

Лабораторная работа №2

студента группы ПИ-211

Зубкова Ильи Леонидовича

Выполнение _____ Защита _____

УСЛОВНЫЕ КОНСТРУКЦИИ. ЦИКЛЫ.

Цель работы: приобрести навыки использования условных конструкций и циклов.

Содержание работы:

Вариант №11

С клавиатуры вводится последовательность натуральных чисел. Признаком конца ввода - 0. Вывести на экран максимальный элемент и его номер.

1. Разработать алгоритм решения задачи, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе.

Ход работы

1. Разработал алгоритм решения, описав его в виде блок-схемы.

Блок-схема:

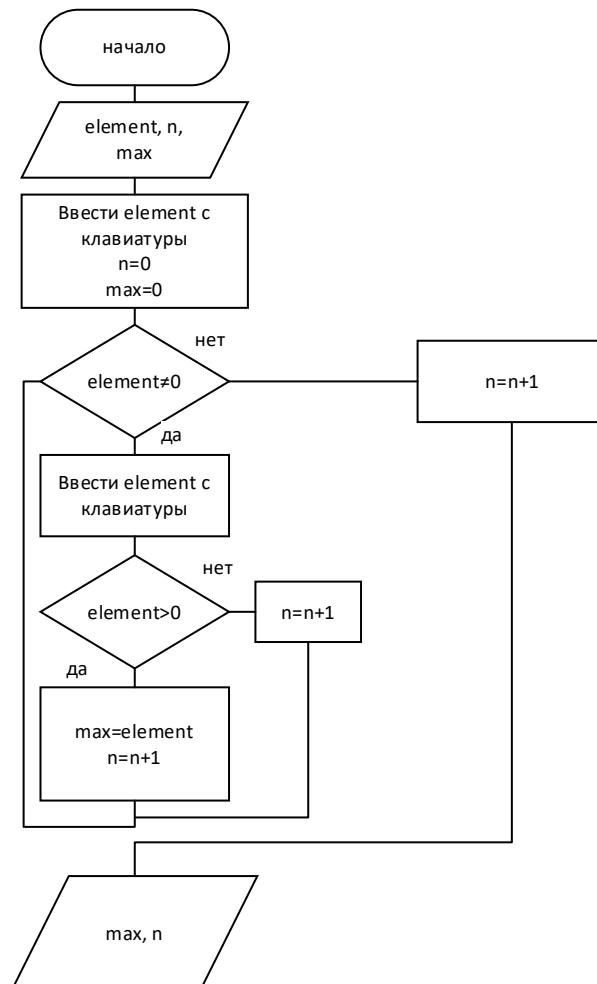


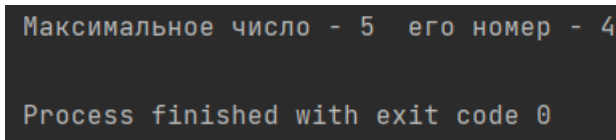
Рис. 1. Блок-схема алгоритма

2. Написал текст программы и отладил её.

Текст программы:

```
max = 0
element = -1
n=0
while element != 0:
    element = int(input())
    if element > max:
        max = element
        n=n+1
    else:
        n=n+1
print('Максимальное число -',max, ' его номер -',n)
```

Результат выполнения программы:



```
Максимальное число - 5 его номер - 4
Process finished with exit code 0
```

Рис. 2. Проверка программы

Вывод: условные конструкции и циклы позволяют решать задачи, без которых они бы не были осуществлены. Их использование в среде программирования PyCharm не вызывает труда, если знать теорию.

Ответы на контрольные вопросы:

1. Какие виды алгоритмов называются разветвляющимися?

Разветвляющийся алгоритм - алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

2. Опишите механизм работы условного оператора.

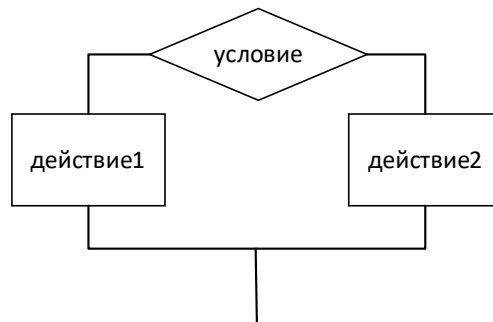
1) Неполный условный оператор:

Если выполняется <условие>, выполнить <действие>.

2) Полный условный оператор:

Если выполняется <условие>, выполнить <действие1>, иначе выполнить <действие2>.

3. Приведите пример описания условного оператора с помощью блок-схемы.



4. Какие алгоритмы называются циклическими?

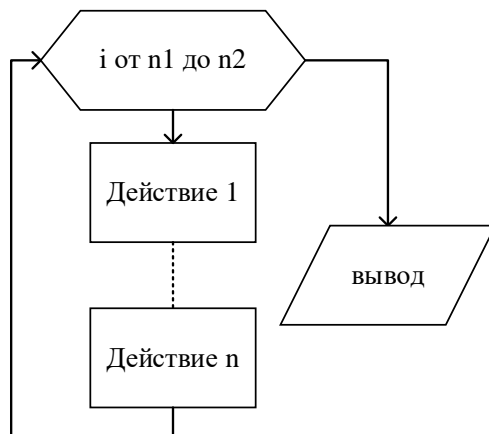
Циклический алгоритм – описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие.

5. Какие виды структур повторения вам известны?

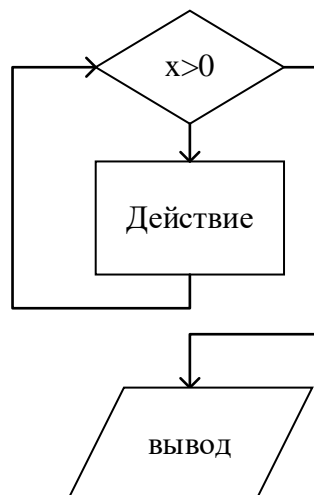
Цикл for, цикл while.

6. Приведите пример каждого вида с помощью блок-схемы.

Цикл for:



Цикл while:



7. Как реализовать цикл со счетчиком?

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

 <действие>

 <вывод>

2) `while i<0:`

 <действие>

 <вывод>

8. Что такое итератор?

Итератор представляет собой объект перечислитель, который для данного объекта выдает следующий элемент, либо бросает исключение, если элементов больше нет.

9. Какие функции, используемые для итерируемых объектов, вы знаете?

`next()`, `iter()`, `range()`, `count()`.

10. Что они возвращают?

iter() – возвращает сам себя, вызывается в самом начале. Позволяет пользоваться циклом `for` и выражением `in`;

next() – перебирает коллекцию по одному элементу и вызывает ошибку `StopIteration`, когда их не остается;

range() – заранее не формирует последовательность из гигантского количества чисел, а когда мы обращаемся к некому ее элементу по индексу, он выводится почти сразу;

count() – формирует бесконечную последовательность чисел с определенным шагом.

11. Чем отличаются разветвляющиеся алгоритмы от линейных?

Линейные алгоритмы выполняются последовательно, а разветвляющиеся содержат хотя бы одно условие, в результате проверки которого происходит переход на один из двух возможных ходов.

12. Почему нельзя выполнить обмен значений двух переменных в два шага: `a=b; b=a`?

Потому что после первого шага **a=b** переменные будут равны друг другу, то есть потеряется значение одной из них.