******

Колледж космического машиностроения и технологий

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей

программного обеспечения для компьютерных систем

специальность 09.02.03 Программирование в компьютерных системах

Выполнил студент:

Батраков Д. С.

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

Гусятинер Л. Б.

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

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

Королев, 2020

Содержание отчёта

[**Раздел 1. Техника решения задач с использованием структурного программирования.** 3](#_Toc58934607)

[1.1 Установка интерпретатора Python 3 и настройка окружения 3](#_Toc58934608)

[1.2 Техника работы в командной строке и среде IDLE 6](#_Toc58934609)

[**1.3** **Техника работы с линейными и разветвляющимися программами** 7](#_Toc58934610)

[**1.4** **Техника работы с циклическими программами, цикл while** 9](#_Toc58934611)

# **Раздел 1. Техника решения задач с использованием структурного программирования.**

## Установка интерпретатора Python 3 и настройка окружения

Для установки интерпретатора Python на компьютер, вам нужно скачать дистрибутив. Загрузить его последнюю версию можно с официального сайта, перейдя по ссылке https://www.python.org/downloads/

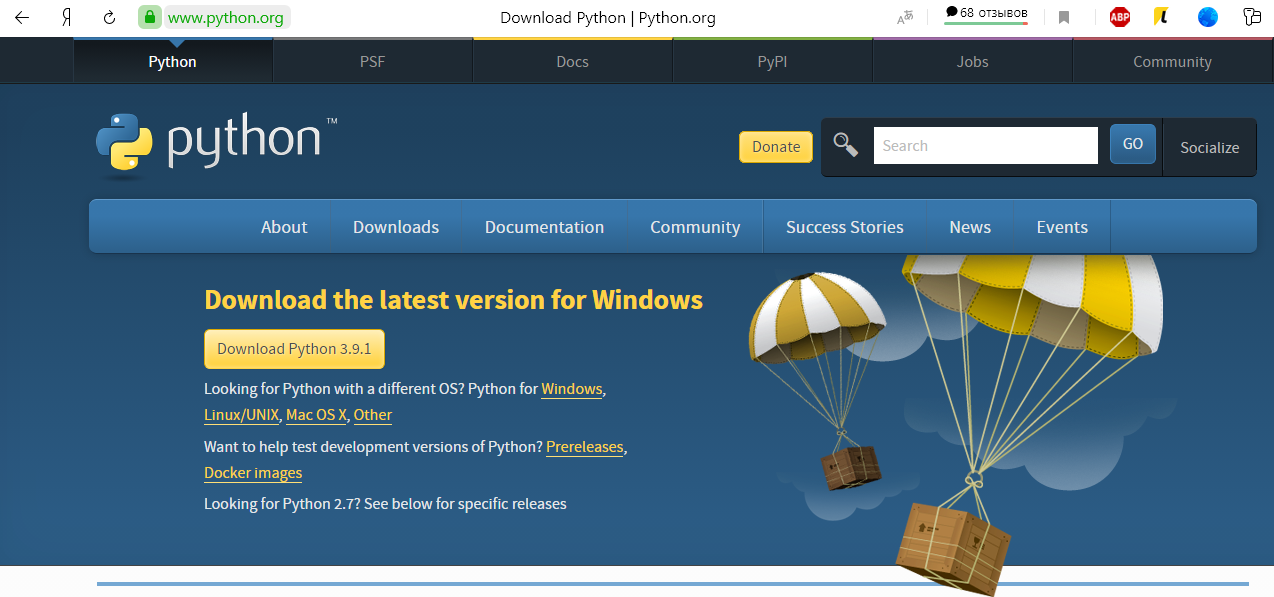


Рис 1. Официальный сайт Python

Порядок установки на Windows:

1. Запустить скачанный установочный файл.

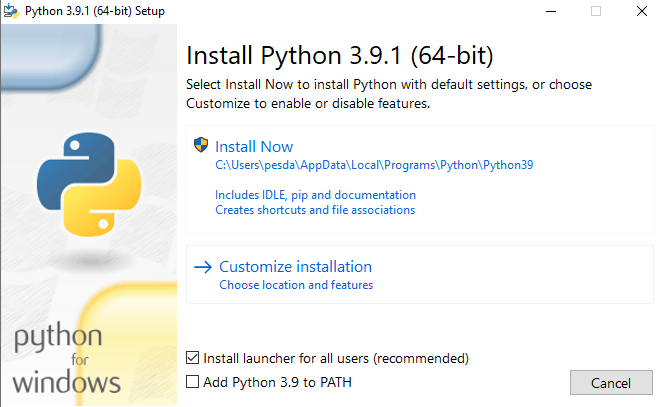
2. Выбрать способ установки.

Рис 2. Установщик Python

3. Отметить необходимые опции установки (доступно при выборе Customize installation)

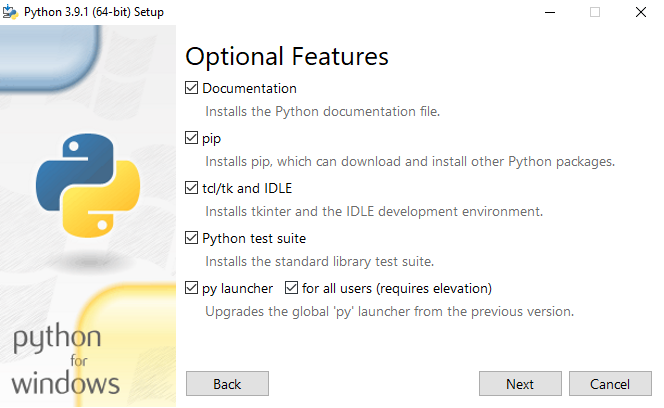


Рис 3. Опции установки

На этом шаге нам предлагается отметить дополнения, устанавливаемые вместе с интерпретатором Python. Выбираю:

* Documentation – установка документаций.
* pip – установка пакетного менеджера.
* tcl/tk and IDLE – установка интегрированной среды разработки и библиотеки.

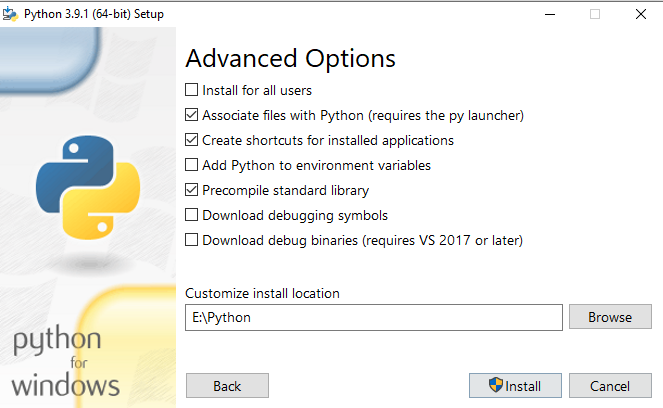
4. Выбираем место установки (доступно при выборе Customize installation)

Рис 4. «Продвинутые» опции установки

5. После успешной установки python:

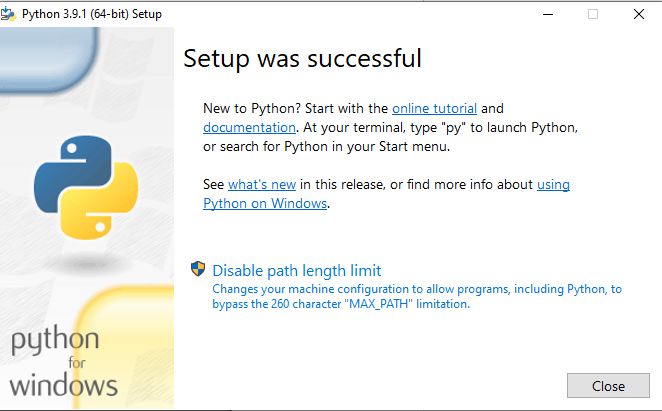


Рис 5. Сообщение об установке python

## Техника работы в командной строке и среде IDLE

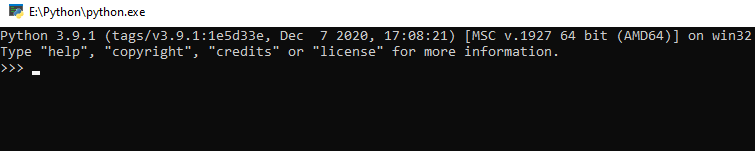
Выполняя (запуская) команду “python” в вашем терминале, вы получаете интерактивную оболочку Python.

Рис 6. Интерактивная оболочка Python

Существует несколько способов закрыть оболочку Python: >>> exit()или же >>> quit()

Теперь напишите в интерактивной оболочке следующий код:

>>> print("Oh, Hi Mark")

Нажмите **Enter** на вашей клавиатуре.

>>> print("Oh, Hi Mark ")

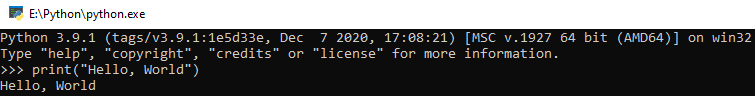
Oh, Hi Mark

Рис 7. Первая программа

## Техника работы с линейными и разветвляющимися программами

**Листинг 1. K4\_1**

**'''**

- input

Функция input() в Python, ввод данных с клавиатура.

https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-input/

- print

Функция print() в Python, печатает объект.

https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-print/

- форматная строка и метод формат

**'''**

print('Введите имя: ')

a = (input())

print('Hi, {0}, {1}!'.format(a, input()))

**Листинг 2. K4\_2\_1**

**'''**

К4\_2. Техника работы с разветвляющимися программами

Задание 1. Разработать программу для печати даты прописью

Пример ввода: 15.12.1983

Пример вывода: Пятнадцатое декабря одна тысяча девятсот восемьдесят третьего года

**'''**

def get\_date(date):

day\_list = ['первое', 'второе', 'третье', 'четвёртое',

'пятое', 'шестое', 'седьмое', 'восьмое',

'девятое', 'десятое', 'одиннадцатое', 'двенадцатое',

'тринадцатое', 'четырнадцатое', 'пятнадцатое', 'шестнадцатое',

'семнадцатое', 'восемнадцатое', 'девятнадцатое', 'двадцатое',

'двадцать первое', 'двадцать второе', 'двадцать третье',

'двадацать четвёртое', 'двадцать пятое', 'двадцать шестое',

'двадцать седьмое', 'двадцать восьмое', 'двадцать девятое',

'тридцатое', 'тридцать первое']

month\_list = ['января', 'февраля', 'марта', 'апреля', 'мая', 'июня','июля', 'августа', 'сентября', 'октября', 'ноября', 'декабря']

#единицы

year\_list\_unit = ['','первого','второго','третьего', 'четвёртого', 'пятого', 'шестого', 'седьмого', 'восьмого', 'девятого']

year\_list\_unit2 = ['', 'одиннадцатого', 'двенадцатого', 'тринадцатого', 'четырнадцатого', 'пятнадцатого',

'шестнадцатого', 'семнадцатого', 'восемнадцатого', 'девятнадцатого']

#десятки

year\_list\_dec = ['','','двадцать','тридцать','сорок','пятьдесят','шестьдесят','семьдесят','восемьдесят','девяносто']

year\_list\_dec2 = ['', 'десятого', 'двадцатого', 'тридцатого', 'сорокового', 'пятидесятого', 'шестидесятого', 'семидесятого', 'восемидесятого',

'девяностого']

#сотни

year\_list\_hun = ['','сто','двести','триста','четыреста','пятьсот','шестьсот','семьсот','восемьсот','девятьсот']

year\_list\_hun2 = ['','сотого','двухсотого','трёхсотого','четырёхсотого','пятьсотого','шестьсотого','семьсотого','восьмисотого','девятьсотого']

#тысячи

year\_list\_th = ['','тысяча','две тысячи','три тысячи']

year\_list\_th2 = ['','тысячного','двух тысячного','трёх тысячного']

date\_list = date.split('.')

date\_y = [int(date\_list[2])//1000, int(date\_list[2])%1000//100, int(date\_list[2])%100//10,int(date\_list[2])%10]

if(date\_y[2] == 1):

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' +

year\_list\_th[date\_y[0]] + ' ' + year\_list\_hun[date\_y[1]-1] + ' ' + year\_list\_unit2[date\_y[3]] + ' ' + 'года')

elif (date\_y[0] == date\_y[1] == date\_y[2] == date\_y[3] == 0):

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' + 'нулевого года')

elif (date\_y[1] == date\_y[2] == date\_y[3] == 0):

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' +

year\_list\_th2[date\_y[0]] + ' ' + 'года')

elif(date\_y[2] == date\_y[3] == 0):

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' +

year\_list\_th[date\_y[0]] + ' ' + year\_list\_hun2[date\_y[1]] + ' ' + 'года')

elif(date\_y[3] == 0):

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' +

year\_list\_th[date\_y[0]] + ' ' + year\_list\_hun[date\_y[1]] + ' ' + year\_list\_dec2[date\_y[2]] + ' ' + 'года')

return (day\_list[int(date\_list[0]) - 1] + ' ' +

month\_list[int(date\_list[1]) - 1] + ' ' +

year\_list\_th[date\_y[0]] + ' ' + year\_list\_hun[date\_y[1]] + ' ' + year\_list\_dec[date\_y[2]] + ' ' + year\_list\_unit[date\_y[3]] + ' ' + 'года')

date = input()

while(date != 'stop'):

print(get\_date(date))

date = input()

**Листинг 3. K5\_1\_2**

**'''**

Задание 2.

Придумать пример(ы) на использование break / continue /else.

**'''**

count = 0

while (1):

a = int(input())

count += 1

if (count >= 5):

break

else:

continue

## Техника работы с циклическими программами, цикл while

**Листинг 4. К5\_2\_1**

**'''**

Задание 1. Вычислить значение sin(x) с точностью до epsilon при помощи разложения в ряд

**'''**

import math

eps = 1.0

n = 0

while eps + 1 > 1:

eps /= 2

n+=1

eps \*= 2

x = int(input())

s = 0

for i in range(n):

s += math.sin(x)

print(s)

**Листинг 5.К5\_2\_2**

**'''**

К5\_2. Техника работы с циклическими программами \_ цикл while;

Задание 2.

https://stepik.org/lesson/3364/step/11?unit=947

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.

**‘’’**

def main():

a = int(input())

um = 0

while a != 0:

um = um + a

a = int(input())

print(um)

if \_\_name\_\_ == “\_\_main\_\_”:

main()

**Листинг 6. К5\_2\_3**

**'''**

К5\_2. Техника работы с циклическими программами \_ цикл while;

Задание 3.

Разработать программу для нахождения наибольшего общего делителя

**'''**

def nod(a, b):

assert a >= 0 and b >= 0

if a == 0 or b == 0:

return max(a, b)

return nod(b % a, a)

if \_\_name\_\_ == "\_\_main\_\_":

a, b = map(int, input().split())

print(nod(a, b))

**Листинг 7. К5\_2\_4**

**'''**

К5\_2. Техника работы с циклическими программами \_ цикл while;

Задание 4.

С использованием результата задания 2 разработать программу для нахождения наименьшего

общего кратного

'**''**

def main():

a, b = map(int, input().split())

c = a \* b

while (n != 0 and c != 0):

if (a > b):

a = a % b

else:

b = b % a

print(c // (a + b))

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Листинг 8. К5\_2\_5**

**'''**

К5\_2. Техника работы с циклическими программами \_ цикл while;

Задание 5.

https://stepik.org/lesson/3369/step/8?unit=952

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ...

(число повторяется столько раз, чему равно).

На вход программе передаётся неотрицательное целое число n — столько элементов

последовательности должна отобразить программа.

На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.

**'''**

def main():

n = int(input())

j = num = 1

for i in range(1, n+1):

print(num, end=' ')

if j < num:

j += 1

else:

j = 1

num += 1

print("\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()

## Техника работы с числами

**Листинг 9. К6\_1\_1**

**'''**

Задание 1. Составить и выполнить по 3 примера использования модулей для работы с дробными числами (fractions), для точных вычислений (decimal).

'**''**

from decimal import Decimal

from fractions import Fraction

#decimal

print("Функции <<Decimal>>")

number = Decimal("0.444")

print(number)

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

print(number)

number = Decimal("0.555678")

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

number = Decimal("0.999")

print()

#fraction

print("Функции <<Fraction>>")

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

print(Fraction(1, 3))

print(Fraction(3.1415))

a = Fraction(1, 7)

b = Fraction(1, 3)

print(a + b)

**Листинг 10. К6\_2**

**'''**

К6\_2. Техника работы с числами cmath

**'''**

import cmath

def main():

print("Сложные функции")

print("cmath.polar(complex(1.0, 1.0)) =", cmath.polar(complex(1.0, 1.0)))

print("cmath.phase(complex(1.0, 1.0)) =", cmath.phase(complex(1.0, 1.0)))

print("abs(complex(1.0, 1.0)) =", abs(complex(1.0, 1.0)))

print("cmath.sqrt(complex(25.0, 25.0)) =", cmath.sqrt(complex(25.0, 25.0)))

print("cmath.cos(complex(25.0, 25.0)) =", cmath.cos(complex(25.0, 25.0)))

print()

if (\_\_name\_\_ == "\_\_main\_\_"):

main()

**Листинг 11. К6\_2**

**'''**

К6\_2. Техника работы с числами math

**'''**

import math

def getsin(x):

multiplier = 1

result = 0

for i in range(1, 20, 2):

result += multiplier \* pow(x, i) / math.factorial(i)

multiplier \*= -1

return result

def main():

print("Арифметические функции")

print("math.pow(3, 2) =", math.pow(3, 2))

print("math.pow(9, 0.5) =", math.pow(9, 0.5))

print("math.sqrt(9) =", math.sqrt(9))

print("math.factorial(5) =", math.factorial(5))

print("sin(pi/2) =", getsin(math.pi / 2))

print()

print("Тригонометрические функции")

print("math.sin(math.pi/4) =", math.sin(math.pi / 4))

print("math.cos(math.pi) =", math.cos(math.pi))

print("math.tan(math.pi/6) =", math.tan(math.pi / 6))

print("math.hypot(12,5) =", math.hypot(12, 5))

print("math.atan(0.5773502691896257) =", math.atan(0.5773502691896257))

print("math.asin(0.7071067811865476) =", math.asin(0.7071067811865476))

print()

print("Гиперболические функции")

print("math.asinh(11.548739357257746) =", math.asinh(11.548739357257746))

print("math.acosh(11.591953275521519) =", math.acosh(11.591953275521519))

print("math.atanh(0.99627207622075) =", math.atanh(0.99627207622075))

print("math.sinh(math.pi) =", math.sinh(math.pi))

print("math.cosh(math.pi) =", math.cosh(math.pi))

print()

print("Логарифмические функции")

print("math.log(148.41315910257657) =", math.log(148.41315910257657))

print("math.log(148.41315910257657, 2) =", math.log(148.41315910257657, 2))

print("math.log(148.41315910257657, 10) =", math.log(148.41315910257657,10))

print("math.log(1.0000025) =", math.log(1.0000025))

print("math.log1p(0.0000025) =", math.log1p(0.0000025))

print("math.exp(5) =", math.exp(5))

print("math.e\*\*5 =", math.e \*\* 5)

print()

print("Округление:")

print("math.ceil(1.001) =", math.ceil(1.001))

print("math.floor(1.001) =", math.floor(1.001))

print("math.factorial(10) =", math.factorial(10))

print("math.gcd(10,125) =", math.gcd(10, 125))

print("math.trunc(1.001) =", math.trunc(1.001))

print("math.trunc(1.999) =", math.trunc(1.999))

print()

if (\_\_name\_\_ == "\_\_main\_\_"):

main()