

## **Лекция № 9. Элементы управления в графических и объектно-ориентированных интерфейсах.**

Под элементами управления (controls) обычно понимаются компоненты графического интерфейса, которые предоставляют пользователю возможность изменять содержимое или форму представления отображаемой информации, а также управлять работой приложения. К элементам управления относятся списки, полосы прокрутки, кнопки и т.д.

Каждый элемент управления имеет уникальный образ и обеспечивает определенную форму взаимодействия пользователя с приложением. Система также поддерживает возможность создания собственных элементов управления. Определяя такие элементы, следует учитывать существующие системные соглашения, принятые для стандартных элементов управления.

Элементы управления обеспечивают обратную связь с пользователем. Для элементов управления характерны следующие состояния:

- нормальное – элемент управления не выбран;
- активное – если указатель мыши (курсор) находится в *горячей зоне* или фокусе ввода;
- состояние выбора – активный элемент управления выбран (запущено соответствующее элементу управления действие);
- «недоступно» - элемент управления недоступен на данном этапе работы.

Горячая зона определяет, будет ли элемент управления реагировать на указатель. Границы горячей зоны зависят от типа элемента. Для некоторых элементов, например, кнопок, горячая зона совпадает с видимой границей элемента. Для других горячая зона может включать графический символ элемента управления и относящуюся к нему текстовую область (это справедливо для флажков и переключателей).

Для большинства элементов управления система обеспечивает вывод текстовой подсказки. Подсказка помогает пользователю определить назначение данного элемента управления. Если элемент управления не имеет подсказки, ее можно реализовать в виде статической текстовой области или в виде всплывающей подсказки (tooltip).

Поскольку некоторые элементы управления могут обеспечивать специфические способы взаимодействия пользователя с приложением, для них целесообразно создавать всплывающие меню (если элемент управления используется для передачи величины, имеющей несколько возможных значений, или для доступа к контекстно-зависимой справочной информации). Такие меню создаются по стандартным правилам, но щелчок

левой клавиши мыши на выбранном пункте не запускает действие, связанное с элементом управления. Т.е. всплывающее меню элемента управления позволяет пользователю определить, какие действия он реализует в текущей ситуации, но не позволяет непосредственно выполнить эти действия.

### **Меню как элемент управления.**

Меню предполагает выбор чего-либо из предложенного списка. Для этого предусмотрены два основных процесса: навигация и выбор.

Навигация (перемещение) реализуется с помощью указателя мыши; клавишами управления курсором.

Выбор реализуется двойным щелчком левой клавиши мыши или нажатием клавиши <ENTER>.

Быстрый выбор действия или раздела меню может быть достигнут при использовании клавиш-акселераторов (горячих ключей), нажимаемых совместно с клавишей <ALT>.

### ***Главное меню окна (main menu).***

Одна из наиболее распространенных форм меню – линейная последовательность команд или разделов. В таком виде выполнено главное меню окна, расположенное непосредственно под полосой заголовка первичного окна (так называемая полоса меню).

Полоса меню содержит названия пунктов меню, каждый из которых предоставляет доступ к выпадающему меню.

Содержание главного меню и связанных с ним выпадающих меню определяется функциональным предназначением приложения и контекстом выполняемого пользователем задания. Если установлена такая конфигурация окна, при которой главное меню не отображается, то необходимо использовать элементы управления, которые обеспечат доступ к тем же функциям приложения, что и меню.

### ***Выпадающее меню (pull-down).***

Отображается как панель с пунктами меню, расположенными в виде столбцов. Хотя система позволяет выводить пункты меню в несколько столбцов, но делать это нежелательно, т.к. это усложняет работу пользователя.

Перемещение по выпадающему меню с помощью клавиш управления курсором рекомендуется организовывать по кругу, т.е. если указатель стоит на последнем пункте меню и курсор перемещен вниз, то произойдет переход курсора на первый пункт, а с первого при перемещении вверх – на последний.

Пользователь может закрыть выпадающее меню нажатием клавиши <ALT>, при этом также деактивируется главное меню.

Нажатие клавиши <ESC> также позволяет закрыть выпадающее меню, но отменяет только меню текущего уровня и оставляет выделенным соответствующий пункт главного меню. Повторное нажатие клавиши <ESC> отменяет выделение пункта главного меню и деактивирует полосу меню, возвращая фокус ввода на информацию, отображаемую в окне.

Использование клавиш-акселераторов позволяют выполнить действие выпадающего меню без его отображения.

### ***Всплывающие меню (pop-up).***

Всплывающее меню предоставляет пользователю эффективный способ доступа к операциям над объектами.

Отображается в текущей позиции (соответствующей положению указателя) – избавляет пользователя от необходимости перемещаться по экрану для выбора действия через меню или панель инструментов.

Всплывающее меню содержит команды, учитывающие специфику выбранного объекта или текущей ситуации, а следовательно сокращает число команд, среди которых пользователь должен сделать выбор.

Позволяют минимизировать объем отображаемой на экране информации, поскольку появляются по требованию пользователя.

Не следует использовать всплывающее меню в качестве единственного доступного средства выполнения каких-то действий. Команды всплывающего меню не должны дублировать содержимое одного из выпадающих меню.

### ***Принципы расположения команд во всплывающем меню:***

- первыми должны располагаться основные команды для работы с объектом (открыть, исполнить, печать), другие команды, поддерживаемые объектом (определяемые непосредственные его свойствами или текущим контекстом), и команда *Что это?*, если она поддерживается системой;
- во вторую группу включают команды, реализуемые через буфер обмена (вырезать, копировать, вставить);
- последними должны идти команды редактирования дополнительных атрибутов объекта, если таковые имеются.

Открывает всплывающее меню щелчок правой клавишей мыши на выбранном объекте. При этом объект изображается как выбранный.

Отображается всплывающее меню так, чтобы его левый верхний угол совпадал с позицией указателя; однако если при этом меню выходит за пределы экрана, его положение должно быть скорректировано.

Если при открытом всплывающем меню пользователь щелкает правой клавишей мыши за пределами области выбора, то открытое ранее всплывающее меню должно быть закрыто и открыто меню, относящееся к новой области выбора.

Всплывающее меню закрывается при нажатии клавиши <ESC> и при щелчке за пределами меню.

Для клавиатурного доступа ко всплывающим меню используется комбинация клавиш <Shift>+F10 или клавиша <Application> для клавиатур Windows.

Не рекомендуется использовать для работы со всплывающими меню клавиши-акселераторы, поскольку для их описания на панели меню требуется дополнительное пространство.

#### ***Каскадные меню.***

Это подменю, на которое распадается пункт меню более высокого уровня. Визуально на наличие каскадного меню указывает треугольник, выводимый рядом с родительским пунктом меню.

Каскадное меню может использоваться для предоставления пользователю возможности дополнительного выбора и для отображения иерархически связанных объектов.

Использование каскадных меню усложняет интерфейс, поэтому старайтесь применять этот вид только в тех ситуациях, где они действительно необходимы; минимизируйте количество уровней (в идеале – единственное подменю); не используйте их для доступа к распространенным, часто используемым командам.

Взаимодействие пользователя с каскадным меню подобно выпадающему меню, но каскадное меню отображается после некоторой задержки.

Если после открытия каскадного меню пользователь перемещает указатель к другому пункту родительского меню, то каскадное меню после короткой задержки закрывается. Эта задержка позволяет пользователю перейти из родительского меню в соответствующее каскадное без нажатия кнопки мыши.

#### **Требования к оформлению меню.**

1. Если меню содержит большое количество пунктов, и они могут быть сгруппированы по некоторому признаку, следует разделить такие группы (стандартный разделитель – горизонтальная прямая линия).

2. Следует визуально выделять недоступные в данный момент пункты. Обычно их «обесцвечивают» или не отображают совсем. Предпочтителен первый способ.
3. Для недоступных выборов следует сохранять функции подсказки (что собой представляет данная команда и почему она недоступна).
4. Если в некоторой ситуации все пункты меню недоступны, то сделайте недоступным все меню в целом.
5. Если команда меню требует ввода дополнительной информации, ее имя должно сопровождаться многоточием (... - эллипсис).
6. Если пункт меню служит для включения/выключения параметра состояния объекта, то слева от пункта меню при включении параметра устанавливается маркер флажка, при выключении – маркер снимается (такие параметры называются взаимонезависимыми).
7. Для визуального отображения выбора взаимозависимых пунктов меню используется маркер переключателя, слева от выбранного пункта меню.
8. Если две команды меню реализуют альтернативные состояния, то после выбора одного из них его название можно заменить в меню на альтернативное.
9. Всегда отображайте значения клавиш-акселераторов.
10. Используйте уникальные имена для пунктов в пределах одного меню; пункты с тем же названием могут повторяться в других меню, чтобы представлять аналогичные действия.

### **Кнопки (button).**

Кнопкой называется элемент управления, всё взаимодействие пользователя с которым ограничивается одним действием – нажатием.

Нажатие на такую кнопку запускает какое-либо явное действие, поэтому правильнее называть такие кнопки «кнопками прямого действия».

#### *Размеры и поля.*

Чем больше кнопка, тем легче попасть в нее курсором (закон Фитса). Кроме того, пользователю должно быть трудно нажать не на ту кнопку. Добиться этого можно либо изменением состояния кнопки при наведении на неё курсора, либо установлением пустого промежутка между кнопками.

Считать экранную кнопку нажатой нужно не тогда, когда пользователь нажимает кнопку мыши, а курсор находится на кнопке, а тогда, когда пользователь отпускает

нажатую кнопку мыши, курсор находится на экранной кнопке и находился на ней в момент нажатия кнопки мыши.

#### *Объем.*

Кнопка должна (или не должна) быть пользователем нажата. Соответственно, пользователю нужно как-то сигнализировать, что кнопка нажимаема. Лучшим способом такой индикации является придание кнопке псевдообъема, т. е. визуальной высоты.

Направление теней во всех элементах управления должно быть одинаковым: снизу справа.

#### *Состояния.*

Кнопка должна как-то показывать пользователям свои возможные и текущие состояния. Количество состояний довольно велико, при этом наборы возможных состояний в ПО и в интернете значительно различаются. Например, кнопка в Windows может иметь шесть состояний: нейтральное, нажатое, нейтральное с установленным фокусом ввода, состояние кнопки по умолчанию, кнопка по умолчанию с установленным фокусом ввода и заблокированное состояние (см. рис. 1).

В интернете обычно используют меньший набор состояний: нейтральное, готовое к нажатию (onMouseOver) и активное (в случаях, когда набор кнопок используется для индикации навигации). Нажатое и заблокированное состояние используются очень редко, а «нейтральное с установленным фокусом ввода» обычно создается браузерами.

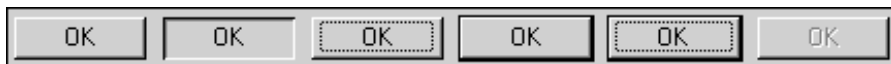


Рис. 1 Состояния кнопки в Windows: нейтральное, нажатое, нейтральное с установленным фокусом ввода, состояние кнопки по умолчанию, по умолчанию с установленным фокусом ввода и заблокированное.

Считается что чем больше набор состояний, тем лучше. Главное правило – состояния не должны дублироваться, т.е. не должно быть разных состояний, выглядящих одинаково. Также очень важно делать заблокированные состояния действительно заблокированными - не только выглядеть менее яркими и значительными, чем обычные, но и не вызывать никаких команд, не нести гипертекстовых ссылок (в веб-интерфейсах).

Никогда не удаляйте элементы, которые нельзя нажать, взамен этого делайте их заблокированными.

#### ***Флажки (чекбоксы) и переключатели (радиокнопки)***

Первое, что необходимо сказать про чекбоксы и радиокнопки, это то, что они являются кнопками отложенного действия, т. е. их нажатие не должно инициировать

какое-либо немедленное действие. С их помощью пользователи вводят параметры, которые скажутся после, когда действие будет запущено иными элементами управления. Нарушать это правило опасно, поскольку это серьезно нарушит сложившуюся ментальную модель пользователей. В этом заключается общность чекбоксов и радиокнопок.

Главное различие заключается в том, что группа чекбоксов даёт возможность пользователям выбрать любую комбинацию параметров, радиокнопки же позволяют выбрать только один параметр.

Из этого различия проистекают все остальные. Например, в группе не может быть меньше двух радиокнопок (как можно выбрать что-либо одно из чего-либо одного?). Еще одно следствие заключается в том, что у чекбокса есть три состояния (выбранное, не выбранное, смешанное), а у радиокнопки только два, поскольку смешанного состояния у неё быть просто не может (нельзя совместить взаимоисключающие параметры).

В группе радиокнопок как минимум одна радиокнопка должна быть проставлена по умолчанию. Всякий раз, когда пользователю нужно предоставить выбор между несколькими параметрами, можно использовать либо чекбоксы, либо радиокнопки (или списки). Если параметров больше двух, выбор прост: если параметры можно комбинировать, нужно использовать чекбоксы (например, текст может быть одновременно и жирным и курсивным); если же параметры комбинировать нельзя, нужно использовать радиокнопки (например, текст может быть выровнен или по левому, или по правому краю).

Если же параметров всего два и при этом параметры невозможно комбинировать (т.е. либо ДА, либо НЕТ), решение зависит от природы выбора: в ряде ситуаций группу из двух радиокнопок можно заменить одним чекбоксом.

#### *Внешний вид.*

Традиционно сложилось так, что чекбоксы выглядят как квадраты, а радиокнопки – как кружки. Нарушать это правило нельзя.

Желательно вертикально располагать чекбоксы и радиокнопки в группе, поскольку это облегчает поиск конкретного элемента.

#### *Текст подписей.*

Каждая подпись должна однозначно показывать эффект от выбора соответствующего элемента.

Поскольку радиокнопки и чекбоксы не вызывают немедленного действия, формулировать подписи к ним лучше всего в форме существительных, хотя возможно

использование глаголов (если изменяется не свойство данных, а запускается какое-либо действие).

Подписи к стоящим параллельно кнопкам лучше стараться делать примерно одинаковой длины.

Все подписи обязаны быть позитивными (т. е. не содержать отрицания).

Повторять одни и те же слова, меняя только окончания подписей (например, «Показывать пробелы» и «Показывать табуляции»), в нескольких кнопках нельзя, в таких случаях лучше перенести повторяющееся слово в рамку группировки.

Если подпись не помещается в одну строку, выравнивайте индикатор кнопки (кружок или квадрат) по первой строке подписи.

#### *Взаимодействие.*

Нажимабельным должен быть не только визуальный индикатор переключения, т. е. кружок или квадратик, но и подпись, просто потому, что закон Фитса однозначно требует больших кнопок.

В интернете первым признаком профессионально разработанного интерфейса являются нажимабельные подписи к чекбоксам и радиокнопкам.

При необходимости заблокировать элемент, желательно визуально ослаблять не только квадрат или круг, но и подпись.

#### *Вариант для панелей инструментов*

Как чекбоксы, так и радиокнопки, бывают двух видов: описанные выше стандартные, и предназначенные для размещения на панелях инструментов.

У них есть определенный недостаток: они не различаются внешне. Это не очень критично, поскольку панелями инструментов пользуются в основном сравнительно опытные пользователи. Тем не менее, на панелях инструментов полезно располагать группы радиокнопок отдельно от групп чекбоксов (чтобы они не смешивались в сознании пользователей).

Графические версии чекбоксов и радиокнопок можно располагать и в диалоговых окнах. Делать это, однако, не рекомендуется, поскольку в окнах они выглядят как командные кнопки, кроме того, такие кнопки не подразумевают подписей.

Обратите внимание, что на панелях инструментов чекбоксы и радиокнопки могут быть кнопками прямого действия.

#### *Списки*

Все часто используемые списки функционально являются вариантами чекбоксов и радиокнопок. Скорость доступа к отдельным элементам и наглядность в них принесены в



жертву компактности (они экономят экранное пространство, что актуально, если количество элементов велико) и расширяемости (простота загрузки в списки динамически изменяемых элементов делает их очень удобными при разработке интерфейса, поскольку это позволяет не показывать пользователю заведомо неработающие элементы).

Списки бывают пролистываемыми и раскрывающимися, причем пролистываемые могут обеспечивать как единственный (аналогично группе радиокнопок), так и множественный выбор; раскрывающиеся же работают исключительно как радиокнопки.

Ширина списка как минимум должна быть достаточна для того, чтобы пользователь мог определить различия между элементами.

В идеале ширина всех элементов должна быть меньше ширины списка, но иногда это невозможно. В таких случаях не стоит добавлять к списку горизонтальную полосу прокрутки, лучше урезать текст элементов.

Для этого нужно определить самые важные фрагменты текста, после чего все остальное заменить эллипсисом (...).

Поскольку нужно максимально ускорить работу пользователей, необходимо сортировать элементы. Идеальным вариантом является сортировка по типу элементов. Если же элементы однотипны, их необходимо сортировать по алфавиту, причем списки с большим количеством элементов полезно снабжать дополнительными элементами управления, влияющими на сортировку или способ фильтрации элементов. Если можно определить наиболее популярные значения, их можно сразу расположить в начале списка, но при этом придется вставлять в список разделитель, а в систему – обработчик этого разделителя.

*Раскрывающиеся (выпадающие) списки.*

Самым простым вариантом списка является раскрывающийся список.

Помимо описанных выше достоинств списков, раскрывающиеся списки обладают еще одним преимуществом: малая высота списка позволяет с большой легкостью визуально отображать команды, собираемые из составляющих (рис.2).



Рис. 2 Пример визуальной сборки команды из составляющих.

Подобно списку единичного выбора, выпадающий список (Drop-down List Box) предусматривает возможность выбора единственного пункта; различие заключается в том, что выпадающий список отображается на экране только по требованию пользователя. Когда список свернут, в его окне отображается только выбранный пункт. Чтобы изменить выбор пользователь должен открыть список.

Раскрывающийся список, как правило, вызывает две проблемы, одна появляется преимущественно в ПО, другая – в интернете.

В раскрывающийся список рекомендуется добавлять «метаэлемент», включающий все элементы из списка. Такой метаэлемент нужно снабжать названием, например, «Все значения» или «Ничего».

#### *Пролистываемые списки*

Другим, более сложным вариантом списка является пролистываемый список. Пролистываемые списки могут позволять пользователям совершать как единственный, так и множественный выбор.

Размер. По вертикали в список должно помещаться как минимум четыре строки, а лучше восемь, но не больше, чем элементов в списке.

*Списки единственного выбора (Single Selection List Box).* Используются для выбора только одного пункта в списке. Список единственного выбора является промежуточным вариантом между группой радиокнопок и раскрывающимся списком. Он меньше группы радиокнопок с аналогичным числом элементов, но больше раскрывающегося списка. Соответственно, использовать его стоит только в условиях «ленивой экономии» пространства экрана.

*Списки расширенного выбора (Extended List Box).* Могут быть использованы для выбора единственного пункта или единственной области, хотя по умолчанию предназначены для выполнения непересекающегося выбора.

*Модифицируемый список (List View Control).* Представляет собой особую форму расширенного списка, который отображает набор пунктов, каждый из которых содержит пиктограмму и текстовую метку. Содержимое модифицируемого списка может быть представлено в одном из четырех видов: пиктограмма, маленькая пиктограмма, список, отчет (таблица). Пример такого списка – содержимое папки Windows.

*Модифицируемое дерево (Tree View Control).* Является частным случаем модифицируемого списка, в котором содержимое отображается с учетом логического и иерархического соотношения между пунктами списка. В таком списке имеются кнопки, которые позволяют изменять форму представления структуры списка в целом и/или отдельных пунктов: они могут отображаться либо в развернутом, либо в свернутом виде. Модифицируемое дерево обычно используется в тех случаях, когда необходимо отобразить отношение между набором контейнеров или других иерархических элементов. (Пример – справочная система Windows).

*Списки множественного выбора (Multiple Selection List Box).* Предназначены для независимого выбора нескольких пунктов (подобен прокручиваемому списку флажков).

*Комбинированный список, выпадающий комбинированный список - Комбобоксы (Combo Box, Drop-down Combo Box).*

Комбобоксами (combo box), называются гибриды списка с полем ввода: пользователь может выбрать существующий элемент, либо ввести свой.

Комбобоксы бывают двух видов: раскрывающиеся и расширенные. Оба типа имеют проблемы.

Главная проблема раскрывающегося комбобокса заключается в том, что такие комбобоксы выглядят в точности как раскрывающиеся списки, визуально отличаясь от них только наличием индикатора фокуса ввода (да и то, только тогда, когда элемент выделен). Это значит, что полноценно пользоваться ими могут только сравнительно продвинутые пользователи.

Проблемы расширенных комбобоксов, напротив, совершенно иные. Они имеют уникальный вид, отличающий их от остальных элементов управления, но при этом расширенный комбобокс потребляет много места на экране.

Поскольку комбобоксы являются гибридами списков и полей ввода, к ним применимы те же требования, что и к их спискам.

### ***Поля ввода – текстовые поля (Check-Box, Rich-Text Box)***

Вместе с командными кнопками, чекбоксами и радиокнопками, поля ввода являются основой любого интерфейса. В результате требований к ним довольно много.

#### ***Размеры.***

Основная часть требований к полям ввода касается размера.

Размер по вертикали должен быть производным от размера вводимого текста – если текста много, нужно добавить несколько строк.

Ширина поля (размер по горизонтали) должна соответствовать объему вводимого текста, поскольку гораздо удобнее вводить текст, который видишь. Менее очевидным является другое соображение: ширина поля ввода не должна быть больше максимальной длины строки

#### ***Подписи.***

Вопрос «где надо размещать подписи к полям ввода?» является одним из самых популярных среди программистов. Один из подходов заключается в том, что, поскольку восприятие подписей занимает определенное время, лучше всего действует следующее простое правило: в часто используемых экранах подписи должны быть сверху от поля (чтобы их было легче не читать), в редко же используемых подписи должны быть слева (чтобы всегда восприниматься и тем самым сокращать количество ошибок).

Кроме того, подписи к полям ввода можно размещать не рядом с элементом, а внутри него, что позволяет экономить пространство экрана. Подпись при этом выводится в самом поле ввода, точно так же, как и текст, который в него нужно вводить. Необходимо только отслеживать фокус ввода, чтобы при установке фокуса в поле убирать подпись.

### ***Дискретное текстовое поле (Spin Box – «Крутилка»)***

Дискретное текстовое поле представляет собой текстовое поле, которое может принимать только ограниченный набор дискретных упорядоченных значений, образующих замкнутую последовательность. Данный элемент интерфейса является комбинацией текстового поля и специального элемента управления, который состоит из двух взаимосвязанных кнопок (он известен также как элемент реверсивного управления – Up-Down Control). В русскоязычных изданиях этот элемент называют: спин, спиннер, крутилка.

Крутилка обладает двумя полезными возможностями. Во-первых, чтобы ввести значение в крутилку, пользователю не обязательно бросать мышь и переносить руку на клавиатуру (в отличие от обычного поля ввода). Поскольку перенос руки с места на место занимает сравнительно большое время (в среднем почти половину секунды), к тому же ещё и сбивает фокус внимания, отсутствие нужды в клавиатуре оказывается большим благом. Во-вторых, при вводе значения мышью система может позволить пользователям вводить только корректные данные, в корректном формате. Это резко уменьшает вероятность человеческой ошибки.

### ***Ползунки***

Как и ранее описанные элементы управления, ползунки позволяют пользователям выбирать значение из списка, не позволяя вводить произвольное значение. Ползунки незаменимы, если пользователям надо дать возможность выбрать значение, стоящее в хорошо ранжирующемся ряду, в ситуациях, когда:

- значений в ряду много;
- нужно передать пользователям ранжируемость значений;
- необходимо дать возможность пользователям быстро выбрать значение из большого их количества (в таких случаях ползунков оказывается самым эффективным элементом, хотя и опасен возможными человеческими ошибками).

Ползунки можно также использовать для выбора текстовых параметров, но только в случаях, когда эти параметры можно понятным образом отранжировать.

### ***Полосы прокрутки***

Когда графических интерфейсов еще не было, пользователи перемещались по документу с помощью клавиатуры. С тех далёких времен на клавиатуре остались клавиши Home и End, равно как Page Up и Page Down. Затем появились графические интерфейсы. Первым делом были придуманы полосы прокрутки. К сожалению, оказалось, что они работают не слишком хорошо.

Проблема полос прокрутки заключается в следующем: для маленьких документов они не очень нужны, поскольку пользователям, держащим руки на клавиатуре, гораздо легче переместиться к нужному фрагменту с помощью клавиш со стрелками. Напротив, в больших документах малое перемещение ползунка приводит к существенному сдвигу области просмотра, так что после перемещения нужно еще и подправлять положение курсора либо клавиатурой, либо стрелками на полосе прокрутки. Более того: во многих случаях невозможно реализовать динамическое изменение области просмотра во время перемещения ползунка, а значит, перемещаться по большим документам приходится в несколько шагов. Кроме того, пользователю приходится постоянно переводить взгляд с документа на ползунок и обратно, и при этом еще следить за областью просмотра, что занимает время, снижает внимание и приводит к ошибкам.

К сожалению, вовсе не использовать полосы прокрутки в ПО затруднительно,

Если всё-таки приходится оставлять полосы прокрутки, крайне желательно добиться нескольких свойств полосок:

- размер ползунка должен показывать общий объем пролистываемого документа;
- стрелки на полосах должны быть спаренными, т. е. обе стрелки должны находиться рядом, а не на разных сторонах полоски (это один из случаев, когда логичность интерфейса вступает в противоречие с эффективностью: если при перелистывании была допущена ошибка, спаренные кнопки позволяют минимизировать перемещение курсора к стрелке, ведущей в обратную сторону);
- если невозможно сделать динамическое изменение области просмотра при пролистывании, необходимо показывать текущее местоположение пользователя во всплывающей подсказке (местоположение подсказки при перемещении курсора должно оставаться неизменным);
- необходимо обеспечить обработку погрешности перемещения курсора (когда пользователь курсором перемещает ползунок, а смотрит в это время на документ, курсор может сойти с полосы; до определённого момента - смещение на 30-70 пикселей - система должна такое смещение игнорировать).

### ***Статус-строка.***

Строка статуса предназначена для двух вещей: она может быть либо собственно строкой статуса, т. е. отображать текущее состояние системы, либо быть панелью инструментов для опытных пользователей (или же делать и то, и другое).

*Отображение текущего состояния системы.* Практически каждая система имеет свойства, либо зависящие от документа, либо изменяющиеся со временем. Когда система долгое время занята, она должна показывать пользователю индикатор степени выполнения. И, наконец, самый простой пример: пользователь текстового процессора имеет право знать, на какой странице документа он сейчас находится. Эффективнее всего выводить всё это в строке статуса.

Пример: Статусная строка Adobe PhotoShop. Слева отображается текущий масштаб отображения документа, вслед за ним объем занимаемой документом памяти (стрелка переключает тип показываемой информации), затем индикатор степени выполнения, а справа – контекстная подсказка (место оставалось, вот его и заполнили).

Строка статуса особенно интересна как место вывода индикатора степени выполнения. Существует занятая закономерность: по месту вывода индикатора выполнения можно определить качество интерфейса системы: если индикатор выводится в строке статуса, то система обладает в целом хорошим интерфейсом, если же индикатор выводится в другом месте – не столь уж хорошим.

*Панель инструментов для опытных пользователей.* Зачастую система обладает функциональностью, которая с одной стороны важна, а с другой – способна свести с ума неподготовленного пользователя. Обычно это касается не столько собственно функций, сколько режимов работы системы. В таких случаях строка состояния является отличным решением проблемы. С одной стороны, делая переключатели режимов непохожими на поля вывода (знаете ли вы, например, что метки ЗАП, ИСПР, ВДЛ и ЗАМ в статусной строке MS Word не только индикаторы?) можно снизить вероятность ошибочного переключения. С другой стороны, если уж пользователь нечаянно щелкнет на переключателе, он сразу же увидит изменение его вида и впоследствии, вероятно, сможет переключиться назад. С третьей стороны, опытный пользователь сможет переключаться между режимами так же легко, как если бы он переключался через панель инструментов.

### ***Индикатор состояния процесса (Progress Indicator, Progress Bar).***

Это элемент интерфейса, который обычно используется для того, чтобы отобразить ход выполнения какой-либо длительной операции (процесса). Он состоит из прямоугольной зоны, которая «заполняется» слева направо.

Индикатор не является интерактивным элементом, он только отображает информацию. Полезно снабдить его текстом, поясняющим его назначение (текст располагается вне индикатора).

Индикатор используется в качестве средства обратной связи для длинных операций или фоновых процессов.

### ***Область сообщений (Status Notification, System Tray).***

Приложение может поместить здесь специальный индикатор или сообщение, уведомляющее пользователя о какой-либо ситуации; выведенное сообщение сохраняется в области сообщений даже тогда, когда приложение находится в неактивном состоянии.

Поскольку Панель задач – коллективный ресурс, используемый всеми активными приложениями, в область сообщений следует выводить только такую информацию, которая носит «глобальный» характер или необходима пользователям при работе с другими приложениями.

Для отображения информации в области сообщений следует использовать пиктограммы.

При отображении информации в области сообщений придерживаются следующих рекомендаций.

1. Обеспечьте появление на экране всплывающего окна, которое содержит дополнительную информацию или средства управления для объекта, представленного индикатором в области сообщений; для вызова окна обычно используют щелчок левой клавишей мыши на изображении индикатора. Если нет необходимости в выводе дополнительной информации, всплывающее окно создавать не стоит.
2. Для объекта, представленного индикатором, должно поддерживаться всплывающее меню. Это меню должно содержать основные команды панели свойств соответствующего объекта.
3. Создайте всплывающую подсказку для индикатора.
4. Предоставьте пользователю возможность не отображать индикатор в области сообщений.

### ***Панели инструментов.***

Все панели имеют следующие достоинства:

- они позволяют пользователям быстро вызывать нужные функции мышью;
- они позволяют пользователям меньше задействовать память;

- они повышают визуальное богатство интерфейса;
- они ускоряют обучение работе с системой благодаря своей высокой наглядности.

Недостаток: занимают много места на экране, так что поместить в них всё, что хочется, невозможно. Решается эта проблема двумя способами. Во-первых, можно (и нужно) помещать в панель только наиболее часто используемые команды (поддерживая это решение возможностью индивидуальной настройки панели пользователем). Во-вторых, панель можно сделать зависимой от контекста действий пользователя. Оба способа не противоречат друг другу, так что использовать стоит оба.

Панель инструментов нежелательно делать единственным способом вызова функции.