**№7**

**Лабораторная работа**

**«Использование стандартных диалоговых окон. Обработка событий. Использование метода bind»**

**Вариант 0**

**Цель работы:**

Получить навыки по использованию в процессе разработки GUI программ на Python стандартных диалоговых окон, обработки событий и использования метода bind

**Краткая теория:**

Диалоговые окна предназначены для формирования дружественного интерфейса взаимодействия пользователя в процессе работы с GUI приложением на Python. Диалоговые окна могут быть модальными и немодальными Диалоговые окна, как элементы графического интерфейса, предназначены для вывода сообщений пользователю, получения от него какой-либо информации, а также управления процессом работы программы.

В tkinter с помощью метода bind() между собой связываются виджет, событие и действие. Например, виджет – кнопка, событие – клик по ней левой кнопкой мыши, действие – отправка сообщения. Другой пример: виджет – текстовое поле, событие – нажатие Enter, действие – получение текста из поля методом get() для последующей обработки программой. Действие оформляют как функцию (или метод), которая вызываются при наступлении события. Один и тот же виджет можно связать с несколькими событиями.

**ХОД РАБОТЫ**

Для реализации графического интерфейса воспользуемся стандартным пакетом Tkinter, входящим в состав Python 3 (если установлен Python, то дополнительно не надо

Для реализации проекта потребуется сделать несколько импортов функций модуля tkinter.

* filedialog -это модуль с функциями открытия и сохранения диалоговых окон.
* messagebox предоставляет базовый класс шаблона, а также множество удобных методов для часто используемых конфигураций.
* simpledialog содержит классы удобства и функции для создания простых модальных диалогов для получения значения от пользователя.

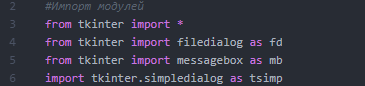


Рисунок 1 – импорт модулей

Создадим функцию, которая будет отвечать за открытие файла и считывание из него текста для записи в текстовое поле. Результат полученный из фала поместим в переменную и сделаем ее глобальной.

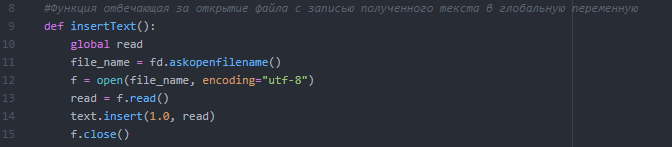


Рисунок 2 – функция работы с открытым фалом

Далее сразу же пропишем возможность сохранения текста из текстового поля в файлы.

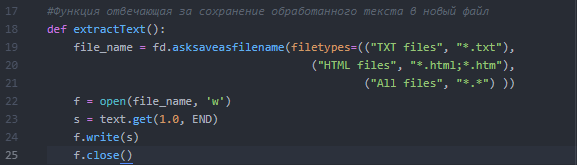


Рисунок 3 – функция сохранения текста в файлы

Далее создадим еще одну функцию, в которой будут производиться преобразования. Также в ней пропишем открытие дополнительного окна, в котором пользователь должен прописать свои данные.

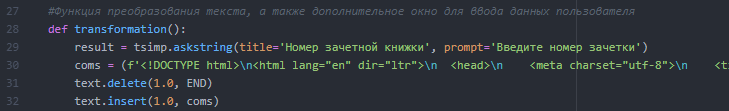


Рисунок 4 – функция преобразования

Теперь вернемся к основному окну программы и пропишем название программы, отображение всех кнопок и текстового поля и т.д.

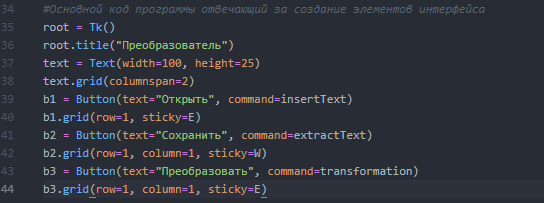


Рисунок 5 - код интерфейса программы

И последняя строчка кода метод mainloop() объекта Tk, запускающий главный цикл обработки событий, что в том числе приводит к отображению главного окна со всеми элементами.



Рисунок 6 – метод mainloop()

**ЛИСТИНГ ПРОГРАММЫ**

|  |
| --- |
| #-\*- coding: utf-8 -\*-#  #Импорт модулей  from tkinter import \*  from tkinter import filedialog as fd  from tkinter import messagebox as mb  import tkinter.simpledialog as tsimp  #Функция отвечающая за открытие файла с записью полученного текста в глобальную переменную  def insertText():  global read  file\_name = fd.askopenfilename()  f = open(file\_name, encoding="utf-8")  read = f.read()  text.insert(1.0, read)  f.close()  #Функция отвечающая за сохранение обработанного текста в новый файл  def extractText():  file\_name = fd.asksaveasfilename(filetypes=(("TXT files", "\*.txt"),  ("HTML files", "\*.html;\*.htm"),  ("All files", "\*.\*") ))  f = open(file\_name, 'w')  s = text.get(1.0, END)  f.write(s)  f.close()  #Функция преобразования текста, а также дополнительное окно для ввода данных пользователя  def transformation():  result = tsimp.askstring(title='Номер зачетной книжки', prompt='Введите номер зачетки')  coms = (f'<!DOCTYPE html>\n<html lang="en" dir="ltr">\n <head>\n <meta charset="utf-8">\n <title>{result}</title>\n </head>\n <body>\n <center><h1>{read}'f'</h1></center>\n </body>\n</html>')  text.delete(1.0, END)  text.insert(1.0, coms)  #Основной код программы отвечающий за создание элементов интерфейса  root = Tk()  root.title("Преобразователь")  text = Text(width=100, height=25)  text.grid(columnspan=2)  b1 = Button(text="Открыть", command=insertText)  b1.grid(row=1, sticky=E)  b2 = Button(text="Сохранить", command=extractText)  b2.grid(row=1, column=1, sticky=W)  b3 = Button(text="Преобразовать", command=transformation)  b3.grid(row=1, column=1, sticky=E)  root.mainloop() |

**КОНТРОЛЬНЫЙ ВАРИАНТ**

