Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
По дисциплине: «Современные системы программирования»

Выполнил:  
студент 3 курса  
группы ИИ-21(I)  
Кирилович А. А.

Проверила:  
Якимук А.В.

Брест 2023

**Цель:** освоить создание консольных приложений; освоить работу с файлами; научиться строить план тестирования приложения.

**Ход работы**

**Задание 1**

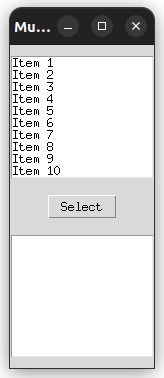
Разработать приложение, обеспечивающее возможность множественного выбора элементов из списка. Выбранные элементы должны образовывать строку текста и помещаться в текстовое поле. Предусмотреть возможность вывода сообщения в диалоговое окно *(Dialog)* в случае, если суммарное количество символов будет превышать 100.

import tkinter as tk

from tkinter import messagebox

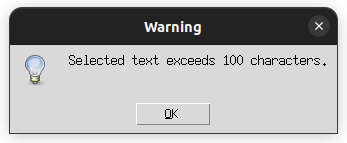
class MultipleChoiceApp:

def \_\_init\_\_(self, master):

self.master = master

self.master.title("Multiple Choice App")

self.available\_items = [f"Item {i}" for i in range(1, 20)]

self.selected\_items = []

self.available\_listbox = tk.Listbox(self.master, selectmode=tk.MULTIPLE)

for item in self.available\_items:

self.available\_listbox.insert(tk.END, item)

self.available\_listbox.pack(pady=10)

select\_button = tk.Button(self.master, text="Select", command=self.select\_items)

select\_button.pack(pady=5)

self.selected\_listbox = tk.Listbox(self.master)

self.selected\_listbox.pack(pady=10)

def select\_items(self):

selected\_indices = self.available\_listbox.curselection()

selected\_items = [self.available\_items[i] for i in selected\_indices]

selected\_text = ', '.join(selected\_items)

if len(selected\_text) > 100:

messagebox.showinfo("Warning", "Selected text exceeds 100 characters.")

else:

self.selected\_listbox.delete(0, tk.END)

self.selected\_listbox.insert(tk.END, selected\_text)

if \_\_name\_\_ == "\_\_main\_\_":

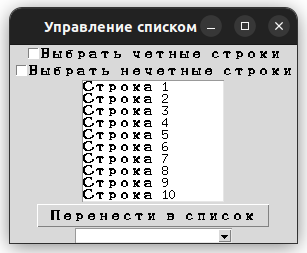
root = tk.Tk()

app = MultipleChoiceApp(root)

root.mainloop()

**Задание 2**

Разработать приложение управления списком. Вывести два флажка *(Checkbox).* При первом включенном флажке осуществляется выбор всех не четных строк, при втором включенном флажке осуществляется выбор всех чет­ных строк и перенос их в раскрывающийся список *(Choice).*

import tkinter as tk

from tkinter import ttk

class ListManagementApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Управление списком")

self.create\_widgets()

def create\_widgets(self):

self.odd\_var = tk.BooleanVar()

self.odd\_checkbox = tk.Checkbutton(self.root,

text="Выбрать четные строки",

variable=self.odd\_var,

command=self.update\_selection)

self.odd\_checkbox.pack()

self.even\_var = tk.BooleanVar()

self.even\_checkbox = tk.Checkbutton(self.root,

text="Выбрать нечетные строки",

variable=self.even\_var,

command=self.update\_selection)

self.even\_checkbox.pack()

self.listbox = tk.Listbox(self.root, selectmode=tk.MULTIPLE)

for i in range(1, 11):

self.listbox.insert(tk.END, f"Строка {i}")

self.listbox.pack()

self.move\_button = tk.Button(self.root,

text="Перенести в список",

command=self.move\_selected)

self.move\_button.pack()

self.choice\_var = tk.StringVar()

self.choice = ttk.Combobox(self.root, textvariable=self.choice\_var)

self.choice.pack()

def update\_selection(self):

self.listbox.selection\_clear(0, tk.END)

if self.odd\_var.get() and self.even\_var.get():

self.listbox.select\_set(0, tk.END)

elif self.odd\_var.get():

for i in range(1, self.listbox.size(), 2):

self.listbox.selection\_set(i)

elif self.even\_var.get():

for i in range(0, self.listbox.size(), 2):

self.listbox.selection\_set(i)

def move\_selected(self):

selected\_indices = self.listbox.curselection()

items\_to\_move = [self.listbox.get(i) for i in selected\_indices]

self.choice['values'] = items\_to\_move

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = ListManagementApp(root)

root.mainloop()

**Задание 3**

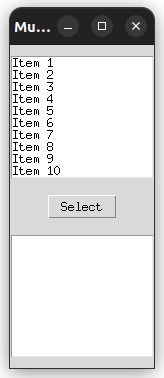
Разработать приложение, обеспечивающее возможность множественного выбора элементов из списка. Выбранные элементы должны образовывать строку текста и помещаться в текстовое поле. Предусмотреть возможность вывода сообщения в диалоговое окно *(Dialog)* в случае, если суммарное количество символов будет превышать 100.

import tkinter as tk

from tkinter import messagebox

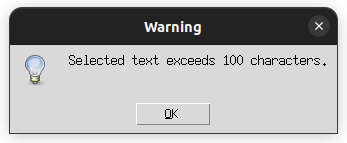
class MultipleChoiceApp:

def \_\_init\_\_(self, master):

self.master = master

self.master.title("Multiple Choice App")

self.available\_items = [f"Item {i}" for i in range(1, 20)]

self.selected\_items = []

self.available\_listbox = tk.Listbox(self.master, selectmode=tk.MULTIPLE)

for item in self.available\_items:

self.available\_listbox.insert(tk.END, item)

self.available\_listbox.pack(pady=10)

select\_button = tk.Button(self.master, text="Select", command=self.select\_items)

select\_button.pack(pady=5)

self.selected\_listbox = tk.Listbox(self.master)

self.selected\_listbox.pack(pady=10)

def select\_items(self):

selected\_indices = self.available\_listbox.curselection()

selected\_items = [self.available\_items[i] for i in selected\_indices]

selected\_text = ', '.join(selected\_items)

if len(selected\_text) > 100:

messagebox.showinfo("Warning", "Selected text exceeds 100 characters.")

else:

self.selected\_listbox.delete(0, tk.END)

self.selected\_listbox.insert(tk.END, selected\_text)

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = MultipleChoiceApp(root)

root.mainloop()

**Вывод:** в ходе лабораторной работы я научился создавать приложения с графическим интерфейсом.