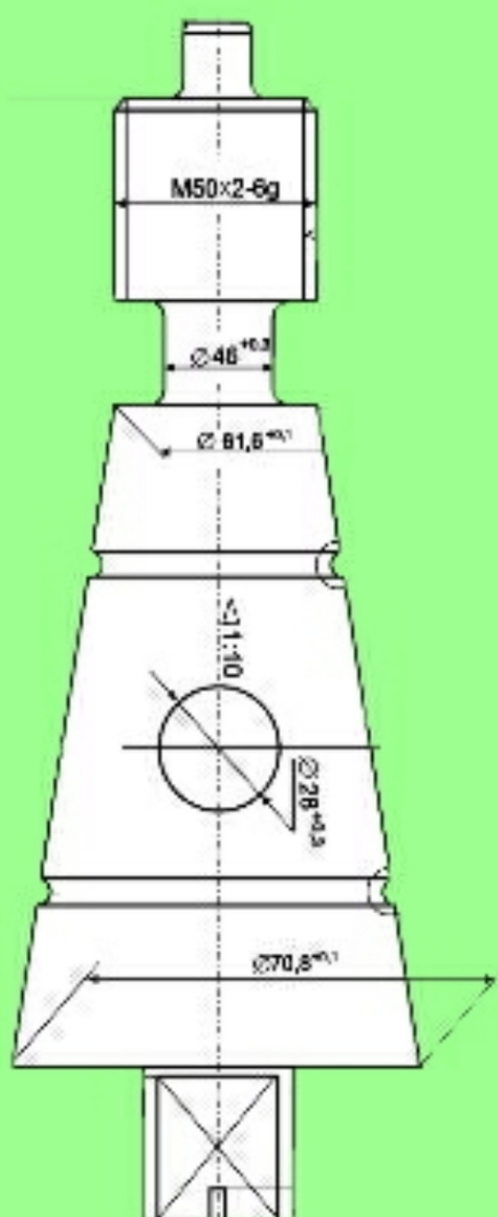


Сергей Гаврилов

Python-3

Расчет веса детали



Сергей Гаврилов

Python 3. Расчет веса детали

«ЛитРес: Самиздат»

2020

Гаврилов С. Ф.

Python 3. Расчет веса детали / С. Ф. Гаврилов — «ЛитРес: Самиздат», 2020

Эта книга для начинающих программистов, желающих начать программировать на языке Python 3. Набор программ создан для цеховых конструкторов-механиков как ежедневный рабочий инструмент. Книга будет полезна для студентов-механиков. Предлагаемые в книге расчеты экономят рабочее время и уменьшают ошибки в расчетах. Данная книга - сборник листингов рабочих программ. Все программы в разное время разработаны самим автором при возникновении необходимости данных расчетов. Все программы проверены и постоянно используются в работе конструктора-механика. Листинги программ снабжены пояснениями.

© Гаврилов С. Ф., 2020
© ЛитРес: Самиздат, 2020

Содержание

Введение	5
Примечание	6
Написание чисел с дробной десятичной частью.	8
Написание комментариев.	9
Листинги программ:	10
Конец ознакомительного фрагмента.	23

Введение

Python 3 Ves.

Инженер конструктор отдела гл.механика Гаврилов Сергей Фёдорович написал эту книгу для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков и для конструкторов механиков. Программа **Ves** используется конструктором каждый день. Закончив разработку детали – необходимо проставить на чертеже **Вес** детали. Значит запускаем программу **Ves**.

Данная книга содержит листинги – набор небольших рабочих программ для расчета веса деталей. Для получения рабочего комплекта – создаем папку **Ves**.

В эту папку копируем в режиме « Блокнот » по одной все программы проведенные в этой книге, сохранять программы следует точно под теми именами, что в книге – все программы сохраняем с расширением .ру..

В основной программе не забудьте выставить нужное количество пробелов в начале строк.

В книге приведен полный комплект программ для расчета веса детали.

Листинги выполнены как есть, без сокращений, и после копирования – готовы к работе. Предлагаемые в книге расчеты существенно экономят время и уменьшают вероятность ошибок. Все программы проверены и постоянно используются в работе конструктора механика. Листинги программ снабжены пояснениями. Листинги удобно использовать в качестве готовых блоков для вновь разрабатываемых программ. Для практического применения листингов программ проверьте – установлен ли на вашем компьютере Python 3.4. или более старшая версия Python 3.8..Python 3.4 распространяется свободно и бесплатно – например на сайте <https://soft-file.ru/python/> Перед скачиванием исходника Python 3.4 проверьте разрядность вашей ЭВМ. Посмотрите вкладку – Компьютер – Свойства системы. Бывают 32 разрядные и 64 разрядные системы. Для каждой системы свой Python 3.4..

Примечание

Программы написаны для Python 3.4 и более новых версий. На Python 2 программы работать не будут, так как в Python 3.4 написание команд отличается от Python 2.

В программе на любом языке очень важен синтаксис – порядок написания символов в строке.

Малейшая ошибка, которую человек даже не заметит, – ЭВМ замечает и отказывается исполнять программу. Поэтому проще копировать блоки уже работающих программ в свою программу, при необходимости, подправить готовые блоки легче, чем писать блок заново – так будет меньше ошибок в программе и экономится время.

Программа в Python состоит из строк, написанных в простом текстовом редакторе.

Текстовый редактор должен быть именно простым, к примеру Блокнот. Редактор Microsoft Word не годится потому, что он вставляет в строку невидимые служебные символы, которые, искажают команды Python и программа отказывается работать. Если листинг программы вы скачали в формате «.doc» или подобном – следует создать пустой текстовый файл в программе «Блокнот» – выделить и скопировать листинг из файла «.doc» и вставить в пустой текстовый файл «Блокнота» с расширением «.txt» и сохранить его с именем программы. Затем открыть этот файл и сохранить уже с расширением «.py». Вам будут нужны оба этих файла.

Важно, строки программы должны начинаться без пробела точно с начала строки.

Если в программе имеются логические операторы, например **while** или **for**, то за таким оператором следуют строки одного или нескольких блоков. Блоки выполняются программой, или пропускаются без исполнения, в зависимости от условия в логическом операторе.

Блоки отделяются от основной программы пробелами в начале строки. Обычно в начале строки блока ставят четыре пробела. Число пробелов в начале строки – важная величина. Мной указано сколько пробелов надо ставить в начале строки.

Сама программа начинается со строки «# -*- coding: cp1251 -*-» и заканчивается строкой «# Конец листинга программы». Копируя листинг в файл .txt – надо копировать только программу, любой текст до тела программы и после тела программы даст сбой в работе.

Если в компьютере Python 3.4 установлен правильно, то при двойном клике мышью на файле с расширением “.py”. программа начнет исполняться. Если программа стопорится или вылетает – Запускаем файл редактора программы

« IDLE(Python GUI) » в этом редакторе щелкаем левой кнопкой мыши на – File – затем Open. В выпавшем меню находим свой файл, выделяем его – щелкаем – Открыть. На экран выводится листинг вашего файла с выделением цветом элементов команд. Находим колонку – Run- щелкнув – открываем выпадающее меню и щелкаем на – Run Module F5 – Ваша программа начинает выполняться и останавливается на месте ошибки. Иногда она останавливается на следующей за ошибкой команде. Выводится комментарий к ошибке на английском.

В некоторых случаях помогает удаление строки в начале программы

«# -*- coding: cp1251 -*-» и программа заработает. Иногда помогает удалить конфликтную строку и забить ее вновь в текстовом редакторе.

« Программа редактор « IDLE(Python GUI) » идет в комплекте с программой Python 3.4.. и служит для нахождения ошибок в разрабатываемой программе.

При верстке книги к печати « умные программы » что-то убирают, а что-то добавляют, искажая оригинал. Для программы маленькие буквы и большие буквы – это разные буквы, а « умные программы » при верстке в начале строки бывает, что заменяют маленькую букву большой или меняют форму ковычек – программа вылетает.

К примеру при верстке удаляются все « лишние» пробелы. Скопировав листинг программы в «Блокноте» в формате «.txt» необходимо восстановить утраченные пробелы. Перед знаком # расположенным после команды надо добавить два пробела. Строка комментариев после значка # не должна переноситься на следующую строку. В блоках- там, где блок выделяется пробелами, перед строками блока будет написана фраза: # Далее Сдвиг – четыре пробела в начале каждой строки. Соответственно в листинг ставим четыре пробела в начало каждой строки. Там, где блок закончился, написано: # Далее Конец Сдвига – четыре пробела в начале каждой строки – и строки начинаются без пробелов в начале строки. Иногда меняется вид кавычек – это тоже не дает работать программе – искать такие ошибки лучше в «IDLE(Python GUI)».

Написание чисел с дробной десятичной частью.

При вводе числа с дробной частью, необходимо отделять дробную часть от целой части числа точкой. Если части числа разделить запятой – программа вылетает без предупреждения.

Величины углов для расчета в программе необходимо задавать в радианах. Если программа в результат расчета выводит величину угла – то он выводится в радианах. Для человека ответ на экран удобнее выводить в градусах – при выводе на экран радианы программно переводят в градусы. При вводе угловых величин- их обычно вводят в градусах и, перед подачей в расчетную часть программы, переводят программно в радианы.

Написание комментариев.

Значок # предваряет начало комментариев. То, что следует за значком программа просто пропускает, переходя на следующую строку. Комментарий программист пишет себе для справки, чтобы потом ему или другому программисту было легче разобраться в работе программы.

Листинги программ:

Для расчета веса детали – деталь мысленно разобьем на ряд типовых элементов: на цилиндры, конуса, трубы и т.д. У элементов запишем все размеры с чертежа и длинны элементов. Затем запускаем программу **Ves_G.py**...

Программа выведет меню из 17 пунктов. Меню примитивное – на мышь меню не реагирует. Для выполнения какого либо пункта меню – вводим число – номер пункта меню и Enter.. Программа закольцована, при исполнении следующего пункта меню – результат расчета веса суммируется с ранее найденным результатом – в итоге мы постоянно имеем суммарный вес уже просчитанных элементов. Проограмма – **Ves_G.py**...головная – в зависимости от выбранного пункта меню – программа автоматически подключает к работе одну из 17 подпрограмм . Главное: Головная программа и детальные программы (подпрограммы) должны располагаться в папке Ves..

Примечание: Python 3.4 имеет особенность – когда головная программа вызывает подпрограммы, то как обычно, подпрограмма забирает исходные данные, рассчитывает результат и отдает его головной программы. Но вот мы повторно вызываем подпрограмму для расчета по изменившимся исходным данным – Python 3.4 игнорирует, то что данные изменились и выдает нам предыдущий результат. Чтобы не попасть на такую ошибку, приходится вести учет применения подпрограмм – при первом применении считаем как обычно,

```
import Krug # Подключили модуль расчета круглого сечения
при втором и последующем применении подпрограмму вызываем функцией:
from imp import reload
reload (Krug) # Вызываем на повторный расчет подпрограмму (Krug)...
```

Только так получаем верные результаты.. (В других языках программирования я такого не встречал !!)..

```
Программа Ves_G.py...
# -*- coding: cp1251 -*-
# Программа расчета веса и других параметров
# объемных геометрических элементов..
# Результаты расчета элемента смотри в файле Rezul.txt
# =====
import sys # Подключили основные библиотеки
import shutil # Подключили модуль копирования файлов
import math # Подключили математич модуль
import time # Подключили модуль времени
import datetime # Подключили модуль даты
Pii=math.pi # Вытащили число " Пи "
# .....
# time.strftime("%d.%m.%Y") # Работа с датой и временем
# a = datetime.datetime.today().strftime("%Y%m%d")
# print(a) # '20170405'
today = datetime.datetime.today()
# print( today.strftime("%m/%d/%Y") ) # '04/05/2017'
# print( today.strftime("%Y-%m-%d-%H.%M.%S") ) # 2017-04-05-00.18.00
Ddd=( today.strftime("%Y-%m-%d-//-%H.%M.%S") )
# Gsf = 461030/2725231222..
print(" ")
```

```
Dds=" "+Ddd+" Расчет веса элементов детали "  
print(Dds)  
# input( ) # Ожидание нажима Ентер  
# .....  
fv = open('Vn.txt', 'r') # Открыли файл для чтения  
# Читаем записанные число часов в текстовом виде  
ct=fv.read()  
fv.close() # закрыли файл  
# .....  
uu=" "  
# print (uu)  
# print (uu)  
u=" Забрали из файла число часов ( строкой ) = "+ct  
# print (u)  
xt=float(ct) # Принудительно в вещественное число  
xtc=int(xt) # Целое число часов из файла  
# print (uu)  
  
# .....  
  
x=time.time() # Нашли число секунд из системы компьютера  
x=float(x) # Принудительно в вещественное число  
xtk=int(x) # Преобразуем секунды к целому числу  
xtmk=int(x/60) # Преобразуем минуты к целому числу  
xtck=int(x/3600) # Преобразуем часы к целому числу  
utck=str(xtck) # Преобразуем число часов в строку  
xtdk=int(xtck/24) # Преобразуем дни к целому числу  
utdk=str(xtdk) # Преобразуем число дней в строку  
  
uk=" Забрали из системы компьютера число часов = "+utck  
# print (uk)  
# print (uu)  
  
y=int(xtck-xtc) # Разность часов из компа и из файла  
uy=str(y) # Преобразуем число в строку  
u=" Разность в часах с временем из файла = "+uy  
# print (u)  
# print (uu)  
  
t1="."  
t2="Ф"  
t3="В"  
t4="о"  
t5="р"  
t6="П"  
t7="С"  
t8="г"  
t9="а"  
t10="."
```

```

t11="Г"
t12="М"
t13="И"
t14="Л"
t15="Т"
t16="С"
t17="2"
t18="0"
t19="Т"
t20="1"
t21=" "
ufp=t6+t5+t4+t8+t5+t9+t12+t12+t13+t16+t15
ufa=t11+t9+t3+t5+t13+t14+t4+t3+t1+t21+t7+t1+t2+t1
ufg=t17+t18+t20+t17+t21+t8+t1

if y>10:
# Далее все строки пишем с отступом в четыре пробела от начала строки...
# .....
fv = open('Vn.txt', 'w') # Открыли файл для записи
# Записываем число часов в текстовом виде
fv.write(utck)
u=" Записали число часов в Файл "
print (u)
print (uu)
fv.close() # закрыли файл
x=xtk/14
y=int(xtk/14)
if y==x:
# Далее все строки пишем с отступом в восемь пробелов от начала строки...
    print (uu)
    print (" "+ufp)
    print (uu)
    print (" "+ufa)
    print (uu)
    print (" "+ufg)
    print (uu)
    print (" ===== ")
    print (uu)
    print (uu)
    input( ) # Ожидание нажима Ентер
# Далее все строки пишем с отступом в четыре пробела от начала строки...
print (uu)
# ..... # Далее все строки пишем без отступа...
u=" Работа с временем "
# print (u)
# print (uu)
# input( ) # Ожидание нажима Ентер
# .....
# .....

```

```
f = open('Rezult.txt', 'w') # Открыли файл для записи ( старое удалим )
# Записываем числа в текстовом виде
u=uu+"\n" # Добавим код перевода строки
f.write(u)

u=Dds+"\n" # Добавим код перевода строки
f.write(u)

u=utdk+"\n" # Добавим код перевода строки
f.write(u) # Запишем – сколько дней прошло от даты X
u=uu+"\n" # Добавим код перевода строки
f.write(u)
u=" ===== "
u=u+"\n" # Добавим код перевода строки
f.write(u) # записали в файл

f.close() # закрыли файл
# .....
# .....
print (uu)
print (uu)
print (uu)
u=" РАСЧЕТ ВЕСА ДЕТАЛИ "+ "\n"
print (u)
print (uu)
u=" Программа рассчитывает вес элементарного фрагмента детали "+ "\n"
print (u)
u=" выводит вес фрагмента на экран, затем суммирует с весом "+ "\n"
print (u)
u=" уже рассчитанных фрагментов. Если фрагмент представляет "+ "\n"
print (u)
u=" собой полость ( пустоту ), то его длина вводится со знаком "+ "\n"
print (u)
u=" минус – вес пустого фрагмента будет вычитаться из суммарного "+ "\n"
print (u)
u=" веса детали. "+ "\n"
print (u)
u=" Одновременно с расчетом веса фрагмента – программа "+ "\n"
print (u)
u=" рассчитывает геометрические характеристики сечения фрагмента "+ "\n"
print (u)
u=" необходимые для расчетов на прочность и других расчетов "+ "\n"
print (u)
u=" На экран эти результаты не выводятся, а записываются "+ "\n"
print (u)
u=" в файл Rezult.txt – который можно посмотреть после "+ "\n"
print (u)
u=" окончания расчета веса детали. "+ "\n"
```

```

print (u)
print (uu)
u=" Гаврилов С.Ф. ноябрь 2012 г. "+"\\n"
print (u)
print (uu)
u=" ===== "+"\\n"
print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# - - - -

global Kodis # Объявили глобальную переменную
global Ves # Объявили глобальную переменную
global VesS # Объявили глобальную переменную
global VesT # Объявили глобальную переменную
VesS=0.000000
VesT=0.000000
# Gsf = 461030/2725231222..
# Pii=math.pi # Вытащили число " Пи "
Kodis=1
Kodp1=0
Kodp2=0
Kodp3=0
Kodp4=0
Kodp5=0
Kodp6=0
Kodp7=0
Kodp8=0
Kodp9=0
Kodp10=0
Kodp11=0
Kodp12=0
Kodp13=0
Kodp14=0
Kodp15=0
Kodp16=0
Kodp17=0
Kodp18=0
Kodp19=0
Kodp20=0
Kodim=0
Kodimp=1
uu=" "
import math # Подключили математич модуль

# - - - - -
while Kodis>0:
# Далее все строки пишем с отступом в четыре пробела от начала строки...
uu=" ===== "
```

```
print (uu)
uu=" "
print (uu)
print (uu)
u=" Расчет веса сечений с суммированием ,"
print (u)
print (uu)
print (uu)
u=" 1 – Цилиндр "
print (u)
u=" 2 – Труба D и d "
print (u)
u=" 3 – Труба D и S "
print (u)
u=" 4 – Шестигранник "
print (u)
u=" 5 – Пластина "
print (u)
u=" 6 – Треугольник прямоугольный "
print (u)
u=" 7 – Треугольник по сторонам "
print (u)
u=" 8 – Трапеция "
print (u)
u=" 9 – Косынка "
print (u)
u=" 10 – Усеченный конус "
print (u)
u=" 11 – Усеченная пирамида "
print (u)
u=" 12 – Горбушка "
print (u)
u=" 13 – Кольцо ( в сечении трапеция ) "
print (u)
u=" 14 – Канавка под клиновой ремень "
print (u)
u=" 15 – Добавить объем в куб. мм. "
print (u)
u=" 16 – Пересчет на другой материал "
print (u)
u=" 17 – Обнулить вес "
print (u)
print (uu)

u=" 0 – Выход из программы ( ноль ) "
print (u)
print (uu)
print (uu)
u=" Размеры вводим в мм. Плотность стали 7,85; "
```

```

print (u)
print (uu)
Kodis=input( )

Kodis=float(Kodis) # Принудительно в вещественное число
# .....
# .....
if Kodis ==1: # Выбор расчета вида сечения ( круг )
# Далее все строки пишем с отступом в восемь пробелов от начала строки...
if Kodp1 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
import Krug # Подключили модуль расчета круглого сечения
VesT=Krug.px # Извлекаем из модуля значение переменной
u=" Вес цилиндрического элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp1=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (Krug)
VesT=Krug.px # Извлекаем из модуля значение переменной
u=" Вес цилиндрического элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)

```

```

print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# .....
# Gsf = 461030/2725231222..
# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==2: # Выбор расчета вида сечения ( Труба D,d )
# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp2 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
import Truba # Подключили модуль расчета Труба D,d
VesT=Truba.px # Извлекаем из модуля значение переменной
u=" Вес трубного элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp2=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (Truba)
VesT=Truba.px # Извлекаем из модуля значение переменной
u=" Вес трубного элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)
print (u)

```

```

print (uu)
input() # Ожидание нажима Enter

# .....
# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==3: # Выбор расчета вида сечения ( Труба D,S )
# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp3 ==0:
# Далее все строки пишем с отступом двенадцать пробел от начала строки...
import TrubaS # Подключили модуль расчета Труба D,S
VesT=TrubaS.px # Извлекаем из модуля значение переменной
u=" Вес трубного элемента S = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp3=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input() # Ожидание нажима Enter
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (TrubaS)
VesT=TrubaS.px # Извлекаем из модуля значение переменной
u=" Вес трубного элемента S = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)
print (u)
print (uu)

```

```

input( ) # Ожидание нажима Enter
# ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,
# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==4: # Выбор расчета вида сечения ( Шестигранник )
# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp4 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
import Sestig # Подключили модуль расчета Sestig_2
VesT=Sestig.px # Извлекаем из модуля значение переменной
u=" Вес шестигранника = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp4=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Enter
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (Sestig)
VesT=Sestig.px # Извлекаем из модуля значение переменной
u=" Вес шестигранного элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Enter
# ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,

```

```

# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==5: # Выбор расчета вида сечения ( Пластина )
# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp5 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
import Prug # Подключили модуль расчета Prug_2
VesT=Prug.px # Извлекаем из модуля значение переменной
u=" Вес пластины = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp5=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (Prug)
VesT=Prug.px # Извлекаем из модуля значение переменной
u=" Вес прямоугольного элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Ентер
# .....
# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==6: # Выбор расчета вида ( Прямоуг. треугольник )

```

```

# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp6 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала
# строки...
import Treu # Подключили модуль расчета Treu_2
VesT=Treu.px # Извлекаем из модуля значение переменной
u=" Вес Прямоуг. треугольника = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
Kodp6=1 # Первое применение произошло
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Enter
# Далее все строки пишем с отступом в восемь пробел от начала строки...
else:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
from imp import reload
reload (Treu)
VesT=Treu.px # Извлекаем из модуля значение переменной
u=" Вес Прямоуг. треугольника = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
VesS=VesS+VesT # Суммарный вес
u=" Суммарный вес = "
t=str(VesS) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" ===== "
print (uu)
print (u)
print (uu)
input( ) # Ожидание нажима Enter
# .....
# Gsf = 461030/2725231222..
# Далее все строки пишем с отступом в четыре пробела от начала строки...
if Kodis ==7: # Выбор расчета вида сечения ( Треугольник )

```

```
# Далее все строки пишем с отступом в восемь пробел от начала строки...
if Kodp7 ==0:
# Далее все строки пишем с отступом в двенадцать пробел от начала строки...
import Treu3 # Подключили модуль расчета Treu3
VesT=Treu3.px # Извлекаем из модуля значение переменной
u=" Вес Треугольного элемента = "
t=str(VesT) # Преобразуем число в строку
u=u+t # сложим строки
```

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.