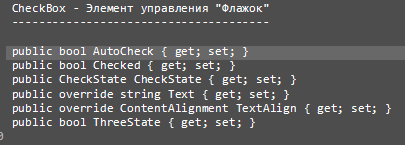
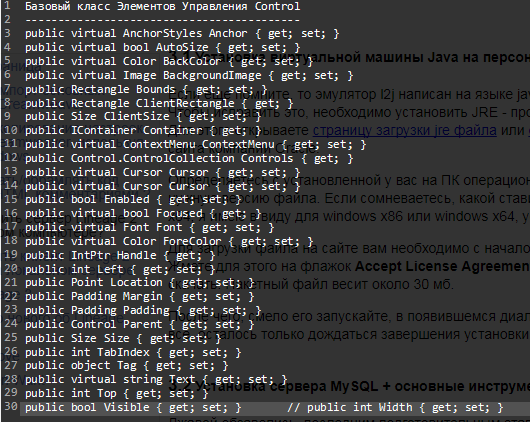


1. Вернет информацию о том, какая кнопка нажата
2. Кол-во кликов
3. Величина прокрутки колесика мыши
4. Расположение мыши в момент события ( клиентские координаты элемента управления, над которым произошло событие)



**Control**

****

* Public virtual AnchorStyles Anchor { get; set; } – якорь. С помощью коря можно прикреплять элемент управления к одной или нескольким сторонам. Делается с целью автоматического изменения размера элемента управления в зависимости от размера области, в которой он находится.
* Public virtual bool AutoSize { get; set; } – не используется
* Public virtual Color BackColor { get; set; } – цвет фона
* Public virtual Image BackgroundImage { get; set; } – фоновая картинка
* Public Rectangle Bounds { get; set; } – размер и расположение элементов управления
* Public Rectangle ClientRectangle { get; } – содержит размеры и координаты клиентской области. Используется для того, чтобы узнать размеры клиентской области.
* Public Size ClientSize { get; set; } – размеры клиентской области
* Public IContainer Container { get; } – возвращает ссылку на контейнер, в котором находится элемент управления
* Public virtual ContextMenu ContextMenu { get; set; } – возвращает ссылку на контекстное меню.
* Public Control.ControlCollection Controls { get; } – коллекция дочерних элементов управления
* Public virtual Cursor Cursor { get; set; } – устанавливает курсор, который будет выводится при наведении
* Public bool Enabled { get; set; } – включает/выключает элемент
* Public virtual bool Focused { get; } – фокус ввода
* Public virtual Font Font { get; set; } - шрифт
* Public IntPtr Handle { get; } – описание элемента управления в операционной системе
* Public object Tag { get; set; } – позволяет задать ссылку на объект, который может содержать дополнительные данные для элемента управления, может идентифцировать элемент управления.

**Менеджер раскладки**

Менеджер раскладки – это контейнер, который автоматически располагает свое дочерние элементы управления по определенному правилу, избавляя программиста от необходимости размещать элементы управления самостоятельно. Особое преимущество использования менеджера раскладок проявляется при изменении размера окон или панелей.

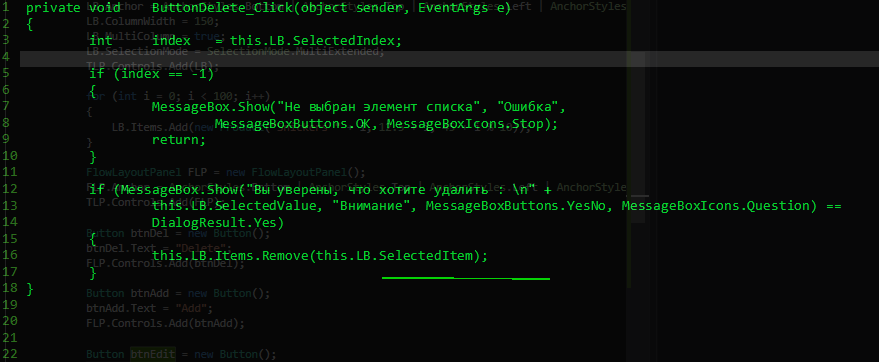
* FlowLayoutPanel – менеджер поточной раскладки
* TableLayoutPanel – менеджер табличной раскладки

*TableLayoutPanel*

* public virtual bool AutoScroll { get; set; } – будет ли прокручиваться содержимое контейнера в случае, когда размер содержимого больше, чем размер контейнера
* public TableLayoutPanelStyle GrowStyle { get; set; } - задает стиль увеличения контейнера в случае добавления в него новых элементов управления. Значения: AddColumns – добавляются столбцы, AddRows – строки, FixedSize – фиксированный размер.
* Public int ColumnCount { get; set; } - задает количество столбцов
* Public int RowCount { get; set; } - задает количество строк

**ListBox**

* Public int ColumnWidth { get; set; } – задает ширину столбца, когда отображение происходит построчно
* Public overriede int SelectedIndex { get; set; } – возвращает индекс выбранного элемента
* Public ListBox.SelectedIndexCollection SelecetedIndexes { get; } – возвращает индексы выбранных элементов
* Public object SelectedItem { get; set; } – возвращает значение выбраенного элемента
* Public ListBox.SelecetedObjectCollection SelectedItems { get; } – возвращает выбранные значение в виде коллекции
* Public object SelectedValue { get; set; } – возвращает выбранное значение
* Public virtual SelectionMode SelectionMode { get; set; } – задает способ выбора элементов



**Диалоговые окна**

* Модальное окно не позволит перейти на другие окна приложения до тех пор, пока не будет закрыто
* Позволяет перейти на другие окна приложения

**Компонент ErrorProvider**

*\* компонент – не элемент управления*

* Public int BlinkRate { get; set; } – частота мигания значка
* Public ErrorBlinkStyle BlinkStyle { get; set; } – значек ошибки: **AlwaysBlink**, **BlinkIfDifferentError**, **NeverBlink**
* public Icon Icon { get; set; } – позволяет задать иконку. [ public Icon(Stream stream) || public Icon(string fileName) ]
* Public void SetError(Control control, String value) – назначает ошибку для элемента управления.
* Public String GetError(Control control) – вращает текст сообщения об ошибки для элемента управления *control.*

**Timer**

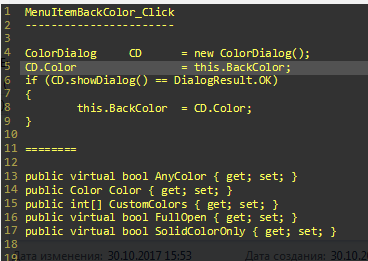
* Public double Interval { get; set; } – интервал
* Public bool Enabled { get; set; } – вкл/выкл
* Public void Start()
* Public void Stop()
* Public event ElapsedEventHandler Elapsed – делегат на метод, который будет вызываться всякий раз по истечении времени

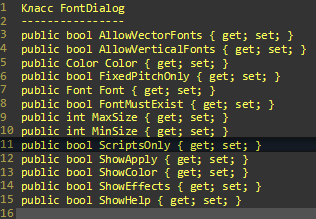
**ProgressBar**

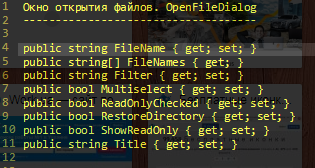
* Public int Maximum { get; set; }
* Public int Minimum { get; set; }
* Public int Step { get; set; } -> public void PerformStep()
* Public ProgressBarStyle Style { get; set; } (Blocks, Continuous, Marquee)
* Public int Value { get; set; }

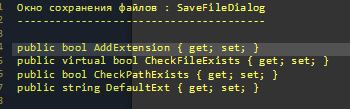
**MenuItem**

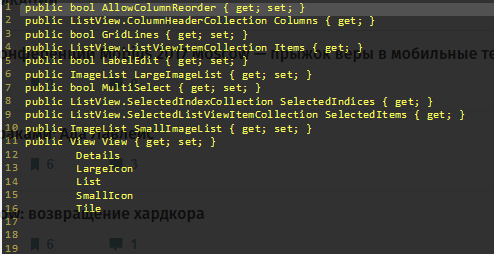
* Public bool BarBreak { get; set; } – будет ли помещатья на новой строке или новом столбце
* Public bool Checked { get; set; } – выбран или нет
* Public bool Enabled { get; set; } – запрещен или разрешен
* Public ShortCut ShortCut { get; set; } – горячия клавиша
* Public bool RadioCheck { get; set } – является ли радиокнопкой
* Public bool ShowShortCut { get; set; }



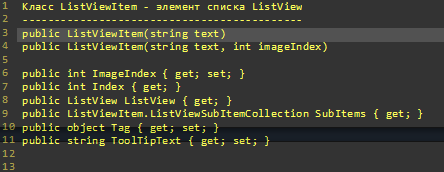




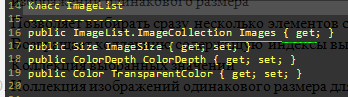




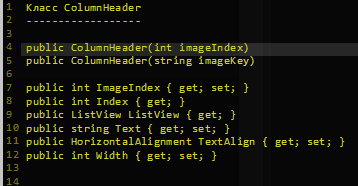
1. Разрешает пользователю перетаскивать столбцы меняя их порядком
2. Возвращает коллекцию столбцов //Column header
3. Если 1, то в режиме таблицы будет отображаться сетка
4. Коллекция элементов (каждый элемент представлен классом ListViewItem)
5. Разрешает редактировать содержание ячеек ListView
6. Коллекция изображений для режима LargeImage //ImageList – класс-коллекция изображений одинакового размера
7. Позволяет выбирать сразу несколько элементов списка
8. Возвращает коллекцию содержащую индексы выбранных элементов
9. Коллекция выбранных значений
10. Коллекция изображений одинакового размера для режима SmallImage
11. Режим отображения элементов списка (Details, LargeIcon, List, SmallIcon, Tile)



1. Текст, который будет отображаться в первом столбце
2. Текст для первого столбца и индекс изображения
3. Вернет индекс изображения, которое связано с выбранным элементом списка, в списке изображений
4. Вернет индекс элемента в списке ListView
5. Вернет ссылочку на ListView, которому принадлежит элемент



1. Коллекция изображений одинакового размера
2. Размер изображения
3. Глубина цвета
4. Содержит цвет, который должен быть прозрачным



1. Принимает индекс изображения которое должно отображаться в заголовке окна
2. Вернет индекс изображения из списка изображений
3. Индекс столбца в коллекции столбцов
4. Вернет ссылку на ListView, к которому относится столбец
5. Заголовок столбца
6. Выравнивание текста
7. Ширина столбца

**TreeView**

Предназначен для отображения списка имеющих иерархическую структуру. Элементы списка TreeView называют узлами. Каждый узел является коллекцией дочерних узлов. Элемент TreeView является объектом класса TreeNode.

