**Задания для изучения и выработки навыков создания оконных приложений для любой платформы.**

**Task 1**

1. Вывести на экран свое (краткое) резюме с помощью последовательности MessageBox'ов (числом не менее трех). Причем на заголовке последнего должно отобразиться среднее число символов на странице (общее количество символов в резюме / количество MessageBox'ов).
2. Написать функцию, которая "угадывает" задуманное пользователем число от 1 до 2000. Для запроса к пользователю использовать MessageBox. После того, как число отгадано, необходимо вывести количество запросов, потребовавшихся для этого, и предоставить пользователю возможность сыграть еще раз, не выходя из программы.
3. Представьте, что у вас на форме есть прямоугольник, границы которого на 30 пикселей отстоят от границ рабочей области формы.

Необходимо создать следующие обработчики:

* + Обработчик нажатия левой кнопки мыши (OnLButtonDown), который выводит сообщение о том, где находится текущая точка: внутри прямоугольника, снаружи, на границе прямоугольника. Если при нажатии левой кнопки мыши была нажата кнопка Control (Ctrl), то приложение должно закрываться.
  + Обработчик нажатия правой кнопки мыши (OnRButtonDown), который выводит в заголовок окна информацию о размере клиентской (рабочей) области окна в виде: Ширина = x, Высота = y, где x и y - соответствующие параметры высшего окна.
  + Обработчик перемещения указателя мыши в пределах рабочей области (OnMouseMove), который должен выводить в заголовок окна текущие координаты мыши x и y.

**Task 2**

1. Разработать приложение, созданное на основе диалогового окна.
   1. Пользователь "щелкает" левой кнопкой мыши по форме и, не отпуская кнопку, ведет по ней мышку, а в момент отпускания кнопки по полученным координатам прямоугольника (вам, конечно, известно, что двух точек на плоскости достаточно для создания прямоугольника) необходимо создать надпись (Label), которая содержит свой порядковый номер (имеется в виду порядок появления на форме).
   2. Минимальный размер надписи составляет 10х10, при попытке создания элемента меньших размеров пользователь должен увидеть соответствующее предупреждение.
   3. При щелчке правой кнопкой мыши над поверхностью надписи в заголовке окна должна появиться информация о ее площади и координатах (относительно формы). В случае, если в точке щелчка находится несколько надписей, то предпочтение отдается надписи с наибольшим порядковым номером.
   4. При двойном щелчке левой кнопки мыши над поверхностью надписи она должна исчезнуть с формы (удалиться). В случае, если в точке щелчка находится несколько надписей, то предпочтение отдается надписи с наименьшим порядковым номером.
2. Разработать приложение "Убегающая надпись”. Суть приложения: на форме расположен элемент управления Label. Пользователь пытается подвести курсор мыши к нему, и, если курсор находиться близко, элемент управления "убегает". Предусмотреть перемещение контрола только в пределах диалогового окна.

**Task 3**

1. Написать программу, которая по введенной дате определяет день недели. Результат выводить в текстовое поле (желательно по-русски).
2. Написать программу, вычисляющую, сколько осталось времени до указанной даты (дата вводится с клавиатуры в текстовое поле). Предусмотреть возможность выдачи результата в днях, минутах, секундах. Для переключения между вариантами желательно использовать переключатели (RadioButton).
3. Написать игру "Пятнашки", учитывая следующие требования:
   * Реализовать для пользователя возможность "сыграть еще раз"
   * Предусмотреть автоматическую перестановку "пятнашек" в начале новой игры
   * Выводить время, за которое пользователь окончил игру (собрал "пятнашки")

**Task 4**

1. Создать приложение, у которого на форме находится один список (ListBox) и три переключателя (RadioButton). В список заносится от 10 до 20 случайных чисел в диапазоне от -10 до 10 (0 не включается). Переключатели обеспечивают отображение пользователю суммы чисел, произведение чисел, среднее арифметическое чисел.
2. Создайте на форме элемент управления CheckListBox (если на вашей платформе такой есть) и заполните его названиями следующих элементов управления: надпись, кнопка, текстовое поле, комбинированный список, список. Когда пользователь ставит "галочку" на какой-либо строчке, то соответствующий элемент управления должен динамически создаваться и отображаться на форме. При снятии "галочки" соответствующий элемент управления уничтожается (естественно, с формы он исчезнет самостоятельно).

**Task 5**

1. Создать приложение, которое при запуске загружает в два списка (ListBox) названия книг и их авторов (список книг и авторов желательно хранить в файле или в файлах). На форме также должен присутствовать элемент выпадающий список (ComboBox), в котором пользователь может выставить фильтр на отображение книг и авторов. Варианты фильтров: все книги и авторы, фильтр по каждому автору (отображаются только книги, принадлежащие данному автору).
2. Создать приложение, которое осуществляет поиск заданного слова (словосочетания) в заданном файле (файлах) и выводит в список наименование файла и позицию(-и) заданной фразы в нем.
3. Написать простейший вариант записной книжки.
   * Записная книжка должна хранить ФИО человека, его адрес и телефон.
   * Необходимо обеспечить возможности добавления, удаление и модификации записей.
   * Для ввода новых данных и модификации данных необходимо использовать вспомогательную форму, причем для модификации предусмотреть вызов вспомогательной формы с полями, заполненными текущими значениями (текущие значения берутся из той записи, которая на данный момент выделена).
   * Вызов режима модификации продублировать с помощью двойного щелчка мыши по списку.
   * Если для хранения данных используется более одного списка, то после выбора какого-либо элемента списка необходимо обеспечить возможность выделения соответствующих элементов в других списках.
   * Обеспечить возможность сохранения данных в файл и загрузку данных из файла.
4. Реализовать простой файловый менеджер с поддержкой функций:
   * определение и получение списка логических дисков, установленных в системе;
   * отображение файловой структуры для каждого найденного диска;
   * получение информации о каждом элементе структуры;
   * просмотр структуры каталогов любого уровня вложенности;
   * Необходимо сохранять настройки окна: выбранный диск, стиль сортировки, текущий путь (например, если последний путь был c:\a\b, то при следующем запуске программы необходимо стартовать от данного пути, если он еще существует);
   * Добавьте на форму несколько переключателей (Radio), с помощью которых должна осуществляться сортировка файлов и папок в списке. Сортировка должна производиться: по имени (по умолчанию), по времени создания, по времени последней модификации. Причем, каталоги всегда отображаются до файлов;
   * Реализовать функцию подсчета размера каталога.

**Task 6**

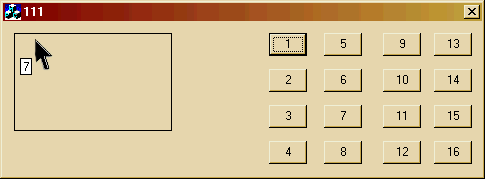
1. Необходимо решить следующую задачу: динамическое создание элементов управления и назначение им функций. По щелчку на форме правой кнопкой выпадает контекстное меню со следующими пунктами: Надпись 3, Текст 3, Кнопка 3. Когда пользователь выбирает пункт меню, то на форме (в точке щелчка) появляется соответствующий элемент управления (Label, TextEdit или Button), а количество элементов уменьшается, например, выставили одну надпись и при следующем отображении меню в нем будут пункты: Надпись 2, Текст 3, Кнопка 3. Если на форме выставлены все элементы одного типа, например, текст, то данный пункт исчезает из меню. После выставления всех девяти элементов (необходимо сделать проверку на это), по щелчку на первой кнопке: Надпись1 = Площадь Окна, Текст1 = площадь окна (посчитанная); по нажатию на вторую: Надпись2 = Площадь Клиентской Области Окна, Текст2 = площадь клиентской области окна (посчитанная); по нажатию на третью: Надпись3 = Площадь Заголовка Окна, Текст3 = площадь заголовка окна (площадь окна - площадь клиентской области окна).
2. Создать произвольное меню (7-8 пунктов, причем из них 3 первого уровня, 2 входят в элементы меню первого уровня, остальные входят в элементы меню 2 уровня). Создать на форме две кнопки. По нажатию на первую кнопку меню должно русифицироваться (т. е. все английские названия переводятся на русский), после повторного нажатия меню вновь становится англоязычным. Вторая кнопка должна делать то же самое. Разница заключается в том, что в первом случае вы должны сделать "подмену" меню (у вас изначально существует два меню: русское и английское), а во втором вы работаете с одним и тем же ресурсом (т. е. осуществляете двухсторонний перевод пунктов одного и того же меню).

**Task 7**

1. Создать маленькую игрушку. На поле находятся 10 кнопок (Button), счетчик (Spin), список (ListBox). Правила: при запуске игры на кнопки помещаются 10 случайных чисел (при этом на счетчике необходимо выставить скорость игры в десятых долях секунды, что будет являться значением для таймера). Задача: необходимо за заданное время щелкнуть по числам на кнопках в порядке возрастания; числа должны добавляться в список (только если данное число является следующим по возрастанию). Если игрок не успевает, то список очищается и игра начинается снова с текущим значением таймера (на счетчике).
2. Дополнить (созданное вами ранее) приложение, которое осуществляет поиск заданного слова (словосочетания) в заданном файле (файлах) и выводит в список наименование файла и позицию(-и) заданной фразы в нем, индикатором, который сообщает пользователю процент выполненной работы.

**Task 8**

1. Создать форму, которая содержит список, содержащий с десяток строк произвольного содержания, текстовое поле и две кнопки. В текстовом поле набирается подсказка и по кнопке назначается на текущую (выделенную) строчку списка. По нажатию другой кнопки подсказка снимается с текущей строчки списка. Учесть возможность наличия полосы вертикальной прокрутки у списка.
2. Создать форму, содержащую несколько произвольных элементов управления. По наведению мыши на любой из них должна отображаться всплывающая подсказка, содержащая размер, стили и расположение (координаты) элемента под курсором относительно главной формы.
3. Создать форму, содержащую в правой половине 4 столбика кнопок по 4 штуки в каждом. Каждая кнопка в качестве заголовка содержит свой порядковый номер. В левой половине формы расположен статик размером 4 ширины кнопки на 4 высоты кнопки, то есть статик условно разбит на прямоугольные области, каждая из которых равна кнопке. Номера областей перемешиваются произвольным образом, и при наведении мышки на какую-нибудь область, должна выпадать подсказка, выводящая номер конкретной области. Задача пользователя заключается в перетаскивании кнопки в область, соответствующую (по номеру) заголовку кнопки. При попытке поставить, например, кнопку с номером 5 в область с номером 3, кнопка должна вернуться на исходную позицию. Примерный вид формы см. на рисунке.



**Task 9**

1. Создать приложение - "Планировщик задач". На форме должен быть список задач и время их выполнения. И то и другое может добавляться, удаляться и модифицироваться. То есть когда подходит заданное время, запускается ассоциированная с этим временем задача. При появлении приложение должно "прятаться" в трэй. После выполнения задачи она помещается в список выполненных задач, где также указывается ее статус, то есть задача была выполнена успешно или неуспешно (в этом случае указать возможную причину).

**Task 10**

1. Создать программу, в которой пользователь может хаотично щелкать левой кнопкой мыши по рабочей области окна (количество щелчков неограниченно), и по нажатию на кнопку, которая также присутствует на форме, на экране должен появится прямоугольник, который содержит в себе все точки, на которые клацал пользователь. Если пользователь нажимает правую кнопку мыши, то на экране также должен появиться прямоугольник, который содержит в себе все точки, на которые клацал пользователь, но при отпускании правой кнопки мыши прямоугольник должен с экрана исчезнуть.
2. Создайте текстовый редактор, который включает в себя элементы:

- строку состояния, которая должна отображать текущую позицию мигающей каретки (то есть текущую строку, текущий столбец, количество символов от начала TextEdit'а до каретки, общее количество символов в файле);

- полноценный диалог для поиска и замены текста (то есть поиск заданного выражения в обе стороны, поиск по отдельному слову, регистрозависимый и регистронезависимый поиск, замену выражений по одиночке и во всем тексте одновременно);

- в редакторе должно быть меню и набор соответствующих акселераторов;

- возможность выбора шрифта.

## Task 11

1. Используя, “Rich Text” контрол реализовать подкраску математических элементов с возможностью управления цветовыми настройками из меню. Например, подкрашиваются числа, названия тригонометрических функций, скобки, арифметические операторы и остальной текст.

## Task 12

1. Дополнить проект из задания Task 11, добавив возможность открывать несколько документов одновременно, для этого можно успешно использовать закладок “Tab” контрол. Также необходимо дополнить проект возможностью настройки текущего или всех одновременно документов. Например, настройка цветов для разных компонент математических формул, настройка отступов в редакторе, настройка выравнивания текста и т. п. Под документом подразумевается вкладка с “Rich Text” контролом.

**Task 13**

Создать класс, имитирующий поведение гиперссылки.

Список минимальных требований:

1. Гиперссылка может содержать текст, иконку или картинку.
2. При наведении мыши, должен изменяться курсор мыши и подчеркиваться текст надписи (изображения должны выделяться рамкой).
3. Если указатель мыши задерживается на гиперссылкой какое-то время, должна появиться всплывающая подсказка, содержащая полный адрес перехода.
4. Цвет надписи определяется состояниями гиперссылки: активная, посещенная, не посещённая.
5. По нажатию на гиперссылку должно выполниться указанное действие.

**Task 14**

Разработать игру "Слова" ("Балда").

Правила игры: имеется игровое поле размером 5х5 клеток, в центральной строке которого расположено произвольное слово (из файла) длиной в 5 букв. Задача играющего дописать к имеющимся на поле буквам одну любую букву таким образом, чтобы получить новое слово, в которое входит данная буква. Слова не могут повторяться, в слове не может использоваться одна и та же буква на игровом поле дважды, слово может читаться в любом направлении, кроме диагонального. За каждую букву составленного слова начисляется одно очко. По желанию играющего ход один раз можно пропустить, при повторном пропуске этому игроку зачисляется поражение. При отсутствии введенного игроком слова в словаре компьютера, компьютер должен запросить уверен ли пользователь, и если да, то компьютер должен внести в словарь это слово.

Требования к оформлению:

1. реализация меню на русском и "буржуйском языках" с возможностью переключения;
2. сворачивание главного окна в SystemTray;
3. буквы должны перетаскиваться на поле мышкой;
4. необходимо предусмостреть возможность настройки "скинов" для кнопок, игрового поля и т. д.;
5. предусмотреть установку времени, которое игрок вправе потратить на ход (по истечении этого интервала ход передается ко второму противнику);
6. реализовать возможность установки паузы;
7. реализовать возможность сохранения (загрузки) текущей игры в файл;
8. сохранять текущие настройки в реестр;
9. реализовать отображения списка лучших игроков;
10. *предусмотреть возможность игры с компьютером (ЖЕЛАТЕЛЬНО !!!).*

Примерный интерфейс игры представлен на следующих рисунках:


**Task 15**

Разработать игру "Линии" ("Lines").

В двух словах: на игровом поле после каждого хода случайным образом появляются 3 шарика разных цветов. Задача играющего собрать вертикальную, горизонтальную или диагональную линию из 5-ти и более шариков одного цвета, в этом случае собранная линия исчезает и за нее начисляются какие-то очки. Игра прекращается при полном заполнении игрового поля.

Можно создать любые "навороты" по желанию программиста.

**Для облегчения жизни вам предлагается алгоритм нахождения кратчайшего пути между двумя точками (пользоваться, естественно, не обязательно).**

**Волновой алгоритм**

Пусть массив P(MxN) отвечает за игровое поле, тогда массив R(MxN) будет рабочим массивом (M и N, соответственно, высота и ширина поля). Число F = M \* N - количество клеток игрового поля.

1. Кaждому элементу рaбочего мaссивa R(i, j) присвaивaется некоторое знaчение из поля P(i, j) по следующим правилам:
   1. Если поле P(i, j) непроходимо, то R(i, j) = F + 2;
   2. Если поле P(i, j) проходимо, то R(i, j) = F + 1;
   3. Если поле P(i, j) является целевой (финишной) позицией, то R(i, j) = 0;
   4. Если поле P(i, j) является стaртовой позицией, то R(i, j) = F.
2. Вводим переменную Ni - счётчик итерaций (повторений) и присвaивaем ей нaчaльное знaчение 0.
3. Построчно проссмaтривaем рaбочий мaссив R (т.е. оргaнизуем двa вложенных циклa: по индексу мaссивa i от 0 до М, по индексу мaссивa j от 0 до N).
4. Если R(i, j) рaвен Ni, то проссмaтривaются соседние элементы R(i+1, j), R(i-1, j), R(i, j+1), R(i, j-1) по следующему прaвилу (в кaчестве примерa рaссмотрим R(i, j+1)):
   1. Eсли R(i, j+1) = F, то переходим к пункту 8;
   2. Eсли R(i, j+1) = F + 1, то выполняется присвaивaние R(i, j+1) = Ni+1;
   3. Во всех остaльных случaях R(i, j+1) остaётся без изменений. Aнaлогично поступaем с элементaми R(i, j-1), R(i-1,j), R(i+1, j).
5. По зaвершению построчного просмотрa всего мaссивa увеличивaем Ni нa 1.
6. Если Ni > F, то поиск мaршрутa признаётся неудачным. Выходим из программы.
7. Переходим к пункту 3.
8. Построение мaршрутa перемещения. Присвaивaем переменным Х и Y знaчения координaт стaртовой позиции.
9. В окрестности позиции R(Х, Y) ищем элемент с нaименьшим знaчением (т.е. для этого проссмaтривaем R(Х+1, Y), R(Х-1, Y), R(Х, Y+1), R(Х, Y-1). Координaты этого элементa зaносим в переменные X1 и Y1.
10. Совершaем перемещение объектa (в нашем случае шарика) по игровому полю из позиции [X, Y] в позицию [X1, Y1]. (По желaнию, вы можете предвaрительно зaносить координaты X1, Y1 в некоторый мaссив, и, только зaкончив построение всего мaршрутa, зaняться перемещением шарика нa экрaне).
11. Если R(X1, Y1) = 0, то переходим к пункту 13.
12. Выполняем присвaивaние X = X1, Y = Y1. Переходим к пункту 9.
13. Конец алгоритма.

**Примечание:** алгоритм можно улучшить или ухудшить по собственному усмотрению.