Тема 6 ЦИКЛИЧЕСКИЙ АЛГОРИТМ WHILE

Лабораторная работа №4

4.1 Цель практической работы

Приобретение практических навыков разработки и программирования вычислительного процесса с использованием оператора while.

4.2 Теоретические сведения

Циклическими называются структуры, в которых предусмотрена возможность многократного повторения выполнения участка алгоритма. Этот участок алгоритма называется телом цикла. Различают циклические структуры двух видов: с заранее известным и с заранее неизвестным числом повторений цикла.

Существуют два вида элементарных циклических структур:

- циклы с параметром;
- циклы с условием.

Циклы с параметром используют тогда, когда количество повторов тела цикла заранее известно. В языке Python циклы с параметром реализуются с помощью оператора **For**.

Циклы с условием используются тогда, когда число повторений заранее неизвестно, но задано условие окончания цикла. В таких циклах число повторений цикла заранее неизвестно и определяется только в процессе выполнения алгоритма. Такие циклические структуры называются **итеративной циклической структурой**. В языке **Python** для реализации конструкции цикла с неизвестным количеством повторений служит оператор **while** или по другому **цикл с предпроверкой условия** (рисунок 4.1).

Синтаксис оператора while следующий:

Инициализация начального значения
while логическое выражение:
  P_1
  P_2

  P_n

где P_1, P_2, ... P_n - операторы; **while** (пока, до тех пор) - служебное слово языка **Python**.

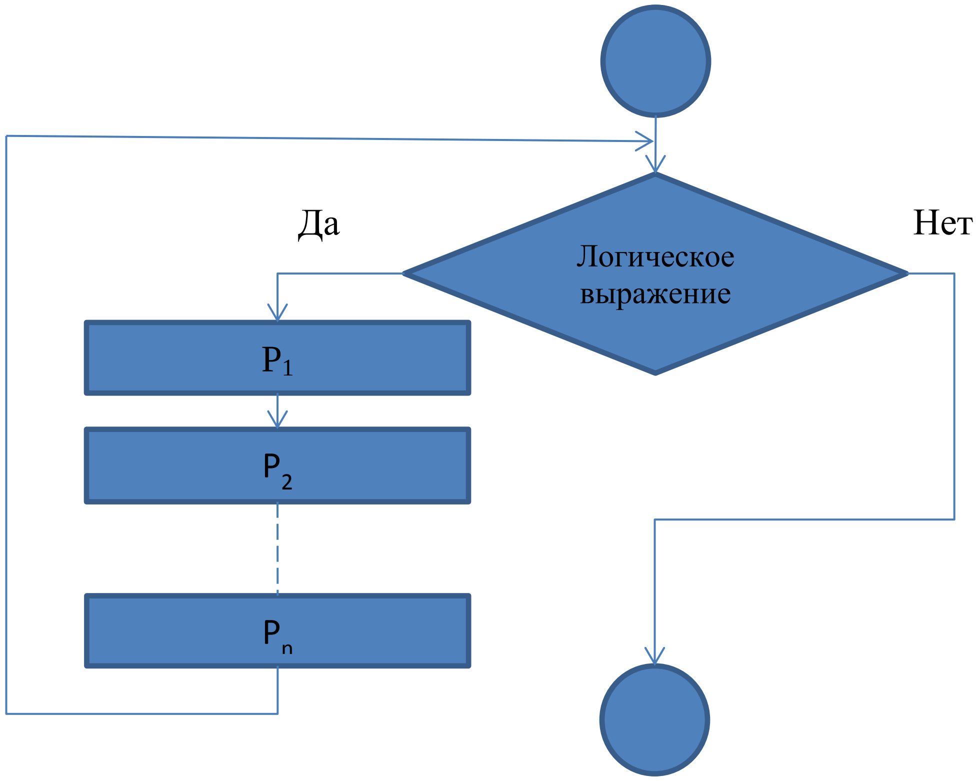

Рисунок 4.1 – Блок-схема алгоритма оператора **while**

Если логическое выражение после служебного слова **while** имеет значение **True** (Истина), то выполняются операторы P_1, P_2, ...P_n, после чего проверка логического выражения повторяется. Если логическое выражение имеет значение **False** (Ложь), то происходит выход из цикла. Если условие в заголовке цикла не является истинным с самого начала, цикл **while** не выполняется ни разу. В цикле **while** отсутствует параметр цикла, который по умолчанию есть в конструкции цикла с оператором **for**. Во многих задачах возникает необходимость в создании такой переменной, которая бы играла соответствующую роль. Причем предусмотреть инициализацию переменной, заменяющую параметр цикла, необходимо до выполнения цикла, а в самом цикле реализовать увеличение этой переменной на определенный шаг.

Например, в листинге приведенном ниже цикл с оператором **while** обрабатывает данные до тех пор, пока не будет введено слово «Python».

```
name=""
while name !=' Python ':
  name=input("Введите любое слово для печати или слово Python для выхода ")
  if name!=' Python ':
    print("Ответ = ", name)
```

4.3 Пример выполнения задания на практическую работу

Задача 1. Найдите сумму целых чисел от 1 до 50, используя оператор цикла

while.

Решение. Ранее в лабораторной работе 3 была разобрана эта же задача, но с оператором **for**. Необходимо для данной задачи ввести какую-нибудь переменную, которая меняется в цикле от 1 до 50. В этом примере такую роль будет играть переменная **k**. В качестве условия выхода из цикла задаем **k!= 50** и применяем в цикле оператор **sum=sum+k.** Этим мы просуммируем все пятьдесят слагаемых и получим в ответе число 1275. Блок-схема алгоритма решения задачи представлена на рисунке 4.2.

В листинге приведен код программы, отвечающий за решение задачи:

```
k=0
sum=0
while k!=50:
  k=k+1
  sum=sum+k
print("Сумма чисел от 1 до 50 sum = ", sum)
```

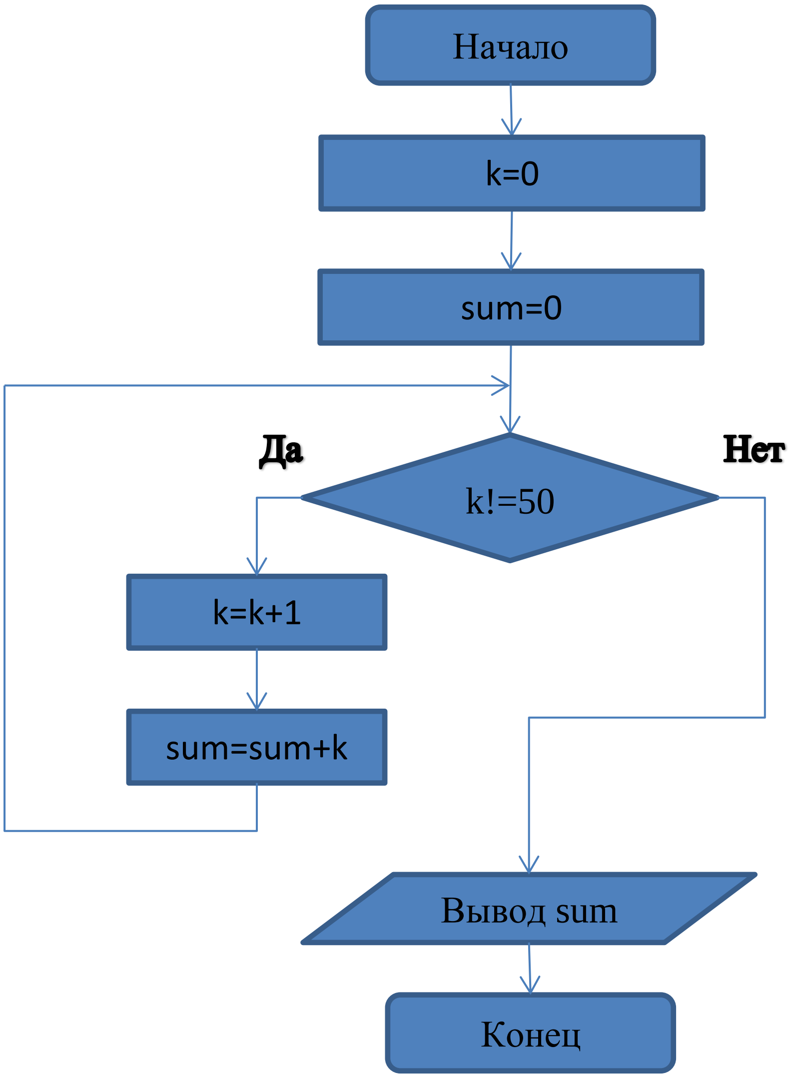

Рисунок 4.2 – Блок-схема алгоритма решения задачи 1

При вычислении значений функций или каких-либо числовых последовательностей (например, арифметической прогрессии), их часто записывают в виде специальных сумм, называемых **рядами**. Многие числа, функции, алгоритмы численных методов могут быть записаны с помощью рядов или итерационных алгоритмов, которые позволяют вычислять их приближенные значения с заданной точностью. При программировании таких задач используются итерационные методы для организации цикла, в котором производится вычисление некоторой **рекуррентной формулы**.

Рекуррентная формула - это такая формула, которая сводит вычисления **n**-го члена последовательности к вычислению нескольких предыдущих членов (или, часто, одного предыдущего члена этой последовательности – **n-1**). В общем случае такая формула имеет вид:

$$S_n = S_{n-1} + U_n,$$

где S_n сумма первых **n** слагаемых ряда, которая образуется из суммы, полученной на предыдущем шаге S_{n-1}; U_n- слагаемое, полученное на текущем шаге.

Рассмотрим пример задачи на применение рекуррентной формулы.

Задание 2. Вычислите сумму членов знакопеременной убывающей последовательности с заданной точностью ε.

$$\frac{(x+1)^0}{1^2} - \frac{(x+1)^1}{2^2} + \frac{(x+1)^2}{3^2} - \cdots + (-1)^n \frac{(x+1)^n}{(n+1)^2} + \cdots$$

Решение. Вычисление с заданной точностью ε означает, что суммирование членов ряда надо продолжать до тех пор, пока очередной вычисленный член ряда не станет меньше по абсолютной величине числа ε.

В нашем случае целесообразно использовать **рекуррентную формулу**, которая позволяет вычислить значение переменной на следующем шаге, используя ее значение на текущем шаге $a_{n+1} = a_n \cdot q$. Выражение для $\boldsymbol{q}$ можно получить, разделив a_{n+1} член на a_n член.

Приведем вывод рекуррентной формулы для заданного в задании ряда. Формула a_nчлена

$$a_n = (-1)^n \frac{(x+1)^n}{(n+1)^2},$$

тогда формула a_{n+1} члена

$$a_{n+1} = (-1)^{n+1} \frac{(x+1)^{n+1}}{(n+1+1)^2} = (-1)^{n+1} \frac{(x+1)^{n+1}}{(n+2)^2}.$$

Разделив a_{n+1} член на a_n получим выражение для **q**

$$q = \frac{a_{n+1}}{a_n} = \frac{(-1)^{n+1}\frac{(x+1)^{n+1}}{(n+2)^2}}{(-1)^n\frac{(x+1)^n}{(n+1)^2}} = \frac{(x+1)^{n+1}\cdot(-1)^{n+1}\cdot(n+1)^2}{(-1)^n\cdot(x+1)^n\cdot(n+2)^2} = -(x+1)\left(\frac{n+1}{n+2}\right)^2.$$

Таким образом, рекуррентная формула для данного ряда примет вид

$$a_{n+1} = -a_n(x+1)\left(\frac{n+1}{n+2}\right)^2.$$

Выбор начального значения номера члена ряда **n** для нашего случая будет **n=0**, так как при подстановке этого значения в формулу **n**-го члена ряда

$$a_n = (-1)^n \frac{(x+1)^n}{(n+1)^2}$$

мы получим значение первого члена, равного **1** или **a= 1**.

Результат работы программы представлен на рисунке 4.3. В листинге приведен код программы, отвечающий за решение задачи:

```
from math import *
x=float(input("Введите значение x = "))
el=float(input("Введите точность вычислений el = "))
n=0
a=1
s=0
print("\n ", n, " ", a)
while abs(a)>el:
  print("\n ", n, " ", a)
  s=s+a
  a=-a*(x+1)*(n+1)*(n+1)/((n+2)*(n+2))
  n=n+1
print("\n Значение итерации не учтенное в сумме ряда ", n, " ", a)
print("\n Сумма ряда, вычисленное с заданной точностью = ", s)
```

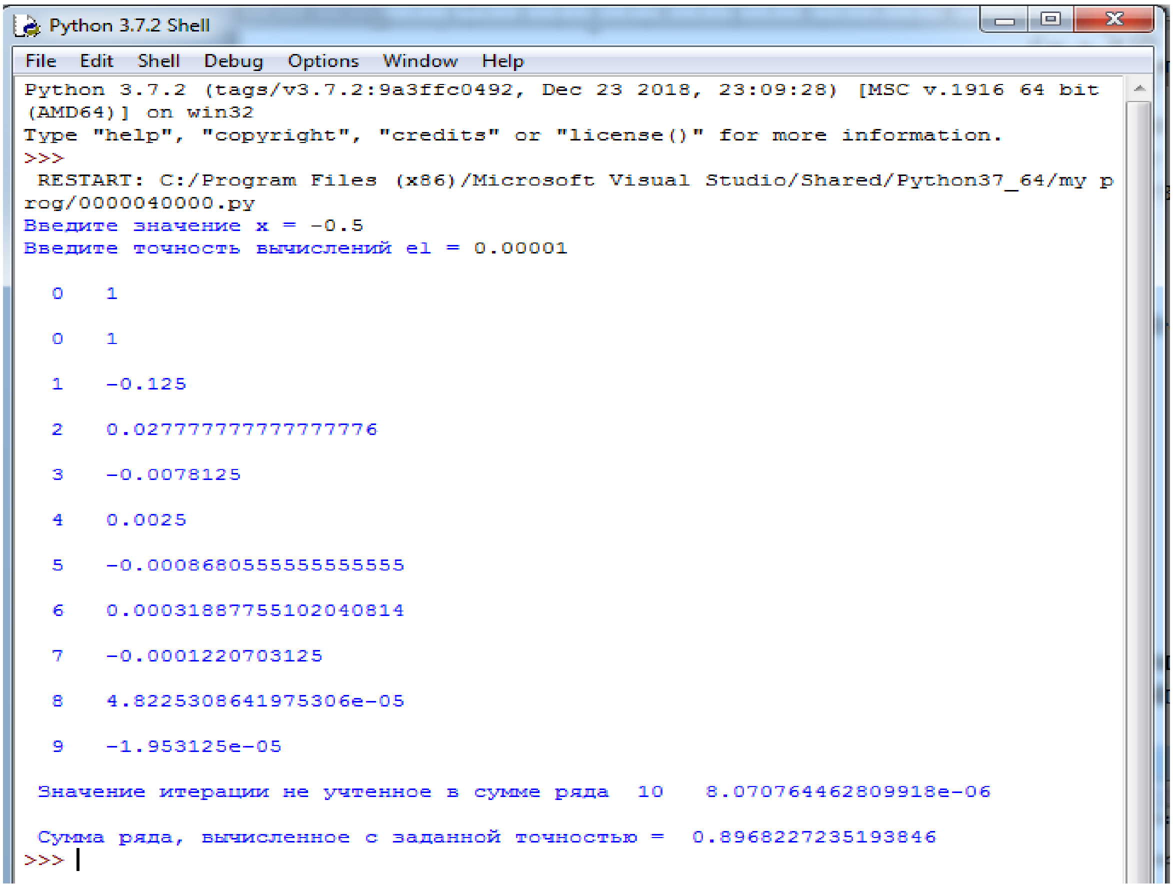

Рисунок 4.3 – Результат работы программы задачи 2

4.4 Задачи для самостоятельного решения

4.4.1. Вводится последовательность вещественных чисел, не равных нулю. Известно, что для завершения последовательности нужно ввести 0. В программе должны вычисляться суммы всех положительных и всех отрицательных чисел. Разработайте алгоритм и программу.

4.4.2. Вычислите сумму четных чисел на отрезке от 10 включительно до 30 включительно. Разработайте алгоритм и программу, используя цикл *while*.

4.5 Задание на практическую работу

Реализовать программно одно из следующих заданий**.**

Варианты заданий

№ варианта	Задание
1	Вычислите с точностью $\varepsilon = 0{,}00001$ константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд: $$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции для $x \in [0; 1]$.
2	Вычислите и выведите те члены последовательности $$x, \frac{x^2}{2!}, \frac{x^3}{3!}, \cdots \frac{x^n}{n!}, \cdots,$$ значения, которых больше $\varepsilon = 0{,}0000001$ при x=0,2.
3	Вычислите с точностью $\varepsilon = 0{,}001$ значение функции $y = \sqrt{x}$ при x =0.2, воспользовавшись рекуррентной формулой: $$y_{i+1} = 0.5\left[y_i + \frac{x}{y_i}\right],\ \text{где}\ i = 0, 1, 2, \cdots, y_0 = \frac{x}{2}.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции.
4	Вычислите и выведите те члены последовательности $$3x, 8x^2, \cdots, n(n+2)x^n, \cdots,$$ значения которых больше $\varepsilon = 0{,}05$ при x=0,6.
5	Вычислите с точностью $\varepsilon = 0{,}000001$ для $x = 1$, воспользовавшись разложением в ряд: $$arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots + (-1)^n \frac{x^{2n+1}}{2n+1} + \cdots$$ Сравните результат со значением, полученным с помощью соотношения $\pi = 4 \cdot arctg(1)$.
6	Вычислите с точностью $\varepsilon = 0{,}01$ значение функции $y = \frac{1}{\sqrt{x}}$ при

	x =0.1, воспользовавшись рекуррентной формулой: $$y_{i+1} = 1.5y_i - 0.5xy_i{}^3, \text{ где } i = 0, 1, 2, \cdots, y_0 = 1.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции.
7	Вычислите $sin0.5$ с точностью $\varepsilon = 0{,}00001$, воспользовавшись разложением в ряд: $$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции.
8	Для функции $z = \frac{x^k}{k(x-0.5)}$ найдите наименьшее целое положительное значение k, при котором $z < 0.001$ ($x = 0.6$).
9	Вычислите $cos0.6$ с точностью $\varepsilon = 0{,}00001$, воспользовавшись разложением в ряд: $$cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции.
10	Вычислите с точностью $\varepsilon = 0{,}0001$ корень уравнения $x - cosx =$ $= 0$, воспользовавшись формулой: $$x_{i+1} = cos(x_i), \text{ где } i = 0, 1, 2, \cdots, x_0 = 0.$$ Проверьте правильность решения подстановкой найденного корня в уравнение.
11	Вычислите и выведите те члены последовательности $$\frac{x^2}{2!}, -\frac{x^3}{3!}, \cdots (-1)^n \frac{x^n}{n!}, \cdots,$$ значения, которых больше $\varepsilon = 0{,}00001$ при x=0,9.
12	Для функции $z = \frac{x^k}{(x-k)^2}$ при $x = 2.5$ найдите наименьшее целое положительное значение k, при котором $z > 100$.
13	Вычислите корень уравнения $f(x) = x^4 + 2x^2 - x - 1 = 0$ с точностью $\varepsilon = 0{,}0001$, воспользовавшись формулой: $$x_{i+1} = x_i - \frac{f(x)}{f'(x)}, \text{где } i = 0, 1, 2, \cdots, x_0 = 0.$$ Проверьте правильность решения подстановкой найденного корня в уравнение.
14	Вычислите $\sqrt{2}$ с точностью $\varepsilon = 0{,}00001$, воспользовавшись разложением представлением в виде цепной дроби: $$\sqrt{2} - 1 = \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \cdots}}}.$$ Значение дроби равно пределу числовой последовательности,

	члены которой вычисляются по рекурентной формуле до достижения заданной точности $$a_n = \frac{1}{2 + a_{n-1}}, \qquad \text{где } n = 1, 2, 3, \cdots, a_0 = 0.5.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции.
15	Вычислите и выведите те члены последовательности $$\frac{x^3}{3}, -\frac{x^5}{15}, \cdots (-1)^{n+1} \ \frac{x^{2n+1}}{4n^2 - 1}, \cdots,$$ значения, которых больше $\varepsilon = 0.00001$ при x=0.9.
16	Вычислите $sh10$ с точностью $\varepsilon = 0{,}00005$, воспользовавшись разложением в ряд: $$sh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \cdots + \frac{x^{2n+1}}{(2n+1)!} + \cdots.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции для вычисления e^x, используя соотношение: $sh(x) = \frac{e^x - e^{-x}}{2}$.
17	Вычислите корень уравнения $x - 0.5(sinx^2 - 1) = 0$ с точностью $\varepsilon = 0{,}00001$, воспользовавшись итерационной формулой: $$x_{i+1} = 0.5(sinx_i{}^2 - 1), \text{где } i = 0, 1, 2, \cdots, x_0 = -0{,}25.$$ Проверьте правильность решения подстановкой найденного корня в уравнение.
18	Вычислите с точностью $\varepsilon = 0{,}00001$ корень уравнения $f(x) = tg(x) - x = 0$, воспользовавшись итерационной формулой: $$x_{i+1} = x_i - \frac{f(x)}{f'(x)}, \text{ где } i = 0, 1, 2, \cdots, x_0 = 4.6.$$ Проверьте правильность решения подстановкой найденного корня в уравнение.
19	Вычислите $ch0.7$ с точностью $\varepsilon = 0{,}00001$, воспользовавшись разложением в ряд: $$ch(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots + \frac{x^{2n}}{(2n)!} + \cdots.$$ Сравните результат со значением, полученным с помощью соответствующей встроенной функции для вычисления e^x, используя соотношение: $ch(x) = \frac{e^x + e^{-x}}{2}$.
20	Пусть $y_0 = 0$, $y_k = \frac{y_{k-1}+1}{y_{k-1}+2}$, $k = 1, 2, 3, \cdots$. Дано действительное число $\varepsilon = 0.00001 > 0$. Найдите первый член y_n, для которого выполнено условие $\| \ y_n - \ y_{n-1} \| < \varepsilon$.

4.6 Требования к оформлению отчета по практической работе

При оформлении отчета по практической работе рекомендуется следующая структура и последовательность элементов:

– титульный лист;

– название лабораторной работы;

– цель лабораторной работы;

– индивидуальное задание (по вариантам) на лабораторную работу;

– краткие комментарии по выполнению индивидуального задания и структурная схема алгоритма решения задачи;

– необходимый программный код индивидуального задания;

– результаты работы программы;

– выводы.

Индивидуальное задание на лабораторную работу содержит полный текст индивидуального задания, полученного у преподавателя, описание алгоритма выполнения индивидуального задания, структурную схему алгоритма решения задачи.

Необходимый программный код индивидуального задания содержит полный текст кода программы, разработанный студентом.

Результаты работы программы обычно содержат копии окон работы программы.

Перед выполнением индивидуального задания на лабораторную работу (согласно варианта), необходимо выполнить задачи, приведенные в пункте 4.4. В отчете привести скриншоты окна с программным кодом и окна с результатами работы программы (ввести свои исходные данные).

4.7 Контрольные вопросы для защиты лабораторной работы

4.7.1. В каких случаях применяются циклы с неизвестным числом повторений.

4.7.2. Какая циклическая структура может считаться итеративной?

4.7.3. Нарисуйте общий вид алгоритма оператора цикла while.

4.7.4. Напишите синтаксис оператора цикла while.

4.7.5. Опишите работу оператора цикла while. Приведите примеры.

4.7.6. В каких случаях применяются рекуррентные соотношения? Расскажите об алгоритме вывода рекуррентной формулы.