***В.С. Иванов, А.А. Иванова,   
Кандидат Технических Наук А.А. Андреева***

**ВКЛЮЧЕНИЕ СПЕЦИАЛЬНЫХ ВОЗМОЖНОСТЕЙ WINDOWS ДЛЯ РАБОТЫ СО СТАНДАРТНОЙ АППАРАТУРОЙ**

**Аннотация**. Показан процесс разработки графического интерфейса для включения специальных возможностей Windows. Описаны функциональные возможности интерфейса, их отображение для пользователя.

**Ключевые слова:** Python, Tkinter, Windows API, ОВЗ, операционная система Windows.

***V.S. Ivanov, A.A. Ivanova,   
Candidate of Technical Sciences A.A. Andreeva***

**A WEB APPLICATION FOR BOOKING SPACES  
IN A CO-WORKING SPACE**

**Annotation.** The process of developing a graphical interface for enabling accessibility features in Windows is demonstrated. The functional capabilities of the interface and their representation for the user are described.

**Keywords****:** Python, Tkinter, Windows API, disabilities, Windows operating system.

По данным ВОЗ, сотни миллионов людей по всему миру испытывают различные формы ограничений зрения, слуха и моторики, что затрудняет их полноценное взаимодействие с компьютерными технологиями. Учитывая тенденцию быстрого развития технологий, людям с ОВЗ все больше и больше необходимо особых условий для взаимодействия с компьютером.

В условиях, когда цифровые технологии становятся неотъемлемой частью многих процессов, важность доступного интерфейса приобретает ещё большее значение. Наша программа ориентирована на повышение доступности Windows для людей с ограниченными возможностями здоровья.

Основной целью проекта является разработка программы, которая расширяет возможности доступности операционной системы Windows для пользователей с особыми потребностями, повышая удобство и комфорт при работе с компьютером для людей с нарушениями зрения, слуха и моторики.

Для создания GUI используются язык программирования Python. Основой для графического интерфейса выступила библиотека tkinter, а для взаимодействия с Windows API – ctypes.

В процессе работы над проектом были реализованы включение экранного диктора, увеличительного стекла, экранной клавиатуры, функция высокого контраста и цветные фильтры.

Программа составлена из нескольких функций, каждая из которых отвечает за определенный функционал:

* **start\_narrator()**: Запускает экранный диктор.
* **start\_magnifier()**: Запускает увеличительное стекло.
* **start\_on\_screen\_keyboard()**: Запускает экранную клавиатуру.
* **toggle\_high\_contrast()**: Включает или выключает режим высокого контраста.
* **toggle\_color\_filter()**: Переключает цветной фильтр.

Взаимодействие с Windows API показано на примере взаимодействия с функцией toggle\_hight\_contrast. В начале кода определяются константы для работы с Windows API:

SPI\_SETHIGHCONTRAST = 0x0043

SPI\_GETHIGHCONTRAST = 0x0042

SPIF\_SENDCHANGE = 0x0002

Эти константы используются для получения и установки параметров высокого контраста через функцию SystemParametersInfoW.

Также определяется структура HIGHCONTRAST, которая содержит необходимые поля для работы с режимом высокого контраста:

class HIGHCONTRAST(ctypes.Structure):

\_fields\_ = [

("cbSize", wintypes.UINT),

("dwFlags", wintypes.DWORD),

("lpszDefaultScheme", wintypes.LPWSTR)

]

Сама функция toggle\_high\_contrast для переключения режима контраста обращается к описанным ранее константам и классу:

def toggle\_high\_contrast():

hc = HIGHCONTRAST()

hc.cbSize = ctypes.sizeof(HIGHCONTRAST)

# Получаем текущие параметры

ctypes.windll.user32.SystemParametersInfoW(SPI\_GETHIGHCONTRAST, hc.cbSize, ctypes.byref(hc), 0)

# Проверяем текущее состояние и переключаем

if hc.dwFlags & 0x00000001: # Если высокий контраст включен

hc.dwFlags &= ~0x00000001 # Отключаем высокий контраст

high\_contrast\_button.config(text="Включить высокий контраст")

else: # Если высокий контраст отключен

hc.dwFlags |= 0x00000001 # Включаем высокий контраст

high\_contrast\_button.config(text="Отключить высокий контраст")

# Применяем изменения

ctypes.windll.user32.SystemParametersInfoW(SPI\_SETHIGHCONTRAST, hc.cbSize, ctypes.byref(hc), SPIF\_SENDCHANGE)

В этой функции:

* Создается экземпляр структуры HIGHCONTRAST.
* Вызывается SystemParametersInfoW для получения текущих настроек высокого контраста.
* Проверяется текущее состояние и, в зависимости от него, включается или отключается высокий контраст.
* Изменения применяются с помощью того же вызова SystemParametersInfoW.

Основной элемент графического интерфейса это окно приложения, в котором размещены кнопки для каждой функции:

root = tk.Tk()

root.title("Специальные возможности Windows")

Каждая кнопка связывается с соответствующей функцией:

narrator\_button = tk.Button(root, text="Включить экранный диктор", font=button\_font, command=start\_narrator)

magnifier\_button = tk.Button(root, text="Включить увеличительное стекло", font=button\_font, command=start\_magnifier)

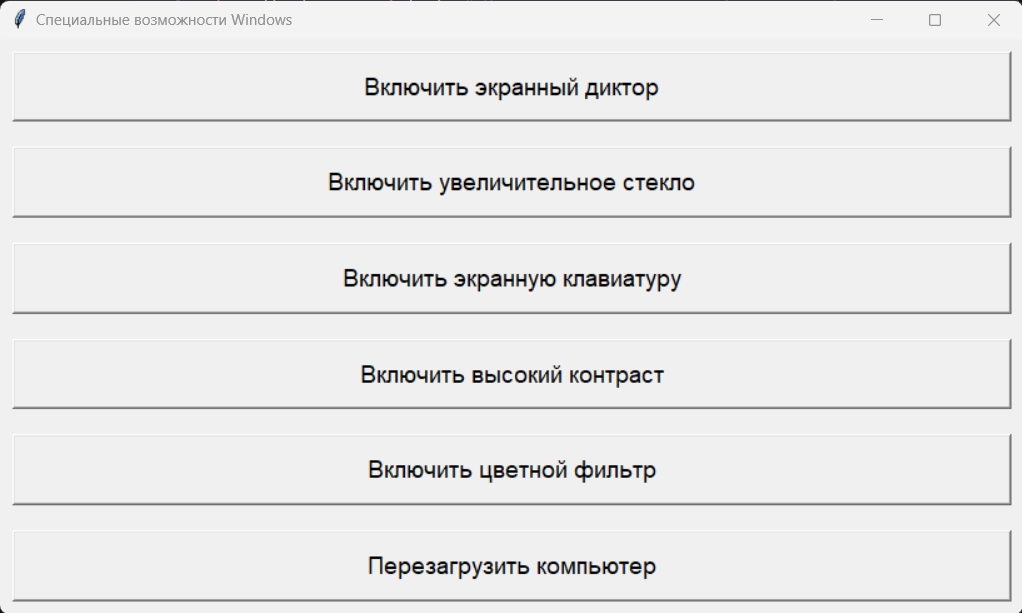
osk\_button = tk.Button(root, text="Включить экранную клавиатуру", font=button\_font, command=start\_on\_screen\_keyboard)

high\_contrast\_button = tk.Button(root, text="Включить высокий контраст", font=button\_font, command=toggle\_high\_contrast)

color\_filter\_button = tk.Button(root, text="Включить цветной фильтр", font=button\_font, command=toggle\_color\_filter)

restart\_button = tk.Button(root, text="Перезагрузить компьютер", font=button\_font, command=restart\_computer)

Ознакомиться с полученным графическим интерфейсом можно на рисунке.



Полученный графический интерфейс

Таким образом, интерфейс приложения интуитивно понятен и удобен в использовании, что является важным аспектом этого интерфейса.

Литература