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

Вызов второй. Создаем окна

Наша цель – научиться создавать окно заданного размера и заданного цвета.

Давайте начнем с некоторых выводов, которые можно сделать по итогам работы с первым вызовом:

- необходимо внимательно вводить текст программы;
- необходимо обращать внимание на ввод таких элементов, как скобки, запятые, кавычки;
- важно не путать заглавные и строчные буквы.

В среде IDLE используют 2 вида окон – окно консоли и окно программы. В окне программы пишется, редактируется и сохраняется программный код. В окне консоли выполняются команды языка Питон. В окне программы пишется текст программы, затем ***сохраняется***, а результат выполнения программы отображается в окне консоли. В окне консоли мы видим и саму программ, и результат ее выполнения.

Какое окно удобнее использовать? Все зависит от того, что мы хотим сделать. Окно *программы* удобно использовать для больших программ, когда нужно сначала набрать все команды, отредактировать, а потом запустить на выполнение командой Run. Окно *консоли* удобно использовать, если хочется попробовать работу разных команд, разобраться с назначением команды.

Выполнение программ

Принцип работы любой программы таков – программа принимает входные данные (ввод), обрабатывает их в соответствии с условиями, заданиями, выводит результаты выполнения (вывод).

Ввод	⇒	Обработка	⇒	Вывод

Переменные в языке Питон

Переменная – это величина, у нее есть имя и значение. Более точно можно сказать, что переменная – это область памяти, имеющая имя. Как только область назвали (дали переменной имя), можно обращаться к данным, которые лежат в этой области.

Для создания новой переменной надо дать ей имя, например, name, или number, или Summa, или _start.

Правила языка требуют, чтобы имя переменной состояло из цифр, букв и знаков подчеркивания. Для удобства работы рекомендуется давать переменным имена, показывающие суть объекта, желательно ограничивать имена 15 символами. Имя переменной должно начинаться с буквы или подчеркивания, имя переменной не может начинаться с цифры. В Питоне различаются регистры, поэтому Text, text или TEXT – это разные переменные.

При задании имен используются символы латинского алфавита, рекомендуется использовать английские слова, а не транслитерацию русских слов, например, правильно использовать speed, а не skorost. В любом случае при чтении текста программы должно быть понятно, что вы имели ввиду.

Для того, чтобы научиться правильно задавать имена переменных, можно изучить коды программ, написанных профессионалами, и сделать выводы.

Оператор присваивания

Операторы – команды на языке программирования.

Любая величина, участвующая в выражении, называется *операндом*.

Выражение – это конструкция, состоящая из переменных, констант и знаков операций и служащая для вычисления какого-либо значения.

Например, в выражении a+b*5 операндами являются переменные a и b и константа 5, операциями являются операция сложения + и операция умножения*.

Оператор присваивания служит для вычисления значения выражения и сохранения его в памяти, для записи оператора присваивания используется символ «=», значение выражения справа от знака «=» передается переменной, стоящей слева, это и называется *присваиванием*.

Оператор присваивания состоит из имени переменной, знака равенства и выражения.

Например, N_team = 17 (в примере переменной *номер команды* присваивается значение 17).

b=”Привет”

d=’Hello’

Т.о. для создания переменной достаточно дать ей имя и присвоить значение.

Использование комментариев

Комментарии – это текстовые сообщения, которые поясняют то, что делает программа, чтобы можно было в ней разобраться. Комментарии не проверяются на ошибки и не выполняются. Чтобы отличить комментарии от текста программы, они начинаются со знака решетки. Комментарии можно писать по-русски.

пример комментария

Основные команды языка Питон

Запустить программу	Меню *Run* или клавиша *F5* в окне программы
Остановить программу	*CTRL+C* на клавиатуре (в окне консоли)
Вывод текста на экран	print('текст')
Задать переменной числовое значение	number = 20
Задать переменной строковое значение	word = 'text'
Считать текст с клавиатуры в переменную	name = input('name?')
Прибавить число к переменной	number = number + 1 или number += 1

Библиотеки языка Питон

Написание новых программ требует времени. Удобно использовать части уже написанных программ. Их можно хранить в так называемых *библиотеках*. В Питоне есть стандартные библиотеки, содержащие фрагменты готового кода. Отдельные части библиотек называются модулями, которые можно использовать при написании программ. Изучим один из таких модулей из стандартной библиотеки.

Создание окна с использованием модуля Tkinter

Tkinter – это графическая библиотека, которая позволяет создавать программы с оконным интерфейсом. Начиная с версии языка питон 3.0 в программах название tkinter пишется с маленькой буквы. Также Tkinter используется для создания кнопок и других графических элементов. В Tkinter визуальные элементы принято называть виджетами (от английского window gadget или widget). Виджет – это компонент графического интерфейса, с которым взаимодействует пользователь.

Окно является одним из основных элементов, в котором можно размещать другие объекты. Его называют главным окном приложения.

Давайте создадим стандартное окно на экране компьютера с помощью модуля Tkinter. Перед использованием модуля в программе, его надо загрузить. Для загрузки модулей используется команда import (импортировать).

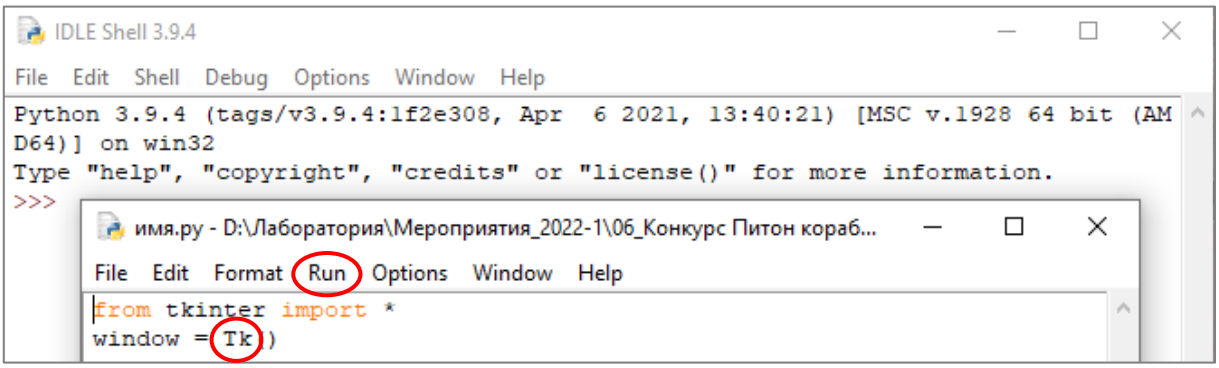
В окне программы наберем следующий текст, программа создаст окно с названием Tk, комментарии можно не набирать:

```
from tkinter import *    # загружает Tkinter из библиотеки
window = Tk()            # создает окно Tkinter с именем windows
```

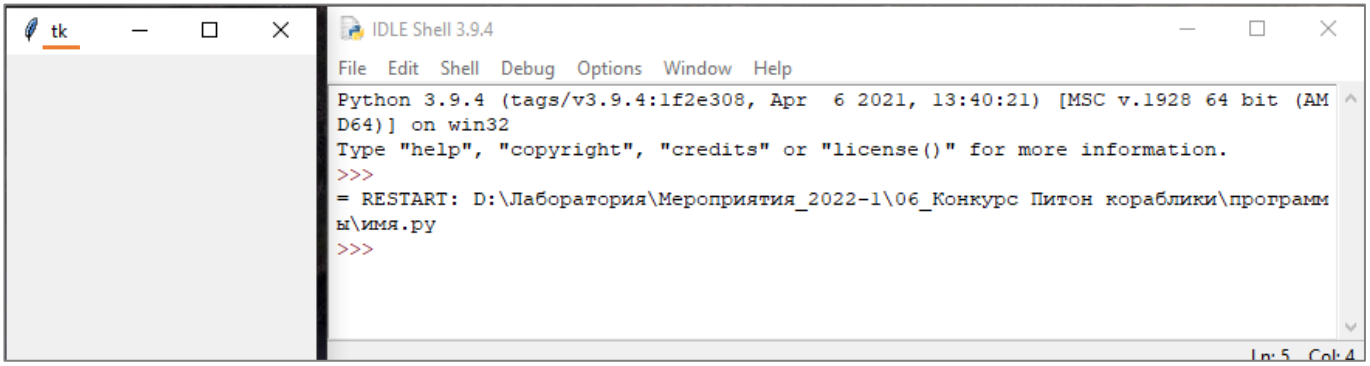
Мы подключаем модуль tkinter с помощью инструкции from, после команды import мы указали звездочку, это означает, что из модуля tkinter будут подключены все имеющиеся атрибуты (классы, функции, объекты). А можно вместо * указать конкретный атрибут, который мы планируем использовать. Т.е. можно загрузить весь

модуль, а можно нужную его часть – только нужные атрибуты, в этом случае надо точно знать, какие атрибуты понадобятся. Если в процессе работы понадобятся другие атрибуты данного модуля, то их придется тоже загружать. Давайте подгрузим весь модуль, в этом случае можно обойтись одной загрузкой.

После набора текста программы ее надо сохранить и запустить на выполнение командой Run/Run Module. В результате работы программы будет создано стандартное окно, назовем его главным окном.



Справа – результат запуска программы в окне консоли, слева – созданное в результате работы программы окно, пока совершенно простое окошко стандартного серого цвета по умолчанию, в его заголовке мы видим tk, как указано в тексте программы. Мы выполнили первый этап создания окна.



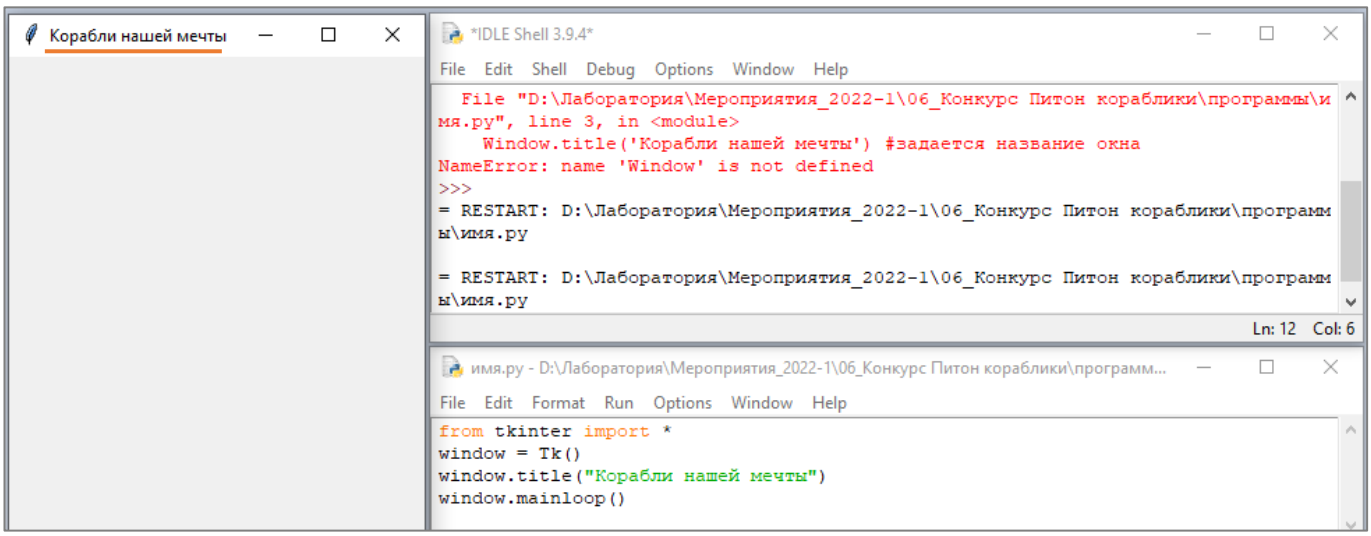
Давайте изменим название в заголовке окна, например, на Корабли нашей мечты. Для этого добавляем в текст программы команду title.

Далее нам понадобится понятие метода. *Метод* – это действие, которое можно совершить с объектом. Обращение к методу осуществляется по схеме *имя_объекта.название_метода*, где имя_объекта – это имя объекта, с которым работаем, ну и название_метода показывает, какой именно метод применяется.

Метод mainloop экземпляра объекта (в нашем случае окна window) в Tkinter запускает главный цикл обработки событий, в частности, приводит к отображению главного окна со всеми расположенными на нем элементами.

```
window.title(«Корабли нашей мечты») #задается название окна в заголовке
window.mainloop() # данная функция обеспечит ситуацию, когда окно будет ждать любого взаимодействия с пользователем, пока окно не будет закрыто
```

У созданного окна изменилось имя на «Корабли нашей мечты».



Обратите внимание, что в тексте программы имя *window* везде пишется с маленькой буквы, во всех строчках одинаково!

Также необходимо обращать внимание на кавычки – используются пары ‘ ’ или “ ” или « ». Если при использовании кавычек программа будет выдавать ошибку, попробуйте выбрать другой тип кавычек.

Таким образом, мы научились создавать окна и давать им названия в заголовке окна.

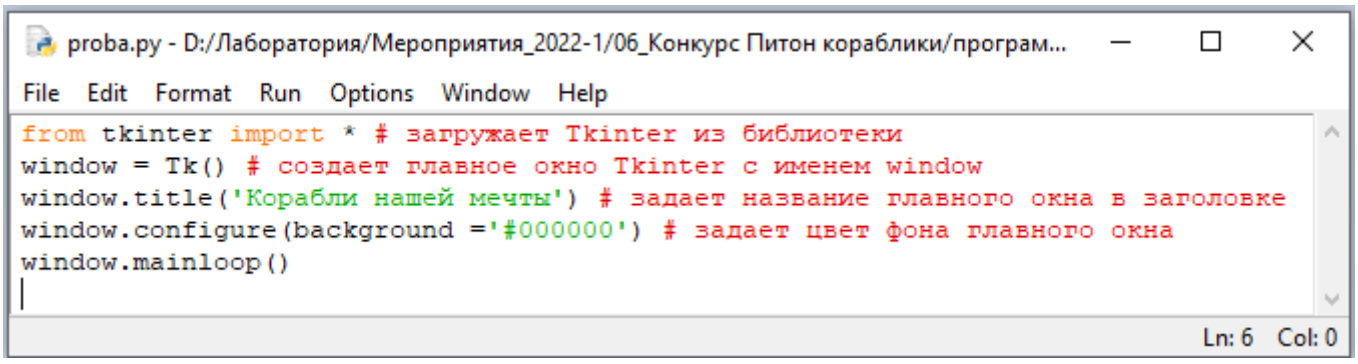
Задаем размеры и цвет окна

При создании виджетов для придания объектам заданных свойств задаются аргументы. Они могут задавать типы шрифтов, цвет, размер и т.п.

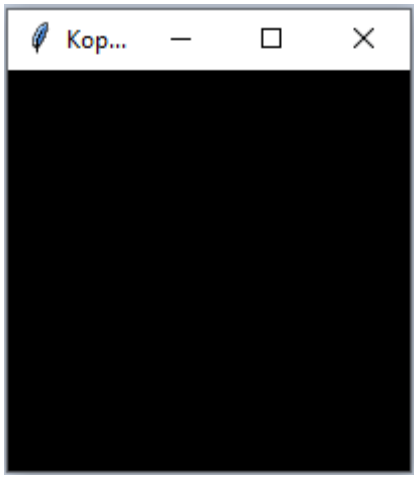
Для задания цвета главного окна используется метод *configure*, в качестве аргумента используется background (переводится как фон):

```
window.configure(background='#000000’) #задается белый цвет главного окна
```

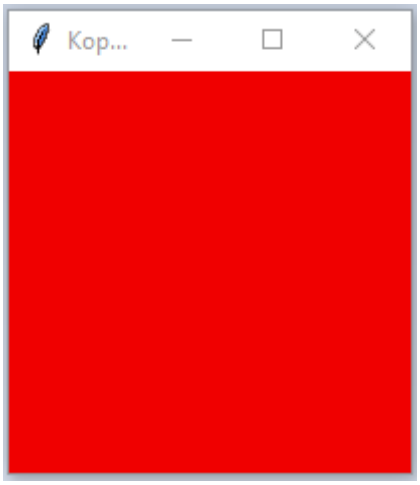
Эту команду необходимо ввести перед командой, содержащей **mainloop.** Иначе работа программы остановится и не будут выполняться команды, расположенные в тексте после **mainloop.**



В результате работы данной программы будет создано окно черного цвета, размер окна задан по умолчанию, в заголовке окна отображается «Корабли нашей мечты» (пока видно не все название в связи с малым размером окна).

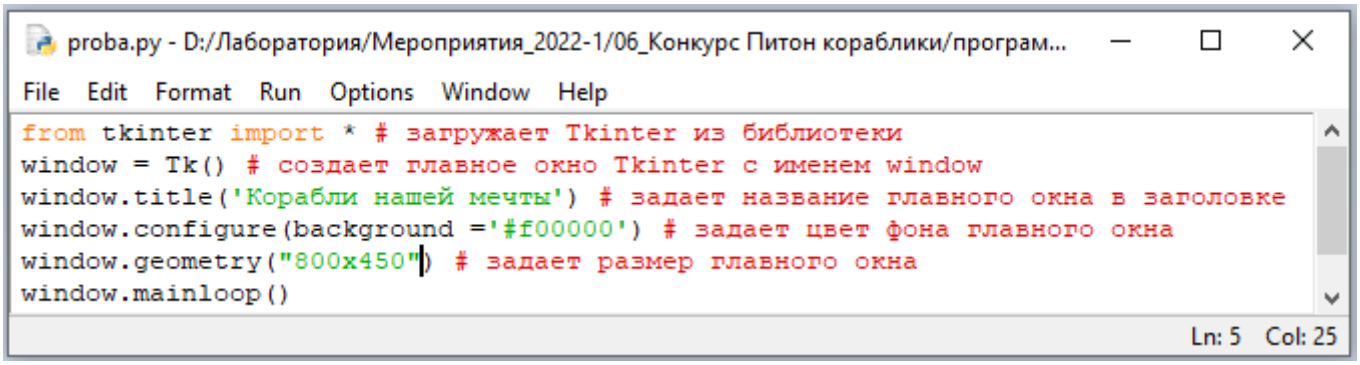


Попробуйте поэкспериментировать с цветом окна, меняя значение аргумента background=’#000000’. Например, используя значение background ='#f00000' получаем главное окно красного цвета.



Теперь давайте разберемся с размером окна. Создадим окно размером 800 пикселей по ширине и 450 пикселей по высоте.

Для задания размера главного окна используется метод *geometry*, в качестве аргумента используются значения ширины и высоты главного окна, аргумент указывается в скобках, обратите внимание на кавычки:



В результате работы данной программы будет создано главное окно красного цвета с размерами 800 пикселей н 450 пикселей.

Попробуйте поэкспериментировать с размерами окна, меняя значения ширины и высоты окна. Например, если задать ширину и высоту одинаковыми, то главное окно будет квадратным.

Для того, чтобы запретить изменение размеров окна, добавим в программу следующий текст:

```
window.resizable(width=False, height=False)
```

Эту команду также необходимо ввести перед командой, содержащей *mainloop*. В результате программа будет иметь вид:

```
from tkinter import * # загружает Tkinter из библиотеки
window = Tk() # создает главное окно Tkinter с именем window
window.title('Корабли нашей мечты') # задает название главного окна в заголовке
window.configure(background ='#f00000') # задает цвет фона главного окна
window.geometry("800x450") # задает размер главного окна
window.resizable(width=False, height=False) # запрещает изменение размеров окна
window.mainloop()
```

А в результате ее выполнения создается главное окно красного цвета размером 800 на 450 пикселей, закрепленный размер окна можно проверить по кнопке «развернуть окно» в правом верхнем углу, эта кнопка стала недоступной.

Создание холста для рисования

Для того, чтобы рисовать в Питоне (рисовать фигуры, размещать графические объекты в окне), необходимо создать область для рисования, эту область называют

холстом. При создании холста (создаем объект Canvas) необходимо задать ширину, высоту и цвет окна.

Фрагмент программы, создающий холст шириной 800 пикселей, высотой 400 пикселей белого цвета, выглядит следующим образом:

```
c = Canvas(window, width=800, height=400, background=’#ffffff’)
```

В этом фрагменте width – ширина, height – высота, bg от слова background, означает фон, эта опция позволяет задать цвет холста. Если размер холста совпадает с размером главного окна, то холст полностью закроет главное окно. Если размер холста задать меньше размеров главного окна, то холст ляжет сверху и будет видна часть главного окна.

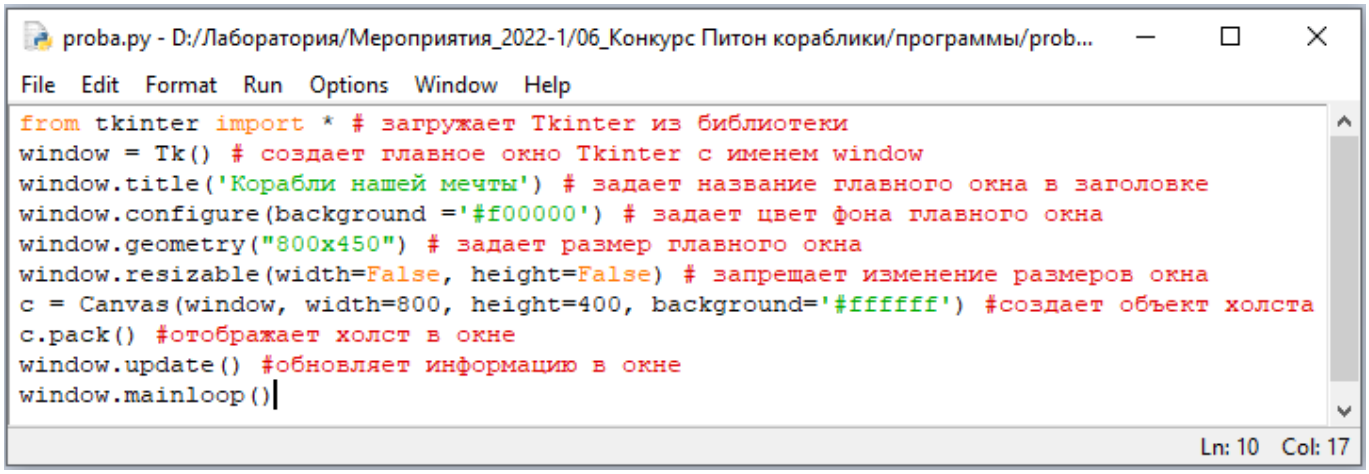
Цвет задается в 16-ричном коде. Более подробно о работе с цветом смотрите в дополнительном материале. Или просто укажите цвет. Мы задали белый цвет ffffff.

Для отображения запрограммированного нами окна с холстом на экране необходимо применить метод *pack*, для обновления информации в окне применяется метод *update*.

Наша программа теперь будет иметь вид:

```
from tkinter import *    # загружает Tkinter из библиотеки
window = Tk()            # создает окно Tkinter с именем window
window.title(«Корабли нашей мечты») #задает название окна в заголовке
window.configure(background ='#f00000') # задает цвет фона главного окна
window.geometry("800x450") # задает размер главного окна
window.resizable(width=False, height=False) # запрещает изменение размеров окна
c = Canvas(window, width=800, height=400, bg=’#ffffff’) #создает объект холста
c.pack() #отображает холст в окне
window.update() #обновляет информацию в окне
window.mainloop()
```

В окне программы наша программа имеет вид:



В результаты работы данной программы будет создано главное окно размером 800x450 пикселей красного цвета и холст в нем размером 800x400 белого цвета:

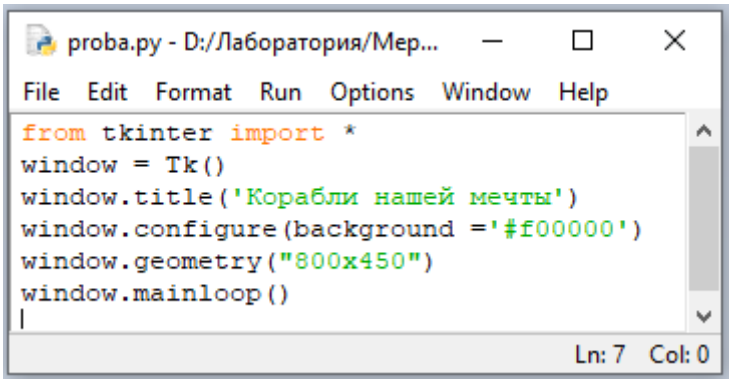
Поэкспериментируйте с размерами холста, меняя значения его ширины и высоты, и с цветом холста, изучив задание цвета в 16-ричном коде.

Для выполнения итогового конкурсного задания как раз понадобится создать окно и холст заданного размера. А цвет команды будут выбирать самостоятельно в соответствии со своей задумкой рисунка.

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

После выполнения первого вызова необходимо заполнить Google форму, в которой надо указать номер команды, фамилию наставника (наставников), ответить на вопросы «Удалось создать главное окно с использованием Tkinter?», «Удалось создать холст?», ответы необходимо дать в форме Да или Нет.

Далее ответьте на следующий вопрос. Дана следующая программа, укажите номер рисунка из трех, который соответствует результату работы данной программы.

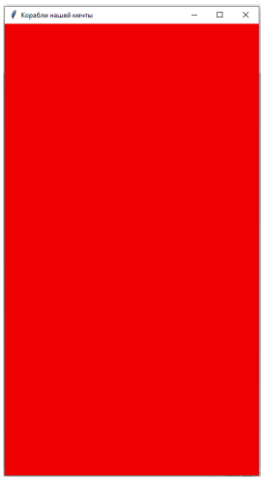


Результату работы данной программы соответствует рисунок:

1

2

3



В заключительной колонке таблицы необходимо написать, с какими трудностями столкнулись при выполнении второго вызова.

Удачной работы!