

## Курс

### «Разработка Windows-приложений с использованием Win API»

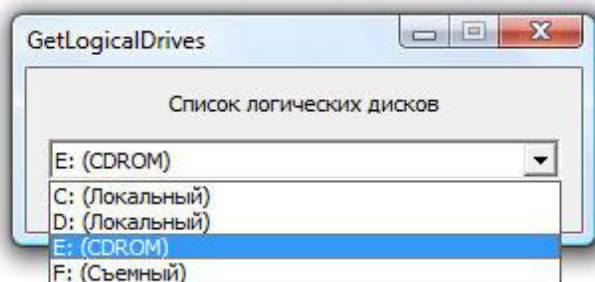
#### Встреча № 9,10

Тема: Списки

Домашнее задание

#### Задание:

Написать приложение, формирующее список логических дисков, которые доступны в данный момент.



При разработке приложения рекомендуется воспользоваться следующими функциями API:

```
DWORD WINAPI GetLogicalDrives(void);
```

Функция `GetLogicalDrives` возвращает битовую маску логических дисков, которые доступны в данный момент. При этом каждый бит маски представляет собой конкретный диск. Если бит установлен, то логический диск доступен, в противном случае бит будет сброшен. Бит 0 соответствует диску A, бит 1 – диску B, бит 2 – диску C и т.д.

```
UINT WINAPI GetDriveType( LPCTSTR lpRootPathName );
```

Функция `GetDriveType` возвращает тип накопителя по заданному имени корневого пути (например, "C:\\"). Перечень типов накопителей приведён в следующей таблице.

Возвращаемый код	Тип накопителя
DRIVE_UNKNOWN	Неизвестный тип накопителя
DRIVE_NO_ROOT_DIR	Корневой каталог не существует
DRIVE_REMOVABLE	Съемный диск
DRIVE_FIXED	Жесткий диск
DRIVE_REMOTE	Сетевой диск
DRIVE_CDROM	Компакт-диск
DRIVE_RAMDISK	Виртуальный диск

```
DWORD WINAPI GetLogicalDriveStrings(  
    __in DWORD nBufferLength, // максимальный размер буфера, на который  
    // указывает второй параметр  
    __out LPTSTR lpBuffer // указатель на буфер, который принимает набор строк,  
    // завершающихся нулевым символом, причем на каждый действительный логический  
    // диск в системе приходится по одной строке  
);
```

Функция `GetLogicalDriveStrings` заполняет буфер строками, которые определяют действительные логические диски в системе.

## Курс

### «Разработка Windows-приложений с использованием Win API»

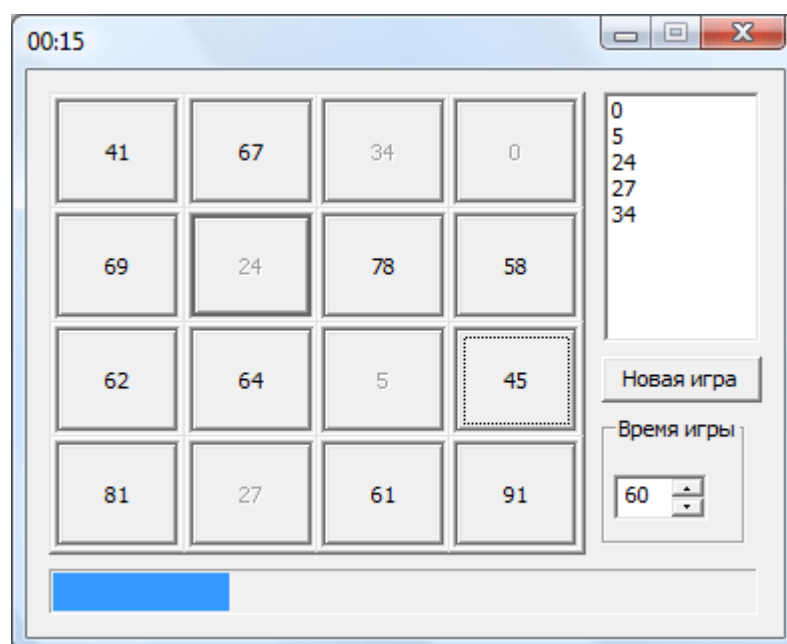
#### Встреча № 11

Тема: Прогресс-бар, слайдер и др.

Домашнее задание

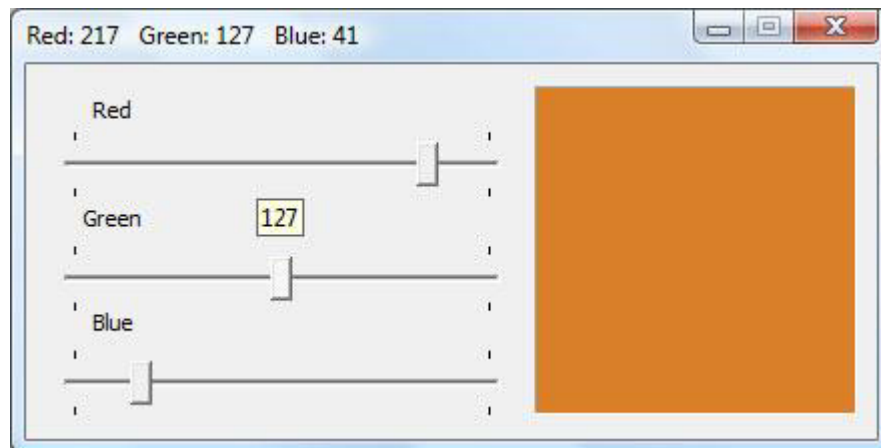
#### Задание 1:

Написать игру, смысл которой состоит в следующем. На игровом поле имеются 16 кнопок, список, счётчик и индикатор. При запуске игры (кнопка «Новая игра») на кнопки помещаются 16 случайных чисел из диапазона от 0 до 100. Задача состоит в том, чтобы за указанное время успеть (пока не заполнится весь индикатор) щелкнуть по всем кнопкам в порядке возрастания чисел. При нажатии на кнопку число должно добавляться в список только в том случае, если это число является следующим по возрастанию. Время продолжения игры (в секундах) устанавливается с помощью счетчика.



## Задание 2:

Разработать приложение, позволяющее с помощью трёх ползунков настраивать цвет фона индикатора.



Напомним, что цвет фона индикатора устанавливается посылкой сообщения PBM\_SETBKCOLOR. При этом в wParam передаётся 0, а в lParam – цвет фона (тип COLORREF), который можно задать с помощью макроса RGB:

```
#define RGB(r,g,b) \
((COLORREF) (((BYTE) (r) | ((WORD) ((BYTE) (g)) << 8)) | (((DWORD) (BYTE) (b)) << 16)))
```

Например:

```
SendMessage(hProgress, PBM_SETBKCOLOR, 0, LPARAM(RGB(red,green,blue)));
```