

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
дисциплины «Объектно-ориентированное программирование»
Вариант 3

Выполнил:
Говоров Егор Юрьевич
3 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

(подпись)

Проверил:
Воронкин Роман Александрович,
доцент департамента цифровых,
робототехнических систем и
электроники

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

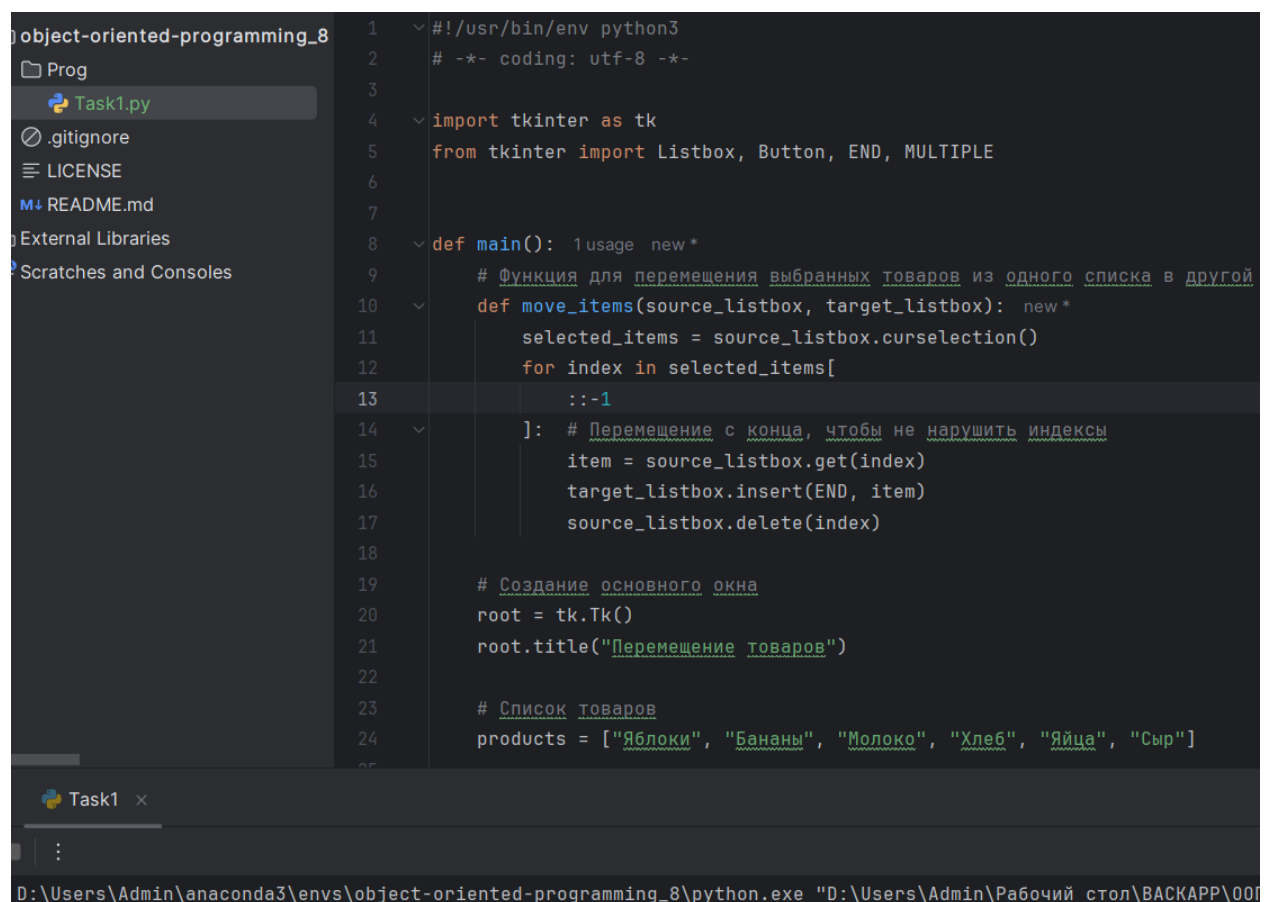
Тема: Обработка событий и рисование в Tkinter

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

Ссылка на репозиторий: https://github.com/Artorias1469/object-oriented-programming_8.git

Ход работы:

Задача №1. 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.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import tkinter as tk
5  from tkinter import Listbox, Button, END, MULTIPLE
6
7
8  def main():
9      # Функция для перемещения выбранных товаров из одного списка в другой
10     def move_items(source_listbox, target_listbox):
11         selected_items = source_listbox.curselection()
12         for index in selected_items[
13             :-1
14         ]:
15             # Перемещение с конца, чтобы не нарушить индексы
16             item = source_listbox.get(index)
17             target_listbox.insert(END, item)
18             source_listbox.delete(index)
19
20     # Создание основного окна
21     root = tk.Tk()
22     root.title("Перемещение товаров")
23
24     # Список товаров
25     products = ["Яблоки", "Бананы", "Молоко", "Хлеб", "Яйца", "Сыр"]
```

Рисунок 1. Код выполнения программы №1

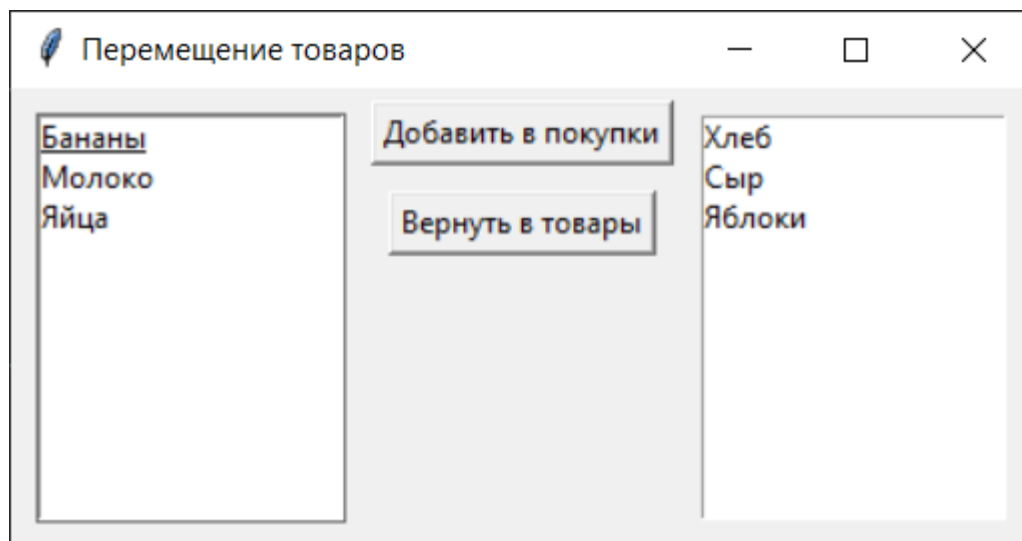


Рисунок 2. Результат выполнения программы №1

Задача №2. 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.

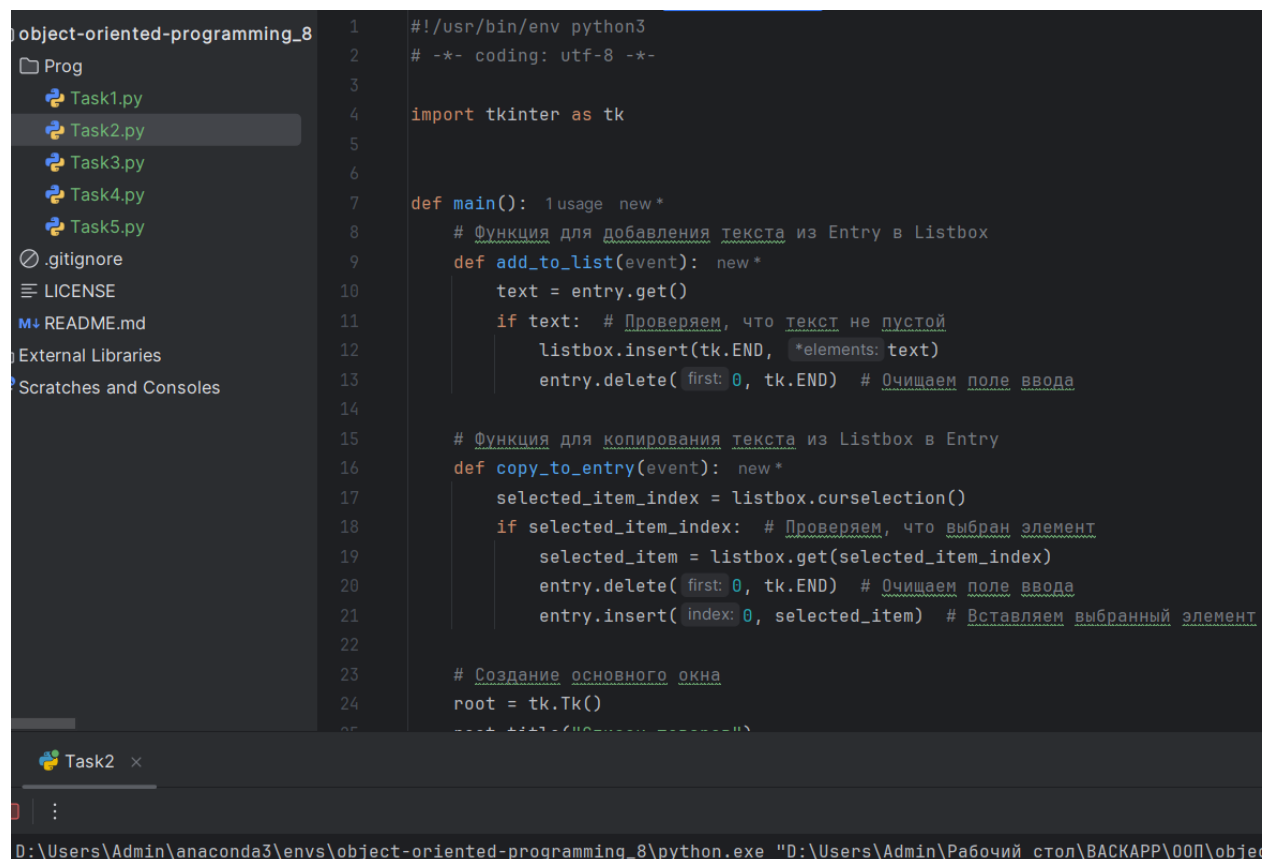


Рисунок 3. Код выполнения программы №2

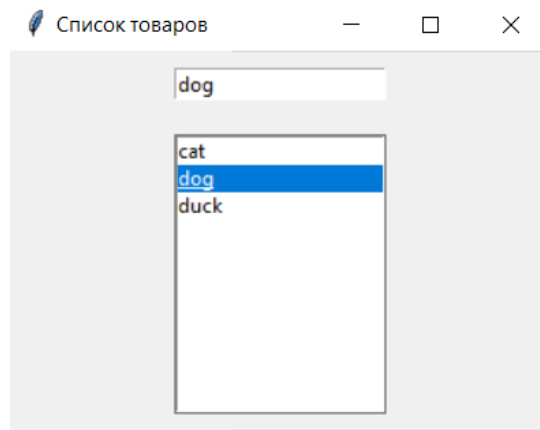


Рисунок 4. Результат выполнения программы №2

Задача №3. 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).

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import tkinter as tk
5
6
7  def main():
8      usage = new *
9      def update_text_area():
10         new *
11         try:
12             rows = int(rows_entry.get())
13             cols = int(cols_entry.get())
14             text_area.config(height=rows, width=cols)
15         except ValueError:
16             pass # Игнорируем ошибки преобразования
17
18     def on_focus_in(event):
19         new *
20         text_area.config(bg="white")
21
22     def on_focus_out(event):
23         new *
24         text_area.config(bg="lightgrey")
25
26     # Создаем главное окно
27     root = tk.Tk()
28     root.title("Изменение размера текстового поля")
29
30     # ... (rest of the code for creating widgets and binding events)

```

Рисунок 5. Код выполнения программы №3

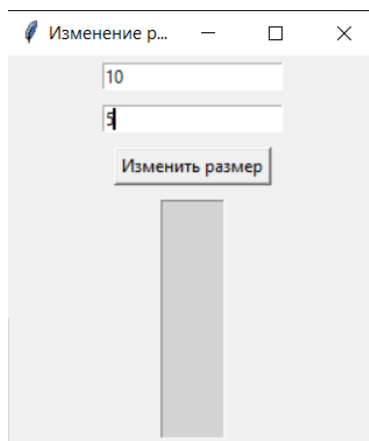


Рисунок 6. Результат выполнения программы №3

Задача №4. Создайте на холсте подобное изображение:



Рисунок 7. Образец выполнения

```

object-oriented-programming_8  1  #!/usr/bin/env python3
                                2  # -*- coding: utf-8 -*-
                                3
                                4  import tkinter as tk
                                5
                                6
                                7  def main(): 1 usage new *
                                8      # Создаем окно
                                9      root = tk.Tk()
                               10      root.title("Рисунок с домом и травой")
                               11
                               12      # Создаем холст
                               13      canvas = tk.Canvas(root, width=300, height=300, bg="white")
                               14      canvas.pack()
                               15
                               16      # Рисуем дом (основа и крыша)
                               17      canvas.create_rectangle(
                               18          100, 150, 200, 250, fill="lightblue", outline="lightblue"
                               19      ) # Основа дома
                               20      canvas.create_polygon(
                               21          60, 150, 150, 100, 240, 150, fill="lightblue", outline="lightblue"
                               22      ) # Крыша
                               23
                               24      # Рисуем солнце
                               25      canvas.create_oval(200, 20, 220, 70, fill="yellow", outline="yellow")

```

Рисунок 8. Код выполнения программы №4

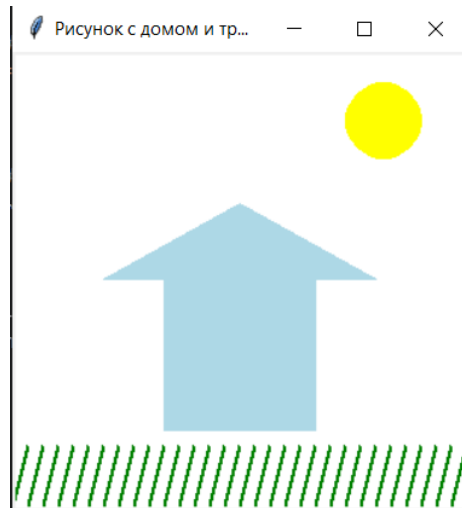


Рисунок 9. Результат выполнения программы №4

Задача №5. 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 `y`. 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`).

```

object-oriented-programming_8 1  #!/usr/bin/env python3
                                2  # -*- coding: utf-8 -*-
                                3
                                4  from tkinter import *
                                5
                                6
                                7  def main(): 1 usage new *
                                8      def move_to_click(event): new *
                                9          """Запускаем движение шара к месту клика."""
                               10          global target_x, target_y
                               11          target_x, target_y = event.x, event.y
                               12          move_ball()
                               13
                               14      def move_ball(): new *
                               15          """Перемещаем шар к указанным координатам."""
                               16          global target_x, target_y
                               17          current_coords = c.coords(ball)
                               18          current_x, current_y = (current_coords[0] + current_coords[2]) / 2, (
                               19              current_coords[1] + current_coords[3]
                               20          ) / 2

```

Task4 x

D:\Users\Admin\anaconda3\envs\object-oriented-programming_8\python.exe "D:\Users\Admin\Рабочий стол\BACKAPP\00П\ob

Рисунок 10. Код выполнения программы №5

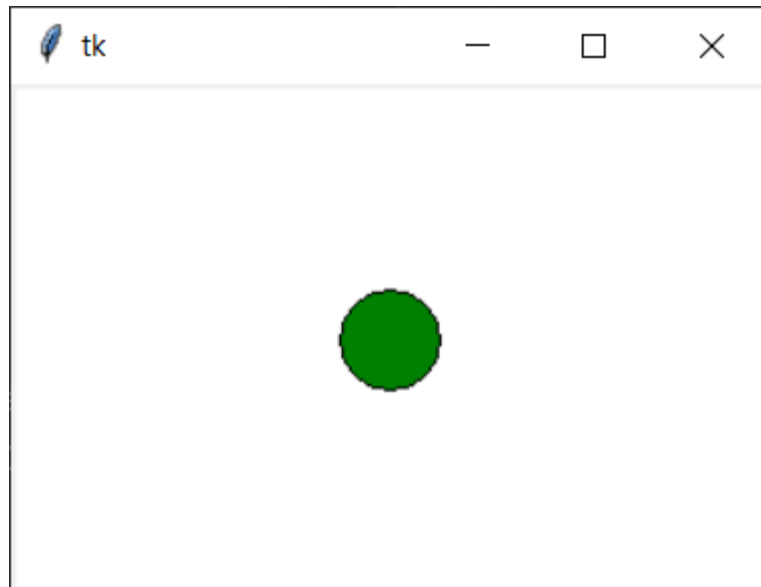


Рисунок 11. Результат выполнения программы №5

Ответы на контрольные вопросы:

1. Каково назначение виджета ListBox?

Виджет Listbox используется для отображения списка элементов, из которого пользователь может выбирать один или несколько элементов. Это удобно для создания интерфейсов, где требуется выбор из множества вариантов.

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

Связывание событий с виджетами осуществляется с помощью метода bind. Этот метод связывает конкретное событие (например, щелчок мыши) с функцией-обработчиком, которая вызывается при возникновении этого события.

3. Какие существуют типы событий в Tkinter? Приведите примеры. События в Tkinter включают:

Мышь:

- 1) <Button-1> — левый щелчок мыши.
- 2) <Button-3> — правый щелчок мыши.
- 3) <Double-Button-1> — двойной щелчок левой кнопкой мыши.
- 4) <Motion> — движение мыши. Клавиатура:

- 1) <Key> — нажатие любой клавиши.
- 2) <KeyPress-a> — нажатие конкретной клавиши (например, "a").
- 3) <Return> — нажатие клавиши Enter. Фокус:
 - 1) <FocusIn> — получение фокуса виджетом.
 - 2) <FocusOut> — потеря фокуса. Окно:
 - 1) <Configure> — изменение размера окна.
 - 2) <Destroy> — закрытие окна.
4. Как обрабатываются события в Tkinter?

События обрабатываются с помощью функций-обработчиков, которые привязываются к конкретным событиям с помощью метода `bind`. При срабатывании события в обработчик передается объект события (`event`), содержащий информацию о типе события, его координатах, кнопках мыши и т.д.

5. Как обрабатываются события мыши в Tkinter?

Для обработки событий мыши используются идентификаторы событий, такие как:

- 1) <Button-1> — левый щелчок.
- 2) <Button-2> — средний щелчок.
- 3) <Button-3> — правый щелчок.
- 4) <Motion> — движение мыши. Эти события можно привязывать к

виджетам с помощью `bind`.

6. Каким образом можно отображать графические примитивы в Tkinter? Для отображения графических примитивов используется виджет `Canvas`.

На холсте можно рисовать линии, прямоугольники, овалы, многоугольники и текст с помощью методов `Canvas`.

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

`create_line(x1, y1, x2, y2, **options)` — рисует линию. `create_rectangle(x1, y1, x2, y2, **options)` — рисует прямоугольник. `create_oval(x1, y1, x2, y2, **options)` — рисует овал. `create_polygon(x1, y1, ..., xn, yn, **options)` — рисует многоугольник. `create_text(x, y, text="Текст", **options)` — рисует текст.

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

Каждая фигура, созданная на холсте, имеет уникальный идентификатор, который возвращается методом создания. К фигурам можно обращаться по этому идентификатору.

9. Каково назначение тэгов в Tkinter?

Тэги используются для группировки объектов на холсте и управления ими одновременно. Тэги задаются с помощью параметра `tags` при создании фигуры.

Вывод: в ходе выполнения 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.