

# ВИЗУАЛЬНАЯ ОТЛАДКА PYTHON-ПРОГРАММ ДЛЯ ПОСТРОЕНИЯ ТРАСС КОСМИЧЕСКИХ АППАРАТОВ

**Становых Владимир Дмитриевич,**  
ученик 9 класса, дружный кружок «Юный физик –  
умелые руки», Благотворительный фонд  
«Образование+» при МБОУ «Гимназия №5» города  
Королёва (Юбилейный) Московской области,  
тел. 8-985-646-31-75; rus1k170206@gmail.com

Научный руководитель Лебедев Владимир Валентинович,  
доктор технических наук, Центр физико-математического образования  
Московского авиационного института (НИУ), Заслуженный деятель науки и  
техники Московской области,  
руководитель школьного кружка «Юный физик – умелые руки»,  
Тел. 8-903184-45-31, 8-925-717-14-37, Lebedev\_v\_2010@mail.ru, личный  
сайт CFMO.UCOZ.RU



**Работа проводится при поддержке Благотворительного фонда «Образование+»**

# **Общая характеристика работы**

**Решаемая проблема:**

**дистанционное зондирование поверхности Земли и Океана в новых промышленных районах (Север)**

**Цель работы:**

**обеспечение информационных потоков новых районов с освоенной территорией**

**Актуальность: перемещение производства на Север**

**Новизна: применение и доработка известных орбит КА**

**Практическая значимость:**

**ускорение промышленного освоения перспективных территорий**

**Решаемая задача: размещение КА в зените над новой промышленной территорией**

# Стратегическое обоснование актуальности



**УКАЗ**

**ПРЕЗИДЕНТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**О Стратегии научно-технологического развития  
Российской Федерации**

В соответствии со статьей 18<sup>1</sup> Федерального закона от 28 июня 2014 г. № 172-ФЗ «О стратегическом планировании в Российской Федерации» постановляю:

1. Утвердить прилагаемую Стратегию научно-технологического развития Российской Федерации.

**ж) необходимость эффективного освоения и использования пространства, в том числе путем преодоления диспропорций в социально-экономическом развитии территории страны, а также укрепление позиций России в области экономического, научного и военного освоения космического и воздушного пространства, Мирового океана, Арктики и Антарктики.**



**Большие  
ВЫЗОВЫ  
(15ж):**

# Анализ литературы

1. Меньшиков В.А., Перминов А.Н., Урлич Ю.М. Глобальные проблемы человечества и космос. – М.: «Изд. МАКД», 2010. – 570 с.

2. Папиашвили Э.Д. 4-й курс, РУДН. Формирование орбиты космического аппарата для дистанционного зондирования вдоль земной параллели / Научно-методическое издание: Материалы XXXI конференции "Современные информационные технологии в образовании". Ред. группа: Алексеев М.Ю. и др. - Фонд новых технологий в образовании "БАЙТИК", ИТО-Троицк-Москва, 2-3 июля 2020. - 572 с. - ISBN 978-5-89513-468-9. - С.534-536.- Эл. ресурс: <https://lk-ito.bytic.ru/uploads/files/materials.pdf>

3. Усатый И.В. Компьютерное моделирование орбиты космического аппарата с необычной трассой / Д25 ХХ Школьные Харитоновские чтения. Межрегиональная олимпиада школьников "Будущие исследователи - будущее науки". Тезисы. Составили Константинова О.В., Селина М.Д., Яшнова В.В. - Саров: ФГУП "РФЯЦ-ВНИИЭФ", 2020. - 255 с. - Ил. - Секция 2 "Информатика". - С.54-56. - УДК 016. - ББК 72. - Д25.

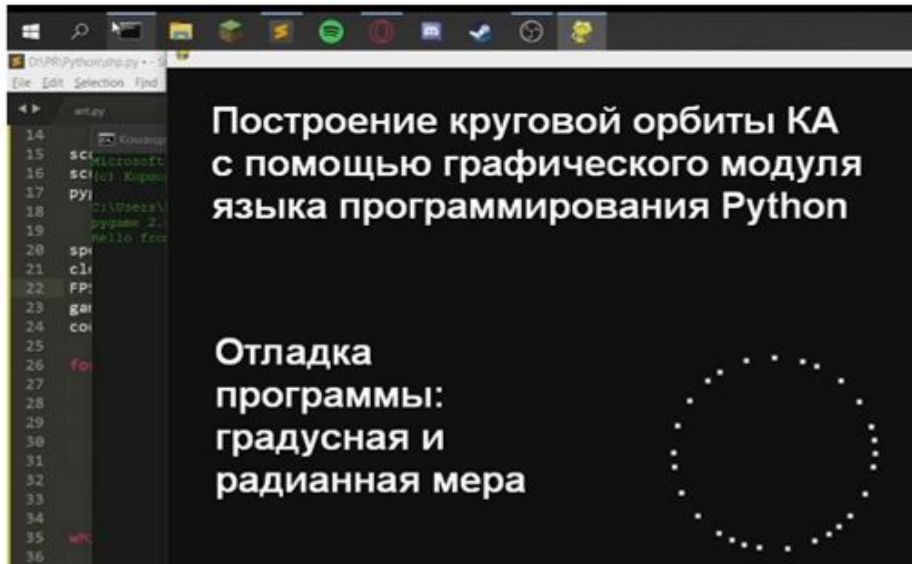
4. Матрица поворота. – Электронный ресурс:  
<https://ru.wikipedia.org>

# Первый шаг отладки программы – моделирование круговой орбиты КА

$$x = r \cdot \cos \varphi$$
$$y = r \cdot \sin \varphi$$

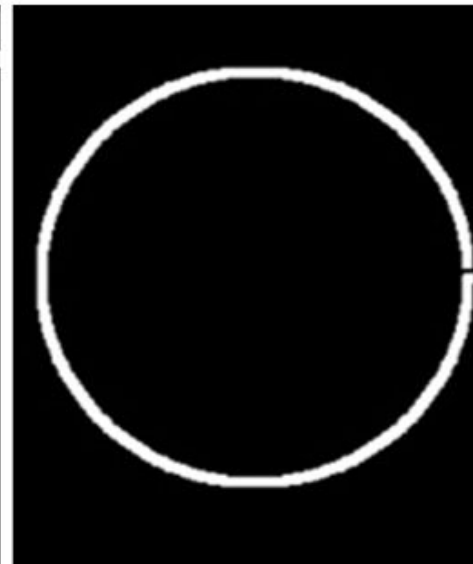
Задача – исправление программных ошибок (градусы и радианы и т.д.)

```
20 speed = 10
21 clock = pygame.time.Clock()
22 FPS = 30
23 game_run = True
24 count = 0
25
26 for i in range(100):
27     x = r * math.cos(i * 3.1416 / 180)
28     y = r * math.sin(i * 3.1416 / 180)
29     x_sp.append(x)
30     y_sp.append(y)
31
32
33
34
35 while game_run:
36     screen.fill((0, 0, 0))
37
38     clock.tick(FPS)
39     for event in pygame.event.get():
40         if event.type == pygame.QUIT:
41             game_run = False
42
43     if count > 360:
44         count = 1
45     <del>
46     count = 0
47
48     for i in range(count):
49         pygame.draw.rect(screen, [255, 255, 255], [x_sp[i] * 400, y_sp[i] * 400, 5, 5])
50
51     pygame.display.flip()
```



Построение круговой орбиты КА с помощью графического модуля языка программирования Python

Отладка программы: градусная и радианная мера



Шаг по углу равномерный, при отладке программы задан 1 градус, потом будет уменьшен для повышения точности

# Второй шаг отладки программы – моделирование эллиптической орбиты КА

$$r = r(\varphi) = \frac{p}{1 + \varepsilon \cdot \cos \varphi}$$

$$p = r_a (1 - \varepsilon)$$

$$r_a = R_3 + h_a$$

$$r_n = R_3 + h_n$$

```

1 import math
2 import time
3 import pygame
4
5 #функция
6 def find_x_y():
7     for t in range(360):
8         r = p / (1 + e * math.cos(2 * 3.14159 * t / 360))
9         x = r * math.cos(2 * 3.14159 * t / 360)
10        y = r * math.sin(2 * 3.14159 * t / 360)
11
12        x_sp.append(x)
13        y_sp.append(y)
14
15
16 pygame.init()
17 print
18 project_3
19 x_sp = []
20 y_sp = []
21 count = 0
22
23
24 #переменные pygame
25 Long = 1220
26 Height = 700
27 speed = 10
28 clock = pygame.time.Clock()
29 FPS = 50
30 game_run = True
31
32
33 #вывод pygame в display
34 pygame.display.set_mode((Long, Height))
35 screen = pygame.display.get_surface()
36 screen.fill((0, 0, 0))
    
```

Построение эллиптической орбиты КА с помощью графического модуля языка программирования Python

Наклонение  $62,8^{\circ}$  ;

высота апогея  $39448000$  м;

высота перигея  $933000$  м;

эксцентриситет  $0,725$

Реальная орбита, КА "Молния"

## Орбита КА «Молния»

наклонение

$i=62,8^{\circ}$

большая полуось

$a=26561$  км

эксцентриситет  $\varepsilon = 0,725$

высота перигея

$h_n=933$  км

высота апогея

$h_a=39448$  км

радиус Земли

$R_3=6371,210$  км

# Расчёт времени нахождения КА в моделируемых точках орбиты

$$S_{el} = \pi a b \quad T = 2\pi \sqrt{a^3 / \mu_3}$$

$$\mu_3 = 398600,4415 \text{ km}^3 / \text{s}^2 = 398600441500000 \text{ m}^3 / \text{s}^2$$

Пропорция  
(Э.Д.Папиашвили,  
РУДН, Космический  
факультет)

$$a = 0,5 (h_a + h_p + 2R_3)$$

$$b = \sqrt{a p} \quad p = r_a (1 - \varepsilon)$$

$$r_a = R_3 + h_a \quad R_3 = 6371210 \text{ м}$$

$$S_{kr} = 0,5 r^2 \Delta \varphi \quad t_i = 0,5 r^2 \Delta \varphi T / S_{el}$$

**Второй закон Кеплера (закон площадей)**

# Третий шаг отладки программы – моделирование времени движения КА

~12 часов

Время от перигея до текущей точки (накопление), необходимо для расчёта угла поворота Земли

0

Апогей

Перигей

Время нахождения КА в моделируемых точках эллиптической орбиты ( $e=0,95$ )

Эксцентриситет эллиптической орбиты увеличен до  $e=0,95$ , вместо  $0,725$ , как у КА "Молния".  
Цель увеличения - повышение наглядности результата работы программы при отладке.

```
D:\PR\Python\anoni\pr_co
File Edit Selection Find
shp.py
21
22
23
24
25 #перем Mi
26 summ_t (c
27 sp_for c:
28 time_spu
29 x_sp = He
30 y_sp = На
31 Number Вы
32 igr = Эк
33 igrd = Пе
34 ha = i
35 hp = i
36 RZ = €
37 e = fl
38 #Вычис
39 muz =
40 omz =
41 ra = F
42 rp = F
43 p = ra
44 a = (h
45 b = (a
46 c = a
47 S = Nu
48
49 T_sek
50 T_hour
51
52 find()
```



# Четвёртый шаг отладки программы – поворот орбиты КА на угол наклонения

$$\begin{cases} x_1 = r \cos \varphi \cos i & ; \\ y_1 = r \sin \varphi & ; \\ z_1 = -r \cos \varphi \sin i & . \end{cases} \quad \text{Источник (ссылка):}$$

Матрица поворота в трёхмерном пространстве

Любое вращение в трёхмерном пространстве может быть представлено композицией соответствующей матрицы, равная произведению

Матрицами вращения вокруг оси декартовой системы координат на углы

• Вращение вокруг оси  $x$ :

$$M_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}.$$

• Вращение вокруг оси  $y$ :

$$M_y(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}.$$

• Вращение вокруг оси  $z$ :

$$M_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$\|x_1 \ y_1 \ z_1\| = \|x \ y \ z\| \cdot M_y(i)$$

$$\begin{cases} x_1 = x \cos i + 0 - z \sin i & ; \\ y_1 = 0 + y + 0 & ; \\ z_1 = -x \sin i + 0 + z \cos i & . \end{cases}$$

$$M_y(i) = \begin{vmatrix} \cos i & 0 & -\sin i \\ 0 & 1 & 0 \\ -\sin i & 0 & \cos i \end{vmatrix}$$

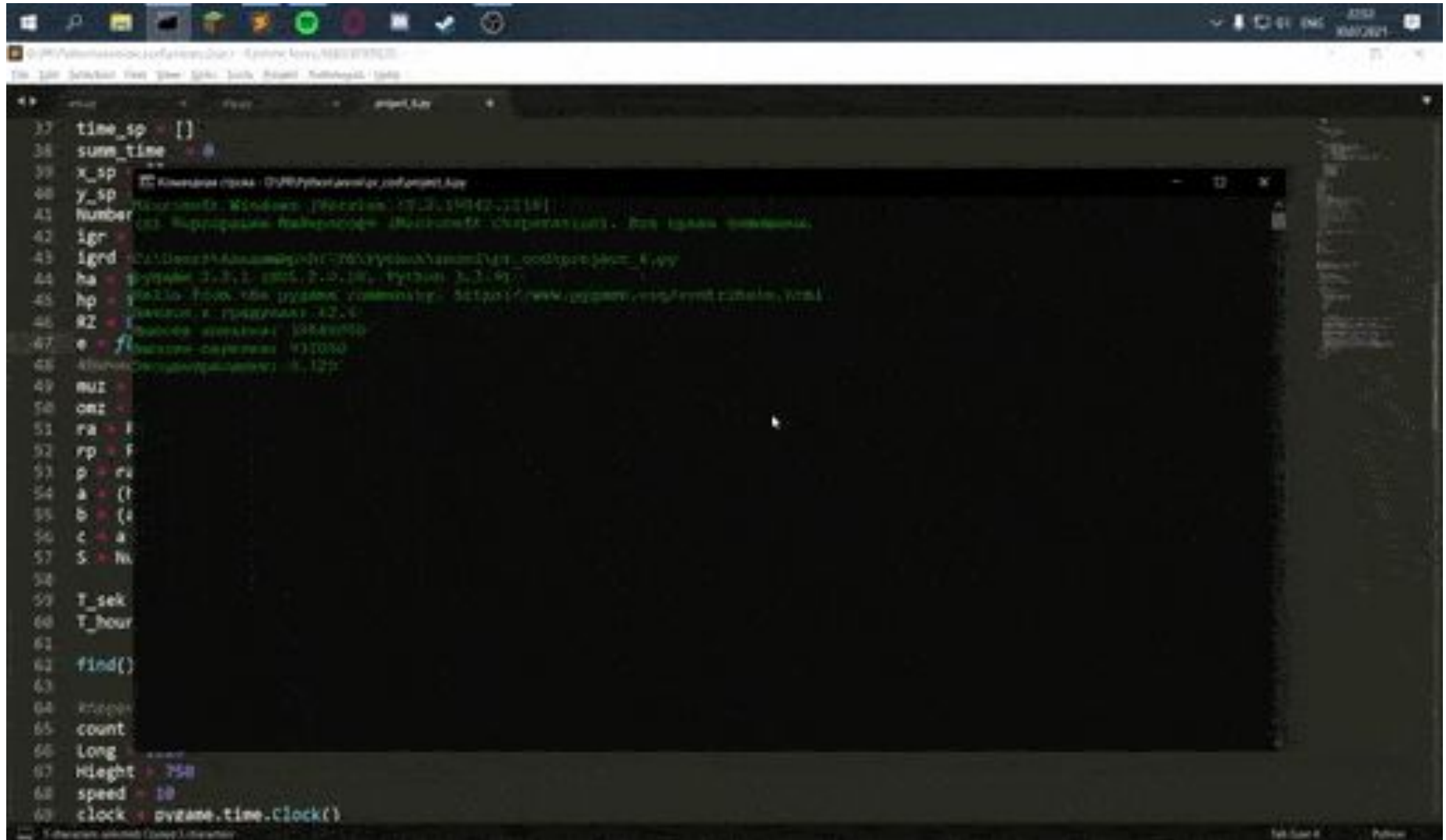
$$x = r \cdot \cos \varphi$$

$$y = r \cdot \sin \varphi$$

$$z = 0$$

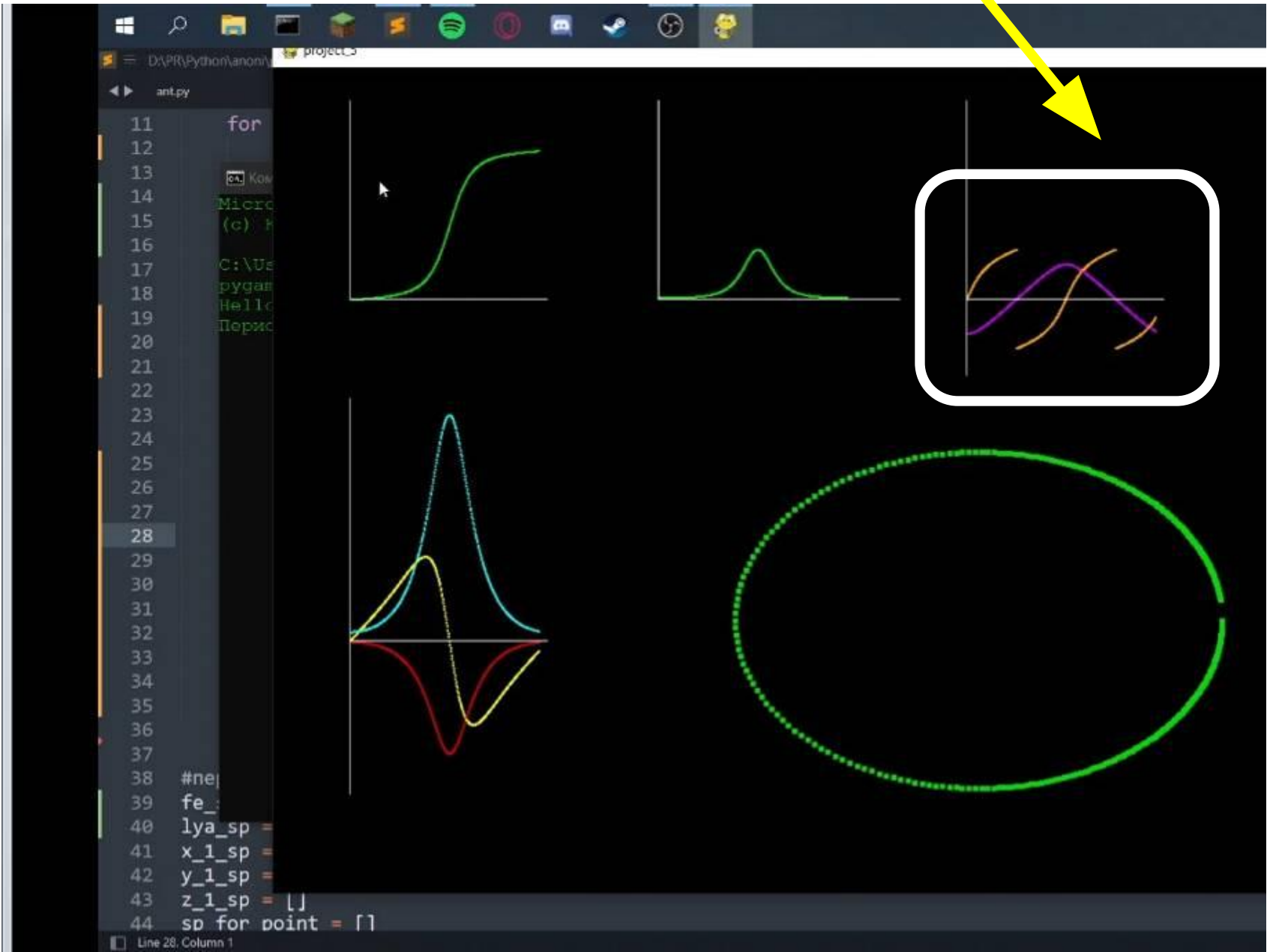
$$\|x_1 \ y_1 \ z_1\| = \|x \ y \ z\| \cdot \begin{vmatrix} \cos i & 0 & -\sin i \\ 0 & 1 & 0 \\ -\sin i & 0 & \cos i \end{vmatrix}$$

# Визуальное программирование времени движения КА по эллиптической орбите и наклона орбиты к плоскости Экватора



```
37 time_sp = []
38 sum_time = 0
39 X_sp = []
40 Y_sp = []
41 Number = 0
42 igr = 0
43 igrd = 0
44 ha = 0
45 hp = 0
46 RZ = 0
47 e = 0
48 mu = 0
49 mu1 = 0
50 om1 = 0
51 ra = 0
52 rp = 0
53 p = 0
54 a = 0
55 b = 0
56 c = 0
57 S = 0
58
59 T_sec = 0
60 T_hour = 0
61
62 find()
63
64 k = 0
65 count = 0
66 Long = 0
67 Hieght = 0
68 speed = 0
69 clock = pygame.time.Clock()
```

# Исправление ошибки в трассе (arctg)



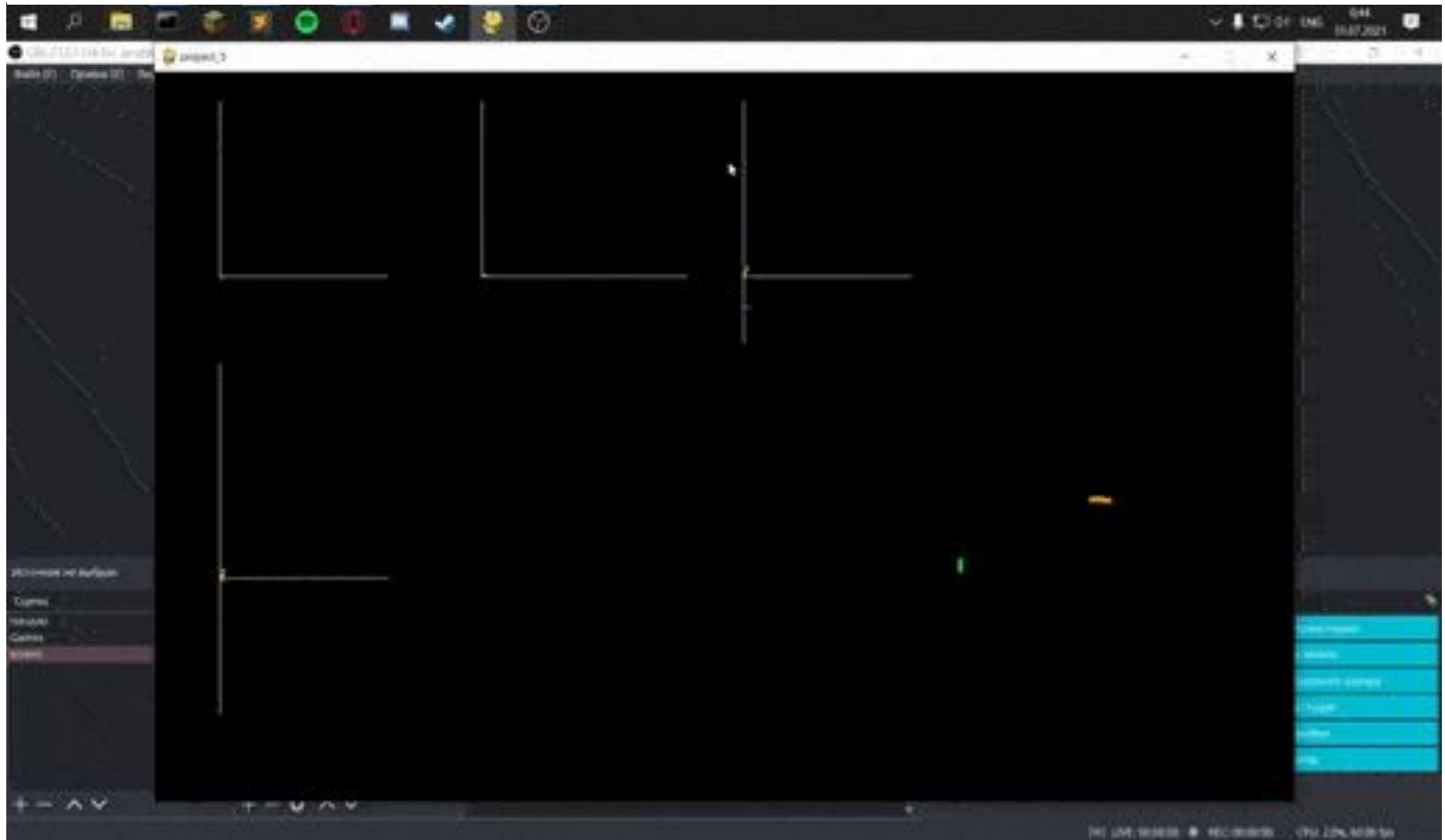
The image shows a code editor window with a Python script on the left and several plots on the right. A yellow arrow points to a plot with multiple overlapping curves, which is highlighted by a white rounded rectangle.

```
11 for
12
13
14   Micro
15   (c) K
16
17   C:\Us
18   pygan
19   Hello
20   Перис
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 #ne|
39 fe_
40 lya_sp =
41 x_1_sp =
42 y_1_sp =
43 z_1_sp = []
44 sp for point = []
```

The plots show various mathematical functions:

- Top-left: A green sigmoid curve.
- Top-right: A green bell-shaped curve.
- Middle-right (highlighted): A plot with multiple overlapping curves in purple, orange, and brown.
- Bottom-left: A plot with multiple overlapping curves in cyan, yellow, and red.
- Bottom-right: A large green oval shape.

# Не должно быть разрыва по географической долготе в трассе (arctg)



Условные операторы на знаки координат после вычисления arctg

# Ошибка в программе найдена очень быстро визуальным программированием

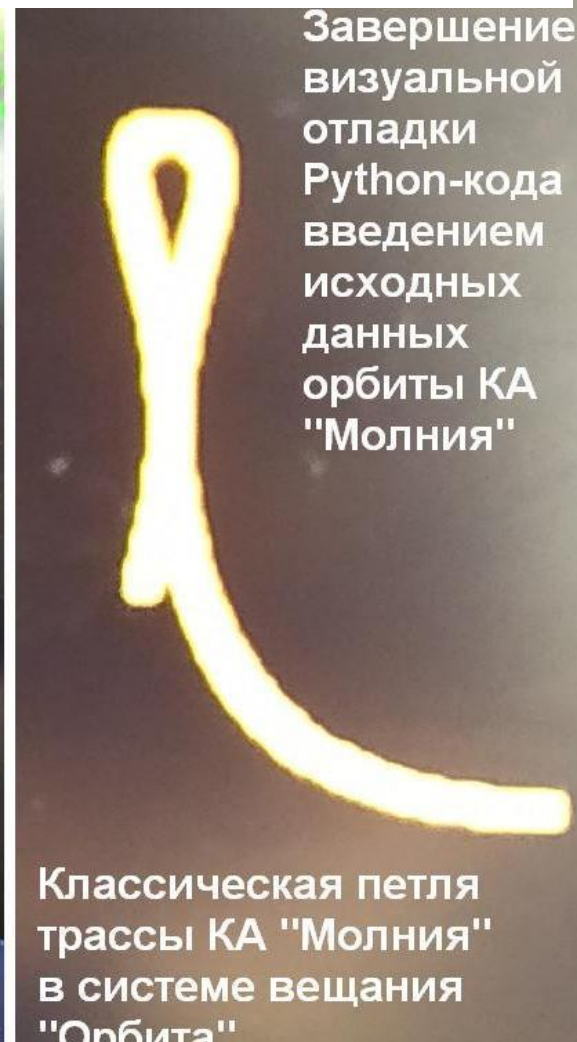
```
D:\PR\Python\anon\pr_cod\project_3.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ant.py x shp.py x project_3.py x
1 import math
2 import time
3 import pygame
4
5 #функции
6 def find_x_y():
7     for t in range(360):
8         r = p/1+e*math.cos(3.1416 * t/180)
9         x = r * math.cos(t*3.1416/180) / 500000
10        y = r * math.sin(t*3.1416/180) / 500000
11
12        x_sp.append(x)
13        y_sp.append(y)
14
15
16
17 #переменные
18 p = 45819210
19 e = 0.725
20 x_sp = []
21 y_sp = []
22 count = 0
23
24 #переменные pygame
25 Long = 1220
26 Hieght = 750
27 speed = 10
28 clock = pygame.time.Clock()
29 FPS = 50
30 game_run = True
31
32
33 #инфа о скрине и старт
34 pygame.init()
35 screen = pygame.display.set_mode((Long, Hieght))
36 screen.fill((0, 0, 0))
```

Программа на языке Python в процессе визуальной отладки

```
9 def
10
11 pygame 2.0.3 (SDL 2.0.1)
12 Hello from the pygame
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

Результат - числовой массив координат трассы КА "Молния". Не иллюстративен.



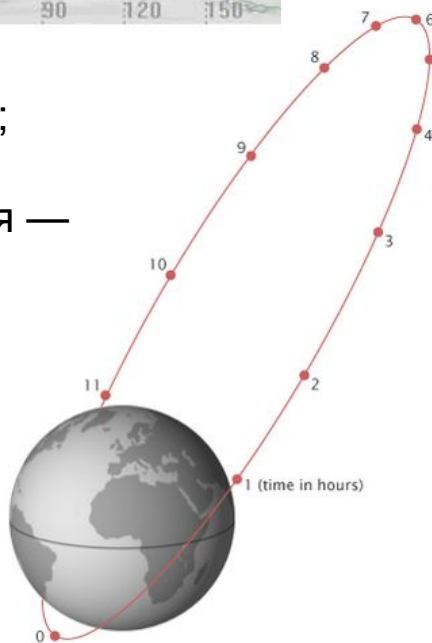
Не было суммирования времени движения КА от перигея, Земля на вращалась

# Верификация программы по трассе КА «Молния»

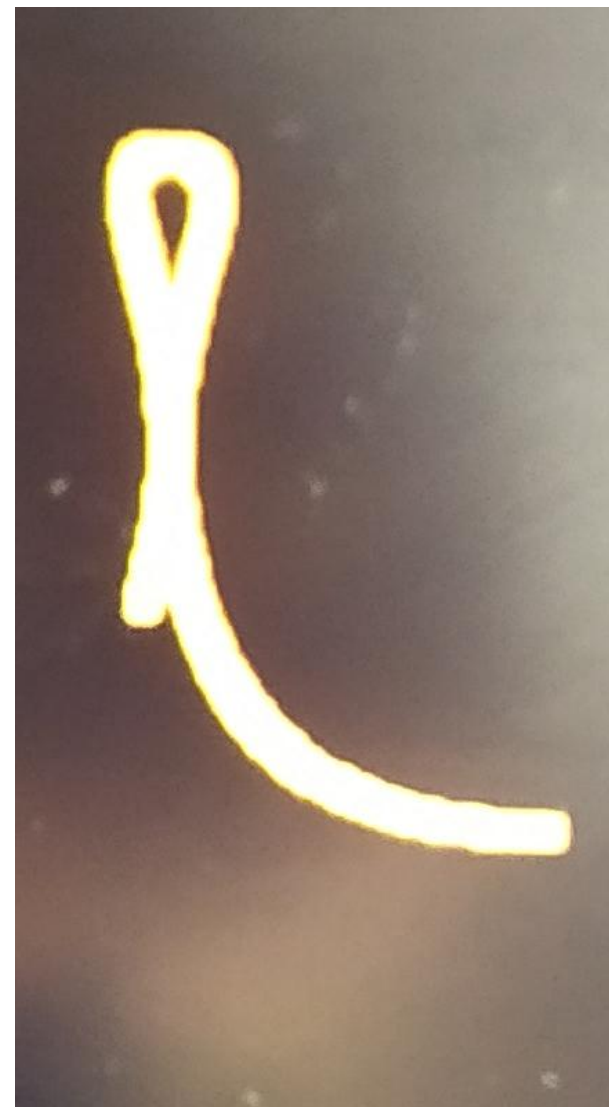


Наземная трасса КА «Молния»

аргумент широты перигея —  $280^\circ$ ;  
наклонение —  $62,8^\circ$ ;  
драконический период обращения —  
11 часов 57 мин. 45 сек.;  
высота — от 500 км в перигее до  
40 000 км в апогее.



Типичная орбита КА «Молния». Красными точками отмечено время движения спутника по орбите в часах



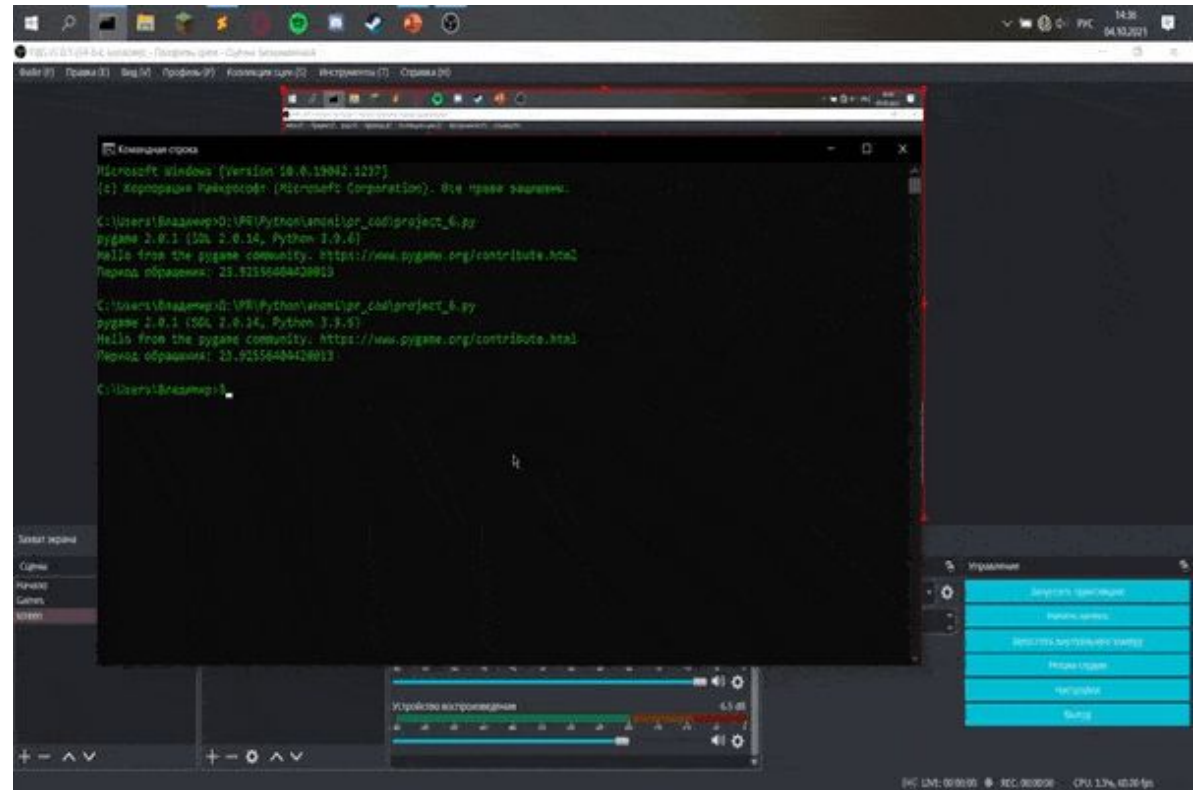
# Верификация программы по трассе группировок КА

## Орбита «Тундра»

## «Тундра» ( $e=0,25$ )



Наземная  
трасса КА  
«QZSS»



- период обращения — 23 часа 56 мин. 04 сек. (1 [Звёздный день](#));
- большая полуось ( ): 42 164 км;
- эксцентриситет ( ): от 0,25 до 0,4;
- высота в перигее: , где (средний радиус Земли) — от 18 900 до 25 240 км;
- высота в апогее: — от 46 330 до 52 660 км;
- наклонение ( ):  $62,15^{\circ}$ <sup>[4]</sup> —  $63,4^{\circ}$ .

# Верификация программы по трассе группировок КА «Тундра» (e=0,4)

The screenshot displays a Windows desktop environment. In the foreground, a Command Prompt window is open, showing the execution of a Python script. The output of the script is visible, including the Pygame version and a greeting message. The background features a software interface with a dark theme. On the right side, there is a vertical timeline or sequence of events, with a red vertical line indicating a specific point in time. Below the timeline, there are several control buttons, including 'Справка', 'Остановка', 'Восстановление', 'Панель задач', 'Настройка', and 'Выход'. At the bottom of the screen, there is a status bar with system information, including CPU usage (4.0%), RAM usage (60.0%), and GPU usage (0.0%).

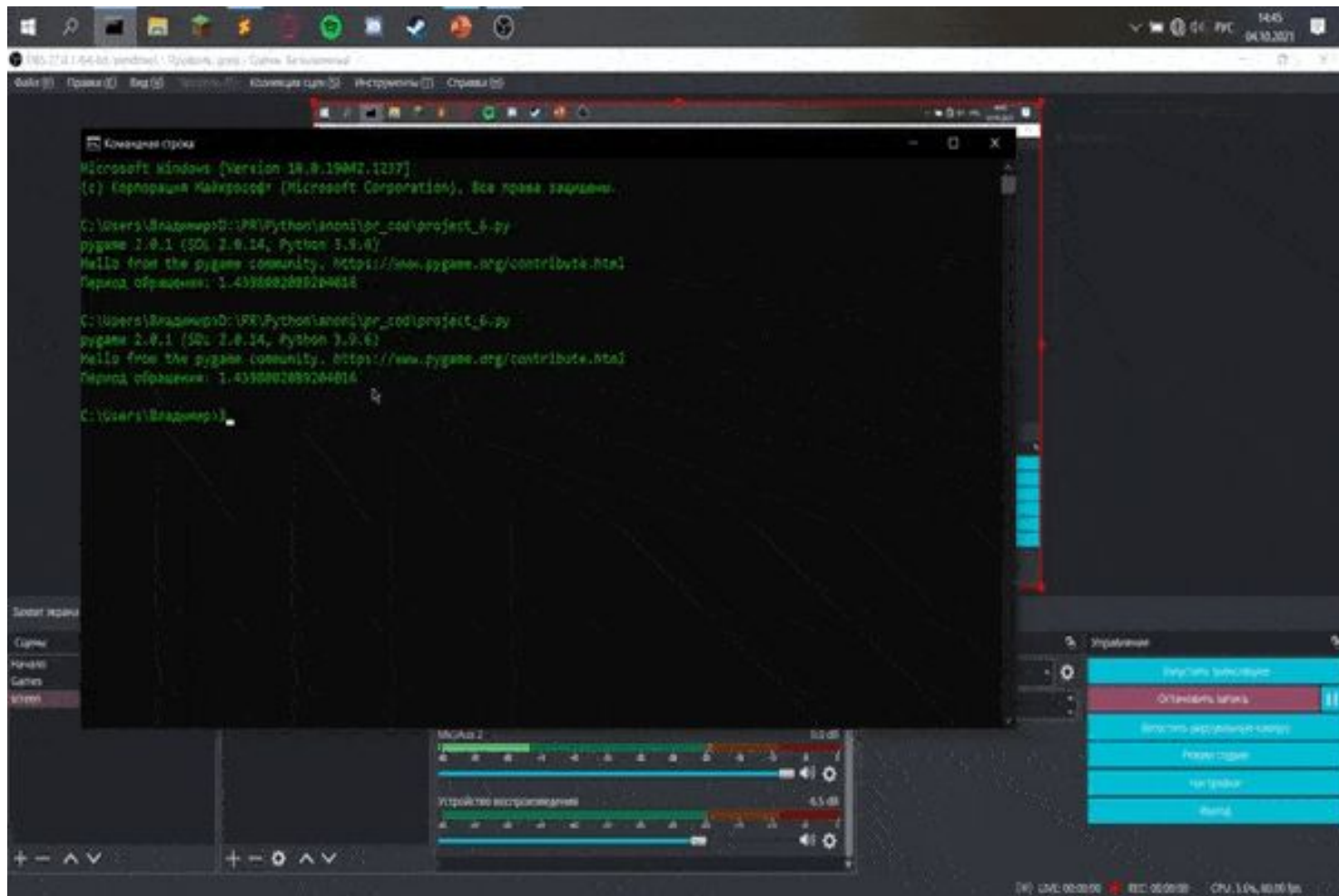
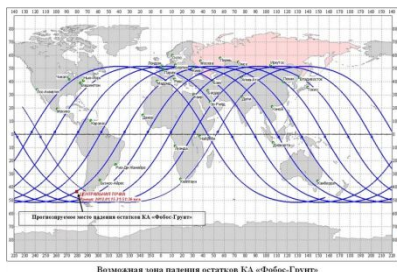
```
Microsoft Windows [Version 10.0.19042.1237]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Владимир>D:\PR\Python\anonilor_cod\project_6.py
pygame 2.0.1 (SDL 2.0.14, Python 3.9.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
Период обращения: 23.91556484428813

C:\Users\Владимир>D:\PR\Python\anonilor_cod\project_6.py
```

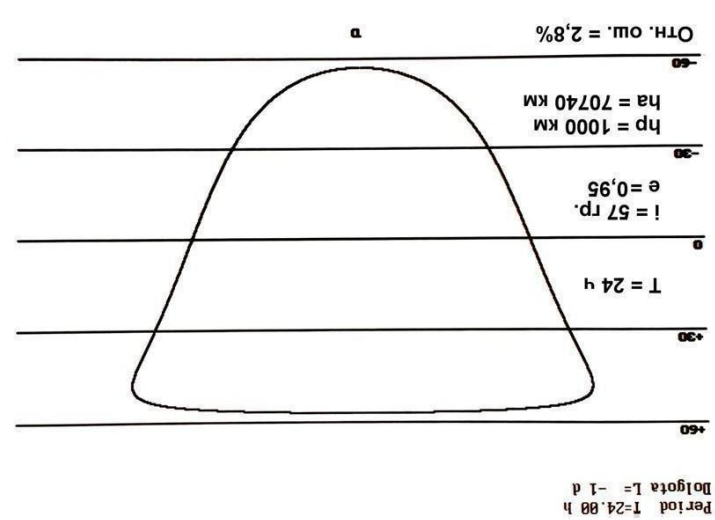
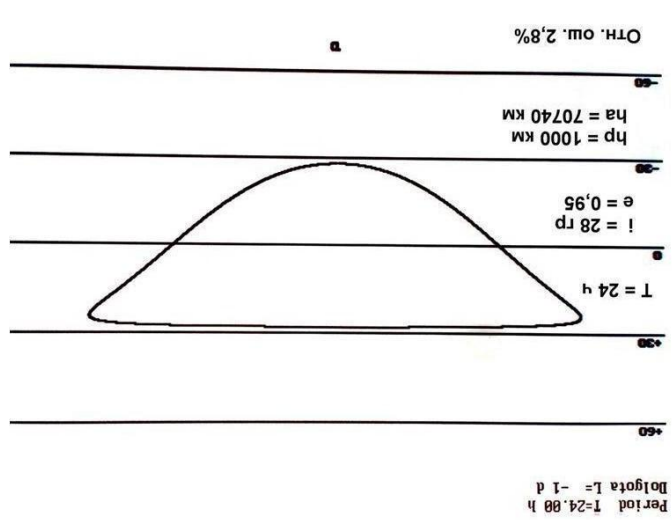


# Верификация программы по трассе группировок КА с низкими орбитами



- высота апогея (максимальная высота) – 174,2 км;
- высота перигея (минимальная высота) – 149,7 км;
- наклонение – 51,44 град.;
- период обращения – 87,57 мин.

# Верификация программы по суточной трассе КА (Усатый И.В.)



```

127 pygame.draw.rect(screen, [206, 0, 255], [1//2+630, 210-fe_sp[1] / 2, 2, 2]) # фн
128 pygame.draw.rect(screen, [255, 162, 0], [1//2+630, 210-ly_sp[1] / 2, 2, 2]) # лямбда
129 # rgb график
130 pygame.draw.rect(screen, [255, 255, 255], [70, 300, 1, 360])
131 pygame.draw.rect(screen, [255, 255, 255], [70, 520, 180, 1])
132 pygame.draw.rect(screen, [225, 0, 0], [1//2+70, 535-x_1_sp[1]/200000, 2, 2])
133 pygame.draw.rect(screen, [255, 255, 0], [1//2+70, 520-y_1_sp[1]/200000, 2, 2])
134 pygame.draw.rect(screen, [0, 255, 255], [1//2+70, 485-z_1_sp[1]/200000, 2, 2])

pygame.display.flip()
    
```

```

1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
    
```

# Выводы

1. Визуальное программирование позволяет быстро создавать программные модули для моделирования движения КА по орбите.

2. Визуальный результат позволяет ускорить процесс отладки программы.

3. Визуальные программы позволяют проверить соответствие модели реальности, а потом перейти к синтезу новых космических орбит для решения практических задач современности.

# Проверка работы в системе «Антиплагиат» Text.ru с показателем 92%

92.70% - Антиплагиат онлайн, п. x

text.ru/antiplagiat/unauthorized

ПРОВЕРКА ТЕКСТА НА УНИКАЛЬНОСТЬ > РЕЗУЛЬТАТЫ ↓

FAQ API проверки

+ Новый текст

Время проверки уникальности: 31.07.2021 9:39 (UTC +03:00)

### Проверка уникальности

Уникальность: **92.70%**

[cfmo.ucoz.ru/load/31\\_ja\\_konferencija\\_sovre...](#) 5%  
[cfmo.ucoz.ru/load/konferencija\\_khkh\\_shkoln...](#) 2%

[Подробнее](#)

### Проверка орфографии

В тексте найдено 37 ошибок:

- восточными.
- не экваториальных орбит задача не
- повтор пробела

[Подробнее](#)

### SEO-анализ текста

Всего символов: **13780** Заспамленность: **65%**  
Без пробелов: **11919** Вода: **10%**  
Количество слов: **1839**

[Подробнее](#)

Вы можете повысить уникальность текста на нашей Бирже рерайтинга.

[Повысить уникальность](#)

Версии текста:

🔒 25 минут назад (UTC +03:00)			
Уникальность	92%	Орфография	37
Всего символов	13780	Заспамленность	65%
Без пробелов	11919	Вода	10%
Количество слов	1839		

Подписаться на регулярную рассылку о русском языке

\*Нажимая кнопку «ОК», вы подтверждаете, что согласны с условиями [Политики конфиденциальности](#) и [обработки файлов Cookie](#)

[Обязательное поле](#) [OK](#)

Напишите нам, мы онлайн!

[Свернуть все окна](#)