**

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. 3](#_Toc57279879)

[Практическое занятие №4. 6](#_Toc57279886)

[Задачи на Begin. 6](#_Toc57279887)

[Задачи на Boolean. 6](#_Toc57279888)

[Задачи на Case. 6](#_Toc57279889)

[Задачи на If. 8](#_Toc57279890)

[Практическое занятие №5. 11](#_Toc57279894)

[Задачи на For. 11](#_Toc57279895)

[Задачи на While. 12](#_Toc57279896)

[Практическое занятие №6. 16](#_Toc57279900)

[Complex. 16](#_Toc57279901)

[Decimal. 16](#_Toc57279902)

[Fraction. 17](#_Toc57279903)

# 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

https://drive.google.com/file/d/1CdA0eJAUHnbn1lGqWnxPOlF6J8u1232Z/view

# 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)**