ЛАБОРАТОРНАЯ РАБОТА №5

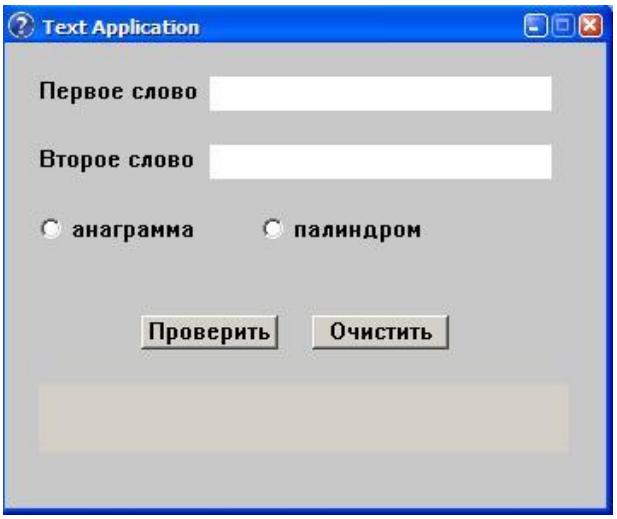
Задание 1

Создайте приложение, которое будет проверять: (1) являются ли анаграммами две строки, введенные пользователем; (2) является ли палиндромом единственная строка, введенная пользователем. Два слова называются анаграммами, если они состоят из одинаковых символов (например, *апельсин* и *спаниель* или *тавро* и *отвар*). Слово называется палиндромом, если оно читается слева направо и справа налево одинаково (например, *поток* или *заказ*).

Окно должно содержать следующие элементы управления, созданные программно:

- Статики (класс `Static`) для надписей и для вывода результата.
- Два текстовых поля типа `Edit` для ввода строк.
- Два переключателя для выбора условия, на соответствие которому проверяются строки (или строка). Когда выбран переключатель «Анаграмма», оба текстовых поля должны быть доступны для ввода, а когда выбран переключатель «Палиндром» – только первое текстовое поле.
- Две кнопки. По нажатию на первую кнопку в статик посылается результат проверки. По нажатию на вторую кнопку текстовые поля и статик с результатом очищаются.

Интерфейс программы может выглядеть примерно так:



Указания

1. Чтобы при создании статика цвет его фона совпадал с цветом окна, нужно обрабатывать сообщение `WM_CTLCOLORSTATIC`. Если полученный ID элемента управления соответствует идентификатору нужного статика, то дальше выполняется следующее:

```
HDC hdcStatic = (HDC) wParam; // создать контекст устройства
SetTextColor(hdcStatic, RGB(0,0,0)); // установить цвет текста
SetBkColor(hdcStatic, RGB(200, 200, 200)); // установить
        нужный цвет фона, обязательно совпадающий с цветом фона окна
```

```
return (INT_PTR)hBrush;
```

Объект кисти типа `HBRUSH` нужно будет создать заранее, например, в функции `WndProc` до `switch`'а:

```
static HBRUSH hBrush = CreateSolidBrush(RGB(200, 200, 200));
```

Подробнее об этом можно посмотреть ответы на StackOverflow: http://stackoverflow.com/questions/4495509.

2. Чтобы сделать текстовое поле недоступным для редактирования, воспользуйтесь функцией `EnableWindow`, которая принимает дескриптор элемента управления и значения `TRUE` (разблокировать элемент) или `FALSE` (заблокировать элемент):

```
BOOL WINAPI EnableWindow(HWND hWnd, BOOL bEnable);
```

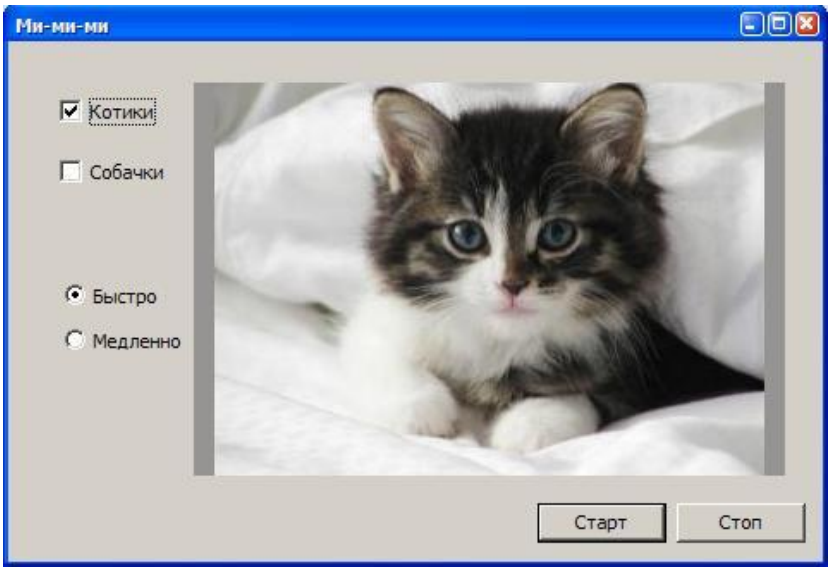
Задание 2

Создайте программу для просмотра картинок на основе модального или немодального диалогового окна.

На форме должно быть два флажка, два переключателя и две обычные кнопки:

- С помощью флажков пользователь выбирает, какие изображения демонстрируются (можно использовать изображения из папки `Img`). Обратите внимание, что одновременно могут быть установлены оба флажка.
- С помощью переключателей задается скорость, с которой сменяются изображения.
- По нажатию на кнопку «Старт» начинается показ картинок, по нажатию на кнопку «Стоп» демонстрация останавливается.
- Дополнительно можно запретить максимизацию окна.

Интерфейс программы может выглядеть примерно так:



Указания

1. Для создания диалогового окна добавьте новый шаблон типа Dialog в ресурсы приложения. Посмотрите его идентификатор через окно свойств или через символы ресурсов. Не забудьте подключить заголовочный файл resource.h в вашем основном cpp-файле.

В функции WinMain создайте диалоговое окно с помощью функции DialogBox или CreateDialog. Отдельно задайте функцию DlgProc, которая будет обрабатывать сообщения от этого окна.

2. Все элементы управления следует перетягивать на оконную форму из панели элементов (ToolBox). В окне свойств элемента изменяйте идентификаторы элементов управления, которые присваиваются по умолчанию, на более понятные (например, IDC_BUTTON1 на IDC_BTN_START и т. п.).

3. В функции DlgProc обязательно предусмотрите обработку сообщения WM_INITDIALOG, в которой можно проинициализовать дескрипторы элементов управления. Для этого воспользуйтесь функцией GetDlgItem:

```
HWND GetDlgItem(HWND hDlgParent, int controlID);
```

4. Для смены картинок создайте массив или вектор (может быть, несколько массивов или векторов) объектов типа BITMAP. Загрузите туда все необходимые картинки с помощью функции LoadBitmap:

```
HBITMAP LoadBitmap(HINSTANCE hInstance, LPCTSTR
lpBitmapName);
```

Сконструировать строку lpBitmapName можно с помощью MAKEINTRESOURCE, который принимает идентификатор картинки из ресурсов.

5. Чтобы изменить картинку в элементе типа Picture Control, пошлите ему сообщение STM_SETIMAGE, например:

```
SendMessage(hPicture, STM_SETIMAGE, (WPARAM)IMAGE_BITMAP,
(LPARAM)hBmp1),
```

где hPicture – дескриптор элемента Picture Control, hBmp1 – дескриптор картинки, которую нужно отобразить в Picture Control.

Смену картинок следует производить по таймеру.