**Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение “Академия управления городской средой, градостроительства и печати”**

Документация к пакету калькулятора

shest\_pack

Студента гр. № 11ИСП-211

Шестакова Егора Павловича

**Санкт-Петербург**

**2022г.**

Оглавление

[**Документация** 3](#_Toc116008930)

[Модуль degrees\_n\_fact 4](#_Toc116008931)

[Модуль oct\_perevod 5](#_Toc116008932)

[Алгоритмы модулей 6](#_Toc116008933)

[Алгоритм модуля oct\_perevod 6](#_Toc116008934)

[Алгоритм модуля degrees\_n\_fact 7](#_Toc116008935)

[**Листинг модулей** 9](#_Toc116008936)

[Листинг модуля oct\_perevod 9](#_Toc116008937)

[Листинг модуля degrees\_n\_fact 10](#_Toc116008938)

**Документация**

Пакет состоит из двух модулей: *degrees\_n\_fact* и *perevod\_8\_ss*, включающих себя перевода чисел из 8 системы счисления в 2, 10 и 16-ные системы счисления и математические операции – возведение в степень, взять под корень любой степени и вычислить факториал.

Вычисления происходят с помощью встроенных функций в python3 и базовых операторов арифметики, а также модуля встроенной библиотеки math.

На ввод в функцию для перевода в систему счисления вводится num(int), а затем внутри функции запрашивается необходимая система счисления osn(int).

Перевод чисел в системы счисления осуществляется через перед сначала в десятичную, а затем, с помощью встроенных функций в python3, переводится в нужную систему счисления.

Подробно расписаны шаги действий каждого модуля расписаны в листинге пакета и закомментированы.

Модуль degrees\_n\_fact

* В модуле 4 функции: check\_num(num), stepen(num), koren(num), fact(num);
* Функции принимают 1 значение – num(float or int);
* Есть функция, проверяющая: являются ли значения числами – check\_num(num), которая принимает 1 значения – num(любой тип);
* В функции stepen(num) и koren(num) вводится дополнительное значение степени, которое также проверяется;
* В функции fact вызывается модуль стандартной библиотеки math – factorial, также в этой функции организована проверка, не пропускающая числа < 0;
* Все функции при ожидаемом результате вернут – float or int, при ожидаемых ошибках – строку.

**Ожидаемые ошибки:**

Error\_01: появляется когда значение не является int or float

Error\_01$1: появляется когда число не является int or float или меньше нуля.

Модуль oct\_perevod

* В модуле 2 функции: check\_oct(num) и preveod\_8\_ss(num);
* Функции принимают значения num – int (de facto – любое, но будет отклонено проверкой встроенной в check\_oct);
* Функция check\_oct проверяет – является ли число подходящей для восьмеричной системы (1234567), путем проверки каждой цифры в числе, проходя по нему с помощью цикла.
* В логику функции perevod\_8\_ss уже встроена функция check\_oct
* Функция perevod\_8\_ss при вызове запрашивает основание, в которое необходимо провести (это основание также проверяется – если ведено основание не 2 8 16, то будет выведена строка с надписью)

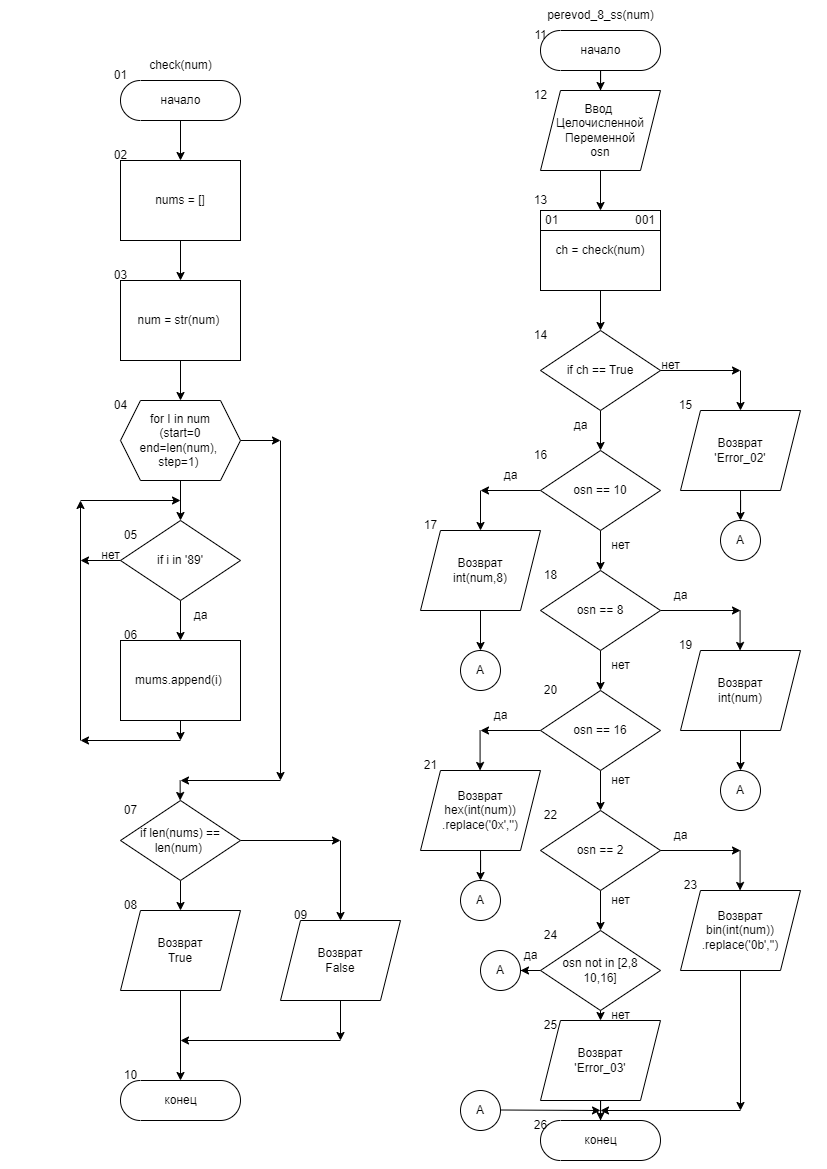
**Ожидаемые ошибки**

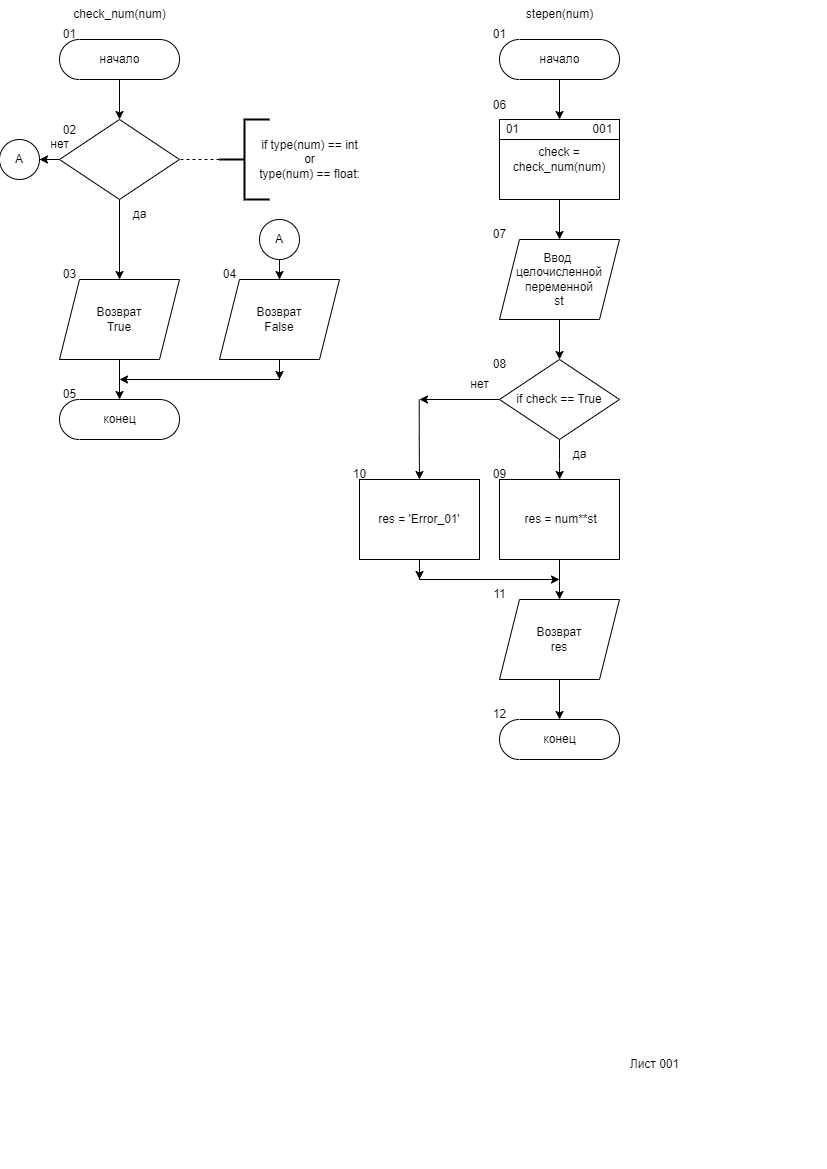
Error\_02: появляется когда число не в восьмеричной системе счисления или введенное число меньше нуля.

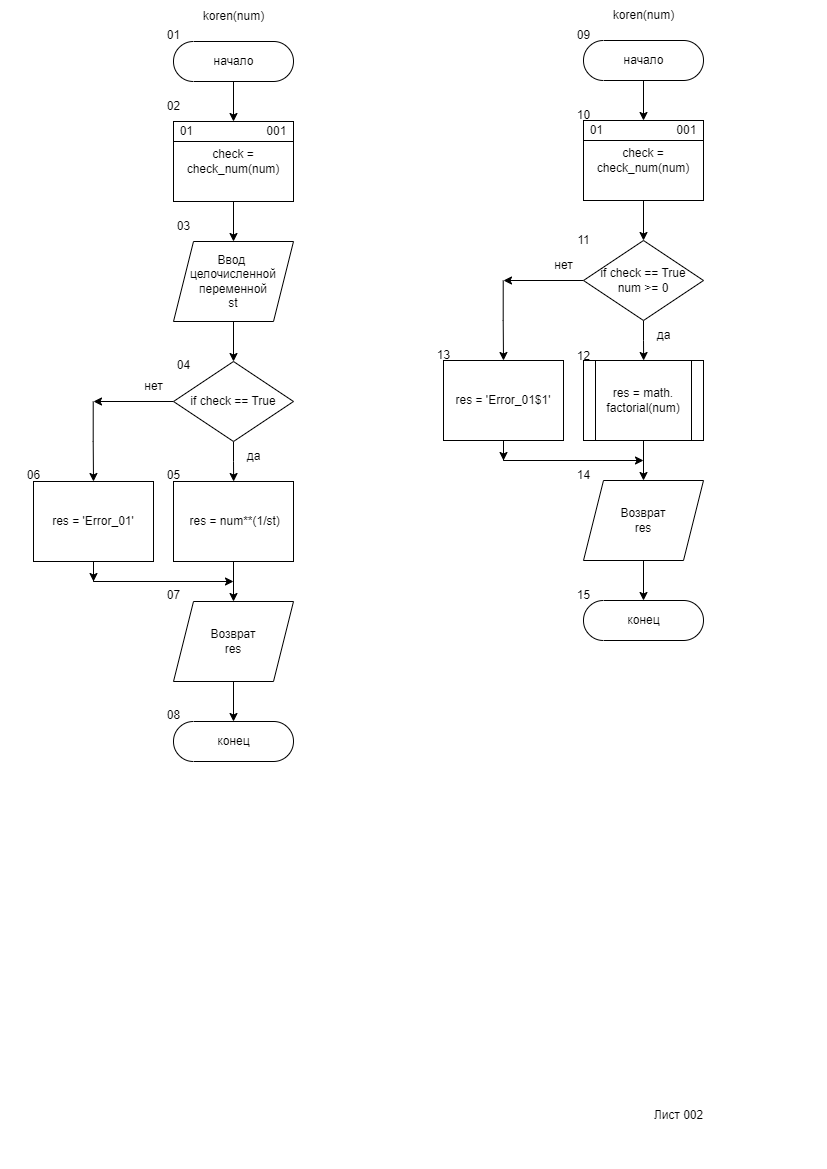
Error\_03: появляется когда желаемая степень не является 2,8,16,10.

Алгоритмы модулей

Алгоритм модуля oct\_perevod

Алгоритм модуля degrees\_n\_fact





**Листинг модулей**

Листинг модуля oct\_perevod

#Перевод из 8 сс в другие

def check\_oct(num): # функции для проверки число в восьмеричной системе или нет

numbers = [] # объявление пустого списка

num = str(num) # перевод числа в строку

for i in num: # проход по элементам строки

if i not in ['8','9']: # если элемент не 8 и не 9

numbers.append(i) # то добавляется в список numbers

if len(numbers) == len(num): # если длина списка = длине изначального числа

return True # возвращает True(bool)

else: # если нет

return False # возращает False(bool)

# Сам перевод из 8 системы счисления

def perevod\_8\_ss(num): # принимает число

convert\_system = int(input('Введите желаемую в выводе систему счисления (2,10,16): '))

if check\_oct(num): # если функция с принятым значением передает True(bool)

if int(num) > 0:

if convert\_system == 8: # если желаемая система счислений - 8

return num # возращает само число

if convert\_system == 10: # если желаемая система счислений - 10

return int(num,8) # возравщает число в 10 сс

if convert\_system == 2: # если желаемая система счислений - 2

ans = int(num,8) # перевод в десятичную сс

ans = bin(int(ans)) # перевод в 2-сс

return ans.replace('0b','') # уберается кодировка с помощью replace

if convert\_system == 16: # если желаемая система счислений - 16

ans = int(num,8) # перевод в десятичную сс

ans = hex(int(ans)) # перевод в 16-сс

return ans.replace('0x','').upper() # уберается кодировка с помощью replace

if convert\_system not in [2,8,16,10]: # если желаемая сс не 2 10 16

return 'Error\_03'

else:

return ('Error\_02') # возращает ошибку, если число не в 8 сс или меньше нуля

else:

return ('Error\_02') # возращает ошибку, если число не в 8 сс или меньше нуля

Листинг модуля degrees\_n\_fact

#Функция проверки является ли переменные числами

def check\_num(num): # принимает любое значение num

if type(num) == int or type(num) == float: # проверяет является ли num числом

return True # возвращает True(bool)

else:

return False # возращает False(bool)

#Степень

def stepen(num): # принимает любое значение num

if check\_num(num): # проверяет число, если функция возвращает True -

st = (input('Введите степень: ')) # вводится степень

if check\_num(st): # проверяется, является ли степень числом

return (num\*\*st) # возвращает число в степени st

else:

return ('Error 01') # если нет, выдает строку

else:

return ('Error 01') # если нет, выдает строку

#Корень

def koren(num): # принимает любое значение num

if check\_num(num): # проверяет число, если функция возвращает True -

st = (input('Введите степень: ')) # вводится степень

if check\_num(st): # проверяется, является ли степень числом

return (num\*\*(1/st)) # возвращает число в степени 1/st

else:

return ('Error 01') # если нет, выдает строку

else:

return ('Error 01') # если нет, выдает строку

#Факториал

def fact(num): # принимает любое значение num

if check\_num(num) and num >= 0: # проверяет является ли num числом и больше ли оно нуля

from math import factorial # импорт функции factorial из библотеки math

return(factorial(num)) # возращает значение функции factorial(num)

else:

return ('Error 01$1') # если нет, выдает строку