

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уфимский университет науки и технологий»**

Кафедра информатики

ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ В PYTHON

**Лабораторный практикум
по дисциплинам
«Языки программирования»,
«Информатика»**

Уфа 2024

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уфимский университет науки и технологий»

Кафедра информатики

ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ В PYTHON

Лабораторный практикум
по дисциплинам
«Языки программирования»,
«Информатика»

Уфа 2024

Составители: Н. А. Гарифуллина, Е. А. Кузьмина, Г.Ф. Низамова, С. Н. Султанова

Циклические конструкции в Python: Лабораторный практикум по дисциплинам «Языки программирования», «Информатика» / [Электронный ресурс] / Уфимск. ун-т науки и техн.; Сост.: Н. А. Гарифуллина, Е. А. Кузьмина, Г.Ф. Низамова, С. Н. Султанова – Уфа: УУНиТ, 2024.

Содержит сведения о языке программирования Python, а также описание создания приложений циклической структуры.

Предназначен для студентов первого и второго курсов технических и экономических специальностей и направлений УУНиТ, изучающих дисциплины «Языки программирования» и «Информатика» в рамках базового курса, а также всех тех, кто изучает язык программирования Python .

Рецензент: к.т.н., доцент каф. информатики Р. Р. Каримов

© Уфимский университет
науки и технологий, 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ЦИКЛИЧЕСКИЕ СТРУКТУРЫ.....	6
1.1.ЦИКЛЫ С ПАРАМЕТРОМ	6
1.2.ЦИКЛЫ С УСЛОВИЕМ.....	8
1.3.ОПЕРАТОР BREAK	9
1.4.ОПЕРАТОР CONTINUE	10
1.5.ОПЕРАТОР PASS	10
1.6.КОНСТРУКЦИЯ ELSE ЦИКЛА	11
1.7.ВЛОЖЕННЫЕ ЦИКЛЫ	12
2. ПРАКТИЧЕСКАЯ ЧАСТЬ	13
3. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ	20
4. ПРИЛОЖЕНИЯ	34
5. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	37
СПИСОК ЛИТЕРАТУРЫ	38

ВВЕДЕНИЕ

Цель и задачи лабораторной работы:

Целью работы является изучение основ программирования с использованием циклических конструкций.

Требования к выполнению лабораторных работ:

1. Перед началом лабораторной работы повторить лекционный материал по данной теме.
2. Получить у преподавателя вариант для выполнения индивидуальных заданий.
3. В порядке, описанном в практикуме, изучить разобранные задачи лабораторной работы.
4. Выполнить индивидуальные задания, приведенные в конце лабораторной работы.
5. Подготовить отчет по лабораторной работе. Ответить на контрольные вопросы, приведенные в конце лабораторной работы.
6. Продемонстрировать выполненную работу преподавателю, ответить на поставленные вопросы, уметь внести изменения в созданный проект.

Отчетность:

Отчетом (ПЗ) по лабораторной работе является документ, подготовленный с использованием текстового процессора MS Word, который должен быть оформлен в соответствие с требованиями ГОСТ-2.105-2019. ЕСКД. Общие требования к текстовым документам и ГОСТ 2.104-2023. ЕСКД. Основные надписи. Алгоритмы задачи должны быть представлены блок-схемами, выполненными согласно ГОСТ 19.701-90. ЕСКД. Схемы алгоритмов программ, данных и систем.

Структура ПЗ:

Введение

1. Задание 1

1.1. Постановка задачи

1.2. Алгоритм решения задачи

1.3. Программирование

1.4. Тестирование и отладка, результаты выполнения

2. Задание 2

2.1. Постановка задачи

2.2. Алгоритм решения задачи

2.3. Программирование

2.4. Тестирование и отладка, результаты выполнения

...

Заключение

Список использованных источников

Приложение

Контрольные вопросы

Защита лабораторных работ:

При защите лабораторной работы студент должен сформулировать постановку задач, уметь объяснить алгоритмы решения задач, используемые операторы языка программирования и конструкции, необходимые для программной реализации алгоритма, а также уметь внести изменения в созданный проект.

1. ЦИКЛИЧЕСКИЕ СТРУКТУРЫ

Циклические структуры обеспечивают многократное выполнение одной и той же последовательности действий для различных значений входящих в них переменных. Изменяющиеся в цикле переменные называются переменными цикла. Алгоритм циклической структуры должен содержать:

- 1) подготовку цикла – задание начальных значений переменных цикла перед первым его выполнением;
- 2) тело цикла – действия, которые выполняются при различных значениях переменных цикла;
- 3) изменение значений переменных цикла перед его новым повторением;
- 4) управление циклом – проверку условия окончания цикла или перехода на начало тела цикла.

Существует два типа элементарных циклических структур:

- цикл с параметром или с повторением,
- цикл с условием.

1.1. Циклы с параметром

Циклы с параметром используют тогда, когда количество повторов тела цикла заранее известно. Цикл повторяется, пока значение параметра цикла k лежит в интервале $[a, b]$, причем, если a и b являются выражениями, они вычисляются только один раз в начале цикла. Параметр цикла, начальное и конечное значения должны быть одинакового типа. Схематично такой цикл изображен на *Рис. 1*.

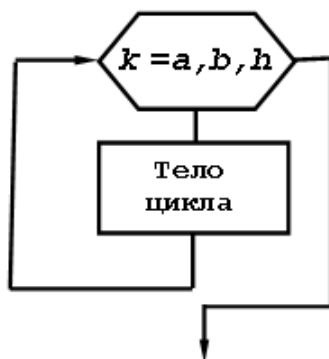


Рис. 1. Алгоритмическая структура «Цикл с параметром»:

- k – параметр цикла, a – начальное значение параметра цикла,
 b – конечное значение параметра цикла, h – шаг изменения параметра цикла

Оператор `for...in` является оператором цикла, который осуществляет итерацию по последовательности объектов, т.е. проходит через каждый элемент в последовательности.

Последовательность – это упорядоченный или неупорядоченный набор элементов.

В общем виде формат оператора цикла с параметром:

```
for <переменная> in итерируемый_объект:
    <тело цикла>
else:
    <дополнительный блок инструкций>
```

Блок `else` не является обязательным.

Пример 1. Вывести квадрат переменной `c`. Переменная цикла `c` по очереди принимает каждое значение последовательности, которая записана после служебного слова `in`.

```
for c in 1,5,4,8:
    print(c ** 2)
# вывод      1
              25
              16
              64
```

Пример 2. Переменная цикла `c` по очереди принимает каждое значение последовательности, которая записана после служебного слова `in`.

```
for c in 1, "Hello", False:
    print(c)
else:
    print('end')
# вывод      1
              Hello
              False
              end
```

Для задания диапазона (последовательности) целых чисел используется функция `range()`.

Формат оператора

```
for a in range(start, finish, step):
    <тело цикла>
```

`start` – начальное значение, по умолчанию – 0

`finish` – конечное значение, последнее значение `a=finish-1`

step – шаг изменения переменной-параметра цикла, по умолчанию, шаг =1.

Пример 3. Вывести числа из диапазона [0,10).

```
for i in range(10): #10 не включительно
    print(i, end=' ')    # end=' ' – печать в
                        # строку через пробел
# вывод      0 1 2 3 4 5 6 7 8 9
```

Пример 4. Вывести числа из диапазона [0,11) шагом 2.

```
for i in range(0, 11, 2): # 2 – шаг
    print(i, end=' ')
# вывод      0 2 4 6 8 10
```

1.2. Циклы с условием

Циклы с условием используются в тех случаях, когда число повторений тела цикла заранее неизвестно. Такой цикл продолжает работу, пока условие окончания цикла является true (истинным). Схематично такой цикл изображен на *Рис. 2*.



Рис. 2. Алгоритмическая структура «Цикл с предусловием»

В Python цикл с условием программируется с помощью конструкции while.

Формат оператора:

```
while <Выражение>:
    <тело цикла>
else:
    <дополнительный блок инструкций>
```

Оператор `while` многократно повторяет блок операторов до тех пор, пока Выражение в заголовочной части оценивается как истина.

Управление продолжает возвращаться к началу оператора, пока проверка не даст ложное значение. Когда результат проверки становится ложным, управление переходит на оператор, следующий после блока `while`.

Если проверка оценивается в ложное значение с самого начала, тогда тело цикла никогда не выполнится и оператор `while` пропускается.

Цикл `while` можно использовать:

- в математических итерационных алгоритмах для проведения вычислений с заданной точностью;
- при вводе данных, когда их количество заранее неизвестно, а условие завершения ввода определено некоторым введенным значением;
- при поиске нужного элемента в какой-либо структуре данных.

Пример 5. Вывод чисел в диапазоне [1, 10) с шагом 1.

```
a, b = 0, 10      # задаем значения a и b

while a < b:
    print(a, end=' ')
    a += 1
# вывод          0 1 2 3 4 5 6 7 8 9
```

1.3. Оператор `break`

Оператор `break` выполняет немедленный выход из цикла, т.е. остановку выполнения команд, даже если условие выполнения цикла еще не приняло значение `False` или последовательность элементов не закончилась.

Пример 6. Вводим и выводим имя и возраст до тех пор, пока не ввели слово 'стоп'.

```
while True:
    name = input('Введите имя: ')
    if name == 'стоп':
        break
    age = input('Введите возраст: ')
    print('Здравствуйте, ', name, '=>', int(age))
# вывод
Введите имя: Наталья
```

```
Введите возраст: 48
Здравствуйтесь, Наталья => 48
Введите имя: Тимур
Введите возраст: 17
Здравствуйтесь, Тимур => 17
Введите имя: стоп
```

1.4. Оператор `continue`

Оператор `continue` начинает следующий проход цикла, минуя оставшееся тело цикла (`for` или `while`), т.е. завершает текущую итерацию цикла и переходит к следующей.

Пример 7. Вывод четных чисел.

```
x = 10

while x:
    if x % 2 != 0:    # нечетное, то пропускаем
        x -= 1
        continue
    print(x, end=' ')
    x -= 1
# вывод    10 8 6 4 2
```

Без `continue`:

```
x = 10

while x:
    if x % 2 == 0: # четное, то выводим
        print(x, end=' ')
    x -= 1
# вывод    10 8 6 4 2
```

1.5. Оператор `pass`

Оператор `pass` – это заполнитель, обозначающий отсутствие действий, используемый в ситуациях, когда синтаксис требует оператора, но нет возможности выполнить что-либо полезное.

Данный оператор часто применяется для кодирования пустого тела для составного оператора.

К примеру, с помощью `pass` можно написать бесконечный цикл, который на каждом проходе ничего не делает:

```
while True:
    pass # для прекращения работы нажмите <Ctrl+C>
```

1.6. Конструкция `else` цикла

Полная форма записи циклов `while` и `for` выглядит следующим образом:

```
while condition(): # или for x in collection:
    operators

    # выход с пропуском else
    if exit_test():
        break

    # переход к заголовку цикла
    if skip_test():
        continue

# выполняется, если не было break
else:
    operators
```

Если циклы `while` или `for` прервать оператором `break`, соответствующие им блоки `else` выполняться не будут.

Пример 8. В приведенном примере выполняется проверка, является ли положительное целое число `y` простым, за счет поиска сомножителей больше 1.

```
y = int(input('Введите число y: '))
x = y // 2 # Для y > 1

while x > 1:
    if y % x == 0: # Остаток от деления
        print(y, "делится на", x) #имеет сомножитель
        break # Пропуск else
    x -= 1

else: # Нормальный выход
    print(y, "является простым числом") # простое

Введите число y: 7
7 является простым числом
Введите число y: 15
15 делится на 5
```

1.7. Вложенные циклы

Операторы цикла `for` могут быть вложены друг в друга на произвольную глубину (вложенный цикл – это цикл, внутри которого находится как минимум один цикл).

Синтаксис вложенного цикла параметром:

```
for element1 in sequence1:
    for element2 in sequence2:
        # тело внутреннего цикла
    # тело внешнего цикла
```

Каждая итерация внешнего цикла заставляет внутренний цикл выполнить все свои итерации.

Внешний цикл не выполняется до завершения внутреннего цикла.

Вложенными могут быть не только циклы `for`.

Можно размещать циклы `while` внутри циклов `while`, циклы `while` внутри циклов `for` и наоборот.

Для прерывания вложенных циклов можно использовать оператор `break`.

Если оператор `break` содержится в теле вложенного внутреннего цикла, то он останавливает только его.

Пример 9. Вывод значений переменной `j`.

```
for i in range(6):
    print('строка ', i)
    for j in range(6):
        print(j ** 2, end=' ')
    print()
# вывод
строка  0
0 1 4 9 16 25
строка  1
0 1 4 9 16 25
строка  2
0 1 4 9 16 25
строка  3
0 1 4 9 16 25
строка  4
0 1 4 9 16 25
строка  5
0 1 4 9 16 25
```

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Пример 10. Вычисление факториала числа.

Необходимо написать программу для вычисления $n!$:

$n! = 1*2*3*\dots*(n-1)*n$, где n – положительное целое число.

С использованием цикла `while`:

```
n = int(input('Введите положительное число n: '))
i = 1
f = 1
while i <= n:
    f *= i
    i += 1
print(' n! =', f)
# вывод
Введите положительное число n: 5
n! = 120
```

С использованием цикла `for`:

```
n = int(input('Введите положительное число n: '))
f = 1
for i in range(1, n + 1):
    f *= i
print(' n! =', f)
# вывод
Введите положительное число n: 25
n! = 15511210043330985984000000
```

Пример 11. Табулировать функцию $y = f(x)$. Отрезок табулирования $[a, b]$ и шаг h задается пользователем. Используйте форматный вывод. Выполните двумя способами: используя цикл `for` и `while`.

$$y = \sin\left(\frac{x-1}{5^{x+1}}\right) - e^x + \cos(2^{2x-1})$$

Используя `for`:

```
from math import sin, cos, pow, exp

a = float(input('Введите начало отрезка a -> '))
b = float(input('Введите конец отрезка b -> '))
h = float(input('Введите шаг табулирования h-> '))
n = int((b - a) / h) + 1
print('x\ty')
# \t - представляет 4 пустых символа, эквивалентно Tab
```

```

x = a

for i in range(1, n + 1):
    y = sin(pow(5, (x - 1) / (x + 1))) - exp(x) +
cos(pow(2, 2 * x - 1))
    print(x, "{:.5f}".format(y), sep="\t")
    x += h
# sep="\t" - символ табуляции между аргументами (x и y)

# ВЫВОД
Введите начало отрезка a -> 1
Введите конец отрезка b -> 3
Введите шаг табулирования h -> 0.5
x          y
1.0        -2.29296
1.5        -4.15353
2.0        -6.54423
2.5        -12.22806
3.0        -18.46456

```

Используя while:

```

from math import sin, cos, pow, exp

a = float(input('Введите начало отрезка a -> '))
b = float(input('Введите конец отрезка b -> '))
h = float(input('Введите шаг табулирования h-> '))
print('x\ty')
x = a
while x <= b:
    y = sin(pow(5, (x - 1) / (x + 1))) - exp(x) +
cos(pow(2, 2 * x - 1))
    print(x, "{:.5f}".format(y), sep="\t")
    x += h

# ВЫВОД
Введите начало отрезка a -> 1
Введите конец отрезка b -> 10
Введите шаг табулирования h -> 1
x          y
1.0        -2.29296
2.0        -6.54423
3.0        -18.46456
4.0        -54.79845
5.0        -149.19413
6.0        -402.49439

```

7.0	-1096.54109
8.0	-2980.93263
9.0	-8103.50566
10.0	-22026.03623

Пример 12. Найти сумму числового ряда $\sum \frac{k+0.5}{3k^2+5}$ с заданным количеством членов ряда n .

```
n = int(input('Введите n -> '))
summ = 0 # начальное значение равно 0
for k in range(1, n + 1):
    a = (k + 0.5) / (3 * k ** 2 + 5)
    print('k = ', k, ', a = ', a)
    summ += a
else:
    print('summa = ', summ)

# ВЫВОД
Введите n -> 10
k = 1 , a = 0.1875
k = 2 , a = 0.14705882352941177
k = 3 , a = 0.109375
k = 4 , a = 0.08490566037735849
k = 5 , a = 0.06875
k = 6 , a = 0.05752212389380531
k = 7 , a = 0.049342105263157895
k = 8 , a = 0.04314720812182741
k = 9 , a = 0.038306451612903226
k = 10 , a = 0.03442622950819672
summa = 0.8203336023066607
```

Пример 13. Найти сумму числового ряда $\sum \frac{k+0.5}{3k^2+5}$ с заданной точностью ε , т.е. $\sum_1^\infty \frac{k+0.5}{3k^2+5}$.

Будем использовать следующий алгоритм:

вычисляем сумму тех членов ряда, абсолютное значение которых больше или равно заданной точности, т.е. $|a| \geq \varepsilon$, где ε – точность решения задачи, представляет собой вещественное число меньше 1.

```
eps = float(input('Введите точность -> '))
summ = 0 # начальное значение равно 0
k = 1
a = (k + 0.5) / (3 * k ** 2 + 5) # первое слагаемое
```



```

while abs(a) >= eps:
    print('k = ', k, ', a = ', a)
    summ += a
    k += 1
    a = (k + 0.5) / (3 * k ** 2 + 5)
else:
    print('summa = ', summ)
    print('количество слагаемых', k - 1)

# вывод
Введите точность -> 0.01
k = 1 , a = 0.1875
k = 2 , a = 0.14705882352941177
...
k = 32 , a = 0.010562235944101397
k = 33 , a = 0.01023838630806846
summa = 1.2155055899145732
количество слагаемых 33

```

Пример 14. Вычислить частичную сумму бесконечного ряда с использованием рекуррентной формулы $\sum_{k=1}^{\infty} (-1)^k \frac{1}{k!(k+1)!}$

Алгоритм:

Найдем коэффициент c , разделив $(k+1)$ -ое слагаемое на k -ое:

$$c = \frac{(-1)^{k+1}}{(k+1)!(k+2)!} \cdot \frac{k!(k+1)!}{(-1)^k} = -\frac{k!}{(k+2)!} = -\frac{1}{(k+1)(k+2)}$$

Вычислим первое слагаемое ($k=1$): $a_k = -\frac{1}{2} = -0.5$

Остальные слагаемые можно найти по формуле:

$$a_{k+1} = c \cdot a_k, \quad \text{где } c = -\frac{1}{(k+1)(k+2)}$$

```

eps = float(input('Введите точность eps -> '))
summa = 0; k = 1
a = -0.5 # число a вычислить при k = 1

while abs(a) > eps:
    summa += a
    k += 1
    c = -1 / ((k + 1) * (k + 2))
    a *= c
else:
    print('Сумма =', '{:.6f}'.format(summa))

```

```

print('Количество слагаемых', k - 1)

# вывод
Введите точность eps -> 0.001
k = 2 a = 0.041667
k = 3 a = -0.002083
k = 4 a = 0.000069
Сумма = -0.460417
Количество слагаемых 3

Введите точность eps -> 0.000001
k = 2 a = 0.041667
k = 3 a = -0.002083
k = 4 a = 0.000069
k = 5 a = -0.000002
k = 6 a = 0.000000
Сумма = -0.460349
Количество слагаемых 5

```

Пример 15. Определить сумму цифр заданного натурального числа N .

Алгоритм:

- с помощью операции нахождения остатка найдем последнюю цифру числа;
- найденную цифру прибавляем к сумме;
- отбрасываем эту цифру с помощью целочисленного деления на 10.

Указанные шаги повторяются, пока число N не станет равно 0.

Число	Последняя цифра	Сумма
1234	4	4
123	3	7
12	2	9
1	1	10
0		

```

N = int(input('Введите натуральное число N -> '))
summa = 0

while N > 0:
    d = N % 10
    summa += d
    N //= 10
print('Сумма цифр числа равна', summa)

```

```
# вывод
Введите натуральное число N -> 123456
Сумма цифр числа равна 21
```

Пример 16. Найти количество четных цифр в восьмеричной записи десятичного натурального числа N .

Алгоритм:

Чтобы перевести целую часть числа из десятичной системы в систему с основанием p , необходимо разделить ее на основание p . Остаток даст младший разряд числа.

Полученное частное необходимо снова разделить на p – остаток даст следующий разряд числа и т.д. Процесс деления продолжается, пока в целой части не получим 0.

$200 : 8 = 25$	(0),	↑
$25 : 8 = 3$	(1),	
$3 : 8 = 0$	(3)	

```
N = int(input('Введите число N -> '))
counter = 0

# нуль считать четной цифрой
while N > 0:
    if (N % 8) % 2 == 0:
        counter += 1
    N //= 8
print('Количество четных цифр равно', counter)

# вывод
Введите число N -> 123 # 123(10) = 173(8)
Количество четных цифр равно 0

Введите число N -> 256 # 256(10) = 400(8)
Количество четных цифр равно 3
```

Пример 17. Определить наибольший общий делитель (НОД) двух заданных натуральных чисел на основе алгоритма Евклида.

Алгоритм нахождения НОД при помощи вычитания:

- из большего числа вычитается меньшее;
- если результат вычитания:
 равен 0, то числа равны друг другу и являются НОД;

не равен 0, в таком случае большее число заменяется на результат вычитания.

- переход к пункту 1.

```
n = int(input('Первое число -> '))
m = int(input('Второе число -> '))

while n != m:
    if n > m:
        n -= m
    else:
        m -= n
print('НОД = ', n)

# ВЫВОД
Первое число -> 28
Второе число -> 21
НОД = 7
```

Пример 18. Вывести все простые числа из диапазона $[1, N]$. N задается пользователем.

```
N = int(input('Введите число N -> '))

for i in range(2, N + 1):
    k = 2
    for d in range(2, i):
        if i % d == 0:
            k += 1
            if k > 2:
                break
    if k == 2:
        print(i, end=' ')

# ВЫВОД
Введите число N -> 25
2 3 5 7 11 13 17 19 23
```

3. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальные задания выполняются по вариантам.

Задание 1. Табулировать функцию $y = f(x)$. Отрезок табулирования $[a, b]$ и шаг h задается пользователем. Используйте форматный вывод. Выполните двумя способами: используя оператор цикла с параметром `for` и оператор цикла с предусловием `while`. Варианты заданий приведены в Таблица 1.

Таблица 1

№ вар.	Функция
1.	$y = x^2 - 5x + 6 - 0.5$
2.	$y = 2 \cos^2 x + 5 \sin x + 1$
3.	$y = \cos x - e^{-\frac{x^2}{2}} + x - 1$
4.	$y = (x^2 - 4x + 4)^{x-1.5} - 1$
5.	$y = \sin^4 x - \cos^4 x - 0.5$
6.	$y = \sqrt{3} \sin 3x + \cos 3x - 1$
7.	$y = e^x \sin x$
8.	$y = 0,1x^2 - x \ln x$
9.	$y = x - 1/(3 + \sin 3,6x)$
10.	$y = x - 0.5$

<i>№ вар.</i>	<i>Функция</i>
11.	$y = \lg(x) + \sqrt[3]{x} - 1.56$
12.	$y = x + \cos(x^{0,52} + 2)$
13.	$y = \frac{\sin x}{x} + 0.03$
14.	$y = (\sqrt{3x})^{x-1} - 1$
15.	$y = 3 \cdot 2^{2x} + 3^{2x+3} - 10$
16.	$y = 27^{\frac{2x-1}{x}} - \sqrt{9^{2x-1}}$
17.	$y = \lg^2 x - \lg x^3 + 2$
18.	$y = \sqrt{10 - 18 \cos x} - 6 \cos x - 5$
19.	$y = 3 \sin \sqrt{x} + 0,35x - 3,8$
20.	$y = x + \sqrt{x} + \sqrt[3]{x} - 2,5$
21.	$y = \sin(\ln x) - \cos(\ln x) + 2 \ln x$
22.	$y = \cos \frac{2}{x} - 2 \sin \frac{1}{x} + \frac{1}{x}$
23.	$y = 3x - 4 \ln x - 5$
24.	$y = \sqrt{1-x} - \cos \sqrt{1-x}$
25.	$y = 3 \ln^2 x + 6 \ln x - 5$

<i>№ вар.</i>	<i>Функция</i>
26.	$y = 3x - 14 + e^x - e^{-x}$
27.	$y = \sqrt{1-x} - \operatorname{tg} x$
28.	$y = \operatorname{tg} x - \frac{1}{3} \operatorname{tg}^3 x + \frac{1}{5} \operatorname{tg}^5 x - \frac{1}{3}$
29.	$y = \sin x - \cos x^2 + 0,25$
30.	$y = e^x \cos x$
31.	$y = x-1 - 0.5$

Задание 2. Вычислить сумму числового ряда с заданным количеством членов ряда. Значение N задается пользователем.

Варианты заданий приведены в *Таблица 2*.

Задание 3. Вычислить сумму числового ряда с заданной точностью. Точность задается пользователем.

Варианты заданий приведены в *Таблица 2*.

Таблица 2

<i>№ вар.</i>	<i>Член послед.</i>	<i>№ вар.</i>	<i>Член послед.</i>	<i>№ вар.</i>	<i>Член послед.</i>
1.	$\frac{1}{(2k-1)(2k+1)}$	11.	$\frac{k+4}{(k^2+2)(k+8)}$	21.	$\frac{k+2}{k^2+4}$
2.	$\frac{k}{(k+1)^2+3}$	12.	$\frac{3(k+1)}{7k^2+9}$	22.	$\frac{2k}{3k+k^2+4}$
3.	$\frac{2k}{(k^2+1)(k+2)}$	13.	$\frac{1}{\sqrt{k^3}+15}$	23.	$\frac{k}{k^2+k+1}$
4.	$\frac{k+1}{k^3-\sqrt{k+2}}$	14.	$\frac{1}{k^2+3k+4}$	24.	$\frac{4k}{5k^2+8k-1}$
5.	$\frac{k}{(3k^2+7)(k^2+1)}$	15.	$\frac{k+1}{k(k+2)(k+3)}$	25.	$\frac{2k+1}{(2k^2+3)k}$

<i>№ вар.</i>	<i>Член послед.</i>	<i>№ вар.</i>	<i>Член послед.</i>	<i>№ вар.</i>	<i>Член послед.</i>
6.	$\frac{k}{k^3 - k + 4}$	16.	$\frac{k}{(5 + k^3 - k)}$	26.	$\frac{k + 1}{k^3 - 2}$
7.	$\frac{k}{k^2(k + 1)}$	17.	$\frac{k}{(k + 1)^2}$	27.	$\frac{k + 3}{(2k)^2 - 1}$
8.	$\frac{k + 1}{k^3 + 2k + 1}$	18.	$\frac{k + 2}{(2k)^2}$	28.	$\frac{1}{k(k + 1)}$
9.	$\frac{1}{(k + 3)^2 - 1}$	19.	$\frac{k}{(2k - 1)^2}$	29.	$\frac{1}{k^2 - 2k + 6}$
10.	$\frac{1}{(k + 2)(k + 3)}$	20.	$\frac{k}{k^2 - k + 5}$	30.	$\frac{\sqrt{k + 1}}{2\sqrt{k^4 + 1}}$

Задание 4. Решить арифметическую задачу:

<i>№</i>	<i>Задача</i>
1.	Найти все четырехзначные числа, у которых сумма крайних цифр равна сумме средних цифр, а само число делится на 6 и 27.
2.	Найти НОД двух целых чисел X и Y используя алгоритм Евклида с делением.
3.	Найти количество трехзначных чисел, сумма цифр которых равна A , а само число заканчивается цифрой B . A и B задаются.
4.	Найти все трехзначные числа, которые при делении на 15 дают в остатке 11, а при делении на 11 - в остатке 9.
5.	Найти все натуральные числа a , b и c из интервала от 1 до 20, для которых выполняется равенство $a^2 + b^2 = c^2$.
6.	Найти наибольшую и наименьшую цифры в записи данного натурального числа.
7.	Найти количество единиц в двоичной записи числа N .
8.	При каком натуральном числе N произведение предшествующего числа и числа, следующего за N , равно 2208?
9.	Найти количество четных цифр натурального числа N .
10.	Найти сумму квадратов всех натуральных чисел на отрезке $[a; b]$.
11.	Найти все делители натурального числа N .

№	Задача
12.	Дано натуральное число. Верно ли, что оно начинается и заканчивается одной и той же цифрой.
13.	Дано натуральное число. Найти сумму первой и последней цифры этого числа.
14.	Вывести 10 первых степеней двойки.
15.	Определить все двузначные числа, сумма квадратов цифр которых кратны числу 15.
16.	Найти сумму чисел от 100 до 200, кратных 17.
17.	Дано n – целое число. Вычислить произведение: $P = 3 \cdot 6 \cdot \dots \cdot (3 \cdot n)$.
18.	Дана последовательность из 10 действительных чисел. Найти количество элементов последовательности, превышающих по модулю число 7.
19.	Дано n вещественных чисел. Найти сумму тех из них, которые являются четными.
20.	Введите с клавиатуры шесть чисел и определите их среднее арифметическое.
21.	С клавиатуры вводятся действительные числа и последовательно суммируются. Когда их сумма превысит 150, найти сумму введенных чисел.
22.	С клавиатуры вводится десять двузначных нечетных чисел. Если число кратно 13, то найти произведение чисел, из которого оно состоит. Результат вывести на экран.
23.	Дано n вещественных чисел. Найти количество тех из них, которые имеют остаток от деления на некоторое число m , равный 0.
24.	Напишите программу, которая будет суммировать вводимые с клавиатуры числа до тех пор, пока они положительны.
25.	Дано число n . Напечатать те натуральные числа, квадрат которых не превышает n .
26.	Напишите программу, которая запрашивает у пользователя числа до тех пор, пока каждое следующее число больше предыдущего. В конце программа сообщает, сколько чисел было введено.
27.	Напишите программу, которая запрашивает у пользователя числа до тех пор, пока каждое следующее число целое. В конце программа сообщает, сколько чисел было введено.

№	Задача
28.	Напишите программу, вычисляющую сумму всех чётных чисел в диапазоне от 1 до 90 включительно.
29.	Найти сумму всех целых чисел от а до 100 включительно (значение а вводится с клавиатуры).
30.	Найти произведение всех целых чисел от 10 до 100 включительно.

Задания повышенного уровня сложности

Задание 5. Табулировать кусочно-заданную функцию. Отрезок табулирования и шаг задается пользователем. Варианты заданий приведены в Таблица 3.

Таблица 3

№ варианта	Функция	Исходные данные	Отрезок
1.	$y = \begin{cases} a \cos(a + \pi x), & x \leq -4 \\ a - be^{ax} + 3 , & -4 < x \leq 4 \\ (x - 4), & x > 4 \end{cases}$	$a = 3$ $b = 4$	[-5; 5]
2.	$y = \begin{cases} x \sin(x - a)^4, & x > 2a \\ (ax)^2 \sin(-ax), & a \leq x \leq 2a \\ x \cos ax, & x < a \end{cases}$	$a = 2,5$	[1; 7]
3.	$y = \begin{cases} ax^2 + bx + c, & x < 2 \\ a/x + \sqrt{x^2 + 1}, & 2 \leq x \leq 3 \\ (a + bx)/\sqrt{x^2 + 1}, & x > 3 \end{cases}$	$a = 2,8$ $b = -0,3$ $c = 4$	[1; 4]

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
4.	$y = \begin{cases} \sin(bx) + \cos(bx), & bx < 0,6 \\ 1, & 0,6 \leq bx \leq 0,9 \\ bx^3 + e^{b+x^5}, & bx > 0,9 \end{cases}$	$b = 1,5$	$[0,1; 1]$
5.	$y = \begin{cases} x^2 - \sin(x + \pi), & x < 0,5 \\ a\sqrt[3]{x} + a\sqrt{3x + a}, & 0,5 \leq x \leq 1,2 \\ \ln(x + ax), & x > 1,2 \end{cases}$	$a = 1,65$	$[0,7; 2]$
6.	$y = \begin{cases} x\sqrt[3]{x - a}, & x > 3 \\ x \cdot \cos ax \cdot \sin ax, & 2 \leq x \leq 3 \\ e^{-ax} \cos ax, & x < 2 \end{cases}$	$a = 2,5$	$[1; 5]$
7.	$y = \begin{cases} bx - \cos(b + x) \cdot \lg bx, & bx < 0,9 \\ (x + b)^3, & 0,9 \leq bx \leq 1,5 \\ bx + \lg bx, & bx > 1,5 \end{cases}$	$b = 1,5$	$[0,1; 1,5]$
8.	$y = \begin{cases} \sin x \cdot \lg ax, & x > 3 \\ \cos ax^2, & 2 < x \leq 3 \\ \sin x \cdot \cos^2 x, & x \leq 2 \end{cases}$	$a = 2$	$[2; 5]$

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
9.	$y = \begin{cases} \lg(x+1), & x \geq 1,5 \\ \cos^2 \sqrt{ ax }, & 1 < x < 1,5 \\ \sin^2 \sqrt{ ax }, & x \leq 1 \end{cases}$	$a = 20,3$	$[0,5; 2]$
10.	$y = \begin{cases} (\ln^3 x + x^2) / \sqrt{ x+t }, & x < 0,5 \\ \sqrt{x+t} + 1/x, & 0,5 \leq x \leq 1,5 \\ \cos x + t \sin^2 x, & x > 1,5 \end{cases}$	$t = 2,2$	$[0,2; 3]$
11.	$y = \begin{cases} \frac{a+b}{e^x + \cos x}, & x < 2,8 \\ (a+b)/(x+1), & 2,8 \leq x < 6 \\ \lg(x) + \sin x \cdot \cos(x), & x \geq 6 \end{cases}$	$a = 2,6$ $b = -0,39$	$[0; 7]$
12.	$y = \begin{cases} a \lg x + \sqrt[3]{ x }, & x > 2 \\ a \sin x + 3x^2, & 1 < x \leq 2 \\ 2a \cos x + 3x^2, & x \leq 1 \end{cases}$	$a = 0,9$	$[0,8; 3]$
13.	$y = \begin{cases} \frac{a}{i} + bi^2 + c, & i > 6 \\ bi\sqrt{i}, & 4 \leq i \leq 6 \\ \left \sin(ai) + \cos(bi^3) \right , & i < 4 \end{cases}$	$a = 2,1$ $b = 1,8$ $c = -20,5$	$[0; 12]$

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
14.	$z = \begin{cases} \frac{a+x}{b+x^2}, & x < 0, \\ \sqrt{1+\frac{a}{1+x}}, & x \in [0, 1), \\ b \sin(3x) , & x \geq 1. \end{cases}$	$a = 2,6$ $b = -0,39$	[-2; 1,8]
15.	$y = \begin{cases} \sqrt{at^2 + b \sin t + 1}, & t < -0,5 \\ (a+3b)t, & -0,5 \leq t \leq 0,5 \\ \sqrt[3]{at^2 + b \cos^2 t}, & t > 0,5 \end{cases}$	$a = 2,5$ $b = 0,4$	[-1; 1]
16.	$y = \begin{cases} (x - \cos(2x)) + 2, & x < 2 \\ 2a + \sin\left(\frac{\pi}{6}x\right), & 2 \leq x \leq 4 \\ -3x, & \text{в остальных случаях} \end{cases}$	$a = 1$	[1;5]
17.	$y = \begin{cases} \left(\cos\left(\frac{\pi}{13}x\right) + a\right), & x < -7 \\ 2\cos\left(\frac{\pi}{6}x + \frac{2\pi}{17}\right) + b^2, & -7 \leq x \leq -3 \\ 2x\sin\left(\frac{\pi}{2}x\right), & \text{в остальных случаях} \end{cases}$	$a = 1$ $b = 2$	[-8;2]
18.	$y = \begin{cases} 3\sqrt{ax^3} - \frac{5}{4}(x-7), & x < 3 \\ \cos\left(\frac{x}{4}\right), & 3 \leq x \leq 12 \\ x-3, & \text{в остальных случаях} \end{cases}$	$a = 3$	[0;15]

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
19.	$y = \begin{cases} \frac{x(x-5)}{\sin x} + 3, & x < 12 \\ t\sqrt{t^2+1} + \frac{5x^4-2x}{4} - 1, & x \geq 12 \end{cases}$	$t = 5$	[3;20]
20.	$y = \begin{cases} 15x + 5t, & x \leq 0 \\ x^{2t} + 3\ln(x), & 0 < x \leq 5 \\ 0, & \text{в остальных случаях} \end{cases}$	$t = 2,5$	[3;8]
21.	$y = \begin{cases} 13a, & x \leq 0 \\ x^2 - bx, & 0 < x \leq 1 \\ \sin(\pi x^2), & \text{в остальных случаях} \end{cases}$	$a = 3$ $b = 2,5$	[-1;3]
22.	$y = \begin{cases} 2x^3 - 7x^2 + 5x + 3^{ab}, & x < 3 \\ x + \ln \cos(2x) - 1 , & x \geq 3 \end{cases}$	$a = 1,5$ $b = 4$	[2;4]
23.	$y = \begin{cases} \frac{5}{x}, & x \leq -1 \\ 5a^2 + 3x^2, & -1 < x \leq 1 \\ 9e^{-0,5x}, & \text{в остальных случаях} \end{cases}$	$a = 0,5$	[-2;2]
24.	$y = \begin{cases} 4^{a-1,5}(x - \sin(x)), & x < 1 \\ \cos\left(\frac{\pi}{12}x\right), & -1 \leq x \leq 2 \\ -3, & \text{в остальных случаях} \end{cases}$	$a = 5,5$	[0;2,4]

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
25.	$y = \begin{cases} \sin^2(9x-3)+0,7x, & x \leq 0 \\ 5t^a \cos^3(25x+0,37)-5x^2, & x > 0 \end{cases}$	$\begin{aligned} a &= 5 \\ t &= 2 \end{aligned}$	$[-0,1;0,1]$
26.	$y = \begin{cases} 1-\sqrt{\cos(2x)}, & x > 2,5 \\ x^{2a} - x, & 1 \leq x \leq 2,5 \\ 1+ax^2, & x < 1 \end{cases}$	$a = 1,5$	$[0;3]$
27.	$y = \begin{cases} 1-\frac{2x^3}{1-x^2}, & x > 3,5 \\ \sqrt{\cos(2x-1)}, & 0 \leq x \leq 3,5 \\ a^t e^{-\cos(2x)}, & x < 0 \end{cases}$	$\begin{aligned} a &= -0,5 \\ t &= 2 \end{aligned}$	$[-0,5;4,5]$
28.	$y = \begin{cases} (\sin(2,3x-1))^a, & x > 2,5 \\ 1-3\ln 1-x , & 0 \leq x \leq 2,5 \\ \frac{ax^2}{2-x}, & x < 0 \end{cases}$	$a = 3$	$\left[-\frac{\pi}{5}; \frac{9\pi}{5}\right]$
29.	$y = \begin{cases} \frac{t}{\ln(x^3)}, & x > 4,5 \\ 2x+0,1a, & 0 \leq x \leq 4,5 \\ 3, & x < 0 \end{cases}$	$\begin{aligned} a &= 2,5 \\ t &= 1 \end{aligned}$	$[-0,5;5]$

<i>№ варианта</i>	<i>Функция</i>	<i>Исходные данные</i>	<i>Отрезок</i>
30.	$y = \begin{cases} a \sin(3x), & x \leq -\frac{\pi}{3} \\ 2tg(x), & -\frac{\pi}{3} < x \leq \frac{\pi}{6} \\ 7x^2 - \pi x, & \text{в остальных случаях} \end{cases}$	$a = 2$	$\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$

Задание 6. Вычислить сумму числового ряда с заданным количеством членов ряда. Значение N задается пользователем.

Варианты заданий приведены в *Таблица 4*.

Задание 7. Вычислить сумму числового ряда с заданной точностью. Точность задается пользователем.

Варианты заданий приведены в *Таблица 4*.

Таблица 4

<i>№</i>	<i>Ряд</i>	<i>№</i>	<i>Ряд</i>
1.	$\sum_{k=1} \frac{k^{k+5}}{2(2k)!};$ $n = 8, \quad \varepsilon = 0,0001$	2.	$\sum_{k=1} (-1)^{k+1} \frac{k^{k+1}}{(2k-1)!}$ $n = 5, \quad \varepsilon = 0,0001$
3.	$\sum_{k=0} \frac{k^{k+1}}{k!(k+1)!}$ $n = 8, \quad \varepsilon = 0,0001$	4.	$\sum_{k=1} \frac{(k)^k}{(2k-1)!}$ $n = 5, \quad \varepsilon = 0,00001$
5.	$\sum_{k=0} \frac{(k+5)!}{(2k)!}$ $n = 7, \quad \varepsilon = 0,0001$	6.	$\sum_{k=1} \frac{(k+1)^{k+1}}{(2k+1)!}$ $n = 5, \quad \varepsilon = 0,000001$
7.	$\sum_{k=1} (-1)^k \frac{10^k}{(k!)^2}$ $n = 5, \quad \varepsilon = 0,00001$	8.	$\sum_{k=0} \frac{(-1)^k (k+3)!}{(2k)!}$ $n = 5, \quad \varepsilon = 0,000001$

<i>№</i>	<i>Ряд</i>
9.	$\sum_{k=0} \frac{k^{k+1}}{(k+1)!(k+2)!}$ $n = 5, \quad \varepsilon = 0,0001$
11.	$\sum_{k=1} \frac{k!}{(2k+1)!}$ $n = 4, \quad \varepsilon = 0,00001$
13.	$\sum_{k=1} \frac{k^k}{(k+1)!(k-1)!}$ $n = 7, \quad \varepsilon = 0,0001$
15.	$\sum_{k=0} \frac{(k+1)^k}{(k)!^2}$ $n = 7, \quad \varepsilon = 0,0001$
17.	$\sum_{k=1} (-1)^k \frac{(k-1)^k}{(2k-1)!}$ $n = 4, \quad \varepsilon = 0,000001$
19.	$\sum_{k=1} \frac{k+2^{k+4}}{(2k-1)!}$ $n = 5, \quad \varepsilon = 0,000001$
21.	$\sum_{k=0} \frac{(-1)^k (2k+8)^k}{2(2k+1)!}$ $n = 6, \quad \varepsilon = 0,000001$
23.	$\sum_{k=1} (-1)^k \frac{2(k-1)!}{5(k+3)!}$ $n = 5, \quad \varepsilon = 0,00001$
25.	$\sum_{k=0} \frac{(-1)^k 3^{k+1}}{k!(k+1)!}$ $n = 5, \quad \varepsilon = 0,000001$

<i>№</i>	<i>Ряд</i>
10.	$\sum_{k=0} (-1)^k \frac{k!}{(k+5)!}$ $n = 4, \quad \varepsilon = 0,000001$
12.	$\sum_{k=0} \frac{k+2^{k+1}}{(2k+1)!}$ $n = 5, \quad \varepsilon = 0,000001$
14.	$\sum_{k=1} (-1)^k \frac{(k+6)!}{k^{k+1}(2k-1)!}$ $n = 5, \quad \varepsilon = 0,000001$
16.	$\sum_{k=1} (-1)^{k+1} \frac{2k+3!}{(2k+1)!}$ $n = 5, \quad \varepsilon = 0,00001$
18.	$\sum_{k=1} \frac{(-1)^k}{(k+1)!}$ $n = 7, \quad \varepsilon = 0,00001$
20.	$\sum_{k=1} \frac{(2k+1)^k}{(2k+1)!}$ $n = 5, \quad \varepsilon = 0,000001$
22.	$\sum_{k=1} \frac{(-1)^k (k+2)!}{(2k-1)!}$ $n = 6, \quad \varepsilon = 0,000001$
24.	$\sum_{k=1} (-1)^k \frac{2k^k}{k!(k+1)!}$ $n = 5, \quad \varepsilon = 0,000001$
26.	$\sum_{k=1} (-1)^k \frac{2^k}{(k+2)!}$ $n = 5, \quad \varepsilon = 0,000001$

\mathcal{N}°	$P_{\text{я} \partial}$
27.	$\sum_{k=1} (-1)^k \frac{3^k}{(k+2)!}$ $n = 7, \quad \varepsilon = 0,000001$
29.	$\sum_{k=2} \frac{k^{k-1}}{(k+1)!(k-1)!}$ $n = 5, \quad \varepsilon = 0,000001$

\mathcal{N}°	$P_{\text{я} \partial}$
28.	$\sum_{k=1} (-1)^k \frac{4^k}{(k+2)!}$ $n = 7, \quad \varepsilon = 0,0001$
30.	$\sum_{k=1} \frac{(-1)^{k+1}}{3k!}$ $n = 5, \quad \varepsilon = 0,000001$

4. ПРИЛОЖЕНИЯ

Операция	Обозначение	Пример использования
Унарный минус	-	-10
Сложение	+	2 + 2 # 4 False + 3 # 3
Вычитание	-	10 - 18 # -8 True - 5 # -4
Умножение	*	4*5 # 20
Деление (с сохранением остатка)	/	10 / 4 # 2.5
Возведение в степень	**	2**4 # 16
Целочисленное деление (деление с округлением в меньшую сторону)	//	10 // 2 # 5 13 // 5 # 2 -12 // 7 # -2 13 // -6 # -3 -10 // -3 # 3 10 // 2.6 # 3.0
Остаток от деления (деление по модулю)	%	10 % 2 # 0 13 % 5 # 3 -12 % 7 # 2 13 % -6 # -5 -10 % -3 # -1 10 % 2.6 # 2.2
Взятие модуля (абсолютное значение)	abs(x)	abs(-10) # 10 abs(1000) # 1000
Округление	round(x)	round(2.512) # 3 round(2.462) # 2 round(2.516, 2) # 2.52
Суммирование	sum(x1, x2)	sum(3, 1, 2, 4, 5) # 15
Минимальное значение	min(x1, x2)	min(3, 1, 2, 4, 5) # 1
Максимальное значение	max(x1, x2)	max(3, 1, 2, 4, 5) # 5
Пара (x//y, x%y)	divmod(x, y)	divmod(5, 2) # (2, 1)

Операция	Обозначение	Пример использования
Равно	==	4+2 == 3+3 # True False == True # False
Не равно	!=	3+1 != 2+2 # False
Больше	>	2 > 2 - 3 # True

Операция	Обозначение	Пример использования
Больше или равно	\geq	$2 \geq 1+1$ # True
Меньше	$<$	$3 < 10$ # True
Меньше или равно	\leq	$5 \leq 1$ # False $\text{True} \leq \text{False}$ # False

Операция	Обозначение	Эквивалент
Сложение	$a += b$	$a = a + b$
Вычитание	$a -= b$	$a = a - b$
Умножение	$a *= b$	$a = a * b$
Деление	$a /= b$	$a = a / b$
Целочисленное деление	$a //= b$	$a = a // b$
Остаток от деления	$a \% = b$	$a = a \% b$
Побитовое умножение	$a \&= b$	$a = a \& b$
Побитовое сложение	$a = b$	$a = a b$
Побитовое исключаяющее или	$a \wedge = b$	$a = a \wedge b$

Функция	Описание
<code>math.cos(a)</code>	Вычисление косинуса числа a. Угол в радианах
<code>math.sin(a)</code>	Вычисление синуса числа a. Угол в радианах
<code>math.tan(a)</code>	Вычисление тангенса числа a. Угол в радианах
<code>math.acos(a)</code>	Вычисление арккосинуса числа a. Результат в радианах
<code>math.asin(a)</code>	Вычисление арксинуса числа a. Результат в радианах
<code>math.atan(a)</code>	Вычисление арктангенса числа a. Результат в радианах
<code>math.exp(a)</code>	Вычисление экспоненты ($e^a = e^{**}a$)
<code>math.log(x[, base])</code>	С одним аргументом возвращает натуральный логарифм
<code>math.log10(a)</code>	Вычисление десятичного логарифма ($\lg(a)$)
<code>math.sqrt(a)</code>	Вычисление корня квадратного числа a ($a \geq 0$)
<code>math.pow(a, b)</code>	Возведение числа a в степень b (a^b)
<code>math.comb(n, k)</code>	Возвращает количество способов выбрать k элементов из n элементов без повторений и без учета порядка (число сочетаний из n элементов по k элементов)
<code>math.factorial(a)</code>	Возвращает целое – факториал x. Если аргумент действительное или отрицательное число

Функция	Описание
	возбуждается исключение ValueError
<code>math.floor(a)</code>	Возвращает ближайшее целое число большее, чем <code>a</code> , <code>floor(1.5) == 1</code> , <code>floor(-1.5) == -2</code>
<code>math.ceil(a)</code>	Округление числа вверх, <code>ceil(1.5) == 2</code> , <code>ceil(-1.5) == -1</code>

Запись некоторых математических выражений

Математическая запись	Python
$y = \sqrt{\pi x}$	<code>y = Math.Sqrt(Math.pi*x);</code>
$z = \sin\left(-\frac{\pi}{2}\right)$	<code>z = Math.Sin(-Math.pi/2);</code>
$a = \sin^2\left(-\frac{\pi}{2}\right)$ $a = \left(\sin\left(-\frac{\pi}{2}\right)\right)^2$	<code>a=Math.Pow(Math.Sin(-Math.pi/2), 2);</code>

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каково назначение операторов цикла?
2. Какие виды элементарных циклических структур языка Python вы знаете?
3. Когда используется цикл с параметром и каков его формат?
4. Когда используются циклы с условием?
5. Как работают вложенные циклы?
6. Для чего при вычислении суммы ряда нужно использовать рекуррентные формулы?
7. Какие символы используются в блок-схеме алгоритма для обозначения цикла с параметром?
8. Какие символы используются в блок-схеме алгоритма для обозначения цикла с условием?
9. Расскажите, в каких случаях применяются циклы с неизвестным числом повторений.
10. Какая циклическая структура может считаться итеративной?
11. Напишите синтаксис оператора цикла `while`.
12. Какой цикл называется внешним, а какой – внутренним?
13. Опишите Python-синтаксис и принцип работы цикла `for`.
14. В чем особенность использования циклической конструкции `for` в языке Python по сравнению с использованием в других языках программирования, к примеру, в Pascal или C/C++?
15. Для чего используется стандартная функция `range()`?
16. Обязательно ли использовать переменную цикла `for` внутри самого цикла?
17. Какова роль оператора `break` в теле цикла?
18. Какова роль оператора `continue` в теле цикла?
19. Какова роль оператора `pass` в теле цикла?

СПИСОК ЛИТЕРАТУРЫ

1. Карчевская, М. П. Основы алгоритмизации инженерных задач [Электронный ресурс]: [учебное пособие для студентов специальностей 27.05.01 "Специальные организационно-технические системы"; 10.05.05 "Безопасность информационных технологий в правоохранительной сфере"; 09.05.01 "Применение и эксплуатация автоматизированных систем специального назначения"] / М. П. Карчевская, О. Л. Рамбургер; Уфимский государственный авиационный технический университет (УГАТУ). — Электронные текстовые данные (1 файл: 6,06 МБ). — Уфа: УГАТУ, 2019.
2. Воробьев, А. В. Методы языка Python для представления знаний в информационных системах : лабораторный практикум по дисциплине "Представление знаний в информационных системах" / А. В. Воробьев, Н. А. Мирьянов ; Уфимский государственный авиационный технический университет (УГАТУ) .— Учебное электронное издание .— Уфа : УГАТУ, 2020 .— 1 электрон. опт. диск (CD-ROM) ; 12 см .— Текст: электронный. — Систем. требования: Pentium 300 МГц, Windows 98, MS Internet Explorer 6.0, CD-ROM 12x и выше, 32 Mb RAM, видеокарта и монитор, поддерживающий режим 800x600 16 бит, мышь, звуковая карта (дата обращения: 26.01.2024).
3. Основы разработки программного обеспечения на языке программирования высокого уровня Python: учебное пособие для студентов очной и заочной форм обучения, обучающихся по направлению подготовки бакалавров 09.03.03 Прикладная информатика / Л. А. Кромина, Л. Е. Родионова, А. Р. Фахруллина, Р. А. Ярцева; Уфимский государственный авиационный технический университет (УГАТУ). — Электронные текстовые данные (1 файл: 1,04 МБ). — Уфа: УГАТУ, 2020. — Электронная версия печатной публикации. — Заглавие с титул. экрана. — Доступ из сети Интернет по логину и паролю. Анонимный доступ из корпоративной сети УГАТУ. — Систем. требования: Adobe Reader. — <URL:http://e-library.ufa-rb.ru/dl/lib_net_r/Kromina_L_A_Osn_razr_progr_obesp_na_yaz_2020.pdf>. — Текст: электронный.