Лабораторная работа №9

**Цель работы**: рассмотреть возможности библиотеки Tkinter, её базовые виджеты и изучить основные принципы создания приложений с графическим интерфейсом с помощью данной библиотеки.

**Задание:** 16)Написать GUI приложение, которое имеет кнопку «Сделать скриншот», по нажатию на которую создается скриншот экрана и вставляет в окно приложение и появляются возможности настройки яркости изображения, некоторые фильтры и поворот изображения. По нажатию на кнопку «Сохранять» должно сохраняться сконфигурированное изображение.

**Выполнение задания:** Ознакомившись с материалом приступаем к работе. В коде потребуется использовать pyautogui, который может делать скриншоты.  
  
Листинг кода:

import tkinter as tk  
from tkinter import filedialog, messagebox  
from PIL import Image, ImageTk, ImageEnhance, ImageFilter  
import pyautogui  
import os  
  
  
class ScreenshotApp:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Screenshot App")  
  
 self.image\_label = tk.Label(root)  
 self.image\_label.pack()  
  
 self.brightness\_scale = tk.Scale(root, from\_=0, to=2, resolution=0.1, orient=tk.HORIZONTAL, label="Brightness")  
 self.brightness\_scale.set(1) # Default brightness  
 self.brightness\_scale.pack()  
  
 self.filter\_var = tk.StringVar(value="None")  
 self.filter\_menu = tk.OptionMenu(root, self.filter\_var, "None", "BLUR", "CONTOUR", "DETAIL")  
 self.filter\_menu.pack()  
  
 self.rotate\_button = tk.Button(root, text="Rotate 90°", command=self.rotate\_image)  
 self.rotate\_button.pack()  
  
 self.screenshot\_button = tk.Button(root, text="Сделать скриншот", command=self.take\_screenshot)  
 self.screenshot\_button.pack()  
  
 self.save\_button = tk.Button(root, text="Сохранять", command=self.save\_image)  
 self.save\_button.pack()  
  
 self.image = None  
 self.screenshot\_path = None # Путь к сохраненному скриншоту  
  
 def take\_screenshot(self):  
 screenshot = pyautogui.screenshot()  
 # Указываем путь для сохранения скриншота  
 self.screenshot\_path = "screenshot.png"  
 screenshot.save(self.screenshot\_path) # Сохраняем скриншот в файл  
 screenshot = screenshot.resize((400, 300), Image.LANCZOS)  
 self.image = ImageTk.PhotoImage(screenshot)  
 self.image\_label.config(image=self.image)  
  
 def apply\_filters(self):  
 if self.image is None:  
 return  
  
 img = Image.open(self.screenshot\_path) # Открываем сохраненный скриншот для применения фильтров  
 img = img.resize((400, 300), Image.LANCZOS) # Уменьшаем размер изображения  
 brightness\_factor = self.brightness\_scale.get()  
 enhancer = ImageEnhance.Brightness(img)  
 img = enhancer.enhance(brightness\_factor)  
  
 filter\_choice = self.filter\_var.get()  
 if filter\_choice == "BLUR":  
 img = img.filter(ImageFilter.BLUR)  
 elif filter\_choice == "CONTOUR":  
 img = img.filter(ImageFilter.CONTOUR)  
 elif filter\_choice == "DETAIL":  
 img = img.filter(ImageFilter.DETAIL)  
  
 return img  
  
 def rotate\_image(self):  
 if self.image is None or self.screenshot\_path is None:  
 return  
  
 img = Image.open(self.screenshot\_path) # Открываем сохраненный скриншот для поворота  
 img = img.resize((400, 300), Image.LANCZOS) # Уменьшаем размер изображения  
 img = img.rotate(90, expand=True)  
 img = self.apply\_filters() or img # Применяем фильтры после поворота  
 self.image = ImageTk.PhotoImage(img)  
 self.image\_label.config(image=self.image)  
  
 def save\_image(self):  
 if self.image is None:  
 messagebox.showwarning("Warning", "Нет изображения для сохранения.")  
 return  
  
 file\_path = filedialog.asksaveasfilename(defaultextension=".png",  
 filetypes=[("PNG files", "\*.png"),  
 ("JPEG files", "\*.jpg"),  
 ("All files", "\*.\*")])  
 if file\_path:  
 img\_to\_save = Image.open(self.screenshot\_path) # Открываем сохраненный скриншот для сохранения  
 img\_to\_save = img\_to\_save.resize((400, 300), Image.LANCZOS) # Уменьшаем размер изображения  
 img\_to\_save = self.apply\_filters() or img\_to\_save # Применяем фильтры перед сохранением  
 img\_to\_save.save(file\_path)  
 messagebox.showinfo("Success", "Изображение сохранено!")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 root = tk.Tk()  
 app = ScreenshotApp(root)  
 root.mainloop()

Результаты работы кода:



Рис. 1 - Интерфейс



Рис. 2 - Сохраненный Скрин