

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Python. Простейшие циклические программы. Оператор цикла с предусловием

Методические указания к лабораторной работе



Рязань 2017

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Python. Простейшие
циклические программы.
Оператор цикла с предусловием**

Лабораторная работа №5

Методические указания к лабораторной работе

Рязань 2017

УДК 004.432

Python. Простейшие циклические программы. Оператор цикла с предусловием: методические указания к лабораторной работе. / Рязан. гос. радиотехн. универ.; Сост.: А.Н. Пылькин, Н.Н. Степанов, Н.А. Тярт. – Рязань, 2017 г.

Рассмотрены понятия циклического алгоритма и его разновидностей. Приведены синтаксис и описание действия оператора цикла с предусловием while. Рассмотрен механизм форматирования вывода. Рассмотрены примеры применения оператора while.

В качестве практических заданий предлагается составить программу, связанную с задачей табулирования функции одной переменной.

Ил.: 9. Библиогр.: 4 назв.

Печатается по решению Научно-методического совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра информатики, информационных технологий и защиты информации ФГБОУ ВО «Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского» (зав. каф., к.т.н., доц. Скуднев Д.М.).

Цель работы

Знакомство с простейшими примерами программирования циклических программ.

Структура и состав цикла

Под **циклическостью** понимают такую форму повторяемости, при которой одни и те же действия многократно выполняются на одном этапе обработки информации для различных значений обрабатываемых данных. Процесс циклическости реализуется в программах с помощью *циклов*, которые в зависимости от особенностей их организации, управления и структуры можно разделить на следующие типы:

- цикл с постусловием;
- цикл с предусловием;
- цикл с параметром;
- итерационный цикл;
- вложенные циклы.

Любой из перечисленных циклов имеет общие структурные особенности. Цикл состоит из четырех частей, расположение которых и взаимосвязь зависят от типа цикла:

- 1) подготовка цикла;
- 2) тело цикла;
- 3) модификация/изменение параметра/параметров цикла;
- 4) проверка условия окончания / продолжения цикла.

Подготовка цикла необходима для задания начальных значений и подготовки правильного выполнения многократно повторяющегося *тела цикла* на первом шаге его реализации.

Модификация параметров подготавливает и задает те значения переменных, при которых реализуется следующее выполнение тела цикла. Многократное повторение тела цикла выполняется до выполнения *условия окончания цикла*.

В большинстве случаев используется один *параметр цикла*, с помощью которого осуществляется управление циклическим вычислительным процессом. Пусть x – параметр цикла, значение которого изменяется в процессе вычислений от начального значения x_0 с постоянным шагом h_x до конечного (граничного) значения x_n .

В дальнейшем такой закон изменения параметра x будем обозначать в следующем виде:

$$x = x_0(h_x)x_n.$$

В языке Python имеются операторы, которые непосредственно реализуют цикл с предусловием и цикл с параметром (цикл с заголовком). Пусть переменная x будет параметром цикла, т.е. той переменной, которая изменяется в цикле и значение которой используется для формирования условия продолжения (или окончания) цикла. На рис. 1 показан алгоритм, реализующий цикл с предусловием. В данном случае проверка условия продолжения цикла осуществляется перед реализацией тела цикла (ТЦ).

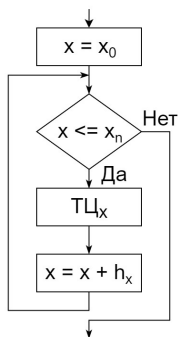


Рис. 1. Алгоритм цикла с предусловием

Разновидностью цикла с предусловием является **цикл с заголовком/параметром**. При реализации такого цикла закон изменения параметра задается в заголовке. Алгоритм цикла с заголовком приведен на рис. 2. В языке Python имеются широкие возможности задания различных законов изменения параметра цикла в заголовке.

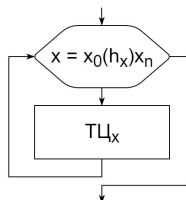


Рис. 2. Алгоритм цикла с заголовком

На рис. 3 показан алгоритм цикла с постусловием, в котором проверка условия окончания цикла осуществляется после выполнения тела цикла. В языке Python нет специального оператора для реализации цикла с постусловием, но, как показано далее, циклическая программа с постусловием может быть реализована достаточно просто.

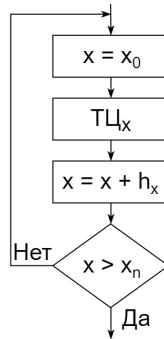


Рис. 3. Алгоритм цикла с постусловием

Другой особенностью циклов с постусловием, предусловием и заголовком, которые реализуют закон изменения параметра по схеме $x = x_0(h_x)x_n$, является возможность заранее определить число повторений тела цикла по формуле

$$N_x = \left[\frac{x_n - x_0}{h_x} \right] + 1,$$

где $[]$ означает целую часть числа.

Алгоритмы, показанные на рис. 1 – 3, являются базовыми при проектировании алгоритмов решения большого множества задач:

- табулирование функции;
- вычисление конечных сумм и произведений;
- приближенное вычисление определенного интеграла;
- построение графика функции и др.

Оператор цикла с предусловием

Оператор цикла `while` позволяет выполнить одну и ту же последовательность операторов (команд), пока проверяемое условие выполняется (истинно/имеет значение `True`). Условие выполняется перед выполнением тела цикла.

Общий вид оператора цикла с предусловием:

```
while <условие>:  
    <команда>  
    ...  
    <команда>
```

В результате выполнения следующей простейшей программы вычисляются целые степени двойки от 0 до 5.

```
i = 0  
while i <= 5:  
    print("2^%d = %d " % (i, 2**i))  
    i += 1
```

На экране дисплея будет следующий результат:

```
2^0 = 1  
2^1 = 2  
2^2 = 4  
2^3 = 8  
2^4 = 16  
2^5 = 32
```

Выше вы также можете наблюдать работу оператора %, используемого для форматирования выводимой строки. Работает он для любых строк (не только внутри оператора print()). Базовый синтаксис оператора % следующий:

```
<строка со спецификаторами> % (<подставляемые значения>)
```

Например, так можно вставить одну строку в другую:

```
'Hello, %s!' % 'World'  
# Результат = 'Hello, World!'
```

В данном примере строка 'World' подставляется в строку 'Hello, %s!' вместо *спецификатора преобразования* %s. Спецификатор преобразования определяет, к какому типу и формату записи будет приведено подставляемое значение при помещении в исходную строку. Существуют следующие основные спецификаторы:

- %d, %i, %u – десятичное число;
- %f – вещественное число;
- %e – вещественное число в научной нотации (с экспонентой);

- **%g** – вещественное число, в научной нотации только если экспонента больше точности или 4 порядков;
- **%c** – символ;
- **%s** – строка;
- **%%** – вывод знака «%».

Также можно указать точность, с которой будет выводиться число. Запись в формате **%.2f** указывает, что число будет выведено с 2 знаками после запятой.

Более подробно работа оператора **%** будет рассмотрена далее в лабораторной работе по строкам.

Оператор **while** позволяет реализовать алгоритм цикла с постусловием по следующей схеме:

```
while True
    <команда>
    ...
    <команда>
if not <условие>:
    break
```

Соответствующая программа для вычисления целых степеней двойки имеет вид:

```
i = 0
while True:
    print("2^%d = %d " % (i, 2 ** i))
    i += 1
    if i > 5:
        break
```

Результат выполнения программы тот же, что и в предыдущем примере. Отметим, что условие **i > 5** соответствует условию **not i <= 5**, т.е. два события **i > 5** и **i <= 5** образуют полную группу.

Табулирование функции одной переменной

При табулировании функции одной переменной (например, переменной x) требуется вычислить значения функции $y = f(x)$ при различных значениях аргумента $x = x_0(h_x)x_n$ и вывести значения функции y и аргумента x , при которых получены значения функции.

Пример 1. Решим задачу табулирования функции

$$f(x) = \begin{cases} 0, & x \leq 0; \\ x, & 0 < x < 1; \\ 1, & x \geq 1. \end{cases}$$

Алгоритм табулирования функции $y = f(x)$ при изменении аргумента x по закону $x = x_0(h_x)x_n$ показан на рис. 4.

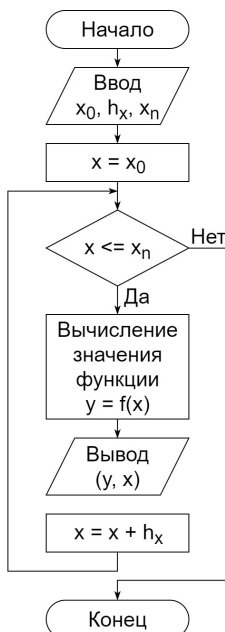


Рис. 4. Алгоритм табулирования функции

Программа табулирования функции может иметь следующий вид:

```

# coding=utf-8
import string

# Цель - табулирование функции y = F(x)
# с помощью оператора цикла с постусловием.
# Переменные:
# x - аргумент (параметр цикла);
# x0, xn - начальное и конечное значения;
# hx - шаг изменения;
  
```

```

#      у - функция.
# Программист: Степанов Н.Н.
# Дата написания: 12.02.2017.

print("Введите исходные данные:")

print("x0 = ", end='')
x0 = float(input())
print("hx = ", end='')
hx = float(input())
print("xn = ", end='')
xn = float(input())

print("Вы ввели:")
print("x0 = %.2f \t hx = %.2f \t xn = %.2f" % (x0, hx,
xn))
print("


|   |   |
|---|---|
| x | y |
|---|---|


")
print("


|   |   |
|---|---|
| x | y |
|---|---|


")
print("


|   |   |
|---|---|
| x | y |
|---|---|


")

x = x0
y = 0

while x < xn + hx / 2:
    if x < 0:
        y = 0
    elif x < 1:
        y = x
    else:
        y = 1

    print("|%-7.2f|%-7.2f|" % (x, y))
    x += hx

print("


|   |   |
|---|---|
| x | y |
|---|---|


")

```

Существует целый ряд задач, решение которых реализуется по схеме, аналогичной алгоритму табулирования функции одной переменной.

Пример 2. Пусть требуется определить максимальное значение функции в точках x_0, x_1, \dots, x_n . Цель поставленной задачи поясняется на рис. 5. Решение данной задачи реализуется с помощью циклического алгоритма, в теле которого определяется наибольшее значение функции $y = f(x)$ на данный момент и запоминается это значение функции и значение аргумента. Алгоритм нахождения максимального значения функции показан на рис. 6.

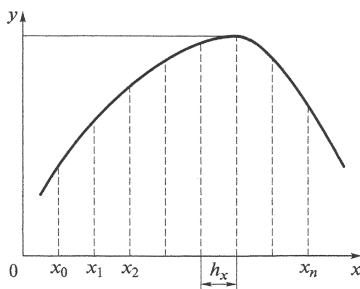


Рис. 5. Нахождение максимального значения функции

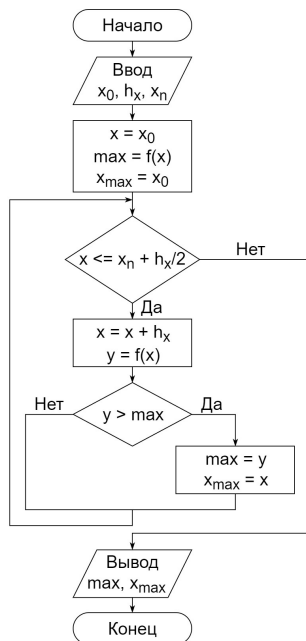


Рис. 6. Алгоритм нахождения максимального значения функции

В качестве условия продолжения цикла вместо неравенства $x \leq x_n$ используется условие $x \leq x_n + h/2$. Прибавление половины шага позволяет исключить проверки равенства двух вещественных значений, что из-за ошибок представления может привести к неправильному (досрочному) выходу из цикла.

Программа имеет следующий вид:

```
# coding=utf-8
import math

# Цель - вычисление максимального значения функции
# Переменные:
#   x - аргумент, изменяющийся от x0 с шагом hx до xn;
#   y = 2 * sin(x) + 3 * cos(x) - функция;
#   max - максимальное значение функции.
# Программист: Степанов Н.Н.
# Дата написания: 12.02.2017.

print("Введите исходные данные:")
print("x0 = ", end='')
x0 = float(input())
print("hx = ", end='')
hx = float(input())
print("xn = ", end='')
xn = float(input())
print("Вы ввели:")
print("x0 = %.2f \t hx = %.2f \t xn = %.2f" % (x0, hx,
xn))
x = x0
xmax = x0
max = 2 * math.sin(x) + 3 * math.cos(x)
while x < xn + hx / 2:
    x += hx
    y = 2 * math.sin(x) + 3 * math.cos(x)
    if y > max:
        ymax = y
        xmax = x

print("ymax = %.3f при xmax = %.3f" % (ymax, xmax))
```

Контрольные вопросы

1. Из каких элементов состоит цикл?
2. Что такое параметр цикла и какие функции он выполняет?
3. Как определить число повторений тела цикла при изменении параметра цикла z по закону $z = z_0(h_z)z_n$?
4. Какие виды циклов используются в программировании?
5. Может ли тело цикла с постусловием не выполниться ни разу?
6. Какова структура цикла с предусловием?
7. В чем смысл задачи табулирования функции?
8. Значения каких переменных выводятся при табулировании функции?
9. Каким образом можно исключить проверку на равенство двух вещественных значений при формировании условия продолжения цикла?

Задания

Для задания в соответствии со своим вариантом составить алгоритм и написать программу, имеющие структуру цикла с предусловием и осуществляющие обработку тела цикла при изменении аргумента от начального значения до конечного значения с постоянным шагом.

Варианты задания:

1. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \begin{cases} 2x + 4, & -2 \leq x \leq -1; \\ 2x^2, & -1 < x \leq 1; \\ -2x + 2, & 1 < x \leq 2. \end{cases}$$

При $x < -2$ и $x > 2$ функция $f(x)$ не определена.

2. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \begin{cases} 1, & x \leq -1; \\ 2x^2 - 1, & -1 < x \leq 1; \\ x^2, & 1 < x \leq 2. \end{cases}$$

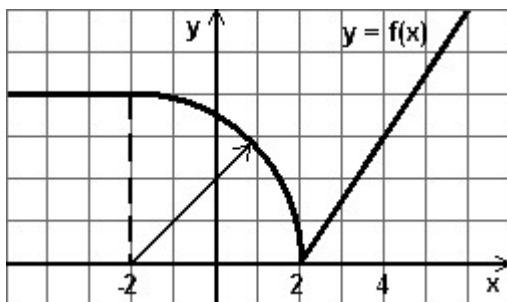
При $x > 2$ функция $f(x)$ не определена.

3. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \begin{cases} 2\sin(\frac{3x}{4}) & x \leq \frac{\pi}{2}; \\ \frac{x}{2} \operatorname{tg}(\frac{x+1}{3}) & x > \frac{\pi}{2}. \end{cases}$$

В тех случаях, когда тангенс не имеет значений, вывести сообщение «функция не существует».

4. Выполнить табулирование функции, заданной графиком, для $x = x_0(h_x)x_n$:



5. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

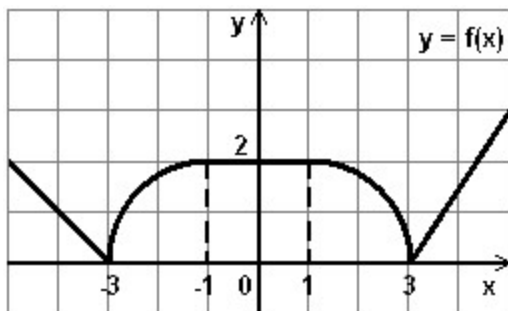
$$f(x) = \begin{cases} \frac{8}{x}, & x \leq -2; \\ x^3 + 4, & -2 < x \leq 0; \\ \frac{4}{x^2 + 1}, & x > 0. \end{cases}$$

6. Значения функций $f_1(x) = 2 + \sin(3x)$ и $f_2(x) = \cos(x/3)$ определены в точках $x = x_0(h_x)x_n$. Определить, при каком x расстояние между $f_1(x)$ и $f_2(x)$ минимально.

7. Значения функции $f(x) = 1/2 + \sin(x/2)$ определены в точках $x = x_0(h_x)x_n$. Определить, сколько значений функции $f(x)$ при заданных x являются отрицательными и сколько положительными.

8. Значения функции $f(x) = 3\sin(x/3) + 2\cos(x/2)$ определены в точках $x = x_0(h_x)x_n$. Определить максимальное и минимальное значения функции $f(x)$.

9. Выполнить табулирование функции, заданной графиком, для $x = x_0(h_x)x_n$:



10. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \begin{cases} 2 \sin\left(\frac{x}{2}\right), & x < \frac{\pi}{2}; \\ \frac{x}{2} \cos 2x, & \frac{\pi}{2} < x \leq \pi; \\ \sin x + \cos x, & x > \pi. \end{cases}$$

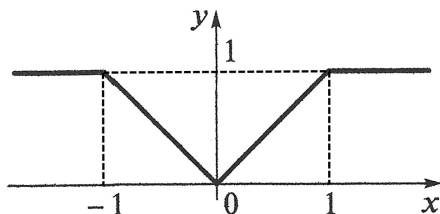
11. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \begin{cases} 2 \sin x, & x \leq -\frac{\pi}{2}; \\ A \sin x + B, & -\frac{\pi}{2} < x < \frac{\pi}{2}; \\ \cos x, & x \geq \frac{\pi}{2}, \end{cases}$$

где $x_0 = -2$; $h_x = 0,2$; $x_n = 2$; $A = -5$; $B = 12$.

12. Определите, сколько четных целых чисел лежит в пределах отрезка $[a, b]$, где $a = \sin(x^2)$; $b = x^4$; x – произвольное (вводимое) значение.

13. Вычислите значение функции $f(x)$ по графику для значения аргумента $x = x_0(h_x)x_n$, где $x_0 = -2$; $h_x = 0,5$; $x_n = 2$.



14. Вычислить значение функции $f(x)$ для $x = x_0(h_x)x_n$:

$$f(x) = \frac{\sin^2 x}{x^2 + 4} + \frac{\cos(x^2)}{(x - 2)(x - 5)},$$

где $x_0 = -3$; $h_x = 0,5$; $x_n = 6$. Точки разрыва исключить.

15. Значения функций $f_1(x) = 3x^3 + 2x^2 + x + 5$ и $f_2(x) = 2x^3 - 4x^2 + 2x - 5$ определены в точках $x = x_0(h_x)x_n$. Определить минимальное и максимальное расстояние между функциями $f_1(x)$ и $f_2(x)$ в заданных точках.

16. Значения функций $f_1(x) = ((x + 1) / 2) * \cos(x / 3)$ и $f_2(x) = ((x + 2) / 3) * \sin(x / 4)$ определены в точках $x = x_0(h_x)x_n$. Определить расстояние между максимальными значениями функций $f_1(x)$ и $f_2(x)$.

17. Значения функции $f(x) = 4x^3 - 3x^2 + 2x - 1$ определены в точках $x = x_0(h_x)x_n$. Определить расстояние между максимальным и минимальным значениями функции $f(x)$.

18. Значения функций $f_1(x) = 4x^2 + x - 3$ и $f_2(x) = 8x^3 + x^2 + 2x - 4$ определены в точках $x = x_0(h_x)x_n$. Определить расстояние между минимальными значениями функций $f_1(x)$ и $f_2(x)$.

19. Значения функции $f(x) = 3x^4 + 2x^3 - 3x^2 - 3$ определены в точках $x = x_0(h_x)x_n$. Определить, каких значений функции $f(x)$ больше, положительных или отрицательных.

20. Значения функции $f(x) = (\sin(3x) + \cos(2x)) / (x^2 + 1)$ определены в точках $x = x_0(h_x)x_n$. Определить число точек функции $f(x)$, значения которых превосходят величину $b = 2,345$.

21. Значения функции $f(x) = 2\sin(x/10 + 0,5)$ определены в точках $x = x_0(h_x)x_n$. Определить среднее арифметическое для отрицательных значений функции $f(x)$.

22. Значения функции $f(x) = \sin x + \cos x$ определены в точках $x = x_0(h_x)x_n$. Определить минимальное значение среди положительных значений функции $f(x)$.

23. Значения функции $f(x) = \cos x$ определены в точках $x = x_0(h_x)x_n$. Определить количество точек функции $f(x)$, расположенных выше прямой $y = 0,5$ и ниже прямой $y = -0,5$.

24. Значения функции $f(x) = 2\cos(5/4 * x)$ определены в точках $x = x_0(h_x)x_n$. Определить среднее арифметическое для положительных значений функции $f(x)$.

Литература

1. Python на примерах. Практический курс по программированию. / А.Н. Васильев. – СПб.: Наука и Техника, 2016. – 432 с.
2. <http://pythonicway.com>
3. <https://pythonworld.ru>
4. http://www.python-course.eu/python3_course.php

Содержание

Цель работы	3
Структура и состав цикла	3
Оператор цикла с предусловием	5
Табулирование функции одной переменной.....	7
Контрольные вопросы.....	12
Задания	12
Литература.....	16