

Упражнения по программированию главы 13

coding: utf-8

Упражнение по программированию 13.1. ФИО и адрес

```
import tkinter
from tkinter.ttk import *

class ShowInfoGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать две рамки.
        self.top_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать пустое поле в верхней рамке.
        self.value = tkinter.StringVar()
        self.address_label = Label(self.top_frame,
                                   textvariable= self.value)

        # Создать две кнопки в нижней рамке.
        self.address_button = Button(self.bottom_frame,
                                     text = 'Показать инфо',
                                     command = self.show_info)
        self.quit_button = Button(self.bottom_frame,
                                   text = 'Выйти',
                                   command = self.main_window.destroy)

        # Упаковать элемент Label.
        self.address_label.pack()

        # Упаковать элементы Button.
        self.address_button.pack(side = 'left')
        self.quit_button.pack(side = 'left')

        # Упаковать элементы Frame.
        self.top_frame.pack()
        self.bottom_frame.pack()

        # Войти в главный цикл tkinter.
        tkinter.mainloop()

# Определить функцию show_info.
def show_info(self):
    self.value.set('Стивен Маркус\n274 Бейли\n' \
                  'Уэйнзвилль, Северная Каролина 27999')
```

```
# Создать экземпляр ShowInfoGUI
show_info = ShowInfoGUI()
```

Упражнение по программированию 13.2. Переводчик с латинского

```
import tkinter
from tkinter.ttk import *

class LatinTranslatorGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать две рамки.
        self.top_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать пустое поле в верхней рамке.
        self.value = tkinter.StringVar()
        self.word_label = Label(self.top_frame,
                                textvariable= self.value)

        # Создать кнопки в нижней рамке.
        self.sinister_button = Button(self.bottom_frame,
                                       text = 'sinister',
                                       command = self.show_word1)
        self.dexter_button = Button(self.bottom_frame,
                                     text = 'dexter',
                                     command = self.show_word2)
        self.medium_button = Button(self.bottom_frame,
                                     text = 'medium',
                                     command = self.show_word3)

        # Упаковать элемент Label.
        self.word_label.pack()

        # Упаковать элементы Button.
        self.sinister_button.pack(side = 'left')
        self.dexter_button.pack(side = 'left')
        self.medium_button.pack(side = 'left')

        # Упаковать элементы Frame.
        self.top_frame.pack()
        self.bottom_frame.pack()

        # Войти в главный цикл tkinter.
        tkinter.mainloop()

        # Определить функцию show_word.
    def show_word1(self):
        self.value.set('левый')
```

```

def show_word2(self):
    self.value.set('правый')
def show_word3(self):
    self.value.set('центральный')

# Создать экземпляр LatinTranslatorGUI.
latin_translator = LatinTranslatorGUI()

```

Упражнение по программированию 13.3. Калькулятор миль на галлон бензина

```

import tkinter
from tkinter.ttk import *

class MilesGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать четыре рамки.
        self.gallons_frame = Frame(self.main_window)
        self.miles_frame = Frame(self.main_window)
        self.mpg_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать элементы интерфейса для рамки gallons_frame.
        self.gallons_label = Label(self.gallons_frame,
                                    text = 'Введите количество галлонов:')
        self.gallons_entry = Entry(self.gallons_frame, width = 10)

        # Упаковать элементы рамки gallons_frame.
        self.gallons_label.pack(side = 'left')
        self.gallons_entry.pack(side = 'left')

        # Упаковать элементы рамки miles_frame.
        self.miles_label = Label(self.miles_frame,
                                   text = 'Введите количество миль:')
        self.miles_entry = Entry(self.miles_frame, width = 10)

        # Упаковать элементы рамки miles_frame.
        self.miles_label.pack(side = 'left')
        self.miles_entry.pack(side = 'left')

        # Создать элементы интерфейса для поля result_label.
        self.result_label = Label(self.mpg_frame,
                                    text = 'Мили на галлон (MPG) = ')

        # Создать пустое поле.
        self.mpg = tkinter.StringVar()
        self.mpg_label = Label(self.mpg_frame,

```

```

        textvariable= self.mpg)

# Упаковать элементы интерфейса для рамки.
self.result_label.pack(side = 'left')
self.mpg_label.pack(side = 'left')

# Создать две кнопки в нижней рамке.
self.mpg_button = Button(self.bottom_frame,
                        text = 'Вычислить MPG',
                        command = self.calculate_mpg)
self.quit_button = Button(self.bottom_frame,
                        text = 'Выйти',
                        command = self.main_window.destroy)

# Упаковать элементы интерфейса в нижней рамке.
self.mpg_button.pack(side='left')
self.quit_button.pack(side='left')

# Упаковать рамки.
self.gallons_frame.pack()
self.miles_frame.pack()
self.mpg_frame.pack()
self.bottom_frame.pack()

# Войти в главный цикл tkinter.
tkinter.mainloop()

# Определить функцию show_info.
def calculate_mpg(self):
    # Получить введенные значения.
    self.gallons = float(self.gallons_entry.get())
    self.miles = float(self.miles_entry.get())

    # Вычислить MPG.
    self.miles_per_gallon = float(self.miles) / self.gallons

    # Обновить поле mpg_label.
    self.mpg.set(self.miles_per_gallon)

# Создать экземпляр MilesGUI.
mpg = MilesGUI()

```

Упражнение по программированию 13.4. Из шкалы Цельсия в шкалу Фаренгейта

```

import tkinter
from tkinter.ttk import *

class CelsiusGUI:
    def __init__(self):
        # Создать главное окно.

```

```

self.main_window = tkinter.Tk()

# Создать рамки.
self.top_frame = Frame(self.main_window)
self.mid_frame = Frame(self.main_window)
self.bottom_frame = Frame(self.main_window)

# Создать элементы интерфейса для верхней рамки.
self.celsius_label = Label(self.top_frame,
                            text = 'Введите температуру в градусах Цельсия:')
self.celsius_entry = Entry(self.top_frame, width = 10)

# Упаковать элементы верхней рамки.
self.celsius_label.pack(side = 'left')
self.celsius_entry.pack(side = 'left')

# Создать элементы интерфейса для средней рамки.
self.results_label = Label(self.mid_frame,
                            text = 'Температура по шкале Фаренгейта: ')

# Создать пустое поле.
self.fahr = tkinter.StringVar()
self.fahrenheit_label = Label(self.mid_frame,
                              textvariable= self.fahr)

# Упаковать элементы средней рамки.
self.results_label.pack(side = 'left')
self.fahrenheit_label.pack(side = 'left')

# Создать две кнопки в нижней рамке.
self.fahrenheit_button = Button(self.bottom_frame,
                                text = 'Преобразовать в градусы Фаренгейта',
                                command = self.convert)
self.quit_button = Button(self.bottom_frame,
                           text = 'Выйти',
                           command = self.main_window.destroy)

# Упаковать элементы нижней рамки.
self.fahrenheit_button.pack(side='left')
self.quit_button.pack(side='left')

# Упаковать рамки.
self.top_frame.pack()
self.mid_frame.pack()
self.bottom_frame.pack()

# Войти в главный цикл tkinter.
tkinter.mainloop()

```

```

# Определить функцию show_info.
def convert(self):
    # Получить введенные значения.
    self.celsius = float(self.celsius_entry.get())

    # Вычислить градусы Фаренгейта.
    self.fahrenheit = 9.0 / 5.0 * float(self.celsius) + 32

    # Обновить поле fahrenheit_label.
    self.fahr.set(self.fahrenheit)

# Создать экземпляр MilesGUI.
celsius = CelsiusGUI()

```

Упражнение по программированию 13.5. Налог на недвижимость

```

import tkinter
from tkinter.ttk import *

class PropertyGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать рамки
        self.value_frame = Frame(self.main_window)
        self.assess_frame = Frame(self.main_window)
        self.tax_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать элементы интерфейса для рамки value_frame.
        self.value_label = Label(self.value_frame,
                                text = 'Введите стоимость имущества: $')
        self.value_entry = Entry(self.value_frame, width = 10)

        # Упаковать элементы рамки value_frame.
        self.value_label.pack(side = 'left')
        self.value_entry.pack(side = 'left')

        # Создать элементы интерфейса для рамки assess_frame.
        self.assess_results_label = Label(self.assess_frame,
                                           text = 'Оценочная стоимость: ')

        # Создать пустое поле для оценочной стоимости.
        self.assess = tkinter.StringVar()
        self.assess_label = Label(self.assess_frame,
                                   textvariable= self.assess)

        # Создать элементы интерфейса для рамки assess_frame.
        self.assess_results_label.pack(side = 'left')

```

```

self.assess_label.pack(side = 'left')

# Создать элементы интерфейса для рамки tax_frame.
self.tax_results_label = Label(self.tax_frame,
                                text = 'Налог на имущество: ')

# Создать пустое поле для величины налога на имущество.
self.tax = tkinter.StringVar()
self.tax_label = Label(self.tax_frame,
                        textvariable= self.tax)

# Упаковать элементы рамки tax_frame.
self.tax_results_label.pack(side = 'left')
self.tax_label.pack(side = 'left')

# Создать две кнопки в нижней рамке.
self.display_button = Button(self.bottom_frame,
                              text = 'Вычислить',
                              command = self.calculate)
self.quit_button = Button(self.bottom_frame,
                           text = 'Выйти',
                           command = self.main_window.destroy)

# Упаковать элементы в нижней рамке.
self.display_button.pack(side='left')
self.quit_button.pack(side='left')

# Упаковать рамки.
self.value_frame.pack()
self.assess_frame.pack()
self.tax_frame.pack()
self.bottom_frame.pack()

# Войти в главный цикл tkinter.
tkinter.mainloop()

# Определить функцию show_info.
def calculate(self):
    # Получить введенные значения.
    self.value = float(self.value_entry.get())

    # Вычислить оценочную стоимость.
    self.assessment = 0.60 * self.value

    # Обновить поле с оценочной стоимостью.
    self.assess.set('$' + str(self.assessment))

    # Вычислить налог.
    self.property_tax = float(self.assessment) / 100 * 0.75

```

```

        # Обновить поле с налогом.
        self.tax.set('$' + str(self.property_tax))

# Создать экземпляр PropertyGUI.
prop = PropertyGUI()

```

Упражнение по программированию 13.6. Авторемонтная фирма "Автоцех"

```

import tkinter
from tkinter.ttk import *
import tkinter.messagebox

class AutoGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать рамки.
        self.top_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать переменные для использования с элементами Checkbuttons.
        self.cb_oil_var = tkinter.IntVar()
        self.cb_lube_var = tkinter.IntVar()
        self.cb_radiator_var = tkinter.IntVar()
        self.cb_trans_var = tkinter.IntVar()
        self.cb_inspection_var = tkinter.IntVar()
        self.cb_muffler_var = tkinter.IntVar()
        self.cb_tire_var = tkinter.IntVar()

        # Присвоить переменным значение 0.
        self.cb_oil_var.set(0)
        self.cb_lube_var.set(0)
        self.cb_radiator_var.set(0)
        self.cb_trans_var.set(0)
        self.cb_inspection_var.set(0)
        self.cb_muffler_var.set(0)
        self.cb_tire_var.set(0)

        # Создать флаговые кнопки Checkbutton в верхней рамке.
        self.cb1 = Checkbutton(self.top_frame,
                                text= 'Замена масла - 500 руб.',
                                variable = self.cb_oil_var)

        self.cb2 = Checkbutton(self.top_frame,
                                text = 'Смазочные работы - 300 руб.',
                                variable = self.cb_lube_var)

        self.cb3 = Checkbutton(self.top_frame,
                                text='Промывка радиатора - 700 руб.',

```



```

        variable = self.cb_radiator_var)
self.cb4 = Checkbutton(self.top_frame,
                        text='Замена жидкости в трансмиссии - 1000 руб.',
                        variable = self.cb_trans_var)
self.cb5 = Checkbutton(self.top_frame,
                        text= 'Осмотр - 800 руб.',
                        variable = self.cb_inspection_var)
self.cb6 = Checkbutton(self.top_frame,
                        text='Замена глушителя выхлопа - 1300 руб.',
                        variable = self.cb_muffler_var)
self.cb7 = Checkbutton(self.top_frame,
                        text = 'Перестановка шин - 1300 руб.',
                        variable = self.cb_tire_var)

# Упаковать элементы Checkbutton.
self.cb1.pack()
self.cb2.pack()
self.cb3.pack()
self.cb4.pack()
self.cb5.pack()
self.cb6.pack()
self.cb7.pack()

# Создать две кнопки в нижней рамке.
self.display_button = Button(self.bottom_frame,
                              text = 'Показать стоимость',
                              command = self.calculate)
self.quit_button = Button(self.bottom_frame,
                           text = 'Выйти',
                           command = self.main_window.destroy)

# Упаковать элементы в нижней рамке.
self.display_button.pack(side='left')
self.quit_button.pack(side='left')

# Упаковать рамки.
self.top_frame.pack()
self.bottom_frame.pack()

# Войти в главный цикл tkinter.
tkinter.mainloop()

# Определить функцию show_info.
def calculate(self):
    # Создать переменную для итоговой суммы.
    self.total = 0.0

    # Определить общую стоимость на основе выбранных флаговых кнопок.
    if self.cb_oil_var.get() == 1:

```

```

        self.total += 500.0
    if self.cb_lube_var.get() == 1:
        self.total += 300.0
    if self.cb_radiator_var.get() == 1:
        self.total += 700.0
    if self.cb_trans_var.get() == 1:
        self.total += 1000.0
    if self.cb_inspection_var.get() == 1:
        self.total += 800.0
    if self.cb_muffler_var.get() == 1:
        self.total += 1300.0
    if self.cb_tire_var.get() == 1:
        self.total += 1300.0

    # Показать информационное окно.
    tkinter.messagebox.showinfo('Общая стоимость',
                                'Ваши затраты = ' + \
                                format(self.total, ',.2f') + ' руб.')
```

Создать экземпляр AutoGUI.

```
auto = AutoGUI()
```

Упражнение по программированию 13.7. Междугородные звонки

```
import tkinter
from tkinter.ttk import *
import tkinter.messagebox

class LongDistanceGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать рамки.
        self.top_frame = Frame(self.main_window)
        self.mid_frame = Frame(self.main_window)
        self.bottom_frame = Frame(self.main_window)

        # Создать переменную для использования с радиокнопками.
        self.rb_var = tkinter.IntVar()

        # Присвоить переменной 1.
        self.rb_var.set(1)

        # Создать элементы Radiobutton в верхней рамке.
        self.rb1 = Radiobutton(self.top_frame,
                                text = 'Дневное время (6:00 - 17:59)',
                                variable = self.rb_var, value = 1)
        self.rb2 = Radiobutton(self.top_frame,
                                text = 'Вечернее время (18:00 - 23:59)',
```

```

        variable = self.rb_var, value = 2)
self.rb3 = Radiobutton(self.top_frame,
                        text = 'Непиковый период (00:00 - 5:59)',
                        variable = self.rb_var, value = 3)

# Упаковать элементы Radiobutton.
self.rb1.pack()
self.rb2.pack()
self.rb3.pack()

# Создать элементы интерфейса для средней рамки.
self.minutes_label = Label(self.mid_frame,
                           text = 'Введите количество минут:')
self.minutes_entry = Entry(self.mid_frame, width = 10)

# Упаковать элементы в рамке value_frame.
self.minutes_label.pack(side = 'left')
self.minutes_entry.pack(side = 'left')

# Создать две кнопки в нижней рамке.
self.display_button = Button(self.bottom_frame,
                             text = 'Показать стоимость',
                             command = self.calculate)
self.quit_button = Button(self.bottom_frame,
                          text = 'Выйти',
                          command = self.main_window.destroy)

# Упаковать элементы в нижней рамке.
self.display_button.pack(side='left')
self.quit_button.pack(side='left')

# Упаковать рамки.
self.top_frame.pack()
self.mid_frame.pack()
self.bottom_frame.pack()

# Войти в главный цикл tkinter.
tkinter.mainloop()

# Определить функцию calculate.
def calculate(self):
    # Получить введенное значение.
    self.minutes = float(self.minutes_entry.get())

    # Определить тариф на основе выбранной радиокнопки.
    if self.rb_var.get() == 1:
        self.rate = 10.0
    if self.rb_var.get() == 2:
        self.rate = 12.0

```

```

        if self.rb_var.get() == 3:
            self.rate = 5.0

        # Вычислить стоимость.
        self.charges = self.minutes * self.rate

        # Показать информационное окно.
        tkinter.messagebox.showinfo('Общая стоимость',
                                     'Ваши затраты = ' + \
                                     format(self.charges, ',.2f') + ' руб.')

# Создать экземпляр LongDistanceGUIю
long_dist = LongDistanceGUI()

```

Упражнение по программированию 13.8. Рисунок старого дома

```

# Эта программа рисует контур дома.
import tkinter

# Именованные константы.
CANVAS_WIDTH = 600
CANVAS_HEIGHT = 400
HOUSE_X = 50
HOUSE_Y = 400
HOUSE_WIDTH = 500
HOUSE_HEIGHT = 200
ROOF_X = HOUSE_X - 25
ROOF_Y = HOUSE_Y - HOUSE_HEIGHT
ROOF_WIDTH = HOUSE_WIDTH + 50
ROOF_HEIGHT = 150
ROOF_APEX_X = ROOF_X + int(ROOF_WIDTH / 2)
ROOF_APEX_Y = ROOF_Y - ROOF_HEIGHT
DOOR_WIDTH = 80
DOOR_HEIGHT = 150
DOOR_X = (HOUSE_X + int(HOUSE_WIDTH / 2)) - (int(DOOR_WIDTH / 2))
DOOR_Y = HOUSE_Y
WINDOW_WIDTH = 100
WINDOW_HEIGHT = 100
WINDOW1_X = (HOUSE_X + int(HOUSE_WIDTH / 4)) - (int(WINDOW_WIDTH / 2))
WINDOW1_Y = HOUSE_Y - 50
WINDOW2_X = (HOUSE_X + (int(HOUSE_WIDTH / 4) * 3)) - (int(WINDOW_WIDTH / 2))
WINDOW2_Y = WINDOW1_Y
SUN_WIDTH = 75
SUN_X = CANVAS_WIDTH - SUN_WIDTH - 20
SUN_Y = SUN_WIDTH + 20

class MyGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

```

```

# Создать элемент интерфейса Canvas.
self.canvas = tkinter.Canvas(self.main_window, width=CANVAS_WIDTH,
                              height=CANVAS_HEIGHT)

# Нарисовать корпус дома.
self.canvas.create_rectangle(HOUSE_X, HOUSE_Y, HOUSE_X + HOUSE_WIDTH,
                              HOUSE_Y + HOUSE_HEIGHT)

# Нарисовать крышу.
self.canvas.create_polygon(ROOF_X, ROOF_Y, ROOF_APEX_X, ROOF_APEX_Y,
                            ROOF_X + ROOF_WIDTH, ROOF_Y,
                            ROOF_X, ROOF_Y)

# Нарисовать дверь.
self.canvas.create_rectangle(DOOR_X, DOOR_Y,
                              DOOR_X + DOOR_WIDTH, DOOR_Y + DOOR_HEIGHT)

# Нарисовать окно 1.
self.canvas.create_rectangle(WINDOW1_X, WINDOW1_Y,
                              WINDOW1_X + WINDOW_WIDTH,
                              WINDOW1_Y + WINDOW_HEIGHT)

# Нарисовать окно 2.
self.canvas.create_rectangle(WINDOW2_X, WINDOW2_Y,
                              WINDOW2_X + WINDOW_WIDTH,
                              WINDOW2_Y + WINDOW_HEIGHT)

# Нарисовать солнце.
self.canvas.create_oval(SUN_X, SUN_Y,
                        SUN_X + SUN_WIDTH, SUN_Y + SUN_HEIGHT,
                        fill='yellow')

# Упаковать холст.
self.canvas.pack()

# Запустить главный цикл.
tkinter.mainloop()

# Создать экземпляр класса MyGUI.
my_gui = MyGUI()

```

Упражнение по программированию 13.9. Возраст дерева.

```

import tkinter

# Именованные константы.
CANVAS_WIDTH = 430
CANVAS_HEIGHT = 260

```

```

RING1_X1 = 50
RING1_Y1 = 90
RING1_WIDTH = 100
RING1_HEIGHT = 80

RING2_X1 = 40
RING2_Y1 = 70
RING2_WIDTH = 180
RING2_HEIGHT = 120

RING3_X1 = 30
RING3_Y1 = 50
RING3_WIDTH = 260
RING3_HEIGHT = 160

RING4_X1 = 20
RING4_Y1 = 30
RING4_WIDTH = 340
RING4_HEIGHT = 200

RING5_X1 = 10
RING5_Y1 = 10
RING5_WIDTH = 420
RING5_HEIGHT = 240

class MyGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать элемент интерфейса Canvas.
        self.canvas = tkinter.Canvas(self.main_window, width=CANVAS_WIDTH,
                                     height=CANVAS_HEIGHT)

        # Нарисовать первое кольцо.
        self.canvas.create_oval(RING1_X1, RING1_Y1,
                                RING1_X1 + RING1_WIDTH, RING1_Y1 + RING1_HEIGHT)
        self.canvas.create_text(RING1_X1 + RING1_WIDTH - 10,
                                RING1_Y1 + RING1_HEIGHT - int((RING1_HEIGHT / 2)),
                                text='1 род', anchor=tkinter.E)

        # Нарисовать второе кольцо.
        self.canvas.create_oval(RING2_X1, RING2_Y1,
                                RING2_X1 + RING2_WIDTH, RING2_Y1 + RING2_HEIGHT)
        self.canvas.create_text(RING2_X1 + RING2_WIDTH - 10,
                                RING2_Y1 + RING2_HEIGHT - int((RING2_HEIGHT / 2)),
                                text='2 рода', anchor=tkinter.E)

        # Нарисовать третье кольцо.

```

```

self.canvas.create_oval(RING3_X1, RING3_Y1,
                        RING3_X1 + RING3_WIDTH, RING3_Y1 + RING3_HEIGHT)
self.canvas.create_text(RING3_X1 + RING3_WIDTH - 10,
                        RING3_Y1 + RING3_HEIGHT - int((RING3_HEIGHT / 2)),
                        text='3 года', anchor=tkinter.E)

# Нарисовать четвертое кольцо.
self.canvas.create_oval(RING4_X1, RING4_Y1,
                        RING4_X1 + RING4_WIDTH, RING4_Y1 + RING4_HEIGHT)
self.canvas.create_text(RING4_X1 + RING4_WIDTH - 10,
                        RING4_Y1 + RING4_HEIGHT - int((RING4_HEIGHT / 2)),
                        text='4 года', anchor=tkinter.E)

# Нарисовать пятое кольцо.
self.canvas.create_oval(RING5_X1, RING5_Y1,
                        RING5_X1 + RING5_WIDTH, RING5_Y1 + RING5_HEIGHT)
self.canvas.create_text(RING5_X1 + RING5_WIDTH - 10,
                        RING5_Y1 + RING5_HEIGHT - int((RING5_HEIGHT / 2)),
                        text='5 лет', anchor=tkinter.E)

# Упаковать холст.
self.canvas.pack()

# Запустить главный цикл.
tkinter.mainloop()

# Создать экземпляр класса MyGUI.
my_gui = MyGUI()

```

Упражнение по программированию 13.10. Голливудская звезда

```

import tkinter

# Именованные константы.
CANVAS_WIDTH = 100
CANVAS_HEIGHT = 100
X1 = 50
Y1 = 1
X2 = 20
Y2 = 91
X3 = 97
Y3 = 35
X4 = 2
Y4 = 35
X5 = 79
Y5 = 91
TEXT_X = 50
TEXT_Y = 35

class MyGUI:

```

```

def __init__(self):
    # Создать главное окно.
    self.main_window = tkinter.Tk()

    # Создать элемент интерфейса Canvas.
    self.canvas = tkinter.Canvas(self.main_window, width=CANVAS_WIDTH, height=CANVAS_HEIGHT)

    # Нарисовать звезду.
    self.canvas.create_polygon(X1, Y1, X2, Y2, X3, Y3, X4, Y4, X5, Y5, fill='white')

    # Написать имя
    self.canvas.create_text(TEXT_X, TEXT_Y, text='Ваше имя',
                             anchor=tkinter.N, fill='black')

    # Упаковать холст.
    self.canvas.pack()

    # Запустить главный цикл.
    tkinter.mainloop()

# Создать экземпляр класса MyGUI.
my_gui = MyGUI()

```

Упражнение по программированию 13.11. Контур транспортного средства

```

import tkinter

# Именованные константы
CANVAS_WIDTH = 410
CANVAS_HEIGHT = 300
X1 = 10
Y1 = 250
X2 = 10
Y2 = 150
X3 = 100
Y3 = 140
X4 = 120
Y4 = 80
X5 = 200
Y5 = 80
X6 = 200
Y6 = 140
X7 = 400
Y7 = 140
X8 = 400
Y8 = 250
X9 = 350
Y9 = 250
X10 = 340
Y10 = 220

```



```

X11 = 300
Y11 = 220
X12 = 290
Y12 = 250
X13 = 110
Y13 = 250
X14 = 100
Y14 = 220
X15 = 60
Y15 = 220
X16 = 50
Y16 = 250
BTIRE_X1 = 290
BTIRE_Y1 = 210
BTIRE_X2 = 350
BTIRE_Y2 = 270
FTIRE_X1 = 50
FTIRE_Y1 = 210
FTIRE_X2 = 110
FTIRE_Y2 = 270

class MyGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать элемент интерфейса Canvas.
        self.canvas = tkinter.Canvas(self.main_window,
                                     width=CANVAS_WIDTH,
                                     height=CANVAS_HEIGHT)

        # Нарисовать контур транспортного средства
        self.canvas.create_polygon(X1, Y1, X2, Y2, X3, Y3, X4, Y4, X5, Y5,
                                  X6, Y6, X7, Y7, X8, Y8, X9, Y9, X10, Y10,
                                  X11, Y11, X12, Y12, X13, Y13, X14, Y14,
                                  X15, Y15, X16, Y16)

        # Нарисовать заднюю шину.
        self.canvas.create_oval(BTIRE_X1, BTIRE_Y1, BTIRE_X2, BTIRE_Y2)

        # Нарисовать переднюю шину.
        self.canvas.create_oval(FTIRE_X1, FTIRE_Y1, FTIRE_X2, FTIRE_Y2)

        # Упаковать холст.
        self.canvas.pack()

        # Запустить главный цикл.
        tkinter.mainloop()

```

```
# Создать экземпляр класса MyGUI.  
my_gui = MyGUI()
```

Упражнение по программированию 13.12. Солнечная система

```
import tkinter  
  
# Именованные константы.  
CANVAS_WIDTH = 520  
CANVAS_HEIGHT = 300  
SPACING = 10  
  
SUN_X = 60  
SUN_Y = 150  
SUN_RAD = 50  
SUN_COLOR = 'yellow'  
  
MERC_RAD = 10  
MERC_X = SUN_X + SUN_RAD + SPACING + MERC_RAD  
MERC_Y = 150  
MERC_COLOR = 'orange'  
  
VENUS_RAD = 15  
VENUS_X = MERC_X + MERC_RAD + SPACING + VENUS_RAD  
VENUS_Y = 150  
VENUS_COLOR = 'lightblue'  
  
EARTH_RAD = 10  
EARTH_X = VENUS_X + VENUS_RAD + SPACING + EARTH_RAD  
EARTH_Y = 150  
EARTH_COLOR = 'blue'  
  
MARS_RAD = 8  
MARS_X = EARTH_X + EARTH_RAD + SPACING + MARS_RAD  
MARS_Y = 150  
MARS_COLOR = 'red'  
  
JUPITER_RAD = 30  
JUPITER_X = MARS_X + MARS_RAD + SPACING + JUPITER_RAD  
JUPITER_Y = 150  
JUPITER_COLOR = 'brown'  
  
SAT_RAD = 30  
SAT_X = JUPITER_X + JUPITER_RAD + SPACING + SAT_RAD  
SAT_Y = 150  
SAT_COLOR = 'blanched almond'  
  
URN_RAD = 20  
URN_X = SAT_X + SAT_RAD + SPACING + URN_RAD  
URN_Y = 150
```

```

URN_COLOR = 'deep sky blue'

NEP_RAD = 17
NEP_X = URN_X + URN_RAD + SPACING + NEP_RAD
NEP_Y = 150
NEP_COLOR = 'light steel blue'

PLU_RAD = 5
PLU_X = NEP_X + NEP_RAD + SPACING + PLU_RAD
PLU_Y = 150
PLU_COLOR = 'dark olive green'

class MyGUI:
    def __init__(self):
        # Создать главное окно.
        self.main_window = tkinter.Tk()

        # Создать элемент интерфейса Canvas.
        self.canvas = tkinter.Canvas(self.main_window,
                                     width=CANVAS_WIDTH,
                                     height=CANVAS_HEIGHT)

        # Нарисовать планеты
        self.draw_planet(self.canvas, SUN_X, SUN_Y, SUN_RAD, SUN_COLOR, 'Солнце')
        self.draw_planet(self.canvas, MERC_X, MERC_Y, MERC_RAD, MERC_COLOR, 'Меркурий')
        self.draw_planet(self.canvas, VENUS_X, VENUS_Y, VENUS_RAD, VENUS_COLOR, 'Венера')
        self.draw_planet(self.canvas, EARTH_X, EARTH_Y, EARTH_RAD, EARTH_COLOR, 'Земля')
        self.draw_planet(self.canvas, MARS_X, MARS_Y, MARS_RAD, MARS_COLOR, 'Марс')
        self.draw_planet(self.canvas, JUPITER_X, JUPITER_Y, JUPITER_RAD, JUPITER_COLOR, 'Юпитер')
        self.draw_ringed_planet(self.canvas, SAT_X, SAT_Y, SAT_RAD, SAT_COLOR, 'Сатурн')
        self.draw_planet(self.canvas, URN_X, URN_Y, URN_RAD, URN_COLOR, 'Уран')
        self.draw_planet(self.canvas, NEP_X, NEP_Y, NEP_RAD, NEP_COLOR, 'Плутон')
        self.draw_planet(self.canvas, PLU_X, PLU_Y, PLU_RAD, PLU_COLOR, 'Плутон')

        # Упаковать холст.
        self.canvas.pack()

        # Запустить главный цикл.
        tkinter.mainloop()

# Метод draw_planet чертит круг с центральной точкой в
# x,y на холсте. Параметр radius задает радиус круга,
# параметр color задает цвет заполнения круга.
def draw_planet(self, canvas, x, y, radius, color, name):
    x1 = int(x) - int(radius)
    y1 = int(y) - int(radius)
    x2 = int(x) + int(radius)
    y2 = int(y) + int(radius)
    canvas.create_oval(x1, y1, x2, y2, fill=color)
    canvas.create_text(x, y2, text=name, anchor=tkinter.N)

```

```

# Метод draw_ringed_planet рисует обведенную кольцом планету.
def draw_ringed_planet(self, canvas, x, y, radius, color, name):
    # Нарисовать кольцо.
    ring_x1 = int(x) - int(radius) - SPACING
    ring_y1 = int(y) - int(radius) + SPACING
    ring_x2 = int(x) + int(radius) + SPACING
    ring_y2 = int(y) + int(radius) - SPACING
    canvas.create_oval(ring_x1, ring_y1, ring_x2, ring_y2)

    # Нарисовать планету.
    self.draw_planet(canvas, x, y, radius, color, name)

# Создать экземпляр класса MyGUI.
my_gui = MyGUI()

```