq
      Assign i 0
      Seq
        While < * i i n
          Assign i + i 1
        If == * i i n
          Assign res 1
          Assign res 0
      Return res
ThatsAllFun

Seq
  Read n
  Write Call isSqr With n ThatsAll

```

---

③

Программа вычисляющая наименьшее общее кратное чисел  $a$  и  $b$ :

---

```

WithFun
  Fun mod With a With b ThatsAll
    Nop
    Return - a * b / a b
WithFun
  Fun gcd With a With b ThatsAll
    If == b 0
      Assign res a
      Assign res Call gcd With b With Call mod With a With b ThatsAll ThatsAll
    Return res
WithFun
  Fun lcm With a With b ThatsAll
    Nop
    Return / * a b Call gcd With a With b ThatsAll
ThatsAllFun

Seq
  Read a
  Seq
    Read b
    Write Call lcm With a With b ThatsAll

```

---

## Feedback 2.0 (06.05.2020)



1. Синтаксис функций очень эргономичен и сочетается с естественным стилем написания кода, что позволяет писать программы удобно и быстро.
2. Ошибок в работе парсера не обнаружено.



1. Слишком длинные ключевые слова `That'sAll` и `That'sAllFun` ☹. Возможно стоит их заменить на более краткие формы или на пунктуационные конструкции.

# Feedback (09.04.2020)



1. Язык очень лаконичный, нет обязательных лишних скобок (вообще нет скобок ☺).
2. Простой понятный синтаксис. Даже польская нотация не замедляет написание программ.
3. Ошибок в работе парсера не обнаружено.



1. Документация очень хорошая, есть небольшая просьба к автору языка добавить в документацию предназначение инструкции **Nop**.



1. Единственное, что затрудняло написание программ на этом замечательном языке – это ограничение команды **Seq**, которая позволяет следовать за ней ровно двум инструкциям:
  - При отсутствии должного обрамления кода отступами, сильно затрудняется чтение, а также само создание программы, так как нужно держать в голове, что за каждой команды **Seq** следует ровно две инструкции.
  - Количество самих команд **Seq** увеличивается и становится пропорционально числу остальных команд.

Могу предложить создателю языка следующие решения этой проблемы:

- (a) Можно добавить дополнительный числовой аргумент после ключевого слова **Seq**, который будет содержать информацию о том, сколько инструкций содержит в себе текущая последовательность инструкций. Это не только решит проблему, но и позволит избавиться от ключевого слова **Nop**, заменив его на **Seq 0**. Код первой программы с этой модификацией выглядел бы так:

---

```
Seq 4
  Read n
  Assign a 0
  Assign b 1
  If == n 0
    Write a
    Seq 2
      While >= n 2
        Seq 4
          Assign c + a b
          Assign a b
          Assign b c
          Assign n - n 1
        Write b
```

---

- (b) Можно просто добавить разделители, например фигурные скобки и точки с запятой, но тогда пропадёт изюминка этого замечательного языка программирования.