

## Лабораторная работа № 13

**Тема:** Разработка программ с элементами управления с использованием Win32 API

**Цель:** Освоить создание приложений с графическим интерфейсом с использованием функций Win32 API: создание элементов управления, прием и передача сообщений элементам управления.

### Краткая теория

В операционной системе Windows присутствуют заранее заложенные классы окон, которые реализуют различные элементы управления, используемые на окнах, создаваемых программистами при реализации GUI. К этим классам относятся: кнопки, текстовые поля, списки, комбинированные списки, флажки и переключатели и т.д. и т.п. Как правило эти элементы управления используются на окнах специального вида – диалоговых окнах. Отличительные черты элементов управления:

- для них уже описаны классы окон;
- все они дочерние окна (стиль WS\_CHILD);
- для них описаны дополнительные стили и списки обрабатываемых и получаемых сообщений;
- для них почти всегда необходимо описывать идентификаторы.

Родительское окно от дочерних окон получает сообщения. Полученное сообщение содержит идентификатор дочернего окна – отправителя сообщения.

Имя класса	Описание	Сообщение родительскому окну
“button”	Окно, состояние которого изменяется при выборе и нажатии. Это нажимающиеся кнопки (push button), переключатели (radio button), флажки (check box)	WM_COMMAND
“static”	Текстовое поле, прямоугольник выделения и другие элементы оформления	WM_NCHITTEST
“scrollbar”	Полоса прокрутки – прямоугольник с ползунком и стрелками направления в обе стороны	WM_HSCROLL WM_VSCROLL
“edit”	Окно ввода текста с клавиатуры или редактор	WM_COMMAND
“listbox”	Список символьных строк. При просмотре в каждый момент времени подсвечивается одна строка и информация о ней передается родительскому окну	WM_COMMAND
“combobox”	Элемент управления, состоящий из списка и поля выбора. Список может быть свернут или развернут.	WM_COMMAND

Для создания элементов управления используется функция CreateWindow, где в качестве параметров указываются следующие значения:

- lpClassName – указатель на имя класса элемента управления;
- lpWindowName – строка, содержащая имя элемента управления;
- dwStyle – стиль окна (обязательно WS\_CHILD с необязательными дополнительными параметрами);
- x, y – позиция элемента управления относительно левого верхнего угла рабочей области родительского окна;
- nWidth, nHeight – размеры элемента управления;

- hWndParent – дескриптор родительского окна;
- hMenu – идентификатор дочернего окна;
- hInstance – дескриптор экземпляра приложения;
- lpParam – указатель на данные окна.

Создание элементов управления осуществляется как правило в момент создания родительского окна (WM\_CREATE). Каждому элементу управления назначают свой идентификатор, как правило, с помощью директивы define. Номер идентификатора должен быть более 3000.

Классы кнопок:

BS_CHECKBOX	Флажок, который может находиться в двух состояниях: «выбран» или «не выбран»
BS_AUTOCHECKBOX	То же, что и предыдущий. Автоматически осуществляется перерисовка при изменении состояния
BS_3STATE	Флажок, который может находиться в трех состояниях: «выбран», «не выбран» и «неактивен»
BS_AUTO3STATE	То же, что и предыдущий. Автоматически осуществляется перерисовка при изменении состояния
BS_RADIOBUTTON	Переключатель с текстом справа. В окне или группе кнопок может быть только одна активная кнопка
BS_AUTORADIOBUTTON	То же, что и предыдущий, но перерисовка осуществляется автоматически при изменении состояния
BS_PUSHBUTTON	Стандартная кнопка без рамки
BS_DEFPUSHBUTTON	Кнопка с рамкой, используется как кнопка по умолчанию: вызывается при нажатии ENTER, когда все элементы окна неактивны (не имеют фокуса ввода)
BS_GROUPBOX	Прямоугольник, в котором группируют органы управления. Не принимает сообщений.
BS_LEFTTEXT	Дополняет стиль флажка меняя порядок расположения самого флажка и текста: сначала текст
BS_RIGHTBUTTON	
BS_ICON	Элемент управления является иконкой
BS_MULTILINE	Распределить текст в нескольких строках внутри прямоугольника кнопки
BS_NOTIFY	Орган управления посылает сообщения BN_DBLCLK, BN_KILLFOCUS, BN_SETFOCUS родительскому окну
BS_PUSHLIKE	Флажок или переключатель ведет себя подобно кнопке: приподнимается или притапливается
BS_TEXT	Элемент управления является текстом
BS_RIGHT, BS_LEFT, BS_TOP, BS_BOTTOM	Выравнивание текста по правому, левому, верхнему или нижнему краю соответственно
BS_CENTER	Центрирование текста по горизонтали в прямоугольнике кнопки
BS_VCENTER	Центрирование текста по вертикали в прямоугольнике кнопки

Кнопки могут находиться в двух состояниях: нажатом и отжатом. Для управления этими состояниями необходимо отправить кнопке сообщение BM\_SETSTATE с параметром wParam равным TRUE (если кнопка нажата):  
SendMessage(hButton, BM\_SETSTATE, TRUE, 0L)

и FALSE (если кнопка отпущена):

SendMessage(hButton,BM\_SETSTATE,FALSE,0L)

Для того, чтобы узнать состояние кнопки, ей посылают сообщение BM\_GETCHECK:  
WORD res = (WORD)SendMessage(hButton,BM\_GETCHECK,0,0L)

Возвращаемое значение:

- 0 – отжатая кнопка, выключенный переключатель или флажок,
- 1 – нажатая кнопка, включенный переключатель или флажок,
- 2 – переключатель или флажок находятся в неактивном состоянии.

Кнопки стилей BS\_3STATE, BS\_CHECKBOX, BS\_RADIOBUTTON не перерисовываются при их переключении. Для выполнения этого действия им необходимо послать сообщение BM\_SETCHECK указав в wParam новое состояние:

- 0 – выключен,
- 1 – включен,
- 2 – неактивный.

Пример:

SendMessage(hRbButton,BM\_SETCHECK,1,0L);

Кнопки стилей BS\_AUTOCHECKBOX, BS\_AUTORADIOBUTTON автоматически меняют свое отображение.

Данный элемент управления использую для оформления внешнего вида окна приложения. Он имеет вид закрашенного или незакрашенного прямоугольника или строк текста в заданном прямоугольнике. Пример создания этого элемент управления:

```
HWND staMain = CreateWindow("static",NULL,  
    WS_CHILD | WS_VISIBLE | SS_BLACKFRAME,  
    10, 200, 150, 150, hwnd, (HMENU)0, hInstance,0);
```

Так как этот элемент не посылает никаких сообщений родительскому окну, то в качестве идентификатора для него обычно указывают нулевое значение.

Стили статического элемента:

SS_BLACKFRAME SS_GRAYFRAME SS_WHITEFRAME	Прямоугольная рамка черного, серого и белого цвета соответственно. Внутренняя область остается незакрашенной. Текст не используют.
SS_BLACKRECT SS_GRAYRECT SS_WHITERECT	Закрашенный прямоугольник черного, серого и белого цвета соответственно. Текст не используют.
SS_LEFT, SS_RIGHT, SS_CENTER	В заданном прямоугольнике указанный текст выводит функция DrawText, выравнивая по левому, правому краю или по центру. Слова переносятся на следующую строку, не поместившаяся часть текста не отображается.
SS_LEFTNOWORDWRAP	То же, что и SS_LEFT, но слова не переносятся.
SS_NOPREFIX	Этот стиль дополняет другие стили, когда необходимо отменить специальную обработку символа &.
SS_NOTIFY	Окно этого стиля посылает родительскому окну сообщения STN_CLICKED или STN_DBLCLK о нажатии или двойном щелчке левой кнопки мыши
SS_SIMPLE	Выводит текст в одну строку, выравнивая по левому краю. Без стирания предыдущего текста.

На базе класса “edit” создают однострочный или многострочный текстовый редактор. Его можно использовать для ввода текстовых и числовых значений, а также для создания и редактирования текстовых файлов. Этот редактор также может выполнять функции выделения текста, работать с буфером обмена и многое другое.

Стили класса редактора текста:

ES_AUTOHSCROLL	Автоматически сдвигать текст влево на 10 символов при достижении правой границы редактора
ES_AUTOVSCROLL	Автоматический сдвиг текста вверх на одну строку при достижении нижнего края редактора
ES_CENTER	Выравнивание текста по центру
ES_LEFT	Выравнивание текста по левому краю
ES_RIGHT	Выравнивание текста по правому краю
ES_LOWERCASE	Преобразовывать вводимые символы в строчные
ES_UPPERCASE	Преобразовывать вводимые символы в прописные
ES_MULTILINE	Разрешить многострочный редактор
ES_NOHIDESEL	При потере фокуса ввода выделенный фрагмент текста отображать в инверсном цвете
ES_NUMBER	Разрешить ввод только цифр
ES_PASSWORD	Отображать «звездочку» для каждого вводимого символа
ES_READONLY	Запрет редактирования текста в редакторе
ES_WANTRETURN	При нажатии ENTER переходить на новую строку

Сообщения для редактора текста		
EM_GETFIRSTVISIBLELINE wParam=lParam=0	Получение номера самой верхней видимой строки	Номер строки
EM_GETHANDLE wParam=lParam=0	Получение дескриптора блока памяти, хранящего редактируемый текст	Дескриптор блока памяти
EM_GETLINE wParam – номер строки, lParam – адрес буфера	Копирование строки из редактора текста в буфер	Количество символов или 0
EM_GETLINECOUNT wParam=lParam=0	Получение количества строк в тексте	Количество строк или 1
EM_GETRECT wParam=0, lParam – адрес RECT	Получение координат прямоугольной области редактора	
EM_LIMITTEXT wParam – макс. длина текста, lParam = 0	Ограничение количества символов в окне.	
EM_SETHANDLE wParam – дескриптор, lParam = 0	Установка дескриптора блока памяти для хранения редактируемого текста	
EM_SETREADONLY wParam – признак (TRUE, FALSE) lParam = 0	Установка или сброс режима «только для чтения»	TRUE или FALSE при ошибке

EM_SETRECT wParam = 0 lParam – адрес нового прямоугольника	Изменение прямоугольника редактирования текста	
---	--	--

Сообщения от редактора текста	
EN_CHANGE	Текст в окне редактирования изменился
EN_ERRSPACE	Недостаточно памяти для выполнения действий
EN_HSCROLL	Нажата горизонтальная полоса просмотра редактора, но свертка еще не произошла
EN_KILLFOCUS	Текстовый редактор потерял фокус ввода
EN_MAXTEXT	Превышен заданный для редактора размер текста
EN_SETFOCUS	Текстовый редактор получил фокус ввода
EN_UPDATE	Последняя операция редактирования выполнена, но еще не отразилась на содержимом редактора
EN_VSCROLL	Нажата вертикальная полоса просмотра редактора, но свертка еще не произошла

Списки строк создают на базе класса “listbox”. Строки в списке нумерует операционная система начиная с 0. Списки могут быть одноколоночные и многоколоночные, с вертикальной и горизонтальной полосой просмотра. При создании списка первый параметр функции CreateWindow должен содержать “listbox”, второй параметр – NULL, а третий параметр должен содержать стиль создаваемого списка.

Стили списка строк	
LBS_DISABLENOSCROLL	Запретить убирать вертикальную полосу прокрутки, если она не нужна
LBS_EXTENDEDSEL	Возможность выделения нескольких строк с помощью клавиши shift
LBS_HASSTRINGS	Список посылает родительскому окну сообщение WM_VKEYTOITEM с кодом виртуальной клавиши (LOWORD)wParam, с номером текущей строки (HIWORD)wParam и дескриптором списка lParam
LBS_MULTICOLUMN	Многоколоночный список. Количество колонок задают сообщением LB_SETCOLUMNWIDTH
LBS_MULTIPLESEL	С помощью клавиши shift или мыши можно выделять несколько строк в любом месте списка
LBS_NOINTEGRALHEIGHT	Высота окна не обязательно кратна высоте строк
LBS_NOREDRAW	Не перерисовывать содержимое при добавлении или удалении строк
LBS_NOSEL	Окно списка содержит элементы, которые могут просматриваться, но не выбираться.
LBS_NOTIFY	Сообщить о двойном щелчке мыши по строке
LBS_OWNERDRAWFIXED	Список строк одинаковой высоты, перерисовывается родительским окном
LBS_OWNERDRAWVARIABLE	Список строк переменной высоты, перерисовывается родительским окном
LBS_SORT	Сортировать строки в алфавитном порядке

LBS_STANDART	Комбинация LBS_NOTIFY, LBS_SORT, WS_BORDER, WS_VSCROLL
LBS_USETABSTOPS	При отображении строк преобразовывать символы табуляции
LBS_WAITKEYBOARDINPUT	Родительское окно от списка получает сообщения WM_KEYTOITEM или WM_CHARTOITEM при работе со списком при помощи клавиатуры

Список со стилем LBS\_NOTIFY посылает в окно сообщение MW\_COMMAND. При этом младшее слово wParam равно идентификатору списка, параметр lParam содержит дескриптор списка. А старшее слово wParam содержит одно из следующих значений:

LBN_DBLCLK	Двойной щелчок левой клавишей мыши по строке списка
LBN_ERRSPACE	Ошибка при попытке заказать дополнительную память
LBN_KILLFOCUS	Список потерял фокус ввода
LBN_SELCANCEL	Пользователь отменил выбор с списке
LBN_SELCHANGE	Изменился номер выбранной строки
LBN_SETFOCUS	Список получил фокус ввода

Сообщения для списка строк	
LB_ADDSTRING, 0 lParam = lpzStr	Добавить в список строку lpzStr, возвращает номер строки
LB_DELETESTRING, wParam = nIndex, 0L	Удалить строку с номером nIndex из списка. Возвращает количество оставшихся в списке строк.
LB_DIR, wParam = uAttr, lParam = lpzFileSpec	Заполнить список именами дисков, каталогов и файлов с атрибутами uAttr по шаблоны lpzFileSpec из текущего каталога. Возвращает номер последнего добавленного списка
LB_FINDSTRING, wParam = nStart, lParam = lpzStr	Начиная со строки с номером nStart искать строку с префиксом lpzStr. Возвращает номер или код ошибки
LB_FINDSTRINGEXACT, wParam = nStart, lParam = lpzStr	Начиная со строки с номером nStart искать строку lpzStr. Возвращает номер или код ошибки
LB_GETCARETINDEX, 0, 0L	Определение номера строки, имеющей фокус ввода. Возвращает номер этой строки или код ошибки
LB_GETCOUNT, 0, 0L	Определить количество строк в списке. Возвращает количество строк или код ошибки
LB_GETCURSEL, 0, 0L	Определение номера выделенной строки. Возвращает номер строки или код ошибки
LB_GETSEL, wParam = nIndex, 0L	Определить: выбрана ли строка с номером nIndex. Возвращает положительное число – если выбрана, 0 – если нет или код ошибки
LB_GETSELCOUNT, 0, 0L	Определить количество выбранных строк. Возвращает количество или код ошибки
LB_GETSELITEMS, wParam = cnt, lParam = (int far )lpBuff	Заполнить буфер lpBuff размера cnt номерами выбранных строк. Возвращает количество записанных в буфер номеров или код ошибки
LB_GETTEXT, wParam = nIndex, lParam = lpBuff	Копировать текст строки с номером nIndex в буфер lpBuff. Возвращает длину строки или код ошибки

LB_GETTEXTLEN, wParam = nIndex, 0L	Определить длину строки с номером nIndex. Возвращает длину или код ошибки
LB_GETTPOINDEX, 0, 0L	Определить номер первой отображаемой строки. Возвращает номер строки или код ошибки
LB_INSERTSTRING, wParam = nIndex, lParam = lpBuff	Вставить строку lpBuff на позицию nIndex в список. Возвращает номер или код ошибки
LB_RESETCONTENT, 0, 0L	Удалить все строки из списка.
LB_SELECTSTRING, wParam = nStart, lParam = lpszPrefix	Начиная с позиции nStart найти и выделить строку с префиксом lpszPrefix. Возвращает номер строки или код ошибки
LB_SELITEMRANGE, wParam = (BOOL)fSelect, lParam = MAKELPARAM(wF, wL)	Выделить (fSelect=TRUE) или отменить выделение (fSelect=FALSE) строк с номерами от wF до wL. Возвращает код ошибки
LB_SETCARETINDEX, wParam = nIndex, lParam = MAKELPARAM(fS, 0)	Передать фокус ввода строке с номером nIndex. fS управляет сверткой (частично или полностью видна). Возвращает код ошибки.
LB_SETCURSEL, wParam = nIndex, 0L	Отменить предыдущий выбор и выбрать строку с номером nIndex.
LB_SETTOPINDEX, wParam = nIndex, 0L	Сверка списка до тех пор, пока строка с номером nIndex не станет видимой

Стили комбинированного списка	
CBS_AUTOHSCROLL	Автоматическая свертка текста по горизонтали в окне редактирования
CBS_DISABLENOSCROLL	Вертикальную полосу просмотра отображать в неактивном состоянии даже если в списке помещаются все строки
CBS_DROPDOWN	Список остается невидимым до тех пор, пока пользователь не нажмет пиктограмму, предназначенную для отображения списка
CBS_DROPDOWNLIST	Запрещает редактирование текста в однострочном текстовом редакторе
CBS_LOWERCASE	Преобразовывать все символы в строчные
CBS_UPPERCASE	Преобразовывать все символы в прописные
CBS_SIMPLE	Создается список, который всегда и расположен под окном однострочного редактора, содержащего выделенную в списке строку
CBS_SORT	Автоматически сортирует строки, введенные в окно списка

Сообщения от комбинированного списка	
CBN_CLOSEUP	Список исчез (стал невидимым)
CBN_DBLCLK	Двойной щелчок мыши по строке списка CBS_SIMPLE
CBN_DROPDOWN	Список стал видимым
CBN_EDITCHANGE	Пользователь изменил содержимое окна редактирования и изменения отображены
CBN_EDITUPDATE	Пользователь изменил содержимое окна редактирования

CBN_ERRSPACE	Ошибка выделения дополнительной памяти
CBN_KILLFOCUS	Список теряет фокус управления
CBN_SELENDCANCEL	Пользователь отменил выбор в списке
CBN_SELENDOK	Пользователь выбрал строку в списке
CBN_SEKCHANGE	Изменился номер выбранной строки
CBN_SETFOCUS	Список получает фокус ввода

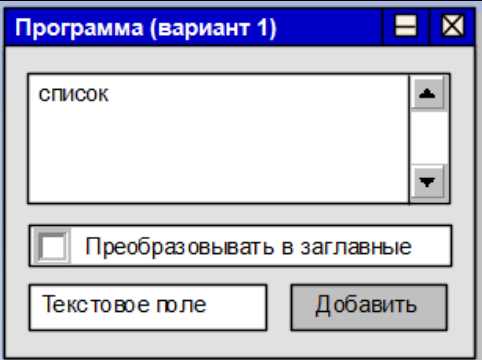
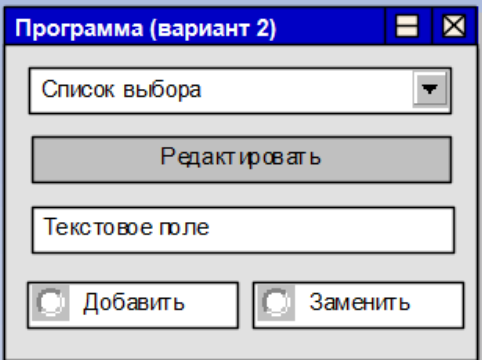
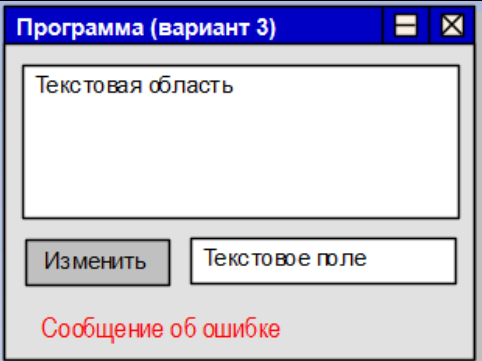
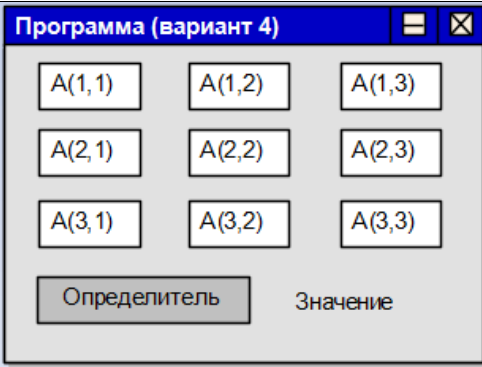
Сообщения для комбинированного списка	
CB_ADDSTRING, 0, lParam = lpzStr	Добавить строку lpzStr. Возвращает номер строки или код ошибки
CB_DELETESTRING, wParam = nIndex, 0L	Удалить строку с номером nIndex. Возвращает количество оставшихся строк в списке или код ошибки.
CB_DIR, wParam = uAttr, lParam = lpzFileSpec	Заполнить список именами файлов с атрибутом uAttr – и шаблона lpzFileSpec и каталогов в текущем каталоге
CB_FINDSTRING, wParam = nIndex, lParam = lpzStr	Начиная со строки с номером nIndex искать строку с префиксом lpzStr. Возвращает номер найденной строки или код ошибки
CB_GETCOUNT, 0, 0L	Определить количество строк в списке. Возвращает количество или код ошибки
CB_GETCURRESEL, 0, 0L	Определить номер выделенной строки. Возвращает номер или код ошибки.
CB_GETDROPPEDSTATE, 0, 0L	Определить, находится ли список в видимом или невидимом состоянии. Возвращает TRUE – если виден, FALSE – если нет.
CB_GETLBTEXT, wParam = nIndex, lParam = lpBuff	Копировать текст строки с номером nIndex в буфер lpBuff. Возвращает длину строки или код ошибки
CB_INSERTSTRING, wParam = nIndex, lParam = lpBuff	Вставка строки lpBuff в список под номером nIndex. Возвращает номер позиции или код ошибки
CB_LIMITTEXT, wParam = cMax, 0L	Определить cMax как максимально допустимое количество символов в окне редактирования.
CB_RESETCONTENT, 0, 0L	Удалить все строки из списка
CB_SELECTSTRING, wParam = nIndex, lParam = lpBuff	Начиная со строки с номером nIndex найти и выделить строку с префиксом lpBuff. Возвращает номер найденной строки или код ошибки
CB_SETCURRESEL, wParam = nIndex, 0L	Выделить строку с номером nIndex. Возвращает код ошибки.
CB_SHOWDROPDOWN, wParam = (BOOL)fExt, 0L	Переключение списка в отображаемое или не отображаемое состояние. fExt = TRUE – отображать, FALSE – не отображать

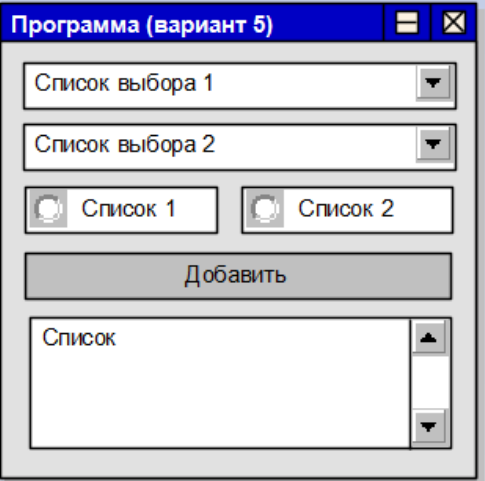
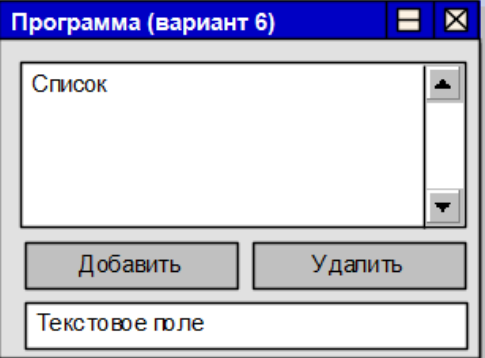
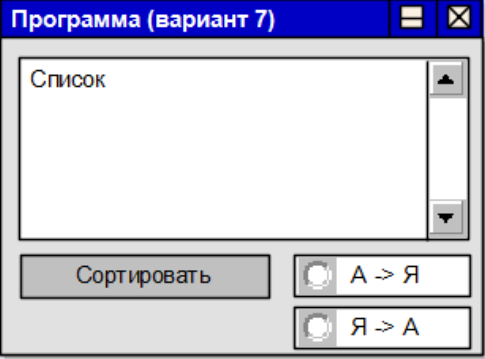
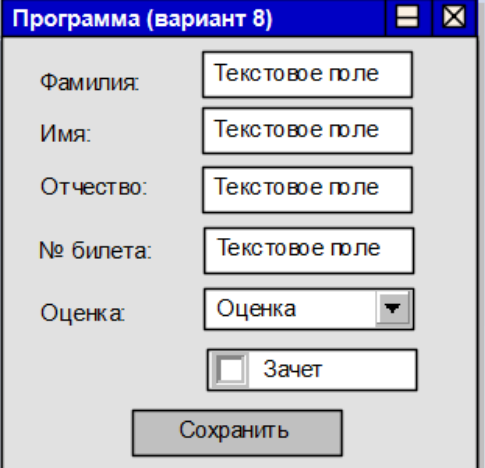
### Ход работы

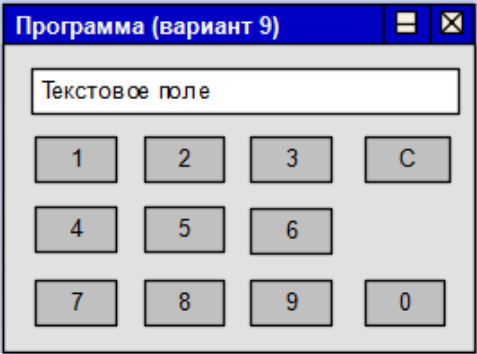
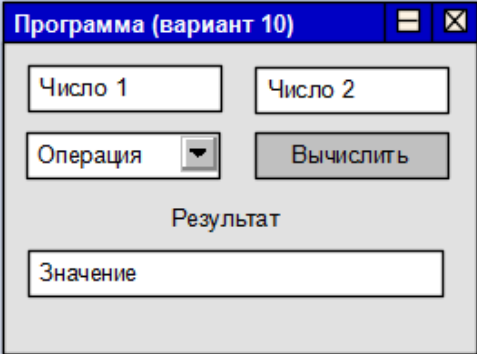
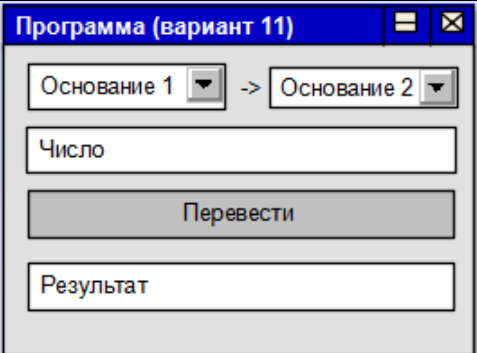
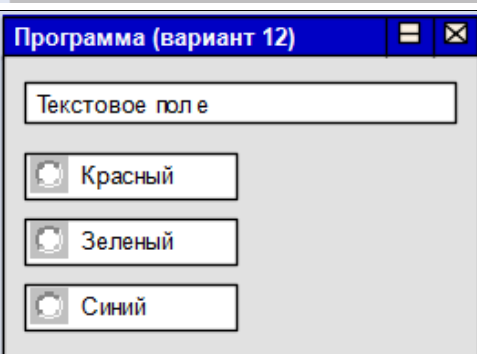
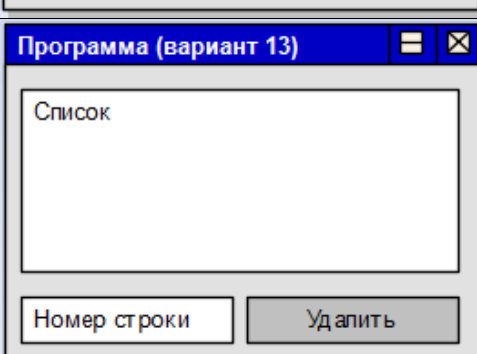
В рамках данной лабораторной работы необходимо, используя Win32API, разработать программу с графическим оконным интерфейсом согласно варианту задания. Внешний вид окна должен соответствовать картинке приведенной в задании. При разработке программы использовать мастера не рекомендуется. Выполнение лабораторной работы можно осуществлять в IDE Pelles C или Visual C++. Для разработки программы необходимо создать проект типа Win32 Application.

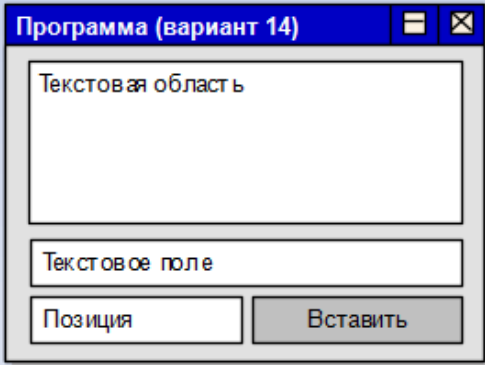
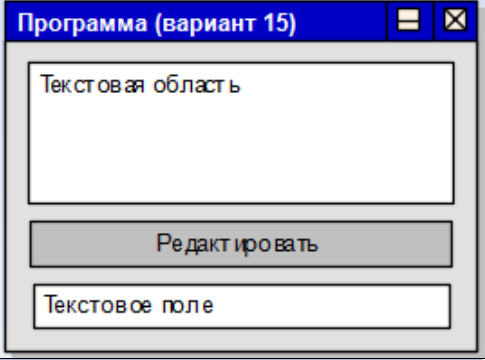
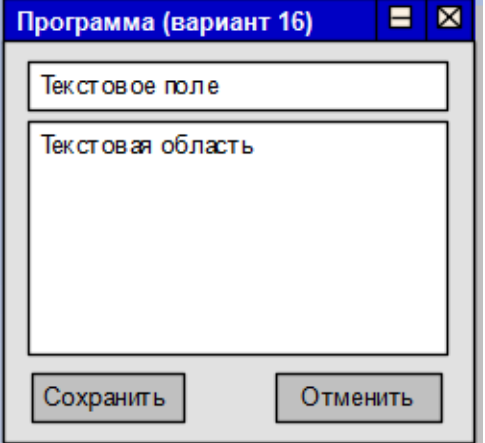
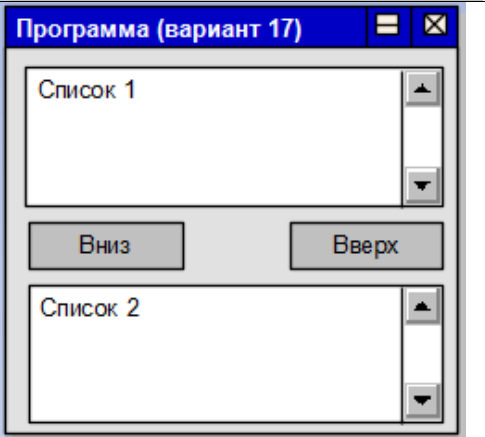


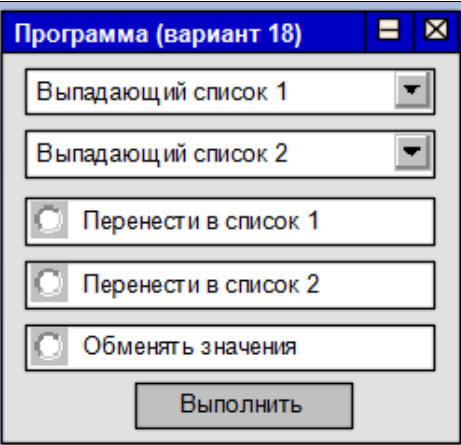
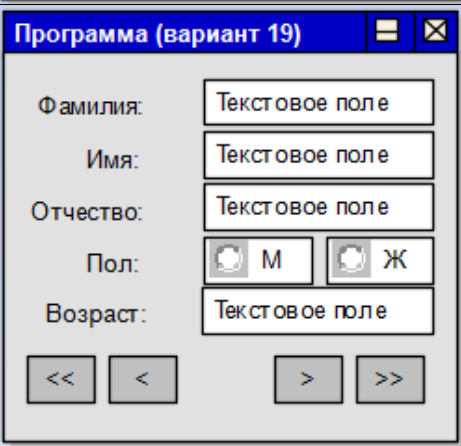
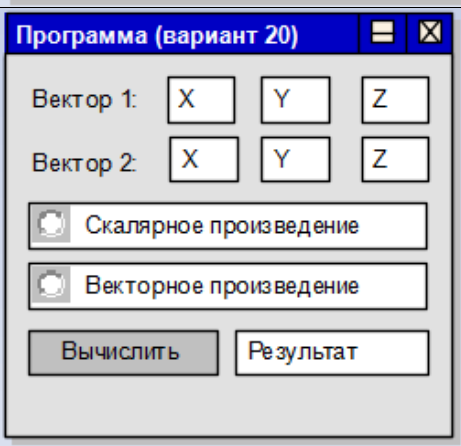
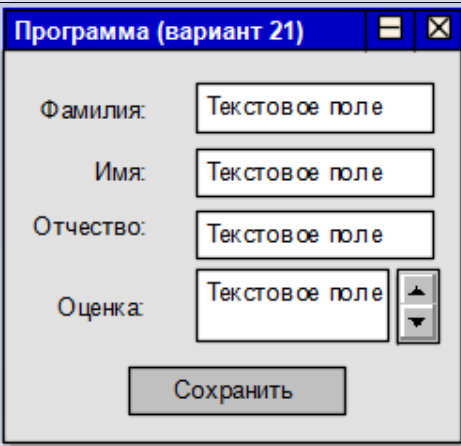
Варианты заданий:

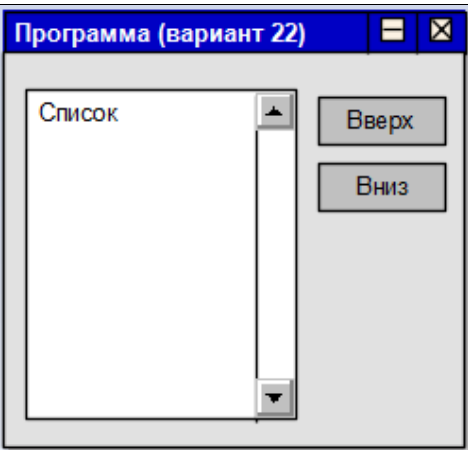
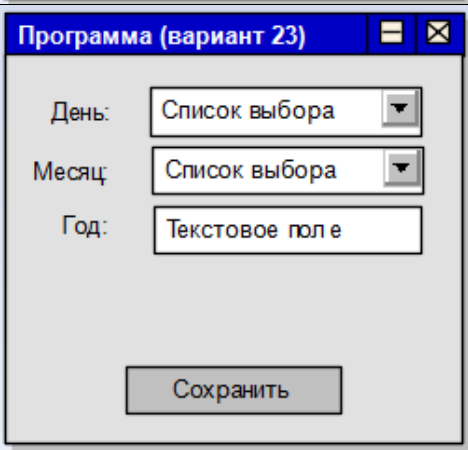
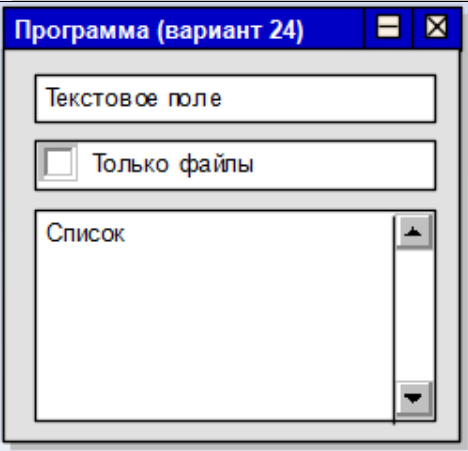
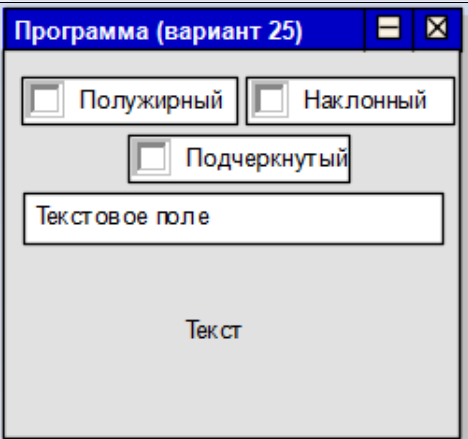
1		<p>При нажатии кнопки добавить в список заносится содержимое текстового поля. При этом если флажок установлен, то все буквы текста преобразуются к заглавным. После занесения в список текстовое поле очищается. При двойном клике по элементу списка, данный элемент должен удаляться. При завершении программы содержимое списка заносится в файл result.txt.</p>
2		<p>Список выбора изначально загружается из файла data.txt. При нажатии кнопки редактировать выбранный элемент заносится в текстовое поле. Пользователь может его редактировать и, по нажатию клавиши ENTER, текст в зависимости от значения переключателей «Добавить» и «Заменить» либо в конец списка выбора, либо в место текущего выбранного значения.</p>
3		<p>Изначально текст в текстовую область загружается из файла text.txt. При нажатии кнопки «Изменить» выделенная строка или ее часть заносятся в текстовое поле. Пользователь может ее отредактировать и по нажатию клавиши ENTER заменить отредактированным значением выделенный фрагмент в текстовой области. Если в текстовой области ничего не выделено или выделен фрагмент из нескольких строк, то вывести соответствующее сообщение об ошибке.</p>
4		<p>Пользователь в текстовых полях задает значения элементов матрицы. При нажатии кнопки «Определитель» вычисляется определитель введенной матрицы. Изначально кнопка «Определитель» запрещена. Она становится доступна, только тогда, когда пользователь укажет корректно все коэффициенты матрицы. Коэффициенты матрицы – вещественные числа.</p>

5		<p>Изначально содержимое списков выбора загружается из файлов list1.txt и list2.txt. При нажатии кнопки «Добавить» в зависимости от состояния флажков «Список 1» и «Список 2» выбранное значение в одном из списков выбора заносится в обычный список. При двойном клике мыши в обычном списке текущий элемент удаляется.</p>
6		<p>Изначально список пуст. При нажатии кнопки «Добавить» содержимое текстового поля, если оно есть, заносится с конец списка. При нажатии кнопки «Удалить» из списка удаляются все выделенные значения. При завершении работы программы содержимое списка заносится в файл result.txt. Список допускает множественный выбор.</p>
7		<p>Изначально список загружается из файла list.txt. В списке разрешен множественный выбор. При нажатии кнопки «Сортировать» строки списке сортируются в направлении согласно состояниям флажков «А-&gt;Я» и «Я -&gt; А». При этом должны сортироваться только те элементы списка, которые не выбраны. Выделенные элементы должны оставаться на своих местах. При завершении программы список записывается обратно файл.</p>
8		<p>Форма ввода данных о сдаче экзамена или зачета. Изначально кнопка «Сохранить» недоступна. Пользователь заполняет поля формы. Когда все поля заполнены кнопка «Сохранить» становится доступной. Номер билета – целое положительное число. Если флажок «Зачет» не установлен, то список оценка содержит оценки от 1 до 10, если не установлен, то слова «зачтено» и «не зачтено». При нажатии кнопки «Сохранить», данные с формы записываются в файл exam.txt дописывая его. При этом все текстовые поля очищаются, а кнопка сохранить блокируется.</p>

9	 <p>Программа (вариант 9) contains a text field labeled 'Текстовое поле' and a numeric keypad with buttons for digits 1-9, 0, and a 'C' (clear) button.</p>	<p>Изначально текстовый редактор содержит значение ноль. При нажатии кнопки с цифрой в конец строки текстового редактора добавляется соответствующая цифра. При нажатии кнопки «С» строка сбрасывается в ноль. Возможность редактирования в текстовом поле должна быть запрещена. Число в текстовом поле не может иметь ведущих нулей, а его длина не может превышать 10 цифр.</p>
10	 <p>Программа (вариант 10) contains two input fields for 'Число 1' and 'Число 2', a dropdown menu for 'Операция', a 'Вычислить' button, and a 'Значение' output field.</p>	<p>В текстовых полях «Число 1» и «Число 2» пользователь указывает числа. В выпадающем списке выбирает операцию (сложение, вычитание, умножение, деление). После нажатия кнопки «Вычислить» в поле «Значение» заносится результат. В поле «Значение» должно быть запрещено редактирование. Кнопка «Вычислить» становится доступной, когда пользователь введет два числа в соответствующие поля.</p>
11	 <p>Программа (вариант 11) contains two dropdown menus for 'Основание 1' and 'Основание 2', a 'Число' input field, a 'Перевести' button, and a 'Результат' output field.</p>	<p>Пользователь в поле «Число» вводит числовое значение, записанное в системе исчисления с основанием указанным в первом выпадающем списке. При нажатии кнопки «Перевести» число переводится в систему счисления с основанием указанным во втором выпадающем списке. Результат преобразования записывается в поле «Результат», которое нельзя редактировать вручную. Если число было указано некорректно, то в поле «Результат» записывается слово «Ошибка».</p>
12	 <p>Программа (вариант 12) contains a 'Текстовое поле' and three color selection buttons: 'Красный', 'Зеленый', and 'Синий'.</p>	<p>Пользователь в текстовое поле вводит текст. Символы текста отображаются цветом, выбранным с помощью флажков. Как только пользователь выбирает другой цвет, он тут же меняется и для текста. При нажатии клавиши ENTER текст из текстового поля записывается в файл res.txt (в режиме записи в конец файла), а само текстовое поле очищается.</p>
13	 <p>Программа (вариант 13) contains a 'Список' (list) area, a 'Номер строки' (line number) input field, and a 'Удалить' (delete) button.</p>	<p>Список изначально загружается из файла list.txt. В текстовом поле пользователь указывает номер строки и нажимает кнопку «Удалить». Указанная строка удаляется из списка. Кнопка «Удалить» должна быть доступна, только тогда, когда в текстовом поле содержится корректный номер строки. При завершении программы оставшийся список записывается в исходный файл, переписывая его.</p>

14		<p>Содержимое текстовой области изначально загружается из файла text.txt. Пользователь в текстовом поле набирает текст, который по нажатию кнопки «Вставить» вставляется в текстовую область, начиная с позиции указанной в поле «Позиция», или в конец, если такой позиции нет. После этого текстовое поле очищается. Кнопка «Вставить» становится доступной только тогда, когда пользователь введет текст и укажет номер позиции.</p>
15		<p>Содержимое текстовой области загружается из файла text.txt. При нажатии кнопки «Редактирование» выделенный в текстовой области текст помещается в текстовое поле. Пользователь может редактировать этот текст и при нажатии кнопки ENTER он замещает выделенный в текстовой области текст, а текстовое поле очищается.</p>
16		<p>Пользователь в текстовое поле указывает имя файла и вводит текст в текстовую область. При нажатии кнопки "Сохранить" текст записывается в файл. При нажатии кнопки "Отменить" - в текстовую область загружается содержимое файла, если он был сохранен. Кнопка "Отменить" должна быть доступна только если пользователь сохранил файл и изменил текст после сохранения. Если имя файла не указано, то оно задается автоматически - text.txt.</p>
17		<p>Содержимое списка 1 изначально загружается из файла list.txt. В обоих списках разрешен множественный выбор. Если пользователь нажимает кнопку "Вниз", то элементы, выбранные в верхнем списке переносятся в нижний (из верхнего списка удаляются). При нажатии кнопки "Вверх" наоборот: из нижнего списка в верхний список. Также можно перемещать по одному элементу в списке двойным щелчком мыши на этом элементе: этот элемент переносится в другой список.</p>

18		<p>Изначально содержимое списков загружается из файлов list1.txt и list2.txt. При нажатии кнопки осуществляется действие в соответствии с выбранным флажком. Если "перенести в список 1", то элемент, выбранный в списке 2, переносится в список 1 (из списка 2 он удаляется). Если "Перенести в список 2" - наоборот. Если "Обменять значения", то выбранные в списках элементы перемещаются в обоих направлениях. Надпись на кнопке изменяется автоматически при выборе флажка: "&gt;", "&lt;", "&lt;&gt;".</p>
19		<p>Список записей задан изначально (не менее 10 записей). При нажатии кнопок осуществляется переход: "&lt;&lt;" - в начало списка, "&lt;" - на предыдущую запись, "&gt;" - на следующую запись, "&gt;&gt;" - в конец списка. Если текущая запись первая, то кнопки "&lt;" и "&lt;&lt;" блокируются, если текущая запись последняя, блокируются кнопки "&gt;" и "&gt;&gt;". Текстовые поля и флажки должны быть заблокированы для изменения.</p>
20		<p>Пользователь задает координаты двух векторов (вещественные числа). Выбирает действие и нажимает кнопку "Выполнить". В поле "Результат" записывается результат действия. Кнопка "Выполнить" становится доступной, только тогда, когда пользователь корректно введет все координаты двух векторов. При вычислении векторного произведения результат выводится в формате (x,y,z).</p>
21		<p>Пользователь заполняет все поля. Поле оценка доступно только для чтения. Изменить его значение можно используя кнопки "вверх-вниз" расположенные справа от поля. Оценка - целочисленное значение от 1 до 10. При нажатии кнопки "Сохранить" данные записываются в файл list.txt (дозапись в конец файла). Кнопка "Сохранить" становится доступной, только когда пользователь укажет значения всех полей.</p>

22		<p>Содержимое списка изначально загружается из файла list.txt. В списке разрешен выбор по одному элементу. При нажатии кнопки "Вверх" выбранный элемент перемещается на одну строчку вверх. При нажатии кнопки "Вниз" - выбранный элемент перемещается на одну позицию вниз. Если выбран первый элемент списка, то кнопка "Вверх" блокируется. Если выбран последний элемент, то блокируется кнопка "Вниз".</p>
23		<p>Пользователь задает дату с помощью двух списков выбора и текстового поля. В первом списке выбора задается день месяца, во втором списке - месяц. При выборе месяца список дней должен меняться в соответствии с количеством дней в месяце. При нажатии кнопки "Сохранить" дата записывается в файл dates.txt в формате DD/MM/YYYY. Если год указан некорректно, то в этом поле должен выводиться текст "Ошибка!"</p>
24		<p>В текстовом поле пользователь указывает имя каталога. При нажатии клавиши ENTER в этом поле список файлов и каталогов отображается в списке. Если установлен флажок "Только файлы", то отображаются только файлы. При изменении этого флажка список также перегружается.</p>
25		<p>Пользователь в текстовом поле набирает текст. При нажатии клавиши ENTER текст отображается в центре нижней части формы (метка "Текст"). При этом шрифт зависит от состояния флажков. При изменении состояния флажка отображение текста также меняется.</p>

### Контрольные вопросы

1. Какие основные элементы управления могут присутствовать на окнах?

2. Какие присутствуют классы для создания элементов управления?
3. В чем заключается отличие создания окон элементов управления от обычных окон?
4. С помощью каких функций осуществляется взаимодействие с элементами управления?
5. Перечислите виды кнопок, в каких случаях они используются?
6. Перечислите сообщения, которые могут поступать от кнопок.
7. Перечислите сообщения, которые могут быть переданы кнопкам.
8. Для чего используются статический элемент управления?
9. Для чего используются элемент управления редактор текста?
10. В каких режимах может функционировать элемент управления редактор текста?
11. Какие сообщения могут быть получены от редактора текста?
12. Какие сообщения могут быть переданы редактору текста?
13. В каких режимах может функционировать элемент управления список?
14. Какие сообщения могут быть получены от списка?
15. Какие сообщения могут быть переданы списку?
16. В каких режимах может функционировать элемент управления комбинированный список?
17. Какие сообщения могут быть получены от комбинированного списка?
18. Какие сообщения могут быть переданы комбинированному списку?