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

Документация к пакету калькулятора

shulya\_impact

Студентки гр. № 11ИСП-211

Гридневой Юлии Юрьевны

**Санкт-Петербург**

**2022г.**

Оглавление

[Документация 3](#_Toc116002870)

[Модуль arifmetika 3](#_Toc116002871)

[Модуль perevod\_2\_ss 3](#_Toc116002872)

[Алгоритмы 4](#_Toc116002873)

[Алгоритмы модуля perevod\_2\_ss 4](#_Toc116002874)

[Алгоритм модуля arifmetika 5](#_Toc116002875)

[Листинг модулей 7](#_Toc116002876)

[Модуль perevod\_2\_ss 7](#_Toc116002877)

[Модуль arifmetika 8](#_Toc116002878)

Документация

Пакет состоит из двух модулей (arifmetika и perevod\_2\_ss), включающих себя перевода чисел из 2 системы счисления в 8, 10 и 16-ные системы счисления и базовые математические операции – сложение, вычитание, умножение и деление.

Вычисления происходят с помощью встроенных функций в python3 и базовых операторов арифметики.

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.

Модуль arifmetika

* В модуле 4 функции: summa(a,b) - **сум**ма, umn(a,b) - **умн**ожение, vich(a,b) - **выч**итание, chast(a,b) - **част**ное;
* Функции вычислений принимают 2 значения – a(int or float) и b(int or float);
* Есть функция, проверяющая: являются ли значения числами – check(a,b), которая принимает 2 значения – a(любой тип) и b(любой тип);
* В функции chast (которая выполняет действие деления) имеется проверка, которая не позволяет разделить число на ноль и при попытке сделать это возвращает ошибку;
* Все функции при нормальной работе функции возвращают число – результат выполнения функции, у sum, umn,vich – float or int(в зависимости от введенных значений) у chast – всегда float. При неправильной работе выдает ошибку.

**Возможные ошибки**

**Error 001**: «Введены не числа»

**Error 002**: «Невозможно делить на ноль»

Модуль perevod\_2\_ss

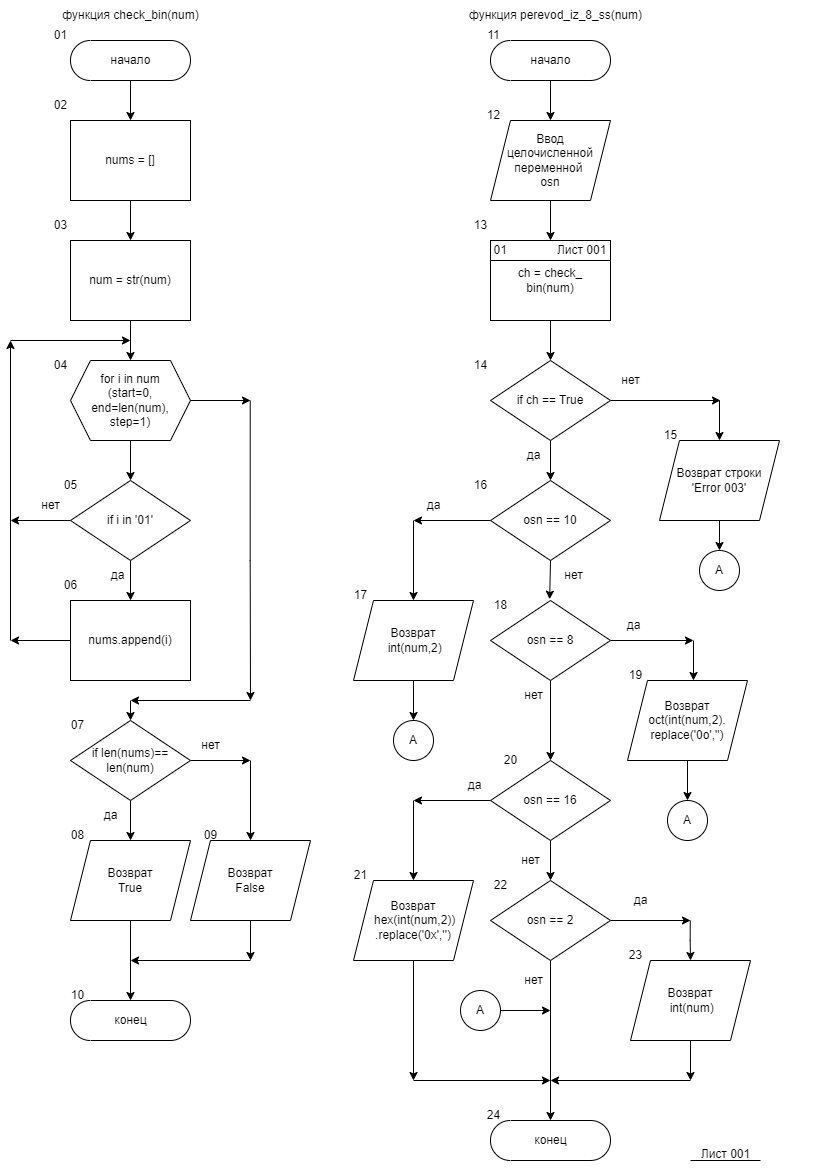
* В модуле 2 функции: check\_bin(num) и perevod\_2\_ss(num);
* Функции принимают значения num – int;
* Функция check\_bin проверяет – является ли число двоичным, путем проверки каждой цифры в числе, проходя по нему с помощью цикла.
* В функции perevod\_2\_ss встроен возврат ошибки, которая вызывается при не пройдённой проверки с помощью check\_bin – возвращая «Число не в двоичной системе счисления»;
* Функция perevod\_2\_ss при запуске запрашивает необходимое основание системы счисления, к которому пользователю необходимо осуществить перевод
* Все функции при нормальной работе функции возвращают: check\_bin – True/False (bool), perevod\_2\_ss – int. При неправильной работе выдает ошибку.

**Возможные ошибки**

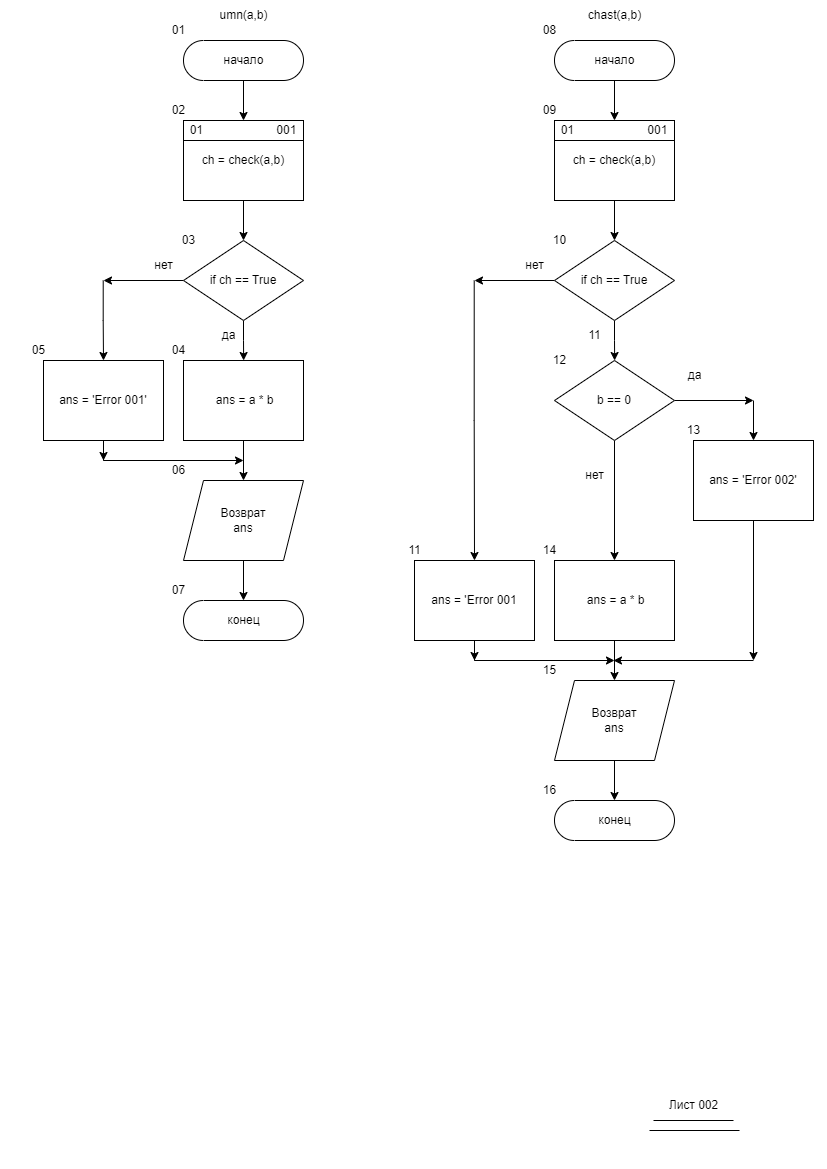
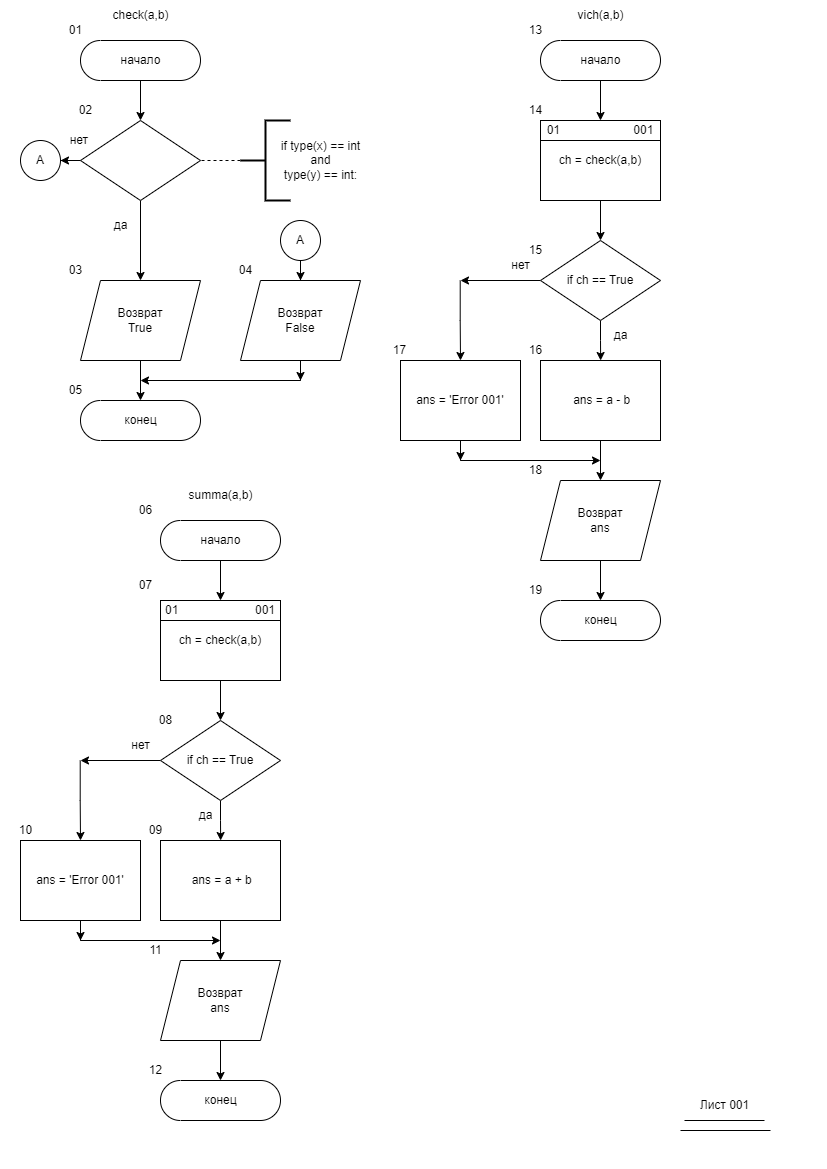
**Error 003**: «Число не в двоичной системе»

# Алгоритмы

## Алгоритмы модуля perevod\_2\_ss



## Алгоритм модуля arifmetika



# Листинг модулей

Модуль perevod\_2\_ss

#Функция проверки на двочиное число

def check\_bin(num): #Принимает num- любое значение

numbers = [] #Объявление списка

num = str(num) #Перевод из числа в строку

for i in num: #Проход по строке

if i in ['1','0']: #Является ли ЦИФРА единицей или нулем

numbers.append(i) #Если да, записывается в список

if len(numbers) == len(num): #Если длина нового списка такая же,как и длина изначального числа

return True #То число двоичное

else:

return False #Если нет, то нет.

#Функция перевода из двоичной сс во все остальные

def perevod(num,convert\_system): #Принимает num- любое значение, convert\_system

if check\_bin(num): #Проверка является ли число двоичным

if convert\_system == 10: #Если нужно перевести в 10

return int(num,2) #Переводит в 10-ую сс

if convert\_system == 8: #Если нужно перевести в 8

ans = int(num,2) #Перевод в 10-ую сс

ans = oct(int(ans)) #Перевод в 8-ую сс

return ans.replace('0o','') #Вывод результата без 0о, которое образуется от функции oct

if convert\_system == 16: #Если нужно в 16-ую сс

ans = int(num,2) #Перевод в 10-ую сс

ans = hex(int(ans)) #Перевод в 16-ую сс

return ans.replace('0x','').upper() #Вывод результата без 0х, котрое образуется от функции hex

if convert\_system == 2: #Если нужно перевести в 2 сс

ans = int(num) #Ответ равен самому числу

return num #Возвращает число

else:

raise Exception("Ваше число не в двоичной системе") #Если число не является двоичный, то выдает ошибку

## Модуль arifmetika

#Функция проверки является ли переменные числами

def check(x, y): #Принимает x, y- любое значение

if type(x) == int or type(x) == float and type(y) == int or type(y) == float: #Если x и y- являются числами

return True #Возвращает True

else:

return False #Возвращает False

#Функция сложения

def summa(a,b): #Принимает a, b- любое значение

if check(a,b): #Проверка a и b на то, являются ли они числами

return a + b #Сумма a и b

else:

return ('Введены не числа') #Если не прошло проверку- возвращает строку

#Функция вычитания

def vich(a,b): #Принимает a, b- любое значение

if check(a,b): #Проверка a и b на то, являются ли они числами

return a - b #Разность a и b

else:

return ('Введены не числа') #Если не прошло проверку- возвращает строку

#Функция умножения

def umn(a,b): #Принимает a, b- любое значение

if check(a,b): #Проверка a и b на то, являются ли они числами

return a \* b #Умножение a и b

else:

return ('Введены не числа') #Если не прошло проверку- возвращает строку

#Функция деления

def chast(a,b): #Принимает a, b- любое значение

if check(a,b): #Проверка a и b на то, являются ли они числами

if b == 0: #Проверяеи b (второе число) на ноль

return('Невозможно делить на ноль') #Если на b является нулем- возвращает строку

else: #Если b не ноль, то:

return a / b #Деление a и b

else:

return ('Введены не числа') #Если не прошло проверку- возвращает строку