

Сергей Гаврилов

Python 3
полезные программы

12+

Сергей Фёдорович Гаврилов

Python 3, полезные программы

http://www.litres.ru/pages/biblio_book/?art=51565391

SelfPub; 2020

Аннотация

Эта книга для начинающих программистов, желающих начать программировать на языке Python 3. В книге приведено большое количество рабочих программ. Книга будет полезна для студентов механиков и для цеховых конструкторов механиков. Предлагаемые в книге расчеты экономят рабочее время. Язык программирования Python хорошо подходит для быстрого решения задач по определению оптимальных размеров и параметров, методом расчета множества вариантов решения. Программы на Python хорошо подходят для быстрого решения типовых задач. Программирование на Python требует меньше времени чем на других языках. Конструкторам в цеху часто приходится быстро выполнять разнообразные расчеты – библиотека небольших программ позволяет значительно экономить время и уменьшить вероятность ошибок. Данная книга – сборник листингов рабочих программ. Все программы, в разное время, разработаны самим автором, при возникновении необходимости данных расчетов. Все программы проверены.

Содержание

Введение.	4
Примечание:	7
Фрагменты программ для справок	11
Математические функции	11
Файловые функции	13
Подключение библиотек	15
Общая геометрия	18
Линейная интерполяция	18
Система уравнений	22
Квадратное уравнение	29
Координаты точек на радиусной кривой	33
Хорда – Прогиб – Радиус	39
Конец ознакомительного фрагмента.	55

Введение.

Книга первая..

Инженер конструктор отдела гл.механика Механического завода Гаврилов Сергей Федорович написал эту книгу для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков и для цеховых конструкторов механиков. Предлагаемые в книге расчеты существенно экономят время.

Язык программирования Python хорошо подходит для быстрого решения задач типа

определения оптимальных размеров, методом расчета множества вариантов решения. Программы на Python хорошо подходят для быстрого решения типовых задач.

Программирование на Python требует меньше времени чем на других языках.

Небольшие программы можно комплектовать в библиотеки или включать их в объединяющую программу. В объединяющую программу можно просто перекопировать текст небольших программ уже готовых и проверенных, или можно просто указать ссылки на эти программы и все будет работать.

Конструкторам в цеху часто приходится быстро выполнять разнообразные расчеты – библиотека

небольших программ позволяет значительно экономить

время и уменьшить вероятность ошибок. 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. Листинги удобно использовать в качестве готовых блоков для вновь разрабатываемых программ.

Python высокоуровневый интерпретирующий язык программирования.

– язык программирования, разработанный для быстроты и удобства использования программистом. Команда на высокоуровневом языке легко запоминается человеком. При интерпретации, внутри ЭВМ, команда высокого уровня разворачивается в целый пакет команд на машинном коде которые очень длинны и сложны для непосредственного понимания. Высокоуровневые языки программирования были разработаны также для платформенной независимости (независимости от марки ЭВМ и операционной системы).

Приспособление к платформе перекладывается на инструментальные программы – трансляторы, компилирующие текст, написанный на языке высокого уровня, в элементарные машинные команды (инструкции) или на интерпретаторы (например, Python).

Python интерпретирующий язык. При запуске программы

Python читает первую строку, переводит прочитанную команду высокого уровня в пакет машинных кодов, затем выполняет команды пакета машинных кодов, после выполнения читает следующую строку и так до конца программы. При этом каждая команда высокого уровня интерпретируется в пакет машинных кодов – затем пакет машинных кодов исполняется.

Для практического применения листингов программ в первую очередь 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/>

При установке проверьте Python только на отсутствие вирусов. 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.

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.

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.

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.

Если в компьютере Python 3.4 установлен правильно, то

при двойном клике мышью на файле с расширением “.ру”. 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 ее 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. Программа редактор « IDLE(Python GUI) » идет в комплекте с программой Python 3.4.. и служит для нахождения ошибок в разрабатываемой программе.

При верстке книги « умные программы » что-то убирают, а что-то добавляют, искажая оригинал.

К примеру при верстке удаляются все « лишние» пробелы, поэтому перед знаком # надо добавить два пробела. В блоках- там, где блок выделяется пробелами, в начале строки пишем:

Далее Сдвиг – четыре пробела в начале каждой строки. И в листинг ставим четыре пробела. Там, где блок закончился, пишем: # Далее Конец Сдвига – четыре пробела в начале каждой строки – и строки начинаются без пробелов в начале

строки.

Если при копировании программа не работает: обращайтесь gsf4610@yandex.ru к автору.

Фрагменты программ для справок

Математические функции

```
u=" Математические функции ( блоки для вставки в про-
граммы ) "
uu=" "
print (uu)
print (uu)
print (u)
u=" ..... "
print (uu)
print (u)
print (" Демонстрация № 1 работа с математическими опе-
раторами ")
print (uu)
print (" Ввести число дробью с двумя десятичными знака-
ми ")
Kt=11.11 # Предварительно задаем – что такое Kt ( это
число )
Kt=input( ) # Вводим число Kt
aus=str(Kt) # Преобразуем число в строку
print (" = "+aus)
Kt=8.64 # Задано значение Kt
```

```
xs=str(Kt) # Преобразуем число в строку
print (xs)
K=float(xs) # Принудительно в вещественное число
print (K)
Kb=math.ceil(K) # Kt округлим до ближайшего большего
целого.
Km=math.floor(K) # Kt округлим до ближайшего меньше-
го целого.
print (Kb)
print (Km)
E=-1 # Присвоили E значение минус один.
K=K*E # Теперь K – отрицательное число
print (K)
Ee=abs(K) # Выводим абсолютное значение K в перемен-
ную Ee.
print (Ee) # Ee – положительное число.
print (" Демонстрация № 1 закончена ")
input( ) # Ожидание нажима Enter
print (uu)
# .....
# K=float(Kt) # Принудительно Kt в вещественное число K
# aus=str(au) # Преобразуем число au – в строку aus.
# Kt=8.64 # Задано значение Kt
# Kb=math.ceil(K) # Kt округлим до ближайшего большего
целого.
# Km=math.floor(K) # Kt округлим до ближайшего мень-
```

шего целого.

Ee=round(Nt,6) # Выводим только 6 знаков после запятой числа Nt ..

Ee=abs(Nt) # Выводим абсолютное значение Nt в переменную Ee.

y=math.cos(C) # COS угла C в радианах

z=math.sin(A) # Sin угла A в радианах

ur=math.tan(X) # Тангенс от X

aur=math.acos(X) # Угол aur в радианах от значения косинуса X

A=aur*180/Pii # Угол A в градусах, aur в радианах.

ur=math.atan(X) # Аркс тангенс от X, ur в радианах.

ur=math.asin(X) # Аркс синус от X

z=math.sqrt(D) # Корень квадратный из D

X=math.log(K) # X – Натуральный логарифм от числа " K "

K=math.exp(X) # K – экспонента от числа X

Файловые функции

f = open('Rezult.txt', 'a') # Открыли файл для дополнения записи

f = open('Rezult.txt', 'w') # Открыли файл для записи

(прежнее удаляем).

```
# f = open('Rezult.txt', 'r') # Открыли файл для чтения  
# f.close() # закрыли файл Rezult.txt
```

```
# f = open('Rezult.txt', 'r') # Открыли файл для чтения  
# t=f.read() # Выводит все оставшиеся строки файла на  
экран
```

```
# print(open('Fa.txt').readline()) # Выводит 1 строку файла  
на экран
```

```
# t=f.readline() # Выводит следующую строку файла в пе-  
ременную и на экран
```

```
# e=" Прочитали 3 = "
```

```
# k=e+t
```

```
# print (k)
```

```
# t=f.read() # Выводит оставшиеся строки файла в пере-  
менную и на экран
```

```
# e=" Прочитали = = "
```

```
# k=e+t
```

```
# print (k)
```

```
# ..... .....
```

```
# fv = open('Rezult.txt', 'r') # Открыли файл Rezult.txt для  
чтения
```

```
# Читаем записанное в файл число в текстовом виде.
```

```
# ct=fv.read()
```

```
# fv.close() # закрыли файл Rezult.txt
```

```
# .....  
# f = open('Rezult.txt', 'r') # Открыли файл для чтения  
# t=f.read() # Выводит все оставшиеся строки файла на  
экран  
# print(open('Fa.txt').readline()) # Выводит 1 строку файла  
на экран  
# t=f.readline() # Выводит следующую строку файла в пе-  
ременную и на экран  
# e=" Прочитали 3 = "  
# k=e+t  
# print (k)  
# t=f.read() # Выводит оставшиеся строки файла в пере-  
менную и на экран  
# e=" Прочитали = = "  
# k=e+t  
# print (k)
```

Подключение библиотек

```
# import sys # Подключили основные библиотеки  
# import shutil # Подключили модуль копирования файлов  
# import math # Подключили математич модуль  
# import time # Подключили модуль времени  
Функции времени  
# x=time.time() # Нашли число секунд из системы компью-  
тера
```

```
# 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
# .....
# ..... Демонстрация № 2 .....
# u=" Пишем слово, извлекаем из слова буквы и пишем
их "
# print (u)
# Sl="Python"
# print (Sl)
# print (Sl[0])
# y=Sl[1]
# print (y)
# u1=" Число символов в строке = "
# x=len(Sl)
# xl=str(x) # Преобразуем число в строку
# u=u1+xl
# print (u)
# u1=" Выведем код третьего символа в строке = "
# y=Sl[2] # Первый символ строки имеет порядковый номер ноль.
# x2=str(y) # Преобразуем число в строку
```


z=ord(y) # Y – Код (число) символа – возвращает символ

x3=str(z) # Преобразуем число в строку

u=u1+x2+" "+x3 # Сложим символы в новую строку..

print (u)

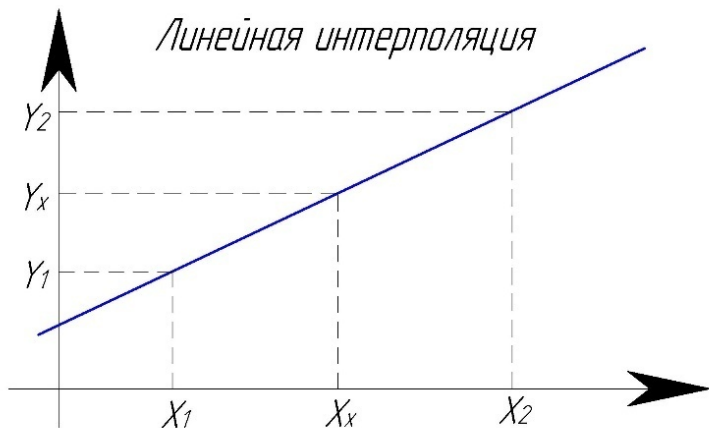
print (uu)

input() # Ожидание нажима Ентер

..... Конец демонстрационной программы № 2

Общая геометрия

Линейная интерполяция



Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль
```

```
# Эти первые две строки листинга определяют таблицу кодировки символов
```

и подключают библиотеку для математических расчетов.

Эти две строки точно без изменения стоят во всех листингах программ.

uu=" " # Далее идет вывод на экран заголовка программы.

```
print (uu)
```

```
print (uu)
```

u=" Расчет линейной интерполяции "

```
print (u)
```

```
print (uu)
```

Далее вводим в программу исходные данные для расчета.

```
print (uu)
```

u=" Введите X1 "

```
print (u)
```

```
print (uu)
```

```
x1=0.00000000
```

```
x1=input( ) # Вводим число
```

```
x1=float(x1) # Принудительно в вещественное число
```

```
print (uu)
```

u=" Введите X2 "

```
print (u)
```

```
print (uu)
```

```
x2=0.00000000
```

```
x2=input( ) # Вводим число
```

```
x2=float(x2) # Принудительно в вещественное число
```

```
print (uu)
u=" Введите Xx "
print (u)
print (uu)
xx=0.00000000
xx=input( ) # Вводим число
xx=float(xx) # Принудительно в вещественное число
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Y1 "
print (u)
print (uu)
y1=0.00000000
y1=input( ) # Вводим число
y1=float(y1) # Принудительно в вещественное число
print (uu)
u=" Введите Y2 "
print (u)
print (uu)
y2=0.00000000
y2=input( ) # Вводим число
y2=float(y2) # Принудительно в вещественное число
# Далее идет расчет по формуле..
yx=((y2-y1)*(xx-x1))/(x2-x1))+y1
```

```

# Далее выводится на экран результат расчета
u=" ..... "
print (u)
print (u)
print (uu)
print (uu)
u=" X1 = ", x1," Y1 = ",y1," "
print (u)
print (uu)
u=" X2 = ", x2," Y2 = ",y2," "
print (u)
print (uu)
u=" Xx = ", xx," Yx = ",yx," "
print (u)
print (uu)
u=" ..... "
print (u)
print (u)
print (uu)
print (uu)
# Pii=math.pi # Вытащили число " Пи "
input( ) # Ожидание нажима Enter
# Далее делаем еще один расчет на тех же исходных дан-
ных и выводим результаты.
Pii=math.pi # Вытащили число " Пи "
cc=((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1))

```

```

c=math.sqrt(cc) # Квадратный корень из " cc "
cs=str(c) # Преобразуем число в строку
u=" Длина линии x1;y1 – x2;y2 = "+cs
print (uu)
print (u)
a=(y2-y1)/(x2-x1)
aur=math.atan(a) # Угол А в радианах через арс тангенс
au=aur*180/Pii # Угол А в градусах
aus=str(au) # Преобразуем число в строку
u=" Угол линии x1;y1 – x2;y2 с осью Х = "+aus+" градус "
print (uu)
print (u)
print (uu)
print (uu)
input( ) # Ожидание нажима Ентер
# ..... Конец листинга программы .....

```

Система уравнений

Листинг программы:

```

# -*- coding: cp1251 -*-
import sys # Подключили основные библиотеки
import math # Подключили математич модуль
uu=" "
print (uu)
print (uu)

```

```
print (uu)
u=" Расчет системы уравнений первой степени "
print (u)
u=" с тремя неизвестными "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Первое уравнение "
print (u)
print (uu)
u="  $A * X + B * Y + C * Z = D$  "
print (u)
print (uu)
A=0.0000
u=" Введите Коэф A "
print (u)
A=input( )
A=float(A) # Принудительно в вещественное число
B=0.0000
u=" Введите Коэф B "
print (u)
B=input( )
B=float(B) # Принудительно в вещественное число
C=0.0000
```

u=" Введите Коэф С "

print (u)

C=input()

C=float(C) # Принудительно в вещественное число

D=0.0000

u=" Введите Коэф D "

print (u)

D=input()

D=float(D) # Принудительно в вещественное число

print (uu)

u=" Второе уравнение "

print (u)

print (uu)

u=" $M \cdot X + N \cdot Y + F \cdot Z = T$ "

print (u)

print (uu)

M=0.0000

u=" Введите Коэф M "

print (u)

M=input()

M=float(M) # Принудительно в вещественное число

N=0.0000

u=" Введите Коэф N "

print (u)

N=input()

N=float(N) # Принудительно в вещественное число

F=0.0000

u=" Введите Коэф F "

print (u)

F=input()

F=float(F) # Принудительно в вещественное число

T=0.0000

u=" Введите Коэф F "

print (u)

T=input()

T=float(T) # Принудительно в вещественное число

print (uu)

u=" Третье уравнение "

print (u)

print (uu)

u=" $Q \cdot X + K \cdot Y + E \cdot Z = V$ "

print (u)

print (uu)

Q=0.0000

u=" Введите Коэф Q "

print (u)

Q=input()

Q=float(Q) # Принудительно в вещественное число

K=0.0000

u=" Введите Коэф K "

print (u)

```
K=input( )
K=float(K) # Принудительно в вещественное число
E=0.0000
u=" Введите Коэф E "
print (u)
E=input( )
E=float(E) # Принудительно в вещественное число
V=0.0000
u=" Введите Коэф V "
print (u)
V=input( )
V=float(V) # Принудительно в вещественное число
aa=str(A) # Преобразуем число в строку
bb=str(B) # Преобразуем число в строку
cc=str(C)
dd=str(D)
nn=str(N)
mm=str(M)
ff=str(F) # Преобразуем число в строку
tt=str(T) # Преобразуем число в строку
qq=str(Q)
kk=str(K)
ee=str(E)
vv=str(V)
print (uu)
u=" ..... "
```

```

print (u)
print (uu)
print (uu)
u=" "+aa+" * X + "+bb+" * Y + "+cc+" * Z = "+dd
print (u)
print (uu)
print (uu)
u=" "+mm+" * X + "+nn+" * Y + "+ff+" * Z = "+tt
print (u)
print (uu)
print (uu)
u=" "+qq+" * X + "+kk+" * Y + "+ee+" * Z = "+vv
print (u)
print (uu)
print (uu)
u=" ..... "
print (u)
print (uu)
j=((A*N*E+B*F*Q+C*M*K-C*N*Q)-A*F*K)-B*M*E
jx=((D*N*E+B*F*V+C*T*K-C*N*V)-D*F*K)-B*T*E
jy=((A*T*E+D*F*Q+C*M*V-C*T*Q)-A*F*V)-D*M*E
jz=((A*N*V+B*T*Q+D*M*K-D*N*Q)-A*T*K)-B*M*V
if j ==0:
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Уравнение не имеет решений "
```

```
print (u)
print (uu)
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
else:
# Далее Сдвиг – четыре пробела в начале каждой строки
X=jx/j
Y=jy/j
Z=jz/j
xx=str(X) # Преобразуем число в строку
yy=str(Y) # Преобразуем число в строку
zz=str(Z) # Преобразуем число в строку
print (uu)
u=" X = "+xx
print (u)
print (uu)
u=" Y = "+yy
print (u)
print (uu)
u=" Z = "+zz
print (u)
print (uu)
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
input( ) # Ожидание нажима Ентер
# ..... .....
```

```
print (uu)
print (uu)
# .....
# ..... Конец листинга программы ..... ..
```

Квадратное уравнение

Листинг программы:

```
# -*- coding: cp1251 -*-
import sys # Подключили основные библиотеки
import math # Подключили математич модуль
uu=" "
print (uu)
print (uu)
print (uu)
u=" Расчет Квадратного уравнения "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u="  $A * X * X + B * X + C = 0$  "
print (u)
print (uu)
A=0.0000
u=" Введите Коэф А "
```

```

print (u)
A=input( )
A=float(A) # Принудительно в вещественное число
B=0.0000
u=" Введите Коэф В "
print (u)
B=input( )
B=float(B) # Принудительно в вещественное число
C=0.0000
u=" Введите Коэф С "
print (u)
C=input( )
C=float(C) # Принудительно в вещественное число
aa=str(A) # Преобразуем число в строку
bb=str(B) # Преобразуем число в строку
cc=str(C)
print (uu)
u=" ..... "
print (u)
print (uu)
print (uu)
u=" "+aa+" * X * X + "+bb+" * X + "+cc+" = O "
print (u)
print (uu)
print (uu)
u=" ..... "

```

print (u)

print (uu)

input() # Ожидание нажима Ентер

$D=(B*B)-(4*A*C)$

xx=str(D) # Преобразуем число в строку

print (uu)

u=" Дискриминант = "+xx

print (u)

print (uu)

if D<0:

Далее Сдвиг – четыре пробела в начале каждой строки

print (uu)

u=" Уравнение не имеет решений "

print (u)

print (uu)

Далее Конец Сдвига – четыре пробела в начале каждой строки

if D==0:

Далее Сдвиг – четыре пробела в начале каждой строки

$x=(-1)*B/(2*A)$

xx=str(x) # Преобразуем число в строку

print (uu)

u=" Уравнение имеет один корень = "+xx

print (u)

print (uu)

Далее Конец Сдвига – четыре пробела в начале каждой строки

строки

```
if D>0:
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
z=math.sqrt(D)
```

```
x=(-1)*B+z/(2*A)
```

```
x1=(-1)*B-z/(2*A)
```

```
xx=str(x) # Преобразуем число в строку
```

```
xx1=str(x1) # Преобразуем число в строку
```

```
print (uu)
```

```
u=" Уравнение имеет первый корень = "+xx
```

```
print (u)
```

```
print (uu)
```

```
u=" Уравнение имеет второй корень = "+xx1
```

```
print (u)
```

```
print (uu)
```

```
print (uu)
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

строки

```
u=" ..... "
```

```
print (u)
```

```
print (uu)
```

```
input( ) # Ожидание нажима Enter
```

```
# ..... 
```

```
print (uu)
```

```
print (uu)
```

```
# ..... Конец листинга программы ..... 
```

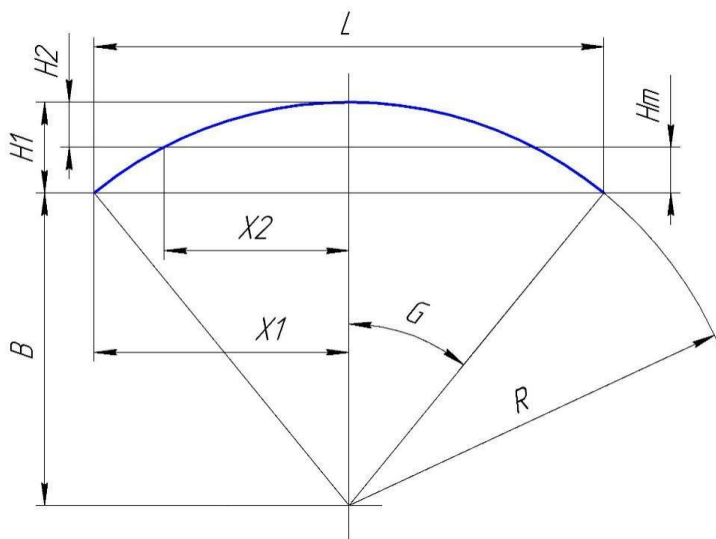

Координаты точек на радиусной кривой

Программа бывает необходима при построении сегмента с радиусом

большой величины – который построить непосредственной разметкой

сложно или невозможно из за мешающих препятствий.

Результаты расчета записываются в файл 'Rezult.txt'.



Листинг программы.

```

# -*- coding: cp1251 -*-
import math # Подключили математич модуль
import sys # Подключили основные библиотеки
# .....
Pii=math.pi # Вытащили число " Пи "
f = open('Rezult.txt', 'w') # Открыли файл для записи
# Записываем числа в текстовом виде
uu=" "
u=uu+"\n" # Добавим код перевода строки
f.write(u)
u1=" Расчет координат точек на радиусной кривой "
u=u1+"\n"
f.write(u)
uu=" "+" \n"
f.write(uu)
uuu=" ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,, "+"\n"
f.write(uuu)
f.write(uu)
print (uu)
u=" Построение большого радиуса методом подъема "
print (u)
print (uu)
u=" Размер по горизонтали отсчитывается от центра хор-
ды "
print (u)
u=" от точки X2 на хорде проводим перпендикуляр Nm

```

расчитанной величины. "

```
print (u)
```

u=" При вводе размера по горизонту = нулю – выход из программы "

```
print (u)
```

```
print (uu)
```

```
input( ) # Ожидание нажима Enter
```

```
print (uu)
```

```
a=0.00000000
```

```
y=0.00000000
```

```
ug=0.00000000
```

```
ugg=0.00000000
```

```
R=0.00000000
```

```
hm=0.00000000
```

```
ht= 0.00000000
```

```
u=" Введите радиус "
```

```
print (u)
```

```
print (uu)
```

```
R=input( ) # Вводим число
```

```
R=float(R) # Принудительно в вещественное число
```

```
u=" Радиус заданной кривой = "
```

```
ss=str(R) # Преобразуем число в строку
```

```
Rad=u+ss
```

```
u=Rad+"\n"
```

```
f.write(u)
```

```
u=" Введите Хорду L "
```

```

print (u)
print (uu)
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Хорда максимальная заданная L = "
ss=str(a) # Преобразуем число в строку
Xord=u+ss
u=Xord+"\n"
f.write(u)
# Находим значения максимального прогиба при  $X^2 = 0 \dots$ 
x=((R*R)-(a*a/4))
b=R-(math.sqrt(x)) # Квадратный корень из " x "
# b – максимальный прогиб...
u=" Подъем максимальный в центре хорды = "
ss=str(b) # Преобразуем число в строку
Progi=u+ss
u=Progi+"\n"
f.write(u)
x=(a/2)/R
y=math.asin(x)
v=math.cos(y)
aur=2*y
au=aur*180/Pii # Угол A в градусах
Lx= 0.0000001 # <Начальное значение>
while Lx>0:
# Далее Сдвиг – четыре пробела в начале каждой строки

```

```

print (uu)
u=" Введите расстояние от центра хорды до перпендикуля-
ра X2 "
print (u)
print (uu)
Lx=input( ) # Вводим число
Lx=float(Lx)
u=" От центра хорды до точки X2 по оси X-X = "
ss=str(Lx) # Преобразуем число в строку
Xord=u+ss
u=Xord+"\n"
f.write(uu)
f.write(u)
x=(R*R)-(Lx*Lx)
z=math.sqrt(x)
y=R-z # Прогиб при хорде = Lx*2
ht=b-y # Расчитали величину подъема
u=" На расстоянии от центра = "
ss=str(Lx) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Величина подъема ( перпендикуляра ) Nm = "
ss=str(ht) # Преобразуем число в строку
u=u+ss
print (u)

```

```
f.write(u)
f.write(uu)
print (uu)
u="
```

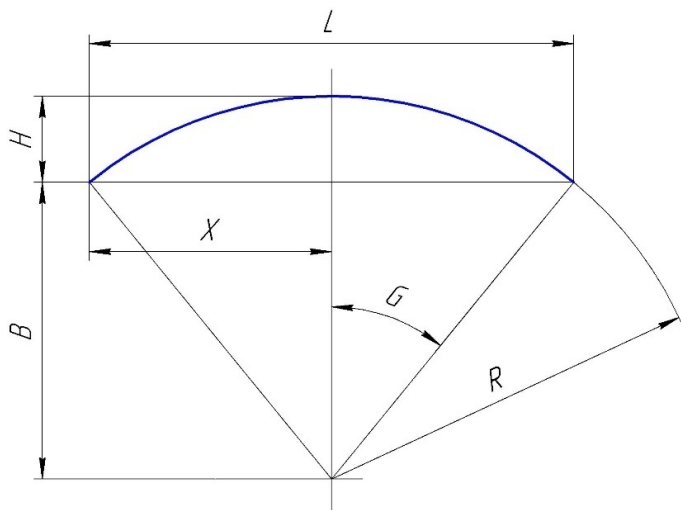
=====

```
"
```

```
    print (u)
    f.write(u)
    print (uu)
    input( ) # Ожидание нажима Enter
    # Далее Конец Сдвига – четыре пробела в начале каждой
строки
```

```
    print (uu)
    input( ) # Ожидание нажима Enter
    print (uu)
    # .....
    f.write(uu)
    f.write(uuu)
    f.close() # закрыли файл
    # .....
    # .....
    u=" ..... Конец программы ..... "
    print (u)
    print (uu)
    input( ) # Ожидание нажима Enter
    # ..... Конец листинга программы .....
```

Хорда – Прогиб – Радиус



В этой программе объединены общим меню двенадцать небольших программ.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль
```

```
# Отступ – четыре пробела..
```

```
# Все проверено – 23-06-2014 г..
```

```
global Rad # Радиус – Объявили глобальную переменную
global Diam # Диаметр
global Xord # Хорда
global Progi # Прогиб
global UgSe # Угол раствора радиусов на хорду
global Dug # Длина дуги над хордой
global PlSe # Площадь сектора на хорду
global PlTr # Площадь Треугольника под хордой
global PlGo # Площадь Горбушки
u=" "
u=" Расчет элементов Хорда – Прогиб – Радиус "
print (uu)
print (u)
print (uu)
u=" ..... "
print (u)
Pii=0.000000
Pii=math.pi # Вытащили число " Пи "
Kvz=0 # Флаг – Если =1 то работа уже выполнена.
print (uu)
u=" 1 – Расчет по Хорде – Радиусу "
print (u)
print (uu)
u=" 2 – Расчет по Хорде и Углу раствора "
print (u)
print (uu)
```


u=" 3 – Расчет координат Радиусной кривой "

print (u)

print (uu)

u=" 4 – Расчет по Хорде и длине Дуги "

print (u)

print (uu)

u=" 5 – Расчет по Хорде и Прогибу "

print (u)

print (uu)

u=" 6 – Расчет по Радиусу и Углу раствора "

print (u)

print (uu)

u=" 7 – Расчет по Радиусу и Прогибу "

print (u)

print (uu)

u=" 8 – Расчет по Радиусу и длине Дуги "

print (u)

print (uu)

u=" 9 – Расчет по Прогибу и длине Дуги "

print (u)

print (uu)

u=" 10 – Расчет по Прогибу и Углу раствора "

print (u)

print (uu)

u=" 11 – Расчет по Углу раствора и длине Дуги "

print (u)

```

print (uu)
u=" 12 – Расчет Правильного многогранника "
print (u)
print (uu)
u=" ..... "
print (u)
q=0
q=input( ) # Вводим число
q=float(q) # Принудительно в вещественное число
# .....
if q==1: # Расчет по Хорде – Радиусу
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Расчет по Хорде – Радиусу "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите Радиус "
print (u)
print (uu)

```

R=0.00000000

R=input() # Вводим число

R=float(R) # Принудительно в вещественное число

x=0.00000000

y=0.00000000

v=0.00000000

Sk=0.00000000

St=0.00000000

S=0.000000000

x=((R*R)-(a*a/4))

b=R-(math.sqrt(x)) # Квадратный корень из " x "

x=(a/2)/R

y=math.asin(x)

v=math.cos(y)

aur=2*y

au=aur*180/Pii # Угол A в градусах

Kvz=1 # Коэф – Расчет уже выполнен

Далее Конец Сдвига – четыре пробела в начале каждой

строки

else:

Далее Сдвиг – четыре пробела в начале каждой строки

Kvz=0

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q==2: # Расчет по Хорде и углу раствора

Далее Сдвиг – четыре пробела в начале каждой строки

```
print (uu)
```

```
print (uu)
```

```
u=" Расчет по Хорде и углу раствора "
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите Хорду "
```

```
print (u)
```

```
print (uu)
```

```
a=0.00000000
```

```
a=input( ) # Вводим число
```

```
a=float(a) # Принудительно в вещественное число
```

```
u=" Введите Угол раствора в градусах "
```

```
print (u)
```

```
print (uu)
```

```
au=0.00000000
```

```
au=input( ) # Вводим число
```

```
au=float(au) # Принудительно в вещественное число
```

```
x=0.00000000
```

```
aur=0.00000000
```

```
v=0.00000000
```

```
y=0.00000000
```

```
Sk=0.00000000
```

```
St=0.00000000
```

```
S=0.0000000000
```

```
aur=au*Pii/180
```



```
print (uu)
u=" Хорда = "
ss=str(a) # Преобразуем число в строку
u=u+ss
print (u)
Xord=u
print (uu)
u=" Стрела прогиба = "
ss=str(b) # Преобразуем число в строку
u=u+ss
print (u)
Progi=u
print (uu)
u=" Радиус = "
ss=str(R) # Преобразуем число в строку
u=u+ss
print (u)
Rad=u
print (uu)
u=" Диаметр = "
ss=str(D) # Преобразуем число в строку
u=u+ss
print (u)
Diam=u
print (uu)
u=" Угол раствора хорды = "
```

ss=str(au) # Преобразуем число в строку

u=u+ss

print (u)

UgSe=u

print (uu)

u=" Длина дуги над хордой = "

ss=str(L) # Преобразуем число в строку

u=u+ss

print (u)

Dug=u

print (uu)

u=" Площадь сектора = "

ss=str(Sk) # Преобразуем число в строку

u=u+ss

print (u)

PlSe=u

print (uu)

u=" Площадь треугольника под горбушкой = "

ss=str(St) # Преобразуем число в строку

u=u+ss

print (u)

PlTr=u

print (uu)

u=" Площадь горбушки = "

ss=str(S) # Преобразуем число в строку

u=u+ss

```

print (u)
PlGo=u
print (uu)
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
# .....
if q==3: # Координаты радиусной прямой
# Далее Сдвиг – четыре пробела в начале каждой строки
q=13 # Переброс в конец программы
# .....
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
if q==4: # Расчет по Хорде и длине Дуги
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Расчет по Хорде и и длине Дуги "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите длину Дуги "

```



```
print (u)
print (uu)
L=0.000000000
L=input( ) # Вводим число
L=float(L) # Принудительно в вещественное число
R=0.000000000
dx=a/200000
rt=(a/2)+dx # Начальный радиус расчета
aa=a/2 # Половина хорды
dl=Pii*rt
while dl>L: # Расчет по Хорде и длине Дуги
# Далее Сдвиг – восемь пробелов в начале каждой строки
rt=rt+dx # Текущий радиус
x=aa/rt
y=2*(math.asin(x)) # Угол А в радианах через арс синус
dl=rt*y # Текущая длина дуги
# Далее Сдвиг – четыре пробела в начале каждой строки
yg=y*180/Pii # Угол " y " в градусах
au=yg
R=rt # Нашли радиус
y=(R*R)-(aa*aa)
v=math.sqrt(y) # Квадратный корень из " y "
b=R-v # Нашли прогиб
D=R+v
Sk=Pii*D*D*yg/(4*360) # Площадь сектора круга с углом
```

```

St=aa*v # Площадь треугольника в секторе
S = Sk-St # Площадь горбушки
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
else:
# Далее Сдвиг – четыре пробела в начале каждой строки
Kvz=0
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
# .....
if q==5: # Расчет по Хорде – Прогибу
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Хорде – Прогибу "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Хорда – Принудительно в вещественное число
u=" Введите Стрелу Прогиба "

```

```
print (uu)
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Стрела прогиба – Принудительно в вещественное число
x=0.00000000
y=0.00000000
v=0.00000000
R=0.00000000
Sk=0.00000000
St=0.00000000
S=0.0000000000
Ex=1.000000
k=a/2
Rt=k+(k/100000)
Rd= k/100000
t=k/100000000
# Уточнение подбором
while Ex > t:
# Далее Сдвиг – восемь пробелов в начале каждой строки
Rt=Rt+Rd
x=((Rt*Rt)-(k*k))
c=math.sqrt(x) # Квадратный корень из " x "
Ex=(Rt-c)-b
```

```
# E=math.abs(x)
```

```
# Конец подбора ..... .....
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
R=Rt
```

```
D=R+R
```

```
x=k/Rt
```

```
sur=math.asin(x) # Угол A в радианах
```

```
su=sur*180/Pii # Угол A в градусах
```

```
au=2*su
```

```
yg=au
```

```
L=(R+R)*Pii*au/360
```

```
# ..... .....
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой строки
```

```
if q==6: # Расчет по Радиусу и Углу раствора
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
print (uu)
```

```
u=" Расчет по Радиусу и Углу раствора "
```

```
print (u)
```

```
print (uu)
```

```
u=" ..... ....."
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите Радиус "
```

```
print (u)
```

```
print (uu)
```

R=0.00000000

R=input() # Вводим число

R=float(R) # Радиус – Принудительно в вещественное
число

u=" Введите Угол раствора в градусах "

print (uu)

print (u)

print (uu)

yg=0.00000000

yg=input() # Вводим число

yg=float(yg) # Угол раствора – Принудительно в веще-
ственное число

yr=yg*Pii/180 # Угол раствора В радианах

x=0.00000000

y=0.00000000

v=0.00000000

Sk=0.00000000

St=0.00000000

S=0.0000000000

Ex=1.000000

x= math.cos(yr/2)

y=R*x

b=R-y

x=(R*R)-(y*y)

c=math.sqrt(x) # Квадратный корень из " x "

a=c+c

au=yg

D=R+R

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

..... .

if q==7: # Расчет по Радиусу и Прогибу

Далее Сдвиг – четыре пробела в начале каждой 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.