**

Колледж космического машиностроения и технологии

ОТЧЕТ

По учебной практике УП.01. Учебная практика по модулю ПМ.01  
Установка интерпретатора Python 3 и настройка окружения

Выполнил студент: Зайцев Никита Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Проверил преподаватель: Гусятинер Леонид Борисович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2020

Оглавление

[Практическое занятие №2. 2](#_Toc57279879)

[Практическое занятие №4. 2](#_Toc57279886)

[Практическое занятие №5. 4](#_Toc57279894)

[Практическое занятие №6. 6](#_Toc57279900)

Практическое занятие №7. 8

Практическое занятие №8. 11

Практическое занятие №14. 12

# Практическое занятие №2.

Материал подготовил студент группы П2-18:

1| Зайцев Никита Евгеньевич

1.Заходим на сайт https://www.python.org/downloads/

2.Жмем на кнопку “Download Python 3.9.0”

3.Ожидаем скачивания программы

4.Запускаем, Ставим галочку “Add Python 3.9 to PATH” и начинаем установку.

5. Дожидаемся установки и начинаем работать с Python



(Приложение 1)

# Практическое занятие №3.

Материал подготовил студент группы П2-18:

1| Зайцев Никита Евгеньевич

## Создание файла с кодом.

Код исходной программы:  
print('Hello, world!') **(code.py)**

## Запуск.

Для того, чтобы открыть существующий файл с кодом, нужно:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. приложение 1)

3. Ввести команду “python название-файла”.

4. Получить результат выполнения программы (см. приложение 2).

## Import.

Чтобы подключить определенную библиотеку, нужно:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. приложение 1)

3. Ввести в консоль команду “python”.

4. Ввести в консоль команду “import название-библиотеки”. (см. приложение 3)

## Reload.

Чтобы перезагрузить определенное нечто, требуется:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. приложение 1)

3. Ввести в консоль команду “python”.  
4. Импортировать нужный файл. (см. приложение 4)  
5. Перезаписать исходный файл.   
Код измененной программы:

print('Hello!!!!!') **(code.py)**

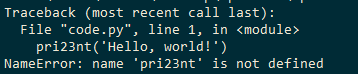
6. Написать команду “reload sample”.

## Отработка ошибок.

Примеры часто встречающихся ошибок:



Данная ошибка говорит о не прописанной команде “python”.



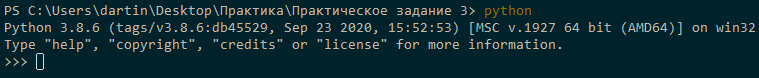
Данная ошибка говорит о том, что интерпретатор не может найти переменную с указанным именем.



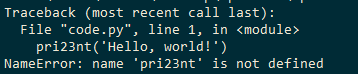
(Приложение 1)



(Приложение 2)



(Приложение 3)



(Приложение 4)

# Практическое занятие №4.

## Задачи на Begin.

**Задача 1: ( begin1.py )**

#Выполнил Зайцев Н. Е. П2-18.

# Дана сторона квадрата a. Найти его площадь S = a2

a = int(input())

a \*= a

print(a)

**Задача 2: ( begin2.py )**

#Выполнил Зайцев Н. Е. П2-18.

# Даны два числа a и b. Найти их среднее арифметическое: (a + b)/2.

a = int(input())

b = int(input())

a =(a+b)/2

print(a)

## Задачи на Boolean.

**Задача 1: ( boolean1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны два целых числа: A, B. Проверить истинность высказывания: «Ровно

#одно из чисел A и B четное»

a = int(input())

b = int(input())

if a % 2 != 0 or b % 2!= 0:

print ('True')

else:

print('False')

**Задача 2: ( boolean2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны три целых числа: A, B, C. Проверить истинность высказывания:

#«Каждое из чисел A, B, C положительное»

a = int(input())

b = int(input())

c = int(input())

if a > 0 and b > 0 and b > 0:

print("True")

else:

print("False")

## Задачи на Case.

**Задача 1: ( case1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 —

#февраль и т. д.). Вывести название соответствующего времени года

#(«зима», «весна», «лето», «осень»).

a = int(input("Введите номер месяца: "))

if 0 < a < 13:

if a == 12 or a == 1 or a == 2:

print("Зима")

elif a == 3 or a == 4 or a == 5:

print("Весна")

elif a == 6 or a == 7 or a == 8:

print("Лето")

elif a == 9 or a == 10 or a == 11:

print("Осень")

else:

print("Ошибка")

**Задача 2: ( case2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 —

#февраль и т. д.). Определить количество дней в этом месяце для

#невисокосного года.

a = int(input("Введите номер месяца: "))

if 0 < a < 13:

if a == 1:

print("январь, 31 день")

elif a == 2:

print("февраль, 28 дней")

elif a == 3:

print("март, 31 день")

elif a == 4:

print("апрель, 30 дней")

elif a == 5:

print("май, 31 день")

elif a == 6:

print("июнь, 30 дней")

elif a == 7:

print("июль, 31 день")

elif a == 8:

print("август, 31 день")

elif a == 9:

print("сентябрь, 30 дней")

elif a == 10:

print("октябрь, 31 день")

elif a == 11:

print("ноябрь, 30 дней")

elif a == 12:

print("декабрь, 31 день")

else:

print("Ошибка")

## Задачи на If.

**Задача 1: ( if1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны три целых числа. Найти количество положительных чисел в исходном

#наборе.

a = int(input())

b = int(input())

c = int(input())

k = 0

if a > 0:

k += 1

if b > 0:

k += 1

if c > 0:

k += 1

else:

print("Все числа отрицательные")

print(k)

**Задача 2: ( if2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны две переменные целого типа: A и B. Если их значения не равны, то

#присвоить каждой переменной сумму этих значений, а если равны, то

#присвоить переменным нулевые значения. Вывести новые значения

#переменных A и B.

A = int(input())

B = int(input())

if A == B:

A += B

B = A

elif A != B:

A = 0

B = 0

else:

print("Ошибка")

print("A =", A,"B =", B)

# Практическое занятие №5.

## Задачи на For.

**Задача 1: ( for1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны целые числа K и N (N > 0). Вывести N раз число K

N = int(input("N = "))

K = int(input("K = "))

for i in range(N):

print(K)

**Задача 2: ( for2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны два целых числа A и B (A < B). Вывести в порядке возрастания все #целые числа,расположенные между A и B (включая сами числа A и B), а

#также количество N этих чисел

a = int(input("a = "))

b = int(input("b = "))

n = 0

for i in range(a, b+1):

print(i)

n = b-a+1

print("\n N =", n)

**Задача 3: ( for3.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны два целых числа A и B (A < B). Вывести в порядке убывания все

#целые числа, расположенные между A и B (не включая числа A и B), а

#также количество N этих чисел

a = int(input("a = "))

b = int(input("b = "))

n = 0

for i in range(b-1, a, -1):

print (i)

n = b-a-1

print("\n N =", n)

**Задача 4: ( for4.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1,

#2, . . . , 10 кг конфет.

a = int(input())

for i in range(1,11):

print(i, 'кг -', i\*a, 'рублей')

## Задачи на While.

**Задача 1: ( while1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Напишите программу, которая считывает со стандартного ввода целые

#числа,по одному числу в строке, и после первого введенного нуля

#выводит сумму полученных на вход чисел.

chisla = int(input())

res = 0

while chisla:

res += chisla

chisla = int(input())

print(res)

**Задача 2: ( while2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Даны целые положительные числа N и K. Используя только операции

#сложения и вычитания, найти частное от деления нацело N на K, а также

#остаток от этого деления.

N = int(input('N = '))

K = int(input('K = '))

n = N

k = 0

while n >= K:

n -= K

k += 1

print("Частное -", k)

print("Остаток -", n)

**Задача 3: ( while3.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дано целое число N (> 0). Если оно является степенью числа 3, то

#вывести TRUE, если не является — вывести FALSE

N = int(input())

while N >= 3:

N /= 3

print((N==1))

**Задача 4: ( while4.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дано целое число N (> 1). Найти наименьшее целое число K, при

#котором выполняется неравенство 3K > N.

N = int(input())

K = 0

a = 1

while a < N:

a \*= 3

K += 1

K -= 1

print('K =', K, '3^K =', 3\*\*K, '3^(K+1) =', 3\*\*(K+1))

# Практическое занятие №6.

## Complex.

a = complex(3, 4)

b = complex(6, 4)

print('Комплексное число a:', a)

print('Комплексное число b:', b)

c = a + b

print('Сложения комплексных чисел:', c)

print('Мнимая часть комплексного числа а:', a.imag)

print('Действительная часть комплексного числа b:', b.real)

print('Проверим на равенство комплексные числа a и b:', a == b)

print('Модуль комплексного числа a:', abs(a))

print('Возведение в четвертую степень комплексного числа b', pow(b, 4))

## Decimal.

# Выполнил: Зайцев Никита Евгеньевич

# Группа: П2-18

# Decimal- вычисления с заданной точностью

3.3 + 4.1 # Ответом будет 7,33333, в нашем же случае 7.4

from decimal import Decimal

a = Decimal('3.3') + Decimal('4.1')

print(a)

3.3 + 4.1 == 7.4 # должно быть True, но...

# Но с числами типа Decimal все верно:

c = Decimal('3.3') + Decimal('4.1')

print(c)

# С помощью дополнительных знаков мы можем определить, сколько будет символов в дробной части числа:

number = Decimal("0.10")

number = 3 \* number

print(number) # 0.30

# Однако нельзя смешивать в операциях дробные числа float и Decimal:

# number = Decimal("0.1")

# number = number + 0.1 # здесь возникнет ошибка

# Округление чисел

# Объекты Decimal имеют метод quantize(),который позволяет округлять числа.

# В этот метод в качестве первого аргумента передается также объект Decimal,

# который указывает формат округления числа:

number = Decimal("0.444")

number = number.quantize(Decimal("1.00"))

print(number) # 0.44

number = Decimal("0.555678")

print(number.quantize(Decimal("1.00"))) # 0.56

number = Decimal("0.999")

print(number.quantize(Decimal("1.00"))) # 1.00

# Как отделить рубли от копеек в вещественных числах?

# Decimal вместо float:

x = Decimal('40.80')

a = int(x)

b = int(100 \* (x - a))

print(a, b) # => 40 80

**(decimal.py)**

## Fraction.

# Зайцев Никита Евгеньевич

# Задание разбор модуля Fraction

# Модуль fractions

from fractions import Fraction

from decimal import Decimal

import math

#Fraction.limit\_denominator(max\_denominator=1000000)- ближайшее

#рациональное число со знаменателем не больше данного.

a = Fraction(3.1415) # Целое и вещественное число, так же можно преобразовать в обыкновенную дробь

print (a)

print (a.limit\_denominator())

# по умолчанию numerator=0, denominator=1

print (Fraction())

# равносильно Fraction(1, 2)

print (Fraction(numerator=1, denominator=2))

#Если указанные числитель и знаменатель имеют общие делители,

#то перед созданием рационального числа они будут сокращены

print (Fraction(8, 16), Fraction(15, 30))

#Если указанные числитель и знаменатель имеют общие делители,

#то перед созданием рационального числа они будут сокращены

print (Fraction(3, Fraction(1, 2)))

#Округляет до ближайшего четного числа.

print (Fraction('1/2').\_\_round\_\_())

#создает обыкновенную дробь, которая является точным представлением

#десятичной дроби указанной в dec, где dec – это экземпляр класса decimal.Decimal

print (Fraction.from\_decimal(Decimal('0.7')))

#принимает flt – число типа float и возвращает обыкновенную дробь отношение числителя

#к знаменателю которой максимально приближается к значению flt.

print (Fraction.from\_float(0.5))

#fractions.gcd(a, b) - наибольший общий делитель чисел a и b.

print (math.gcd(1000, 3))

print (math.gcd(4, 6))  
#Использовался сайт https://pyprog.pro/python/st\_lib/fractions.html

**(fraction.py)**

# Практическое занятие №7.

## Задания на строки.

**Задача 1: ( Задание 1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#С клавиатуры вводятся строки, последовательность заканчивается точкой.

#Выведите буквы введенных слов в верхнем регистре, разделяя их пробелами.

import re #Импорт регуляров

sri = input().upper(); #upper - верхний регистр

sri = re.split('',sri) #Разделяет каждую букву или цифру

print(' '.join(sri))#Выводим из масива и ставим пробел для наглядного разделения

**Задача 2: ( Задание 2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#С клавиатуры вводятся строки, последовательность заканчивается точкой.

#Выведите буквы введенных слов в верхнем регистре, разделяя их пробелами.

import re #Импорт регуляров

login = input();

login = re.sub(' ','\_',login) #Если есть пробел в переменной, то меняем его на \_

print(''.join(login)) #Выводим из масива

**Задача 3: ( Задание 3.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Уберите точки из введенного IP-адреса. Выведите сначала четыре числа через пробел,

#а затем сумму получившихся чисел.

ip = [int(n) for n in input().split('.')] # Разбив строки в массиве

print(\*ip,'\n'+str(sum(ip))) # Сложение и вывод ip

**Задача 4: ( Задание 4.py )**

# Выполнил Зайцев Н. Е. П2-18.

# @@ обозначает ошибку

# !! обозначает предупреждение

# // обозначает информационное сообщение

# \*\* обозначает подробное сообщение

# Напишите программу, которая принимает строки до точки и выводит,

# какого типа это сообщение. Если сообщение не содержит модификаторов,

# проигнорируйте его.

def log(msg =''): # Делаем функцию

if len(msg) >= 2:

type\_ = msg[0]+msg[1]

if type\_ == '!!': # Если в переменное есть значение, то вывести print

print ('предупреждение')

return

if type\_ == '@@':

print ('ошибка')

return

if type\_ == '//':

print ('информация')

return

if type\_ == '\*\*':

print ('подробное сообщение')

return

log(input('write: ')) # Вызов функции

## Форматирование строк.

# Выполнил: Зайцев Никита Евгеньевич

# Группа: П2-18

# Задание 1. Подготовить сравнительную инструкцию по использованию

# форматирования строк

# 1 Форматирование строк “По старинке” (оператор %)

name = input()

print('1) Hello, %s' % name)

# 2 Форматирование строк “По новому” (str.format)

print('2) Hello, {}'.format(name))

# 3 Интерполяция строк / f-Строки (Python 3.6+)

print(f'3) Hello, {name}!')

def greet(name, question):

return f"Hello, {name}! How's it {question}?"

print(greet(name, 'going'))

# 4 Шаблонные строки (Стандартная библиотека Template Strings)

from string import Template

t = Template('4) Hey, $name!')

print(t.substitute(name=name))

# Если для подстановки требуется только один аргумент,

# то значение - сам аргумент:

print('Hello, {}!'.format(name))

# А если несколько, то значениями будут

# являться все аргументы со строками подстановки (обычных или именованных):

print('{0}, {1}, {2}'.format('a', 'b', 'c'))

print('Coordinates: {latitude}, {longitude}'.format(latitude='37.24N', longitude='-115.81W'))

# В общем случае, аргументы могут быть как именованными, так и позиционными:

s = '{x}; {0}; {y}; {1}'

print(s.format('A', 'B', x = 1, y = 2))

# Здесь нужно обратить внимание на два нюанса: первый – позиционные аргументы, должны следовать перед именованными;

# второй – аргументами могут быть данные любого типа:

s = '{int}; {float}; {complex}'

print(s.format(int = 2, float = 2e-5, complex = 2+0.2j))

a = list(range(3))

b = dict([[1,'a'],[2,'b']])

c = set('aabbcc')

s = '{list}; {dict}; {set}'

print(s.format(list = a, dict = b, set = c))

# Мало того, что мы можем подставлять списки и словари, мы еще и можем обращаться к элементам, которые в них расположены.

**(7-2.py)**

# Практическое занятие №8.

## Задания на списки.

**Задача 1: ( 8-1.Задание 1.py )**

# Выполнил Зайцев Н. Е. П2-18.

# Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и

# выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку

# у них недостаточно соседей.

inp = input('write or enter ') # Получаем входные данные от пользователя

if len(inp) > 0:# проверяем данные на валидность

data = inp.split(',') #Разбиваем входящую строку на массив

else:

data = [1,3,2,1,0,6,5,6]#объявляем образец данных

c = len(data) -1

if(c > 3): # проверяем данные на валидность

out = 0

for i in range(c):

if i == 0 : continue # Пропускаем первый элемент

if data[i-1] < data[i] > data[i+1]: out+=1 # производим сложные вычисления

print(out)

else:

print("invalid data")

**Задача 2: ( 8-1.Задание 2.py )**

# Выполнил Зайцев Н. Е. П2-18.

# Дан список чисел. Посчитайте, сколько в нем пар элементов, равных друг другу.

# Считается, что любые два элемента, равные друг другу образуют одну пару, которую

# необходимо посчитать.

inp = input('write or enter ') # Получаем входные данные от пользователя

if len(inp) > 0:# проверяем данные на валидность

data = inp.split(',') #Разбиваем входящую строку на массив

else:

data = [1,3,2,1,0,6,5,6]#объявляем образец данных

c = len(data) -1

if(c > 2): # проверяем данные на валидность

out = 0

data = sorted(data) #сортирую массив

for i in range(c):

if data[i] == data[i+1]: out+=1 # производим сложные вычисления

if i==c : break

print(out)

else:

print("invalid data")

**Задача 3: ( 8-1.Задание 3.py )**

# Практическое занятие №14.

## Задания на Text.

**Задача 1: ( Задание 1.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дана строка S и текстовый файл. Добавить строку S в конец файла.

def file\_get\_contents(path,encoding="utf-8"):

with open(path, "r",encoding=encoding) as f:

out = f.read()

f.close()

return out

f = open('text.txt', 'a+')

s = str(input())

f.write(s)

f.close()

print(file\_get\_contents("text.txt"))

**Задача 2: ( Задание 2.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дана строка S и текстовый файл. Заменить в файле все пустые строки на строку S.

import re

def file\_get\_contents(path,encoding="utf-8"):

with open(path, "r",encoding=encoding) as f:

out = f.read()

f.close()

return out

def file\_put\_contents(path,value,encoding="utf-8"):

with open(path, "w",encoding=encoding) as f:

out = f.write(value)

f.close()

return True

txt = file\_get\_contents("text.txt")

s = ' строка S '

txt = re.sub(r'\s', s, txt)

file\_put\_contents("text.txt",txt)

**Задача 3: ( Задание 3.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дана строка S и текстовый файл. Заменить в файле все пустые строки на строку S.

import re

def file\_get\_contents(path,encoding="utf-8"):

with open(path, "r",encoding=encoding) as f:

out = f.read()

f.close()

return out

def file\_put\_contents(path,value,encoding="utf-8"):

with open(path, "w",encoding=encoding) as f:

out = f.write(value)

f.close()

return True

txt = file\_get\_contents("text.txt")

s = ' строка S '

txt = re.sub(r'\s{2,}', ' ', txt)

file\_put\_contents("text.txt",txt)

**Задача 4: ( Задание 4.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дан текстовый файл, каждая строка которого изображает целое число, дополненное слева и справа

#несколькими пробелами. Вывести количество этих чисел и их сумму.

import re

def file\_get\_contents(path,encoding="utf-8"):

with open(path, "r",encoding=encoding) as f:

out = f.read()

f.close()

return out

def file\_put\_contents(path,value,encoding="utf-8"):

with open(path, "w",encoding=encoding) as f:

out = f.write(value)

f.close()

return True

txt = file\_get\_contents('text1.txt')

sri = re.findall(r"[\n']+?",txt)

num = re.findall(r"\d+",txt)

out = 0

for n in num:

out += int(n)

print('Количество строчек - ', len(sri)+1)

print('Сумма - ', out)

**Задача 5: ( Задание 5.py )**

#Выполнил Зайцев Н. Е. П2-18.

#Дан текстовый файл. Создать символьный файл, содержащий все знаки препинания,

#встретившиеся в текстовом файле (в том же порядке).

import re

def file\_get\_contents(path,encoding="utf-8"):

with open(path, "r",encoding=encoding) as f:

out = f.read()

f.close()

return out

def file\_put\_contents(path,value,encoding="utf-8"):

with open(path, "w",encoding=encoding) as f:

out = f.write(value)

f.close()

return True

txt = file\_get\_contents('text1.txt')

sri = re.findall(r"[\n']+?",txt)

num = re.findall(r"\d+",txt)

txt = file\_get\_contents('text2.txt')

sri = re.findall(r"[^\w\s\d]",txt)

sri = '\n'.join(sri)

file\_put\_contents('test2.txt',sri)

print(sri)