

## Челлендж «Программирование графических объектов на языке Питон. Корабли нашей мечты»

**Вызов четвертый.** Учимся показывать нарисованные объекты друг за другом.  
Рисуем корабль

Задача четвертого вызова – научиться воспроизводить нарисованные фигуры на холсте друг за другой с использованием пауз.

Для начала работы возьмем за основу программу, написанную в третьем вызове, которая создает главное окно *window* размером 800x450, размещает в нем холст размером 800x450. Укажем розовый цвет холста **#faafff**. Цвет окна оставим по умолчанию, т.к. окна видно не будет, его полностью закроет холст.

Добавим несколько команд, которые рисуют четырехугольник и 3 линии. В параметрах команд укажем координаты начала и конца отрезков и вершин четырехугольника. Также укажем толщину и цвет границ, цвет заливки четырехугольника. Программа имеет следующий вид:

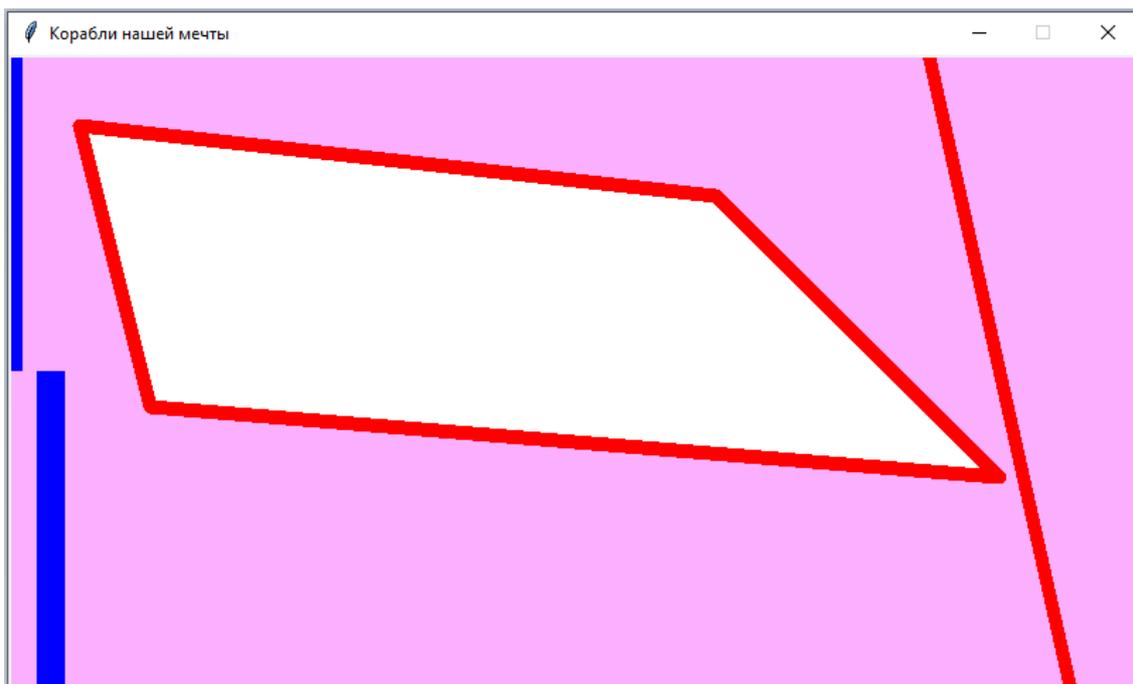
```
from tkinter import *
window = Tk()
window.title('Корабли нашей мечты')
window.configure(background = '#ffffff')
window.geometry("800x450")
window.resizable(width=False, height=False)
c = Canvas(window, width=800, height=450, bg='#faafff')
c.pack()
window.update()
c.create_polygon(50, 50, 500, 100, 700, 300, 100, 250, fill='white', outline='red',
width=10)
c.create_line(650, 0, 750, 450, width=10, fill='red')
c.create_line(0, 0, 0, 450//2, width=20, fill='blue')
c.create_line(30, 450//2, 30, 450, width=20, fill='blue')
window.mainloop()
```

После выполнения данной программы на холсте будут нарисованы четырехугольник и 3 отрезка. Вы можете поэкспериментировать с координатами и параметрами объектов.

Как появляются объекты после работы программы?

Все объекты появляются одновременно.

По итогам работы программы будет создан холст и на нем нарисованы 4 объекта. Получим такой результат:



#### Создание пауз для отображения нарисованных объектов друг за другом

В Питоне есть модуль `time`, который имеет функции для работы со временем. приостанавливает выполнение команды на заданное время. Пауза задается в секундах. Назовем переменную, которая будет задавать паузу, `ts`. Пауза измеряется в секундах. Команда `ts=1` задает паузу длительностью 1 секунда. Для добавления задержки в выполнении программы на заданное время используется функция `time.sleep()`. В качестве аргумента используем переменную `ts`, которая показывает время задержки. В результате получаем команду `time.sleep(ts)`.

В указанной команде `sleep()` – это метод модуля времени в Питоне. Сначала мы должны импортировать модуль времени, а затем применить метод. Импортируем модуль один раз, а задержек можем делать сколько угодно.

```
import time
ts=1
```

Если мы хотим сделать несколько задержек и все они по времени будут одинаковыми, то время задержки достаточно установить один раз и потом использовать во всех командах. Если время задержки для какой-то команды захотим сделать другим, то укажем новую переменную, например, `ts1`, присвоим ей значение, например, `ts1=2` и используем `ts1` в качестве аргумента.

После рисования каждой фигуры применяем команды `time.sleep(ts)` и `window.update()`, где `window` – имя нашего окна, а `update` обновляет информация в окне (`update` переводится как обновить), `ts` – время задержки:

```
time.sleep(ts)  
window.update()
```

Давайте организуем программу так, чтобы каждая фигура рисовалась после предыдущей с задержкой в 1 секунду. Часть программы, посвященная рисованию фигур и выставлению временной задержки, примет вид:

```
import time  
ts=1  
# рисуем четырехугольник и устанавливаем задержку  
c.create_polygon(50, 50, 500, 100, 700, 300, 100, 250, fill='white', outline='red',  
width=10)  
time.sleep(ts)  
window.update()  
# рисуем отрезок и устанавливаем задержку  
c.create_line(650, 0, 750, 450, width=10, fill='red')  
time.sleep(ts)  
window.update()  
# рисуем отрезок и устанавливаем задержку  
c.create_line(0, 0, 0, 450//2, width=20, fill='blue')  
time.sleep(ts)  
window.update()  
# рисуем отрезок и устанавливаем задержку  
c.create_line(30, 450//2, 30, 450, width=20, fill='blue')  
time.sleep(ts)  
window.update()  
window.mainloop()
```

Если какую-то фигуру вы хотите «нарисовать» без временной задержки, вместе с предыдущей, тогда после команды, рисующей данную фигуру, не надо добавлять команды:

```
time.sleep(ts)  
window.update()
```

Поэкспериментируйте со временем задержки, рисуя на холсте различные фигуры.

Для размещения фигур на холсте предварительно рекомендуется рассчитать координаты вершин многоугольников, координаты концов отрезков на бумаге. Используйте бумагу в клеточку, рассчитайте координаты всех точек, а затем напишите программу, выводящую на холст графические объекты, которые вы используете.

Пример расчета и написания программы приведен в видео про создание Андреевского флага.

*Задание **четвертого** вызова челленджа «Корабли нашей мечты»*

Необходимо создать 2 файла – файл программы и файл видео, демонстрирующего работу программы.

*Технические требования к программе:* размер холста 1024x576 пикселей, цвет холста, линий, фигур – произвольный, после рисования каждой линии и/или фигуры (для создания впечатления, что рисуется картина) необходимо поставить задержку по времени в 1 секунду. Допустимо отметить каким-то образом в файле программы название корабля. Время работы программы должно быть не более 5 минут. Имя файла – номер команды.

*Требования к видео:* формат видеофайла mp4, размер окна видео совпадает с размером холста (просим не делать захват лишних областей экрана, только область холста). Имя файла – номер команды.

Готовая работа присылается в оргкомитет по адресу [lab@vcpm.ru](mailto:lab@vcpm.ru). Необходимо выслать оба файла в одном письме в качестве вложения (просьба присылать файлы, а не ссылки на них), в теме письма необходимо написать номер команды и название конкурса «Кораблики».

*Критерии оценки:*

Задания оцениваются по следующим критериям:

- соответствие условию задания;
- сложность представленного объекта;
- эстетический вид представленного объекта;
- качество сопроводительного ролика.

*Удачной работы!*