МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №4.8
Обработка событий и рисование в Tkinter
по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-21-1
Лысенко И.А. « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2024

Цель работы: приобретение навыков улучшения графического интерфейса пользователя GUI с помощью обработки событий и рисования, реализованных в пакете Tkinter языка программирования Python версии 3.x.

Ход работы:

1. Создал репозиторий на GitHub: https://github.com/IsSveshuD/OOP_Lab_4.8.git .
2. Выполнил задание 1.

```python
import tkinter as tk

"""
Решите задачу: напишите программу, состоящую из двух списков Listbox .
В первом будет, например, перечень товаров, заданный программно. Второй
изначально пуст, пусть это будет перечень покупок. При клике на одну
кнопку товар должен переходить из одного списка в другой. При клике на
вторую кнопку - возвращаться (человек передумал покупать).
Предусмотрите возможность множественного выбора элементов списка
и их перемещения.
"""


class ShopApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Список покупок")

        self.available_items = ["Мороженное", "Огурцы", "Помидоры",
                                "Масло", "Какао", "Кофе", "Чай", "Сахар"]

        self.shop_list = []

        self.available_listbox = tk.Listbox(root, selectmode=tk.MULTIPLE)
        for item in self.available_items:
            self.available_listbox.insert(tk.END, item)
        self.available_listbox.pack(padx=10, pady=10, side=tk.LEFT,
                                    fill=tk.BOTH, expand=True)

        self.shop_listbox = tk.Listbox(root, selectmode=tk.MULTIPLE)
        self.shop_listbox.pack(padx=10, pady=10, side=tk.RIGHT,
                               fill=tk.BOTH, expand=True)
```

```
        self.add_btn = tk.Button(root, text=">>>",
                                 command=self.add_shop_list)
        self.add_btn.pack(pady=(10, 0), fill=tk.X)

        self.remove_btn = tk.Button(root, text="<<<",
                                    command=self.remove_shop_list)
        self.remove_btn.pack(pady=(0, 10), fill=tk.X)

    def add_shop_list(self):
        selected_items = list(self.available_listbox.curselection())
        selected_items.reverse()
        for index in selected_items:
            item = self.available_items[index]
            if item not in self.shop_list:
                self.shop_list.append(item)
                self.shop_listbox.insert(tk.END, item)
        for index in selected_items:
            self.available_listbox.delete(index)

    def remove_shop_list(self):
        selected_items = list(self.shop_listbox.curselection())
        selected_items.reverse()
        for index in selected_items:
            item = self.shop_list[index]
            self.shop_list.remove(item)
            self.shop_listbox.delete(index)
            self.available_listbox.insert(tk.END, item)

if __name__ == "__main__":
    root = tk.Tk()
    app = ShopApp(root)
    root.mainloop()
```

Рисунок 1 – Задание 1

3. Получил следующий результат работы задания 1.

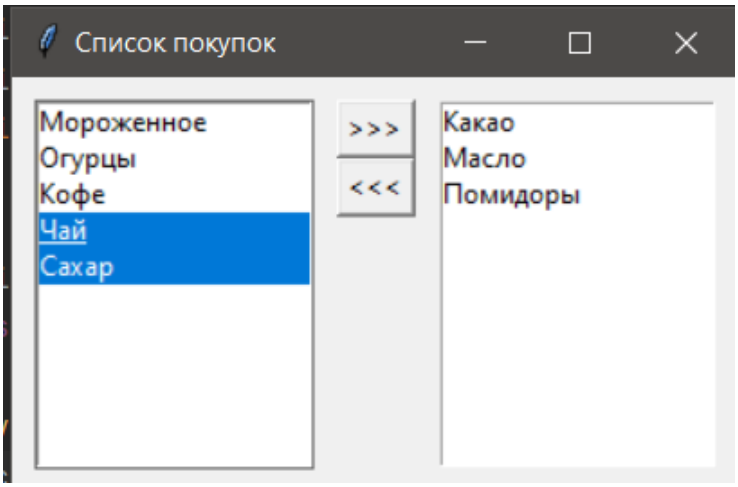

Рисунок 2 – Результат работы задания 1

4. Выполнил задание 2.

```
import tkinter as tk

"""
Решите задачу: напишите программу по следующему описанию. Нажатие
Enter в однострочном текстовом поле приводит к перемещению текста
из него в список (экземпляр Listbox ). При двойном клике
( <Double-Button-1> ) по элементу-строке списка, она должна
копироваться в текстовое поле.
"""


def add_to_listbox(event):
    text = entry.get()
    if text:
        listbox.insert(tk.END, text)
        entry.delete(0, tk.END)


def copy_to_entry(event):
    selected_text = listbox.get(tk.ACTIVE)
    entry.delete(0, tk.END)
    entry.insert(0, selected_text)


root = tk.Tk()
root.title("Пример программы")


entry = tk.Entry(root)
entry.pack(pady=10)
entry.bind("<Return>", add_to_listbox)
```

```
listbox = tk.Listbox(root, selectmode=tk.SINGLE)
listbox.pack(expand=True, fill=tk.BOTH)
listbox.bind("<Double-Button-1>", copy_to_entry)

root.mainloop()
```

Рисунок 3 – Задание 2

5. Получил следующий результат работы задания 2.

Рисунок 4 – Результат работы задания 2

6. Выполнил задание 3.

```python
import tkinter as tk

"""
Решите задачу: напишите программу по описанию. Размеры многострочного
текстового поля определяются значениями, введенными в однострочные
текстовые поля. Изменение размера происходит при нажатии мышью на кнопку,
а также при нажатии клавиши Enter. Цвет фона экземпляра Text светлосерый
( lightgrey ), когда поле не в фокусе, и белый, когда имеет фокус.
Событие получения фокуса обозначается как <FocusIn> , потери - как <FocusOut> .
Для справки: фокус перемещается по виджетам при нажатии Tab, Ctrl+Tab,
Shift+Tab, а также при клике по ним мышью (к кнопкам последнее не относится).
"""


def change_text_size(event=None):
    try:
        width = int(width_entry.get())
        height = int(height_entry.get())
        text.config(width=width, height=height)
    except ValueError:
        pass


def on_focus_in(event):
    text.config(bg='white')


def on_focus_out(event):
    text.config(bg='lightgrey')

root = tk.Tk()
root.title("Resizable Text Field")

width_entry = tk.Entry(root)
width_entry.grid(row=0, column=0, padx=5, pady=5)

height_entry = tk.Entry(root)
height_entry.grid(row=1, column=0, padx=5, pady=5)

text = tk.Text(root, wrap='word', bg='lightgrey', relief='solid')
text.grid(row=2, column=0, columnspan=2, padx=5, pady=5, sticky='nsew')

resize_button = tk.Button(root, text="Изменить", command=change_text_size)
resize_button.grid(row=0, column=2, rowspan=2, padx=5, pady=5, sticky='nsew')

text.bind("<FocusIn>", on_focus_in)
text.bind("<FocusOut>", on_focus_out)

root.bind("<Tab>", lambda e: root.focus_get().event_generate("<FocusOut>"))
root.bind("<Shift-Tab>", lambda e: root.focus_get().event_generate(
    "<FocusOut>"))

root.bind("<Return>", change_text_size)

root.columnconfigure(1, weight=1)
root.rowconfigure(2, weight=1)

root.mainloop()
```

Рисунок 5 – Задание 3

7. Получил следующий результат работы задания 3.

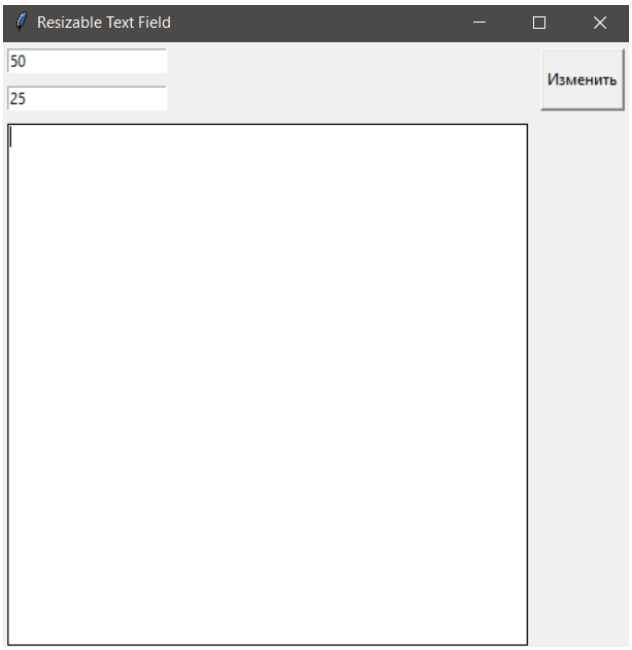

Рисунок 6 – Результат работы задания 3

8. Выполнил задание 4.

```
import tkinter as tk


def draw_house(canvas):
    canvas.create_rectangle(50, 150, 250, 300, fill="lightblue",
                            outline="lightblue")
    canvas.create_polygon(25, 150, 150, 50, 275, 150, fill="lightblue")


def draw_sun(canvas):
    canvas.create_oval(230, 20, 280, 70, fill="orange", outline="yellow")


def draw_grass(canvas, num_lines=20, incline=10):
    x_start = 10
    y_start = 300
    y_end = 275
    color = "green"

    for _ in range(num_lines):
        x_end = x_start + incline
        canvas.create_line(x_start, y_start, x_end, y_end, fill=color)
        x_start += incline + 5
```

```
def main():
    root = tk.Tk()
    root.title("Домик")

    canvas = tk.Canvas(root, width=300, height=300)
    canvas.pack()

    draw_house(canvas)
    draw_sun(canvas)
    draw_grass(canvas)

    root.mainloop()


if __name__ == "__main__":
    main()
```

Рисунок 7 – Задание 4

9. Получил следующий результат работы задания 4.

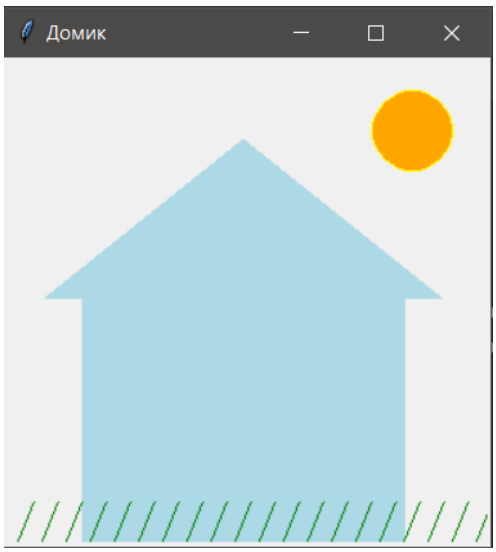

Рисунок 8 – Результат работы задания 4

10. Выполнил задание 5.

```
import tkinter as tk

"""
Решите задачу: в данной программе создается анимация круга, который
движется от левой границы холста до правой:
Выражение c.coords(ball) возвращает список текущих координат объекта
(в данном случае это ball). Третий элемент списка соответствует его
второй координате x. Метод after вызывает функцию, переданную вторым
аргументом, через количество миллисекунд, указанных первым аргументом.
Изучите приведенную программу и самостоятельно запрограммируйте постепенное
движение фигуры в ту точку холста, где пользователь кликает левой кнопкой мыши.
Координаты события хранятся в его атрибутах x и y ( event.x , event.y ).
"""
```

```python
def move_towards_point(target_x, target_y):
    current_x, current_y, _, _ = c.coords(ball)

    delta_x = target_x - current_x
    delta_y = target_y - current_y

    distance = (delta_x ** 2 + delta_y ** 2) ** 0.5

    normalized_delta_x = delta_x / distance
    normalized_delta_y = delta_y / distance

    step = 2

    if distance > step:
        c.move(ball, normalized_delta_x * step, normalized_delta_y * step)

        root.after(10, lambda: move_towards_point(target_x, target_y))


root = tk.Tk()
c = tk.Canvas(root, width=300, height=200, bg="white")
c.pack()

ball = c.create_oval(0, 100, 40, 140, fill='green')


def on_click(event):
    move_towards_point(event.x, event.y)
```

```python
c.bind("<Button-1>", on_click)

root.mainloop()
```

Рисунок 9 – Задание 5

11. Получил следующий результат работы задания 5.

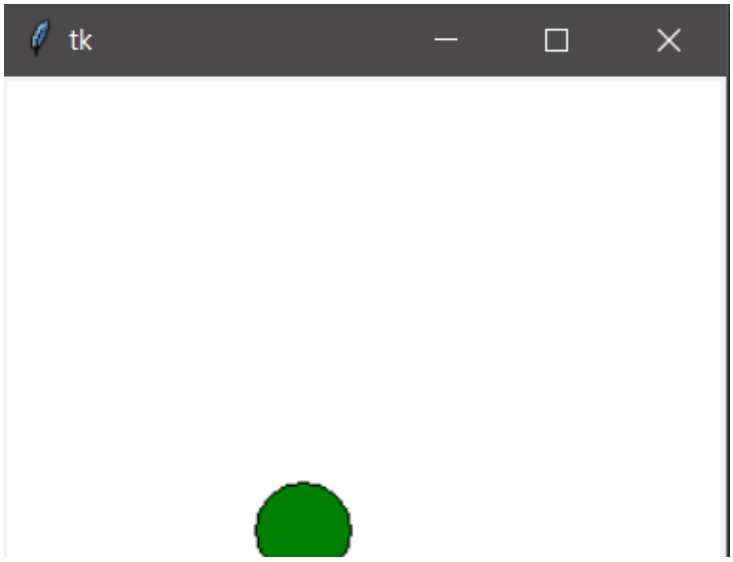

Рисунок 10 – Результат работы задания 5

Ответы на вопросы:

1. Каково назначение виджета ListBox?

Виджет ListBox в Tkinter предназначен для отображения списка элементов, из которых пользователь может выбирать один или несколько. Этот виджет предоставляет прокручиваемый список элементов.

2. Каким образом осуществляется связывание событие или действие с виджетом Tkinter?

Связывание событий в Tkinter осуществляется с использованием метода bind(). Этот метод позволяет привязывать функции-обработчики к определенным событиям, таким как нажатие кнопки, перемещение мыши и другие.

3. Какие существуют типы событий в Tkinter? Приведите примеры.

Существует множество типов событий в Tkinter. Некоторые из них включают:

Button-1: Левая кнопка мыши.

Button-2: Средняя кнопка мыши (если она есть).

Button-3: Правая кнопка мыши.

Motion: Движение мыши.

KeyPress: Нажатие клавиши на клавиатуре.

KeyRelease: Отпускание клавиши на клавиатуре.

4. Как обрабатываются события в Tkinter?

События обрабатываются путем привязки функций-обработчиков к определенным событиям с использованием метода bind() или через метод after() для периодического выполнения действий.

5. Как обрабатываются события мыши в Tkinter?

События мыши, такие как нажатие кнопок или движение мыши, обрабатываются путем привязки функций-обработчиков к соответствующим событиям с использованием метода bind().

6. Каким образом можно отображать графические примитивы в Tkinter?

Для отображения графических примитивов в Tkinter используется виджет Canvas. Этот виджет позволяет рисовать линии, прямоугольники, окружности и другие графические элементы.

7. Перечислите основные методы для отображения графических примитивов в Tkinter.

Некоторые основные методы для работы с графическими примитивами на холсте (Canvas):

create_line(): Создает линию.

create_rectangle(): Создает прямоугольник.

create_oval(): Создает овал.

create_text(): Создает текст.

8. Каким образом можно обратиться к ранее созданным фигурам на холсте?

Каждая фигура на холсте имеет уникальный идентификатор, который возвращается методами создания фигур. Идентификаторы могут использоваться для обращения к ранее созданным фигурам.

9. Каково назначение тэгов в Tkinter?

Тэги в Tkinter используются для группировки и идентификации набора объектов на холсте. Они позволяют применять действия к определенным группам объектов. Тэги могут быть присвоены при создании объекта на холсте или позднее с использованием метода addtag_withtag().

Вывод: в результате выполнения лабораторной работы были приобретены навыки улучшения графического интерфейса пользователя GUI с помощью обработки событий и рисования, реализованных в пакете Tkinter языка программирования Python версии 3.x.