**Особенности проектирования «инфографичных» пользовательских интерфейсов программных приложений для универсальной платформы Windows (UWP)**

В процессе создания сложных технических систем и порождаемых в них технических средах в деятельности операторов наблюдается тенденция «очеловечивания» техники. Работа с техническими устройствами сопровождается появлением у пользователей чувства присутствия. Пользователь в ходе взаимодействия с техническими устройствами используют только ту часть устройства, которая доступна его интерпретациям. По мере развития технических устройств и опыта взаимодействия с техническими устройствами у пользователя растет чувство присутствия. Создается иммерсивная среда - любая искусственная физическая или виртуальная среда в её единстве с включенным, погруженным в неё субъектом [80].

**Новые принципы проектирования**

В связи с необходимостью учета иммерсивности взаимодействия в последнее время все больше внимания уделяется проектированию пользовательских интерфейсов программных приложений с учетом следующих принципов [33]: мастерство; лаконичность; быстрота и плавность; естественная цифровая среда; единство.

При проектировании пользовательских интерфейсов необходимо соблюдать расположение элементов управления с использованием типографской сетки. Она предоставляет эстетические и практические преимущества для соразмерного размещения элементов управления (рис. 44). Разные типы информации должны отображаться с помощью разных значений свойств одних и тех же элементов управления. За счет такого подхода можно строить различные уровни иерархии информации на пользовательском интерфейсе. Например, для разделения разных уровней иерархии использованы разные размеры и цвета шрифтов, расположение и интервалы (рис. 44). При проектировании пользовательского интерфейса необходимо подбирать такие шрифты для отображения сообщений, которые наилучшим образом отражают предпочтения пользователей по восприятию визуальной информации.

Для удобства работы пользователей для отражения названия выбранных элементов управления необходимо использовать крупный текст и высокую цветовую контрастность. На рис. 40 заголовок имеет высокую контрастность (белый цвет на черном фоне). Это означает, что он является выделенным на странице. Остальные заголовки элементов имеют более низкую контрастность (серый цвет на черном фоне). Благодаря этому они не выделяются на странице. Текстом среднего размера на рис. 40 выделен следующий уровень иерархии (имена лиц, с которыми происходит обмен сообщениями). Текстом самого малого размера на рис. 40 выделяются первые одна-две строки в каждом сообщении. Прочитанные сообщения отделяются от непрочитанных сообщений и от имени отправителя цветом (цвет, отличный от серого цвета, на черном фоне). Для подчеркивания иерархии используются различные значения интервалов между уровнями вложения информации. Без иерархичности информации диалоговое окно пользовательского интерфейса теряет смысловую нагрузку, эстетическую привлекательность и удобство использования.



Рисунок 40 Использование типографской сетки и иерархии в пользовательском интерфейсе.

Лаконичность пользовательского интерфейса заключается в том, что он содержит только основные функции программного приложения, с которым пользователь взаимодействует в данный момент. Особое внимание обращается не на внешний вид пользовательского интерфейса, а на его содержимое. Содержимое может существовать в виде изображений, сообщений электронной почты, новых статей и т. п. При этом, необходимо, чтобы пользовательский интерфейс имел в составе, где это возможно, элементы навигации в содержимом и создавал иммерсивное взаимодействие пользователя и программного приложения (функции). Элементы навигации своим видом должны помочь пользователю понять, каким образом лучше взаимодействовать с приложением.

В соответствии принципом быстроты и плавности пользовательский интерфейс, с которым взаимодействует пользователь, должен предлагать пользователю привлекательные анимации, переходы и отклики, которые могут быть неожиданными, но оживляют интерфейс и могут понравиться пользователям.

В случае использования планшета или мобильного телефона пользовательский интерфейс программного приложения должен основываться на сценариях, отражающих пожелания пользователей.

Требованиям быстроты и плавности удовлетворяет использование в пользовательском интерфейсе живых плиток, которые отражают все уведомления, которые доступны пользователю.

Принцип естественной цифровой среды при проектировании пользовательских интерфейсов предусматривает переход к «инфогра- фичности» (рис. 416) взамен используемой ранее «иконографичности» (рис. 41а). Внедрение принципа «инфографики» означает предоставление пользователю только той информации, которая ему необходима, и исключение информации, которая отвлекает пользователя.



Рисунок 41 «Иконографика» и «инфографика»

Принцип единства заключается в использовании при проектировании пользовательского интерфейса широко известных для широкой аудитории пользователей элементов управления, манипуляций с сенсорным экраном и прочих шаблонов взаимодействия. При этом стиль языка общения программного приложения с пользователями должен соответствовать предпочтениям пользователей.

# Планирование программного приложения универсальной платформы Windows (UWP)

При планировании программного приложения UWP необходимо определить концепцию, которая обеспечивает удачную работу пользователей с программным приложением. Далее составляется список всех возможностей программного приложения. Из всех сценариев выбирается наиболее выигрышный, после чего определяется девиз, которым следует руководствоваться при проектировании программного приложения.

При дальнейшей разработке программного приложения используется так называемая «воронка проектирования» [32]. С широкой стороны воронки поступает множество идей по разработке пользовательского интерфейса. Каждая идея реализуется в виде проектного артефакта низкой точности (набросок внешнего вида интерфейса или фрагмент текста на пользовательском интерфейсе). Продвижение к узкому краю «воронки» заключается в сокращении количества идей в результате их тестирования. Точность артефактов, представляющих идеи, увеличивается. Каждый артефакт содержит только ту информацию, которая необходима для сравнения одной идеи с другой или для ответа на такие вопросы, как «насколько это удобно?» или «насколько это интуитивно понятно?». Некоторые идеи отсеются по мере тестирования. Оставшиеся идеи по мере продвижения по воронке будут последовательно подвергаться более тщательной разработке. В результате получается единый проект, представляющий самую удачную идею. После разработки наиболее удачной идеи происходит переход к следующему этапу - созданию проекта приложения. Большинству программных приложений для размещения необходимой информации и взаимодействия с пользователем требуется использовать несколько диалоговых окон. Если в программном приложении более одного диалогового окна, то необходимо выполнить проектирование информационной архитектуры пользовательского интерфейса. Информационная архитектура программного приложения помогает определить модель навигации приложения. Такая модель показывает организацию содержимого пользовательского интерфейса и организацию доступа пользователя к нему. Разработанная информационная архитектура позволяет визуализировать ключевые экраны (диалоговые окна) пользовательского интерфейса.

На этапе динамики производится построение потоков диалоговых окон (страниц) [32], то есть, процессов перемещения из одного места пользовательского интерфейса в другое в рамках программного приложения. Также формируются набор терминов пользовательского интерфейса (названия элементов управления). Каждый поток приложения соответствует выбранному ранее сценарию (девизу). На этом этапе необходимо создать сценарии для создаваемых диалоговых окон (страниц), которые удовлетворяют требованиям пользователей и обеспечивают удобный переход к другим диалоговым окнам. Для разработки потока диалоговых окон необходимо:

в общих чертах разработать схему переходов (как будет выглядеть последовательность действий пользователя применительно к пользовательскому интерфейсу);

определить визуальные эффекты, цветовую палитру, значки и изображения, улучшающие восприятие пользователем программного приложения;

разработать простейшие прототипы диалоговых окон; уточнить перечень действий, которые может выполнить пользователь, работая с пользовательским интерфейсом.

Далее разрабатываются прототипы диалоговых окон, что является частью «воронки проектирования». При этом прототипы являются по содержанию более сложными, чем наброски диалоговых окон, но менее сложными, чем завершенное программное приложение. Прототип может представлять собой несколько нарисованных от руки диалоговых окон, которые увидит пользователь. Разработчик во время тестирования прототипов может реагировать на действия пользователя, размещая перед ним различные диалоговые окна или размещая на диалоговых окнах (или убирая с них) различные элементы управления для того, чтобы имитировать работающее программное приложение. Прототипом также может быть простое программное приложение, которое имитирует некоторые рабочие процессы при условии, что пользователь придерживается сценария работы и нажимает на нужные кнопки. Таким образом, за счет создания и тестирования прототипов различных диалоговых окон программного приложения реализуется циклический процесс усовершенствования и детализации различных компонентов пользовательского интерфейса. При этом прототипы диалоговых окон должны разрабатываться как можно чаще, а разработка прототипов должна начинаться на самых ранних этапах разработки программного приложения.

Следующим этапом является определение набора функций программного приложения, реализованных в пользовательском интерфейсе:

1. Реализация запланированных ранее анимационных эффектов для того, чтобы пользователи смогли отслеживать изменения содержимого пользовательского интерфейса с помощью визуальных переходов. Необходимо использовать [40]:

анимации добавления и удаления;

анимации переходов содержимого;

анимации перетаскивания;

элементы анимации пользовательского интерфейса с использованием края;

анимации исчезания;

анимации перехода между страницами;

анимации щелчков указателя;

анимации всплывающих элементов пользовательского интерфейса;

анимации перемещения.

* 2. Реализация в пользовательском интерфейсе возможности настройки параметров программного приложения UWP (функций геолокации, размеров шрифтов, насыщенности, цветов, отслеживания, интервалов и текстовых) для повышения удобства работы пользователей. Реализация для пользователей возможностей по получению доступа к данным с помощью учетной записи. Настройка локализации программного интерфейса в соответствии с территорией, на которой будет производиться запуск программного приложения.
* 3. Пользовательский интерфейс должен содержать только необходимые элементы. Таким образом, сводятся к минимуму отвлекающие факторы для пользователя, что помогает пользователям сосредоточиться на содержимом пользовательского интерфейса. Для этого необходимо использовать наиболее важные элементы управления. Необходимо выбирать наилучшее расположение для всех элементов пользовательского интерфейса для того, чтобы выполнялись требования пользователей на устройствах любых форм-факторов [40].
* 4. Реализация в пользовательском интерфейсе разных видов взаимодействия пользователя с программным приложением (взаимодействие с помощью речи, клавиатуры, мыши, пера, сенсорных прикосновений к экрану, мультимодального ввода, настройка скольжения по диагонали, визуального масштабирования и изменения размера, сдвига, поворота, выделения текста и изображений, таргетинга, визуальной обратной связи) [40].
* 5. Обеспечение единообразного для пользователя перемещения данных из веб-служб в программное приложение (независимо от типа устройства, на котором запущено программное приложение).
* 6. Предоставление пользователям справок или советов по устранению неполадок, возникающих во время работы с программным приложением. Обучение пользователей эффективному взаимодействию с программным приложением.
* 7. Обеспечение привлекательности программного приложения за счет настраиваемого экрана-заставки. Упрощение навигации. Активное использование живых плиток и уведомлений.

После выполнения этапа определения функций выполняется этап проектирования взаимодействия программного приложения с пользователем:

* 1. Упорядочение содержимого пользовательского интерфейса. Содержимое большинства приложений можно упорядочить в определенные группы или иерархии.
* 2. Определение частей пользовательского интерфейса, которые необходимо представить пользователю в первую очередь.
* 3. Проанализировать потоки диалоговых окон (страниц), определенные ранее. Для каждого потока необходимо создать последовательность действий, предпринимаемых пользователем.

Следующим этапом является определение возможностей для создания хорошего первого впечатления пользователя о программном приложении при первом запуске программного приложения.

Создание привлекательного пользовательского интерфейса для запуска программного приложения может производиться с помощью настраиваемого экрана-заставки, в том числе, и в виде плиток. Живые плитки могут дать пользователю намного больше информации о запускаемом программном приложении, чем традиционный, чаще всего использующийся в настоящее время, значок (иконка). В результате у пользователей может возникнуть ощущение интерактивного контакта с программным приложением. Более подробная информация об использовании живых плиток приведена в главе 5.

Для того чтобы у пользователя возникло желание постоянно работать с программным приложением, необходимо при первом запуске приложения показать пользователю актуальную и внешне привлекательную информацию о предназначении программного приложения и пользе, которую программное приложение может принести. Поэтому одним из диалоговых окон должен быть экран-заставка, который отражает суть программного приложения и остается на экране только во время инициализации программного приложения при запуске программного приложения. Также в пользовательском интерфейсе должна быть домашняя страница (начальное диалоговое окно), которую пользователь видит первой при каждом запуске приложения и на которой кратко демонстрируются основные функции программного приложения.

**Основы работы с универсальной платформой Windows (UWP)**

Платформа UWP предоставляет разработчика встроенные функции и универсальные шаблоны, значительно упрощающие создание пользовательских интерфейсов для нескольких устройств.

Для разработки пользовательских интерфейсов существуют следующие функции и универсальные шаблоны для разработки:

* 1. Эффективные пиксели и масштабирование. Для оптимизации изображений, шрифтов и других элементов пользовательского интерфейса на экранах устройств используется алгоритм, позволяющий оптимизировать воспринимаемый (а не физический) размер элементов управления, учитывает расстояние, на котором осуществляется просмотр, и растровую плотность (количество пикселей на дюйм). Таким образом, разработка программных приложений ведется не в физических, а в эффективных пикселях.
* 2. Универсальные элементы управления могут использоваться для работы на всех устройствах, рассмотренных выше. В набор универсальных элементов управления включены элементы управления, начиная от стандартных переключателей и текстовых полей, до сложных элементов, которые могут генерировать списки на основе потока данных и шаблона. Описание универсальных элементов управления и шаблонов находится в главе 4.
* 3. Универсальные стили предоставляют следующие возможности:

применение к пользовательскому интерфейсу определенного цвета для фона или выделения информации на диалоговом окне;

использование набора шрифтов Segoe, обеспечивающих четкое отображение текста на всех устройствах;

стандартная анимация для взаимодействия пользователя с программным приложением;

поддержка режимов высокой контрастности;

автоматическая поддержка других языков с автоматическим выбором необходимого шрифта для каждого языка;

возможность настройки нестандартных (уникальных) стилей.

**Основы создания пользовательского интерфейса для платформы универсальных приложений для Windows (UWP)**

Пользовательский интерфейс состоит из текста, форм, цветов и анимации, которые, в конечном счете, составляются из отдельных пикселей экрана используемого устройства. Каждое диалоговое окно включает три типа элементов: элементы навигации, командные элементы и элементы содержимого (рис. 42).

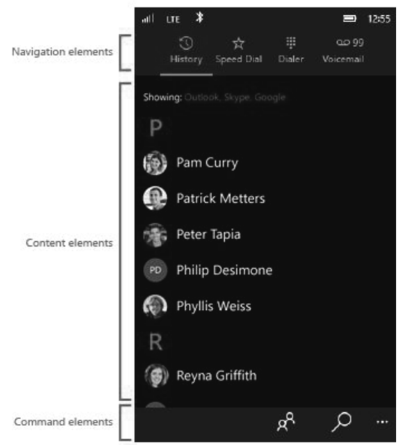


Рисунок 42 Элементы пользовательского интерфейса программного приложения UWP

Элементы навигации - это элементы управления, которые помогают пользователям в процессе работы с пользовательским интерфейсом найти ту информацию, которая им необходима. Более подробно элементы навигации рассмотрены в п.3.5.

Элементы управления (кнопки, панели команд, составные элементы управления) предназначены для выполнения пользователями различных действий (управление, сохранение, предоставление доступа к информации). Командные элементы более подробно описываются в п. 3.6 и главе 4.

Элементы содержимого предназначены для отображения содержимого приложения (рисунки, новостные статьи). Элементы содержимого более подробно описаны в п.3.7.

В пользовательском интерфейсе программного приложения должны быть экран-заставка и домашняя страница, которые определяют пользовательский интерфейс. Обычно пользовательский интерфейс содержит несколько экранов и страниц, при этом элементы навигации, командные элементы и элементы содержимого могут изменяться в зависимости от предъявляемого пользователю диалогового окна (рис. 43).

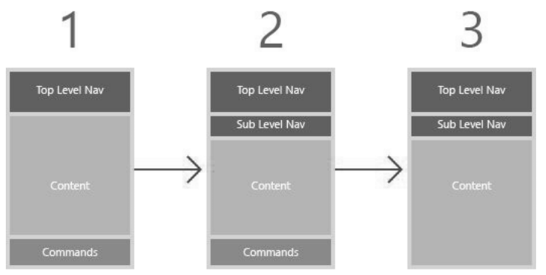


Рисунок 43 Изменение содержимого диалоговых окон

Существуют четыре шаблона пользовательского интерфейса, позволяющие сочетать элементы навигация, командные элементы и элементы содержимого [31].

* 1. Активный холст (рис. 44а) используется для разработки пользовательских интерфейсов программных приложений с одним представлением (браузер, средство просмотра документов, средство просмотра и редактирования фотографий, графический редактор или другие приложения, в которых используется главное представление со свободной прокруткой). Активный холст объединяет в себе элементы «команда» и «содержимое».
* 2. Шаблон основных и подробных данных (рис. 446) используется для формирования пользовательских интерфейсов программных приложений для работы с электронной почтой, списками контактов и адресными книгами (отображает основной список и подробные сведения о выбранном элементе). Шаблон основных и подробных данных объединяет в себе элементы «навигация» и «содержимое».
* 3. Шаблон панель навигации (рис. 44в) состоит из трех основных компонентов: кнопки, панели и области содержимого и позволяет использовать множество элементов навигации верхнего уровня. При этом сохраняется свободное пространство диалогового окна. Кнопка позволяет пользователю открывать и закрывать панель. Панель предназначена для размещения элементов навигации. Шаблон панель навигации объединяет в себе элементы «навигация» и «содержимое».
* 4. Шаблон вкладка (сводка) отображает постоянный список ссылок на диалоговые окна (страницы), который позволяет быстро перемещаться между различными сводками (представлениями или фильтрами) в пределах одного и того же набора данных (рис. 44г). Шаблон объединяет в себе элементы «навигация» и «содержимое».

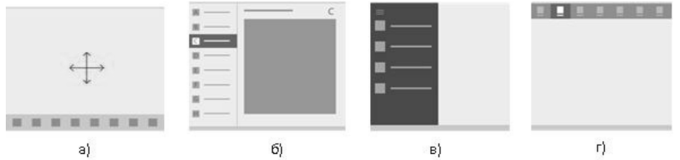


Рисунок 44 Шаблоны пользовательского интерфейса

# Основы проектирования навигации в приложениях универсальной платформы Windows (UWP)

Для построения информационной архитектуры модели навигации необходимо использовать следующие рекомендации:

навигация между диалоговыми окнами должна быть предсказуемой;

элементы навигации позволяют пользователю перейти к нужному содержимому и помогают ему узнать, в какой части программного приложения он находится;

элементы навигации должны быть такими, чтобы удачно подходили к структуре программного приложения;

функции навигации должны обеспечивать ожидания пользователя (единообразие и простота).

Существует два стандартных способа организации навигации между страницами (диалоговыми окнами) [28]: иерархическая (рис. 45а) и одноранговая (рис. 456).

Иерархическая структура перехода между диалоговыми окнами (страницами) подобна дереву. У каждой дочерней страницы имеется только одна родительская, но одна родительская страница может иметь одну или несколько дочерних. Чтобы попасть на дочернюю страницу, сначала необходимо обратиться к родительской странице. При одноранговой структуре перехода диалоговые окна (страницы) располагаются рядом друг с другом. Пользователь может переходить от одного диалогового окна (страницы) к другому диалоговому окну в любом порядке. Одноранговые элементы навигации обеспечивают переходы между страницами, расположенными на одном и том же уровне одного и того же поддерева. Обычно в программных приложениях используются обе модели перехода между страницами (рис. 45в).

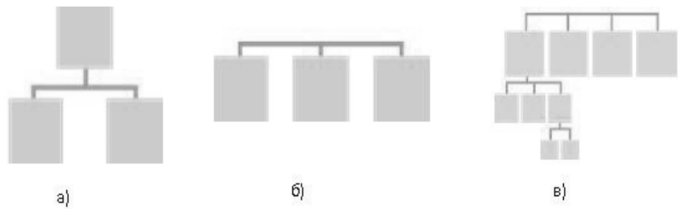


Рисунок 45 Способы навигации между диалоговыми окнами (страницами)

Иерархическая модель организации перехода между диалоговыми окнами применяется в следующих случаях:

предполагается, что пользователь будет просматривать страницы (диалоговые окна) в определенном порядке;

имеются четкие иерархические отношения между одной страницей (диалоговым окном) группы и остальными;

если в группе 7 и более диалоговых окон (страниц).

Одноранговая структура организации перехода между диалоговыми окнами (страницами) применяется в следующих случаях: страницы могут просматриваться в любом порядке; страницы явно отличаются друг от друга и не имеют очевидных отношений типа «предок — потомок»; в группе менее 7 страниц.

Для навигации между одноранговыми элементами рекомендуется использовать вкладки или панель навигации [28].

Вкладки (рис. 46а) отображают постоянный список ссылок на страницы одного уровня и используются в следующих случаях:

количество страниц (диалоговых окон) составляет от двух до пяти;

предполагается, что пользователи будут часто переключаться между страницами (диалоговыми окнами).

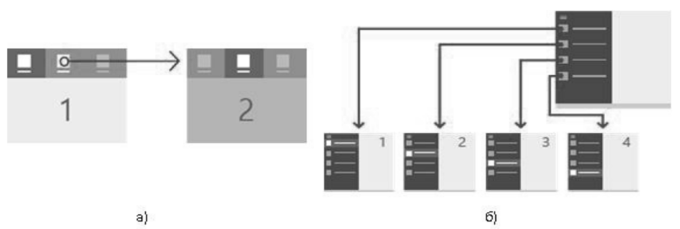
Панели навигации (рис. 466) отображают список ссылок на страницы верхнего уровня и используются в следующих случаях:

предполагается, что пользователи не будут часто переключаться между страницами;

необходимо сохранить пространство на пользовательском интерфейсе за счет более медленной навигации;

страницы (диалоговые окна) располагаются на верхнем уровне.

Если в структуре навигации имеется несколько уровней, то рекомендуется, чтобы одноранговые элементы ссылались друг на друга только в пределах своего текущего поддерева.



**Рисунок 46 Вкладка и панель навигации**

Для перемещения между родительскими и дочерними диалоговыми окнами (страницами) с помощью иерархической модели используются главные разделы (рис. 47а) и шаблоны основных и подробных данных (рис. 476) [28].

Элемент главный раздел обеспечивает предварительный просмотр дочерних страниц (диалоговых окон). В отличие от вкладок или панели навигации, он обеспечивает переход к этим дочерним страницам (диалоговых окон) при помощи ссылок и заголовков секций, встроенных в саму страницу. Данный элемент управления используется в случае, если предполагается, что пользователи захотят просматривать часть содержимого дочерних страниц (диалоговых окон), не переходя к каждой из них.

Элемент шаблоны основных и подробных данных отображает список (основное представление) элементов. Выбор элемента отображает соответствующую ему диалоговое окно (страницу). Также при проектировании пользовательского интерфейса необходимо использовать элементы, позволяющие пользователю вернуться на предыдущую страницу, элементы истории навигации (кнопки «Назад»), а также элементы навигации, встроенные в содержимое диалоговых окон (гиперссылки и кнопки). Такие элементы отображаются в содержимом страницы и могут отличаться в различных диалоговых окнах (страницах).



Рисунок 47 Главные разделы и шаблоны основных и подробных данных

Если пользователь переходит к работе с другим программным приложением, а затем возвращается к выполнявшемуся ранее программному приложению, то рекомендуется открывать последнее диалоговое окно (страницу) в истории навигации выполнявшегося ранее программного приложения.

**Основы проектирования команд на платформе универсальных приложений для Windows (UWP)**

Элементы управления в пользовательских интерфейсах UWP являются интерактивными элементами, которые позволяют пользователю выполнять требуемые действия. В процессе проектирования пользовательского интерфейса необходимо заранее определить, выполнение каких команд понадобится пользователям для выполнения их целей. Для того чтобы пользовательский интерфейс был интуитивно понятным, необходимо включать в состав пользовательского интерфейса элементы управления, которые будут наилучшим образом отражать действия пользователя. В составе платформы универсальных приложений для Windows (UWP) имеется набор командных элементов управления, которые можно использовать при разработке пользовательского интерфейса программного приложения. Командные элементы управления размещаются на поверхностях программного приложения или в специальных командных элементах, которые могут выступать в качестве контейнеров. Список поверхностей для командных элементов управления приведен в табл. 3 [30].

Список поверхностей для командных элементов управления

Таблица 3

|  |  |  |
| --- | --- | --- |
| **Поверхность** | | **Описание** |
| Поверхность приложения (область содержимого) | | Наиболее важные для пользователя команды для выполнения основных сценариев должны быть помещены на поверхность (область содержимого приложения). Элементы управления должны быть размещены возле объектов, на которые они влияют. При этом количество командных элементов управления на холсте не должно перегружать пользователя. Если элемент управления используется редко, то его можно разместить на другой поверхности.  Предоставляют пользователям удобный доступ к командам, доступным пользователю в данный момент времени. Панели команд могут размещаться в верхней части диалогового окна, в нижней части диалогового окна, или в верхней и нижней части диалогового окна |
| https://bstudy.net/htm/img/4/10572/48.png | https://bstudy.net/htm/img/4/10572/49.png |
| https://bstudy.net/htm/img/4/10572/50.png | |

|  |  |
| --- | --- |
| https://bstudy.net/htm/img/4/10572/51.png | Меню предлагают пользователю команды, доступные в текущий момент времени. При этом задействуется меньше пространства диалогового окна. Меню могут содержать интерактивные элементы управления. Контекстные меню могут содержать ярлыки часто используемых команд. Также предоставляется доступ к второстепенным командам, актуальным только при определенных условиях работы пользователя. С помощью контекстных меню могут выполняться следующие команды: «Копировать», «Вырезать», «Вставить», «Проверка орфографии» и т. д.;  команды для выполнения операций с объектами диалогового окна;  отображения команд буфера обмена; пользовательские команды |

|  |  |
| --- | --- |
| https://bstudy.net/htm/img/4/10572/52.png | Диалоговые окна являются в основном модальными и предоставляют пользователю контекстную информацию о приложении. При этом блокируются взаимодействие пользователя с программным приложением, а от пользователя требуется выполнить некоторое действие. До тех пор, пока действие не будет выполнено, диалоговое окно не будет закрыто. |
| https://bstudy.net/htm/img/4/10572/53.png | Диалоговое окно, отображающее элемент пользовательского интерфейса в зависимости от действий пользователя. Всплывающий элемент, рекомендуется использовать в следующих случаях:  для показа меню;  для отображения дополнительных сведений об элементе управления;  для запроса у пользователя подтверждения действия, не блокируя взаимодействия с приложением.  Всплывающий элемент закрывается с помощью, нажатия на участок диалогового окна за его пределами. |

Рекомендации по использованию командных элементов управления [30]:

* 1. Пользователи должны иметь возможность напрямую управлять содержимым пользовательского интерфейса. Поэтому количество командных элементов управления на активном холсте не должно быть таким, чтобы пользовательский интерфейс был перегружен. Если холст перегружен командными элементами управления, то элементы управления необходимо перенести на другие поверхности
* 2. Командные элементы управления лучше всего располагать на панели команд, которая помогает сгруппировать команды и облегчает к ним доступ.
* 3. Если несколько диалоговых окон (страниц) служат для выполнения одной задачи пользователя, то командные элементы управления для обслуживания данной задачи могут быть размещены на холсте (но таких элементов управления должно быть немного).
* 4. В контекстное меню можно поместить команды, обеспечивающие работу буфера обмена (вырезание, копирование и вставка фрагмента текста), или команды, относящиеся к содержимому, которое нельзя выбрать (например, добавление флажка на карту).
* 5. Для того чтобы обеспечить уменьшение количества ошибок пользователя, предусмотреть использование диалоговых окон для подтверждения выполнения действий (имеющих серьезные последствия, которые нельзя отменить) или возможностей для отмены последних действий (для действий пользователя, которые можно отменить). Как правило, приложения UWP используют систему интеллектуального взаимодействия при вводе. Поэтому пользователь, работая с такими программными приложениями, как правило, не задумываются о типе ввода информации для программного приложения (обработка события «нажатие кнопки» производится независимо от того, осуществляется это нажатие с помощью щелчка мыши или прикосновения пальца).

Несмотря на такую универсальную обработку событий, может потребоваться настройка программного приложения на использование следующих типов ввода информации:

* 1. Сенсорный ввод информации позволяет использовать движения одного или нескольких пальцев (жесты) для имитации прямых манипуляций с элементами пользовательского интерфейса.
* 2. Ввод с использованием пера (стилуса) может служить для по- пиксельного указания на пользовательском интерфейсе. Полученные данные используются для распознавания рукописного ввода, сбора информации о силе нажатия сила нажатия, форме, цвете и прозрачности введенной графической информации.

Существуют два типа перьев: активные и пассивные. Активные перья предоставляют подробные входные данные и использующиеся в основном для точного рукописного ввода и указания. Пассивные перья не предоставляют подробных входных данных и только имитируют сенсорный ввод.

* 3. Ввод с помощью мыши лучше всего подходит для повышения производительности пользователей во время работы с программным приложением, которое требует пиксельной точности при работе с элементами пользовательского интерфейса.
* 4. Ввод с помощью клавиатуры - пока еще наиболее удобный для многих пользователей способ ввода информации. Пользователи могут взаимодействовать с универсальными приложениями с помощью аппаратной клавиатуры и двух программных клавиатур: экранной и сенсорной. Экранная клавиатура - это визуальная клавиатура, которая имитирует почти все функции аппаратной клавиатуры и которую можно использовать вместо аппаратной клавиатуры для ввода текста и данных с помощью сенсорного экрана, мыши, пера или стилуса и других указывающих. Экранная клавиатура предназначена для систем, не имеющих аппаратной клавиатуры, или для людей с ограниченными физическими возможностями.

Сенсорная клавиатура - это визуальная клавиатура, позволяющая вводить только текст и появляющаяся только тогда, когда фокус ввода находится в текстовом поле или в другом текстовом элементе управления с поддержкой редактирования.

Экранная клавиатура имеет приоритет над сенсорной клавиатурой, которая не предъявляется пользователю при наличии экранной клавиатуры.

* 5. Ввод информации с помощью речи (голосовых команд). В Windows 10 для обработки голосовых команд и запуска программных приложений используется программа Cortana. Кроме программы Cortana (только в Windows 10) во всех версиях ОС Windows имеются два компонента для распознавания речи и преобразования текста в речь (TTS), которые могут быть встроены в программные приложения с помощью встроенных API.
* 6. Ввод информации с помощью жестов. Жестом называется любая форма движения пользователя, которая распознается в качестве входных данных для управления программным приложением или для взаимодействия с ним.
* 7. Ввод информации с использованием нескольких типов ввода (мультимодальный ввод). При этом комбинированные взаимодействия пользователя и программного приложения должны быть интуитивно понятными и естественными для пользователя.

**Основы проектирования содержимого на платформе универсальных приложений для Windows (UWP)**

Основная функция любого программного приложения заключается в предоставлении пользователю доступа к содержимому пользовательского интерфейса. Существует три сценария использования содержимого [29]:

потребление, заключающееся в односторонних действиях потребления содержимого (чтение, прослушивание музыки, просмотр видеороликов, фотографий и изображений);

создание, заключающееся в односторонних действиях по созданию нового содержимого (съемка фото или видео, создание нового изображения в приложении для рисования или открытие нового документа);

интерактивность, заключающееся в двустороннем взаимодействии (потребление, создание и исправление содержимого).

В приложениях, ориентированных на потребление информации, элементы содержимого получают высочайший приоритет. Вслед за этими элементами идут элементы навигации, которые должны позволять пользователю находить нужное содержимое. При разработке пользовательских интерфейсов для таких программных приложений необходимо придерживаться следующих рекомендаций:

* 1. Для того чтобы пользователи находили в пользовательском интерфейсе нужную информацию, необходимо предусмотреть возможность создания специальных страниц навигации и страниц просмотра содержимого. Это дает возможность просмотра информации на специальной странице.
* 2. Необходимо предусмотреть возможность создания полноэкранного варианта просмотра информации. При этом скрываются все остальные элементы пользовательского интерфейса.

В приложении, ориентированном на создание, самыми важными элементами пользовательского интерфейса являются содержимое пользовательского интерфейса и элементы команд, рассмотренные ранее. Элементы команд позволяют пользователю создавать новое содержимое. При разработке пользовательских интерфейсов для таких программных приложений необходимо придерживаться следующих рекомендаций:

* 1. Необходимо уменьшить до минимума использование элементов навигации.
* 2. Необходимо реализовать в пользовательском интерфейсе функцию просмотра информации (журнал) и отмены команд.

В программном приложении с интерактивным содержимым пользователи создают, просматривают и редактируют содержимое (бизнес- приложения, приложения для управления запасами, различные приложения, позволяющие пользователю создавать или изменять различные документы). Такие программные приложения нуждаются в оптимальном сочетании элементов пользовательского интерфейса, отвечающих за навигацию, фильтрование, сортировку и поиск. При разработке пользовательских интерфейсов такого типа необходимо рассмотреть возможность создания отдельных диалоговых окон для просмотра, создания и редактирования содержимого или предоставления переключателей режима. В таких пользовательских интерфейсах наиболее часто отражаются звуковая информация, видеоинформация, средства просмотра изображений, списки, тексты и текстовый ввод. Элементы управления, предназначенные для отражения данных категорий информации, рассмотрены в главе 4.

**Учет размера экрана устройства для платформы универсальных приложений для Windows (UWP)**

Программные приложения, пользовательские интерфейсы которых предусматривают работу с эффективными пикселями, позволяют сосредоточиться на реально воспринимаемых размерах элементов пользовательского интерфейса [4]. В этом случае пользователю нет необходимости адаптировать программное приложение под размеры экрана устройства (учитывать плотность пикселей или расстояние от глаз пользователя до экрана). При разработке элемента пользовательского интерфейса размером 1 х 1 дюйм, он будет иметь примерно этот же размер на всех устройствах. На очень больших экранах с высокой плотностью пикселей его размер может составлять 200 х 200 физических пикселей. На устройствах с экраном меньшего размера, например, на телефоне, его размер будет 100 х 100 физических пикселей (рис. 48).

Возможности платформы универсальных приложений для Windows предусматривают масштабирование пользовательского интерфейса в величинах, кратных 4. Поэтому чтобы обеспечить четкость отображения, необходимо при разработке пользовательского интерфейса переключиться на сетку с размером ячейки 4><4 пикселя. Необходимо чтобы величина полей, размеры и положения элементов пользовательского интерфейса и положение текста были кратны 4. При этом текст может быть любого размера.

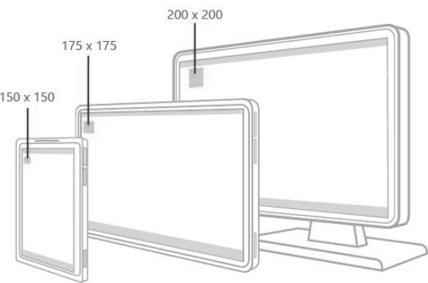


Рисунок 48 Эффективные пиксели

Если программное приложение ориентировано на устройство с маленьким экраном, то при его запуске на компьютере, экран которого намного больше, скорее всего, останется неиспользованное пространство. Программное приложение может быть настроено таким образом, чтобы оно отображало больше содержимого, если размер экрана превышает определенное значение. При выводе большого объема содержимого на экран сокращается количество действий, которые пользователь может совершить для навигации.

Существует шесть методов для настройки пользовательского интерфейса программного приложения для конкретного размера экрана устройства [4].

* 1. Изменение положения. Имеется возможность изменять положение элементов пользовательского интерфейса программного приложения для того, чтобы максимально эффективно использовать экраны каждого из устройств. На рис. 49 показано, что при запуске программного приложения на устройстве, которое позволяет отображать на экране два полных кадра как при вертикальной, так и при альбомной ориентации, кадр В может занимать выделенное для него пространство.
* 2. Изменение размера. Размер экрана оптимизируется одновременно с регулировкой полей и размеров элементов пользовательского интерфейса. Благодаря этому можно расширить пространство для восприятия информации на больших экранах, просто увеличив размер кадра содержимого (рис. 50а).



Рисунок 49 Изменение положения

3. Адаптация. Обеспечивается оптимальное отображение содержимого путем изменения порядка элементов пользовательского интерфейса в зависимости от ориентации. При переходе на устройство с большим размером экрана должны использоваться более крупные контейнеры для отображения информации, могут добавляться новые колонки или видоизменяться элементы списков (рис. 546).



Рисунок 50 Изменение размера (а) и адаптация (б)

* 4. Отображение. Количество элементов управления (их появление и скрытие) и отображение метаданных на пользовательском интерфейсе зависит от свободного экранного пространства и событий, происходящих при выполнении программного приложения (рис. 51 а).
* 5. Замена (рис. 516). Метод позволяет изменять внешний вид пользовательского интерфейса в зависимости от типа устройства или ориентации экрана (на устройствах с небольшими экранами целесообразно применять панель навигации, а на устройствах с экранами большего размера целесообразно применять вкладки).

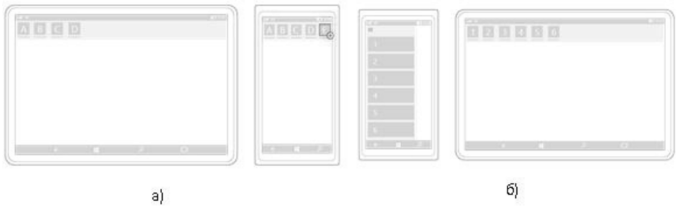


Рисунок 51 Отображение (а) и замена (б)

6. Изменение архитектуры (рис. 52) предусматривает видоизменение визуального представления архитектуры программного приложения на экране (или применение нескольких вариантов визуального представления архитектуры) в зависимости от размера экрана или от типа устройства (например, соединение страниц при переходе от одного устройства к другому).



Рисунок 52 Изменение архитектуры

Для оптимизации размеров пользовательских интерфейсов в соответствии с размером экрана устройств (рис. 53) используются несколько ключевых значений ширины экранов (точки останова): 320, 720 и 1024 эффективных пикселей.

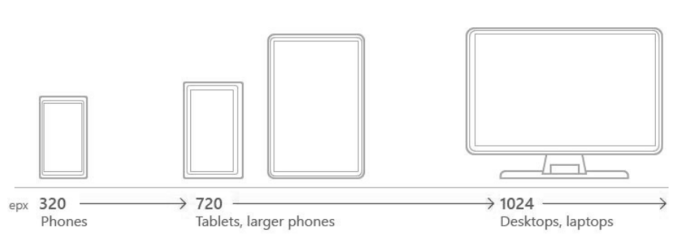


Рисунок 53 Точки останова 320, 720 и 1024 эффективных пикселей

Общие рекомендации по адаптации программных приложений к различным размерам экранов устройств:

1. Устройства с экраном маленького размера (телефон, ширина - 320 эффективных пикселей, стандартный размер - 4-6 дюймов).

Элементы навигации и командные элементы необходимо размещать в нижней части пользовательского интерфейса (при таком расположении элементов управления пользователям будет удобно нажимать их пальцем). Левое и правое поля диалогового окна необходимо сделать равными 12 пикселям (для того, чтобы создать визуальное разделение между левой и правой стороной диалогового окна программного приложения). Также для отображения информации в пределах диалогового окна, предъявляемого пользователю в данных момент времени, могут использоваться по 1 столбцу (или региону диалогового окна). В целях экономии экранного пространства необходимо использовать значок (иконку) для выполнения поиска (вместо текстового поля для поиска). Также в целях экономии экранного пространства область навигации пользовательского интерфейса необходимо включать в режиме наложения для того и использовать расположение элементов управления стопкой.

2. Устройства с экраном среднего размера (фаблет или планшет, ширина - 1024 эффективных пикселя, стандартный размер -13 дюймов и шире).

Вкладки должны быть выровнены по левому краю. Значения левого и правого полей диалогового окна должны быть равными 24 пикселям. Левый и правый края окна приложения должны быть визуально разделены. Для отображения информации в пределах диалогового окна, предъявляемого пользователю в данных момент времени, используется не более 2 столбцов (или регионов диалогового окна). Поле поиска должно быть отражено в диалоговом окне. Панель навигации должна постоянно отображаться в пользовательском интерфейсе.

3. Устройства с экраном большого размера (ПК, ноутбук, Surface Hub, ширина - 720 эффективных пикселей, стандартный размер - 8 дюймов).

Элементы навигации и командные элементы должны быть размещены в верхней части диалогового окна. Элементы вкладки должны выравниваться по левому краю. Значения левого и правого полей диалогового окна должны быть равными 24 пикселям. Левый и правый края диалогового окна должны быть визуально разделены. Для отображения информации в пределах диалогового окна, предъявляемого пользователю в данных момент времени, используется не более 3 столбцов (или регионов диалогового окна). Поле поиска должно быть отражено в диалоговом окне. Панель навигации должна постоянно отображаться в пользовательском интерфейсе.

**Контрольные вопросы по главе 3**

* 1. Перечислите новые принципы проектирования пользовательских интерфейсов для Windows. В чем разница между иконографикой и инфографикой?
* 2. Перечислите типы устройств, для которых могут проектироваться пользовательские интерфейсы для UWP.
* 3. Перечислите функции и универсальные шаблоны для разработки приложений UWP. Какие бывают шаблоны пользовательского интерфейса в приложениях UWP?
* 4. Элементы пользовательского интерфейса программного приложения UWP.
* 5. Перечислите сценарии по предоставлению доступа к содержимому пользовательского интерфейса для пользователя.
* 6. Для чего предназначены элементы навигации, командные элементы и элементы содержимого в приложениях UWP? Какие бывают способы организации иерархической архитектуры навигации?
* 7. Перечислите список поверхностей для командных элементов управления.
* 8. Перечислите рекомендации по использованию командных элементов управления.
* 9. Перечислите сценарии использования содержимого приложения UWP.
* 10. Перечислите рекомендации для разработки пользовательских интерфейсов для программных приложений, ориентированных на потребление информации.
* 11. Перечислите рекомендации для разработки пользовательских интерфейсов для программных приложений, ориентированных на создание.
* 12. Перечислите рекомендации для разработки пользовательских интерфейсов для программных приложений, ориентированных на работу с интерактивным содержимым.
* 13. Перечислите методы для настройки пользовательского интерфейса программного приложения для конкретного размера экрана устройства.
* 14. Перечислите рекомендации по адаптации программных приложений к различным размерам экранов устройств.

**Элементы управления, используемые для построения пользовательских интерфейсов и взаимодействия с пользователями**

В этой главе описываются рекомендации по проектированию как «инфографичных», так и «иконографичных» пользовательских интерфейсов с использованием типовых элементов управления. С одной стороны, остаются задачи проектирования с применением более «привычных» элементов управления, используемых в различных средах разработки программных приложений (на таких элементах управления базируется «иконографичный» пользовательский интерфейс). С другой стороны, появились новые подходы к проектированию пользовательских интерфейсов с учетом «инфографики», сенсорного и речевого взаимодействия.

**Элементы управления для разработки «инфографичных» пользовательских интерфейсов для платформы универсальных приложений для Windows (UWP)**

Элемент управления «Кнопка» [66] предоставляет пользователю возможность немедленно начать выполнение какого-либо действия. Существует несколько состояний кнопок:

«не нажата, но без передачи фокуса» (рис. 54а);

«курсор наведен, не нажата, но без передачи фокуса» (рис. 546); «нажата, но без передачи фокуса» (рис. 54в);

«недоступна» (рис. 54г);

«нажата, с передачей фокуса» (рис. 54д).

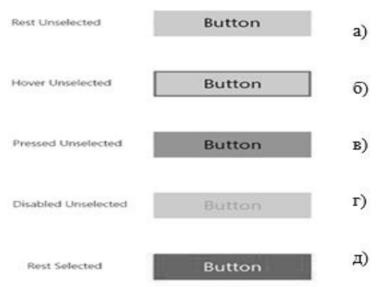


Рисунок 58 Состояние элемента управления «Кнопка»

Элемент управления «Кнопка» не следует использовать в случае, когда необходимо перейти от одного диалогового окна к другому. Для осуществления таких переходов желательно использовать ссылки. При проектировании использования кнопок необходимо придерживаться следующих рекомендаций:

* 1. Назначение кнопки и ее состояние должно быть понятно пользователю. Не допускается размещение большого количества информации на одной кнопке.
* 2. Текст, описывающий действия, которое выполняет кнопка, должен быть кратким, конкретным и не требующим разъяснений пользователю (рекомендуется текст в виде глагола из одного слова). Текст можно (но нежелательно) заменять иконками или сочетанием иконки и текста. Не допускается изменять названия кнопок «Далее» на название «Продолжить».
* 3. Если текст надписи на кнопке динамический, то необходимо, чтобы кнопка могла изменять свои размеры в соответствии с размерами текста, не мешая при этом другим элементам пользовательского интерфейса.
* 4. Для действий, которые требуется выполнить над большим количеством диалоговых окон программного приложения лучше не повторять кнопку с одним и тем же названием в каждом диалоговом окне, а использовать команду в составе элемента управления «панель команд» (см. далее в этой главе).
* 5. Рекомендуется делать одновременно доступными пользователю только одну или две кнопки (например, «Принять» и «Отклонить»). Если требуется предоставить пользователю больший выбор действий, то рекомендуется ввести в пользовательский интерфейс флажки или переключатели, с помощью которых пользователь сможет выбрать нужные действия.
* 6. Кнопка по умолчанию в пользовательском интерфейсе должна использоваться для работы с наиболее часто используемой или рекомендуемой командой. Не допускается менять местоположение кнопок, предусмотренных в пользовательском интерфейсе по умолчанию (например, таких как «Отправить», «Сбросить»).
* 7. При взаимодействии пользователя с кнопкой состояние и внешний вид кнопки должны меняться, отражая отклик на действия пользователя. Выполнение действия начинается в момент, когда пользователь касается кнопки или нажимает ее.
* 8. Кнопки не должны использоваться для задания состояния программного приложения.
* 9. Для обеспечения обратной навигации используется кнопка «Назад» [42]. Расположение кнопки «Назад» должно быть оптимизировано для каждого устройства и способа ввода информации. Для телефона кнопка «Назад» должна быть расположена в нижней части устройства (рис. 55а). Для планшета кнопка «Назад» на панели навигации в нижней части устройства (рис.556). Для ПК и ноутбуков кнопка «Назад» расположена в заголовке диалогового окна программного приложения (рис.55в). Для устройства Surface Hub кнопка «Назад» расположена в нижней части устройства (рис. 55г).

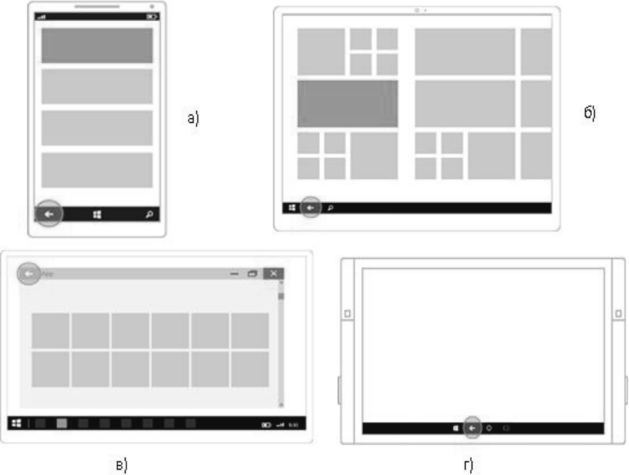


Рисунок 55 Рекомендации по расположению кнопки «Назад» для различных устройств

Элементы управления датой и временем [41] позволяют пользователю просматривать и устанавливать дату и время. В процессе проектирования пользовательского интерфейса можно выбрать любой из четырех элементов управления датой и временем (табл. 4):

Виды элемента управления датой и временем

Таблица 4

|  |  |
| --- | --- |
| Элемент управления | Внешний вид |
| Представление календаря | https://bstudy.net/htm/img/4/10572/62.png |
| Выбор даты в календаре | https://bstudy.net/htm/img/4/10572/63.png |

|  |  |
| --- | --- |
| Управляющий элемент выбор даты | https://bstudy.net/htm/img/4/10572/64.png |
| Сборщик времени | https://bstudy.net/htm/img/4/10572/65.png |

Представление календаря предоставляет пользователю доступ к единственной дате или диапазону дат в формате месяца, года или десятилетия. Этот элемент управления доступен только для чтения и не имеет поверхности выбора для ввода даты.

Выбор даты в календаре совпадает с представлением календаря, при этом в наличии поле ввода даты над календарем. Пользователь может выбрать одну дату или диапазон дат в формате месяца, года или десятилетия. Точка входа отображает замещающий текст, если дата еще не установлена.

Точка входа для выбора даты отображает выбранную дату, а также перед пользователем может разворачиваться поверхность для выбора даты. При выборе точки входа элемент управления разворачивается вертикально от середины (рис. 56). При этом поверхность для выбора даты накладывается на другие элементы управления пользовательского интерфейса, но не вытесняет их.

Сборщик времени используется для выбора пользователем времени для проведения мероприятий. Точка входа отображает выбранное время, а выбор точки входа разворачивает перед пользователем поверхность выбора. При касании точки входа элемент управления также разворачивается вертикально от середины. При этом разворачиваемая поверхность сборщика времени накладывается на другие элементы пользовательского интерфейса, не вытесняя их.

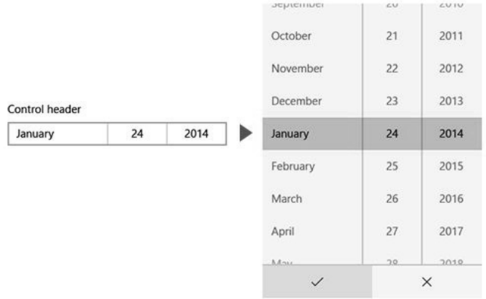


Рисунок 56 Работа с элементом типа «выбор даты»

Элемент управления «поле автозаполнения» [43] - это текстовое поле, в котором отображается список основных вариантов поиска. Такой список формируется автоматически на основе объединения условий поиска и слов, ранее введенных пользователем (рис.57а).



Рисунок 57 Элемент управления «Поле автозаполнения»

Список предъявляется пользователю сразу же после начала ввода текста в элементе управления (список отображается над элементом управления или под ним, при этом появляется кнопка в виде крестика, обозначающая выполнение команды «Очистить все», рис.57а).

В случае использования данного элемента управления в пользовательском интерфейсе необходимо учесть, что если для введенного в элементе управления текста поиск не дает результатов, то должна быть выведена строка сообщения «Результатов нет» (для того чтобы пользователь знал, что поисковый запрос был выполнен, рис. 576).

Элемент управления «флажок» («check box») [44] используются пользователями для выбора (или отмены выбора) одного или нескольких выполняемых действий в предлагаемом пользователям списке. Элемент управления предусматривает три состояния выбора: «не выбран», «выбран» и «не определен», «не доступен» (рис. 58а).



Рисунок 58 Элемент управления «Флажок»

При этом состояние «не определен» возникает, когда в списке возможных вариантов для каждого флажка есть одновременно состояния «не выбран» и «выбран», и при этом пользователь пока не сделал выбора.

При использовании элемента управления «флажок» рекомендуется придерживаться следующих рекомендаций:

* 1. Желательно использовать один элемент управления «флажок» для выбора из двух вариантов «да» или «нет» (например, при подтверждении условий соглашения на обслуживание, рис 586).
* 2. Необходимо использовать несколько элементов управления «флажок» (рис. 62в) в случае множественного выбора (пользователь может выбрать один или несколько элементов, не исключающих друг друга).
* 3. Необходимо, чтобы цель использования элемента управления и их текущие состояния были понятны пользователю. Необходимо добиваться того, чтобы пользователь понимал, что произойдет, если он выберет (снимет) «флажок».
* 4. Длина текстового содержимого, связанного с элементом управления, не должна превышать двух строк. При этом текст должен быть сформулирован как инструкция, в которой выбор флажка соответствует значению «да», а снятие «флажка» обозначает «нет».
* 5. Если пользователю необходимо представить большое количество вариантов выбора, и при этом объем содержимого не помещается в окне просмотра, то окно просмотра, в котором размещаются элементы управления «флажок», можно дополнить элементом управления «средство прокрутки» (класс ScrollViewer) [45] (рис. 59).



Рисунок 59 Элемент управления «средство прокрутки»

Жесты пользователя (касания сенсорного экрана) можно использовать для сдвигания и масштабирования (полосы прокрутки изменяют цвет во время управления), а для прокрутки можно использовать указатель мыши. Быстрое движение пальца пользователя по поверхности экрана заставит содержимое прокручиваться по инерции. Если содержимое пользовательского интерфейса выходит за обе границы окна просмотра (вертикальную и горизонтальную), необходимо применять сдвиг сразу по двум направлениям. Если пользователю предстоит просматривать большой объем текста, то необходимо, чтобы прокрутка осуществлялась только по вертикали. Также «средство прокрутки» может быть использовано в элементах управления списком, раскрывающихся списках, полях текстового ввода, представлениях сетки, представлениях списка и главных разделом.

* 6. Если в названии «флажка» («чекбокса») применяется динамическое текстовое содержимое, необходимо учесть возможное изменение размеров элемента управления и его влияние на другие элементы управления, находящиеся в пользовательском интерфейсе.
* 7. Если «флажки» разделены на группы, то не рекомендуется размещать две группы флажков рядом (вполне возможно, что пользователи в этом случае не смогут определить, какие «флажки к каким группам принадлежат). Поэтому рекомендуется использовать текстовые обозначения для каждой из групп «флажков».
* 8. Элементы управления «флажок» не рекомендуется использовать для отображения диалоговых окон.
* 9. Для одиночного «флажка» нежелательно по умолчанию использование состояния «не определен».
* 10. И «флажок», и «переключатель» («радиокнопка») могут использоваться для отображения двоичного выбора. Группа «переключателей» предоставляет пользователю единственно возможный выбор. Если же используется группа «переключателей», то каждый «флажок» в группе обеспечивает отдельный выбор, независимый от других «флажков». Если в группе выбор применим к нескольким «флажкам», то существует несколько вариантов, выбора «флажков»: ко всем, к некоторым или ни к одному из них. Если выбор применен только к некоторым дочерним «флажкам» (выбраны не все дочерние элементы управления), то, чтобы обозначить смешанный выбор, необходимо использовать состояние «не определен» для родительского «флажка» (рис 60).

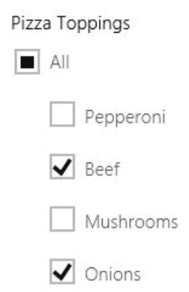


Рисунок 60 Состояние родительского «флажка»

(«не определен») в случае смешанного выбора дочерних «флажков»

«Переключатели» типа RadioButton («радиокнопка») [46] позволяют пользователю выбрать один вариант выполнения задачи из нескольких предлагаемых вариантов. Каждый вариант выбора представлен одним переключателем, пользователь может выбрать только один переключатель из группы. «Переключатели» могут находиться в следующих состояниях (рис. 61):

не нажатое (нормальное и доступное) состояние (рис. 61а); включенное (нажатое) состояние с переданным фокусом (рис.

616);

отключенное (не нажатое и доступное) состояние после покидания элемента управления курсором мыши (рис. 61 в);

отключенное (не нажатое и доступное) состояние после нажатия на другой элемент управления «переключатель» (рис.61г);

не доступное состояние (рис. 61д).

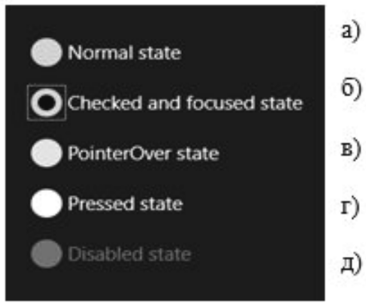


Рисунок 61 Состояния «переключателя»

Все «переключатели» (в пределах одной группы) равнозначны. Если вариантов выбора слишком много (более 8) и при этом они занимают слишком много места в пользовательском интерфейсе, а также отвлекают пользователя от запланированных действий, то рекомендуется вместо группы «переключателей» использовать другие элементы управления (например, списки, в том числе, раскрывающиеся). Если пользователю для выбора предлагаются числа с фиксированной величиной шага (5, 10, 15, ...), то рекомендуется вместо нескольких «переключателей» использовать «ползунки» (элементы управления Slider).

Если доступ к группе «переключателей» производится с помощью клавиатуры, то передача фокуса очередному «переключателю» выполняется с помощью клавиши «ТАВ», при этом пользователи могут циклически и по очереди обращаться к «переключателям», входящих в группу.

Назначение и состояние «переключателей», входящих в состав пользовательского интерфейса, должны быть понятны пользователю.

В процессе работы с «переключателями» пользователь должен получать от программного приложения видимую ответную реакцию (обратную связь).

Текст, поясняющий работу «переключателя» рекомендуется ограничивать одной строкой. Рекомендуется использовать предусмотренные по умолчанию параметры шрифта для отображения текста (если во время работы программного приложения параметры текста не изменяются). При этом поясняющий текст рекомендуется размещать справа от «переключателя». Если текст, поясняющий работу «переключателя», будет изменяться в ходе работы программного приложения, то рекомендуется менять размер элемента управления. При этом необходимо учесть возможное влияние изменения размера элемента управления на другие элементы управления пользовательского интерфейса.

Не рекомендуется размещать рядом группы «переключателей», соответствующих выполнению различных функций программного приложения (пользователю будет сложно определить принадлежность «переключателя»). Для того чтобы явно разделить группы «переключателей», рекомендуется использовать текстовые метки для групп.

Список (элемент управления «выбор») предоставляет пользователю возможность выбрать из предлагаемого списка один или несколько элементов. Элемент управления списком по умолчанию представляет список по вертикали. Список может быть представлен и в горизонтальном виде (например, для размещения фотографий). Существуют три режима выбора элементов списка:

выбор одного элемента (рис. 65, левая часть); выбор нескольких элементов с помощью модификатора (рис. 63);

выбор нескольких элементов без использования модификаторов (рис. 65, правая часть).

Касание пользователя сенсорного экрана в месте расположения элемента списка пометит элемент списка как выбранный, а в режиме с множественным выбором касание выбранного элемента списка отменит его выбор. В режиме с единственным выбором касание другого элемента делает его выбранным. При этом выбор элемента списка, сделанный ранее, отменяется. Скольжение пальца по вертикали позволяет прокручивать список вверх или вниз по инерции. Список имеет полосу прокрутки, положение бегунка которой показывает приблизительное положение пользователя в списке элементов, а размер «бегунка» в полосе прокрутки показывают пропорции представления элементов в списке. Существует несколько шаблонов списков [62]: раскрывающиеся списки; представления списка; представления сетки;

окна со списками.

Списки используются для отображения библиотеки содержимого, основных данных, подробных данных и статических данных.

Библиотека содержимого используется для отображения списка, содержащего изображения и видео. В библиотеке содержимого пользователь должен иметь возможность касаться элемента списка с помощью сенсорного экрана.

Основные и подробные данные размещаются в шаблоне основных и подробных данных (приведен далее в этой главе). Шаблон предоставляет возможности по использованию списка для упорядочивания элементов. Когда пользователь выбирает элемент, в области сведений отображается дополнительная информация об этом элементе. Область сведений, как правило, содержит представление сетки.

Статические данные используются исключительно для представления не интерактивных элементов списка.

Раскрывающиеся списки [47] дают пользователям возможность выбора одного значения какого-либо параметра из нескольких вариантов значений, представленных в виде многострочного списка. Элемент управления рекомендуется использовать в том случае, если для отражения всех элементов списка не хватает места на пользовательском интерфейсе. Элемент управления может находиться в следующих состояниях:

в доступном и не активном состоянии (рис. 62а);

в активном состоянии при прикосновении пальца к сенсорному экрану (рис. 626);

в активном состоянии при передаче фокуса с помощью клавиатуры (рис. 62в);

в не доступном состоянии (рис. 62г).

Справа от элементов списка по умолчанию выделено свободное пространство (27 пикселей) для того чтобы элементы списка не перекрывалось полосами прокрутки.

Значение, предлагаемое пользователю по умолчанию, всегда видно пользователю. Если пользователю необходимо выбрать другое значение из списка, то он должен нажать на кнопку раскрытия списка (рис. 62). При этом если пользователю представляется большое количество элементов в раскрывшемся списке, то в списке появляется полоса прокрутки.

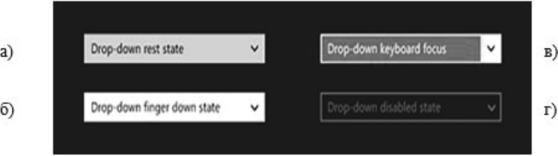


Рисунок 62. Состояния раскрывающегося списка

Если при разработке пользовательского интерфейса используются раскрывающиеся списки, то необходимо обратить внимание на следующие рекомендации:

* 1. Текстовое содержание элемента списка должно состоять только из одной строки.
* 2. Элементы списка должны располагаться логично (за счет применения сортировки: по частоте использования, алфавиту, хронологии и т.д.).

Элементы управления «список» типа «представление списка» используются:

если пользователю необходимо осуществить выбор элемента в коллекции в виде нескольких фрагментов текста;

если необходимо создать основную панель в элементе управления «шаблон основных и подробных данных» (см. далее в этой главе).

Списки типа «представление списка» с заголовками групп отображаются в один столбец (рис.63), читаются сверху вниз и «прокручиваются» по вертикали. При этом все элементы списка должны реагировать одинаково на действия пользователя.

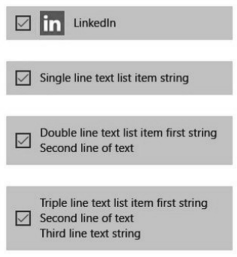


Рисунок 63 Элемент управления «список» типа «представление списка»

В случае разделения списка на группы, может быть использован элемент управления «контекстное масштабирование» (см. далее в этой главе) для облегчения навигации пользователя по группам.

Элементы управления «список» типа «представление сетки» рекомендуется использовать в случае, если пользователю необходимо осуществить выбор изображения в коллекции, предъявляемой пользователю (рис. 64). Для комфортной работы пользователя и комфортного считывания информации элементы списка в элементе управления такого типа должны располагаться справа налево и сверху вниз. Представление сетки является целесообразно применять для коллекции, в которой представлено мультимедийное содержимое.

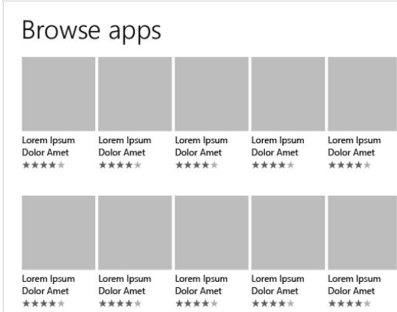


Рисунок 64 Элемент управления «список» типа «представление сетки»

Элемент управления «список» типа «окно со списком» позволяет пользователю выбрать один или несколько элементов из коллекции элементов (рис. 65). Элемент управления «окно со списками» всегда открыт (в отличие от «раскрывающегося списка»). Если для отображения предъявляемого пользователю списка не хватает места, то элементы списка можно прокручивать с помощью линии прокрутки. Элементами «окна со списками» могут быть строки текста, числовые значения, а также любые визуальные элементы. Субъективная удовлетворенность пользователя повышается, если взаимодействует напрямую с содержимым приложения. Поэтому в элементе управления «окно со списками» в качестве элементов списка может быть показано актуальное содержимое программного приложения. При этом каждый элемент списка реагирует на прикосновение пользователя к сенсорному экрану. В результате становится понятно, что элемент списка выбран.

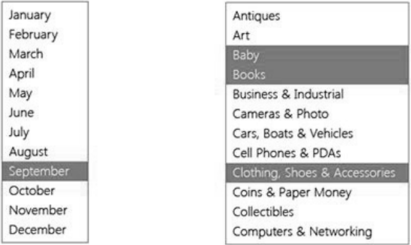


Рисунок 65 Элемент управления «список» типа «окно со списком»

Элемент управления «окно со списком» предпочтителен для использования в пользовательском интерфейсе, если экранного пространства достаточно для отображения всех необходимых для пользователя элементов списка и привлечения внимания пользователя.

В случае использования элементов управления «окно со списком» для построения пользовательского интерфейса следует придерживаться следующих рекомендаций:

* 1. Если списки состоят из 3 или 4 элементов, то возможно применение «переключателей». Оптимальное количество элементов в списке - от 3 до 9. Если каждый элемент списка предусматривает двоичный выбор, то вместо элементов списка могут использоваться «флажки» или «тумблеры» (Toggle Switch, см. далее в этой главе). В случае очень большого количества элементов в списке наилучшим вариантом могут быть элементы управления «представление сетки» или «представление списка» (при этом для сгруппированных списков должны использоваться элементы управления «контекстное масштабирование»). Если список элементов представляет собой ряд числовых значений, то возможно использовать элемент управления «ползунок» (см. далее в этой главе).
* 2. Если элементы списка могут динамически изменяться в ходе работы программного приложения, то хорошо применимы элементы управления «окно со списком». Размер элемента управления «окно со списком» желательно выбирать таким образом, чтобы список элементов не нужно было сдвигать или прокручивать.
* 3. Цель «окна со списком», а также элементов списка должны быть понятны пользователю. Элементы списка должны располагаться логично (группировка зависимых элементов, сортировка элементов по частоте использования, сортировка элементов в алфавитном порядке или по дате).
* 4. Необходимо предусмотреть визуальные эффекты и анимацию в виде изменения состояния элемента управления для отображения реакции программного приложения на прикасания пользователя к сенсорному экрану.
* 5. Поясняющий текст в элементах списка должен быть ограничен одной строкой. Если элементы списка представлены визуальными элементами, то их размер должен настраиваться. При этом если элементы «окна со списком» содержат несколько строк текста или изображений, то необходимо использовать представление сетки или представление списка. Для поясняющего текста должен использоваться шрифт по умолчанию (если не требуется использования другого шрифта).
* 6 Элемент управления «окно со списком» не должен использоваться для выполнения команд, а также скрытия либо отображения других элементов управления.
* 7. При выборе элементов, отвечающих за выполнение команд, которые могут привести к выполнению необратимых действий, рекомендуется предъявлять пользователю диалоговое окно подтверждения.

Элемент управления «контекстное масштабирование» (класс SemanticZoom) [49] дает пользователю возможность работать с двумя представлениями для одного и того же набора данных. Элемент управления (в зависимости от текущего состояния программного приложения) изменяет графическое представление и отображаемую структуру данных, предъявляемых пользователю.

В случае использования элемента управления при разработке пользовательского интерфейса необходимо придерживаться следующих рекомендаций:

* 1. Пользовательский интерфейс может содержать только один элемент управления «контекстное масштабирование».
* 2. Размер представления данных не может выходить за границы элемента управления «контекстное масштабирование».
* 3. Касание пальцем пользователя заголовка группы данных должно переключать представления данных. Например, на рис. 66 касание пальцем заголовка группы (рис. 66, левая часть) вызовет более подробное отображение данных (рис. 66, правая часть).
* 4. Переходы между представлениями данных должны быть последовательными и предсказуемыми.
* 5. Количество экранов в каждом из представлений должно быть не более трех (большее количество экранов снижает практическую ценность применения контекстного масштабирования).
* 6. Элемент управления не должен применяться для изменения объема представления данных (например, фотографии не должны преобразовываться в иконки соответствующих им файлов).
* 7. Для каждого представления необходимо использовать понятную для пользователя структуру и семантику данных. Если данные объединяются в группы, то необходимо использовать понятные для пользователя названия групп. Данные, не объединенные в группы, необходимо упорядочивать (по алфавиту, дате и т.д.).

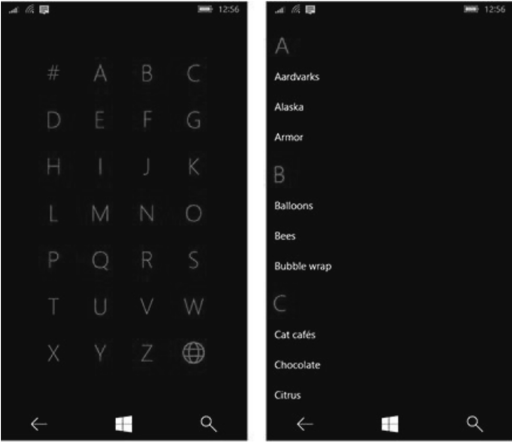


Рисунок 66 Применение элемента управления «контекстное масштабирование»

Элемент управления «тумблер» (Toggle Switch) имитирует физический переключатель (рис. 67), позволяющий пользователям включать и выключать что-либо [50]. Он имеет два состояния: включен (true) или выключен (false).



Рисунок 67 Элемент управления «тумблер» (Toggle Switch)

В случае использования элемента управления «тумблер» при разработке пользовательского интерфейса необходимо придерживаться следующих рекомендаций:

1. Необходимо различать случаи, когда выгоднее использовать «тумблеры», а когда «флажки». Элемент управления «тумблер» необходимо использовать для случаев, когда изменение состояния элемента управления приводит к немедленному выполнению какого-либо действия (на рис. 68а показано выключение и включение WiFi). При использовании элемента управления «тумблер» видно, что беспроводная связь включена (рис 686), и пользователю понятно, что нужно сделать для ее выключения. При использовании «флажка» (рис. 686) пользователю может быть не ясно, что беспроводная связь включена (пользователь может подумать, что нужно дополнительно нажать на «флажок» для ее вк

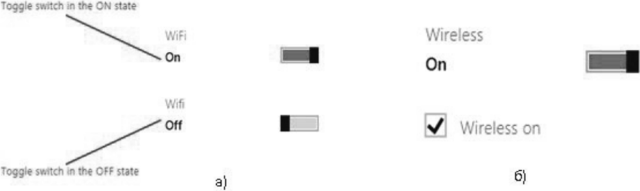


Рисунок 68 Использование элемента управления «тумблер» (Toggle Switch) и «флажок»

* 2. Элементы управления «тумблер» лучше не применять в случаях, когда пользователю необходимо выполнить какие-то дополнительные действия для того чтобы изменения вступили в силу (например, согласовать выполнение какого-либо действия, как это показано на рис. 69а). В таких случаях лучше применять элементы управления «флажок».
* 3. Элементы управления «тумблер» не рекомендуется применять для случаев, если пользователю предъявляется несколько элементов, каждый из которых может принимать лишь два значения «да» и «нет». В этом случае рекомендуется применять списки или «флажки» (рис. 696).

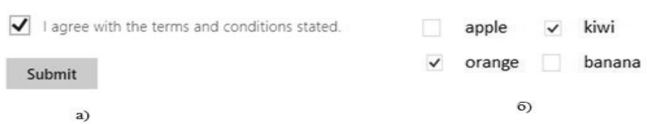


Рисунок 69 Случай, в котором «тумблер» лучше не использовать

4. Текст, поясняющий работу «тумблера», не должен быть длинным. При этом текст «Включено» и «Выключено» рекомендуется не менять на какой-то другой, если в этом нет необходимости. Но предпочтительными являются текстовые обозначения длиной 3-4 символа.

Встроенный пользовательский интерфейс захвата с камеры CameraCaptureUI [51] позволяет пользователям:

фотографировать или записывать видео на устройствах со встроенной или подключенной камерой;

обрезать фотографии и усекать записанные видео перед возвратом их запрашивающему программному приложению;

настраивать параметры камеры (яркость, контрастность и экспозиция).

Пользовательский интерфейс CameraCaptureUI захвата с камеры рекомендуется подключать к программному приложению в том случае, если программное приложение должно работать с фотографиями или видео и при этом использовать небольшое количество программного кода, необходимое для подключения и работы с интерфейсом. Пользовательский интерфейс CameraCaptureUI не рекомендуется использовать в следующих случаях:

* 1. Если планируется управлять изображениями в режиме реального времени, то интерфейс CameraCaptureUI не дает прямого контроля над видеопотоком. Поэтому вместо него следует использовать интерфейс API MediaCapture, который управляет асинхронной операцией захвата видеопотока.
* 2. Если пользовательский интерфейс CameraCaptureUI не совместим с элементами управления, которые необходимо разместить в пользовательском интерфейсе программного приложения. В этом случае следует также использовать интерфейс API MediaCapture.
* 3. Если в программном приложении требуется реализовать низкоуровневое программное управление видеопотоком, включая управление фокусом, вспышкой и стабилизацией изображения. Поэтому вместо интерфейса CameraCaptureUI следует использовать интерфейс API MediaCapture.

Кроме этого, если в программном приложении будут реализованы функции обрезки или монтажа, то в пользовательском интерфейсе интерфейс CameraCaptureUI эти функции должны быть отключены. В этом случае программное приложение не будет дублировать функции интерфейса CameraCaptureUI.

Элемент управления «панель команд» [52] предоставляет пользователям возможность работы с командами или параметрами, относящимися к работе пользователя. Элемент управления «панель команд» состоит из двух компонентов: пространства действий для размещения команд или элементов навигации (должны оставаться видимыми) и области «Дополнительно», представленной в виде многоточия (рис. 70).

Элемент управления «панель команд»

Рисунок 70 Элемент управления «панель команд»

Область «Дополнительно» предназначена для отражения списка для команд и элементов навигации, которые используются реже. При выборе «многоточия» (рис. 70) открывается меню и отображается список, элементами которого являются команды.

Элемент управления «панель команд» может быть размещен в верхней части экрана, в нижней части экрана, а также в верхней и нижней частях экрана (рис. 71).

При использовании элемента управления «панель команд» в процессе разработки пользовательского интерфейса следует придерживаться следующих рекомендаций:

* 1. При использовании только одного элемента управления «панель команд» в программном приложении желательно поместить его в нижней части экрана, что облегчает доступ пользователя.
* 2. Если в нижней части экрана расположен элемент управления «вкладка», то элемент управления «панель команд» необходимо разместить в верхней части экрана для того чтобы пользовательский интерфейс не был загроможден в нижней части экрана.
* 3. При размещении элемента управления «панель команд» на большом экране рекомендуется поместить его в верхней части экрана.
* 4. Элементы управления «панель команд» могут быть встроенными для того чтобы пользователи могли использовать их для контекстных операций.

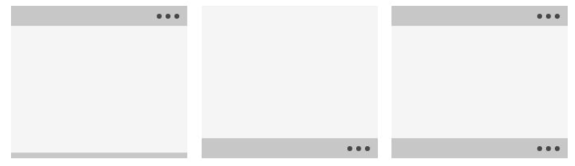


Рисунок 71 Варианты размещения элемента управления «панель команд»

5. Элементы управления «панель команд» могут быть размещены на экране в случае с одним представлением (рис. 72, слева) и в случае с несколькими представлениями (рис. 72, справа).

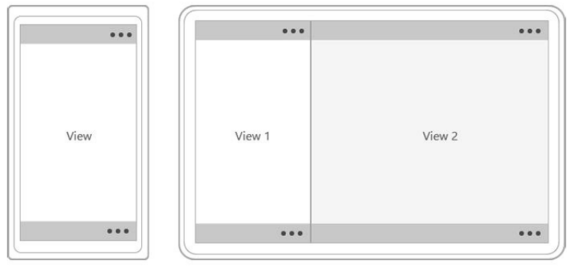


Рисунок 72 Размещение элемента управления «панель команд» в случаях с одним и двумя представлениями

* 6. Элементы управления «панель команд» могут быть размещены в любом месте действий пользователя на экране.
* 7. Приоритет команд в элементе управления «панель команд» зависит от их порядка расположения в элементе управления. На самых маленьких экранах устройств (шириной в 320 ерх, эффективных пикселей) в элементе управления размещаются 2-4 элемента.
* 8. Менее важные команды размещаются в области «Дополнительно» в раскрывающемся меню.
* 9. Команды в пространстве действий элемента управления «панель команд» изображаются с помощью значков или кнопок. В случае использования значков к ним должно быть добавлено текстовое пояснение, которое отображается под значком.
* 10. Команды, которые будут доступны во всех диалоговых окнах, необходимо размещать одном и том же месте в каждом диалоговом окне. Это позволит пользователям более комфортно работать с программным приложением и применять полученный опыт навигации в приложениях с аналогичными пользовательскими интерфейсами.
* 11. Недопустимо, чтобы все команды находились в списке раскрывающегося меню области «Дополнительно». Скрытие всех команд может привести к значительному снижению скорости работы пользователя с программным приложением.
* 12. При использовании элемента управления «панель команд» могут быть использованы всплывающие подсказки, в которых необходимо предусмотреть поиск необходимой команды с помощью логическую группировки или сортировки (рис. 73).

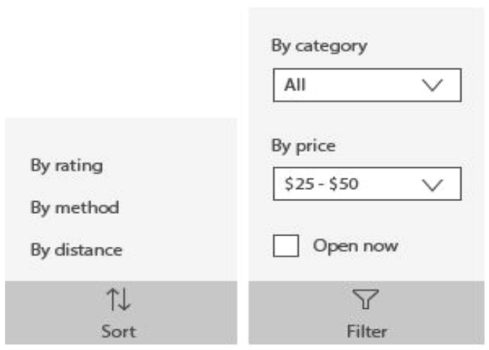


Рисунок 73 Варианты всплывающих подсказок элемента управления «панель команд»

Элемент управления «вкладки» («сводки») используются для перехода к разделам информации, которые часто используются [53]. Шаблон элемента управления состоит из двух или нескольких панелей с информацией и соответствующих им заголовками (рис. 74) [53]. Заголовки располагаются на экране, при этом пользователь (для того чтобы понимать, с какой категорией информации он работает) должен видеть, что заголовок выделен.



Рисунок 74 Элемент управления «Вкладки»

Вкладки и сводки эквивалентны и основаны на элементе управления Pivot.

При разработке программного приложения и использовании элемента управления «вкладка» необходимо учитывать несколько особенностей [53]:

* 1. Элементы управления «вкладки» могут быть размещены в верхней или нижней части экрана (рис. 75).
* 2. У заголовков элементов управления «вкладки» могут быть значки с текстом (или только текст, или только значок, рис. 75).
* 3. Заголовки элементов управления «вкладка» могут выравниваться по левому краю или по центру. Рекомендуется определять выравнивание заголовков в зависимости от размера экрана. Для ширины экрана меньше 720 ерх (эффективных пикселей) лучше всего подходит выравнивание по центру, а выравнивание по левому краю рекомендуется в большинстве случаев для ширины экрана более 720 ерх.
* 4. Элементы управления «вкладка» могут быть использованы для навигации пользователя в пользовательском интерфейсе и размещены на верхнем уровне навигации или подуровне навигации. Заголовки элементов управления «вкладка» верхнего уровня и заголовки подуровней должны визуально различаться.
* 5. Элементы управления «вкладка» могут поддерживать сенсорные жесты. Могут быть использованы следующие способы взаимодействия с элементом управления для навигации между категориями информации:

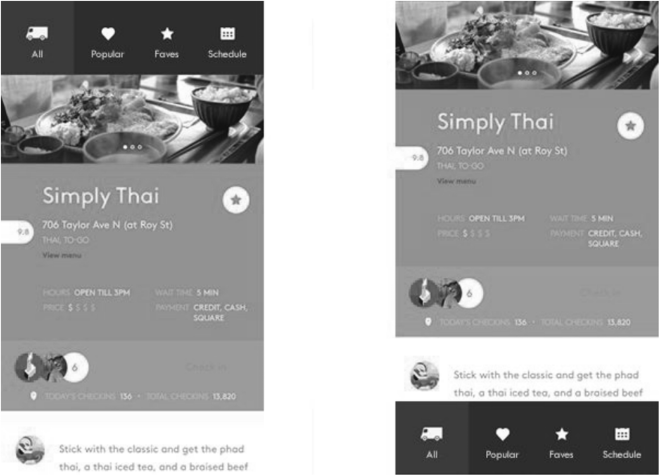


Рисунок 75 Расположение значков и текста в элементах управления «вкладки»

* 1. Прикосновение пользователя к заголовку «вкладки» для перехода к информации, соответствующей «вкладке», или «прокрутка» пальцем для перехода к смежной категории информации.
* 2. Прикосновение пользователя к заголовку «вкладки» для перехода к информации, соответствующей «вкладке» (без «прокрутки»).
* 6. Оптимальное расположение элементов управления «вкладка» зависит от способа взаимодействия пользователя с программным приложением и устройства, на котором работает программное приложение [53].
* 7. Необходимо сохранять одно и то же общее количество заголовков «вкладок» в альбомной и книжной ориентации экрана. В «карусельном» режиме не рекомендуется использовать более 5 заголовков (это может снизить скорость работы пользователя с программным приложением).

Элемент управления «контекстное меню» - это меню, которое позволяет пользователю мгновенно выполнять команды из списка пользовательских команд [54]. Список может предъявляться пользователю в любом месте пользовательского интерфейса, например, при нажатии правой клавиши мыши. Контекстные меню можно закрыть с помощью нажатия левой клавишей мыши (или прикоснувшись к сенсорному экрану) за пределами панели, в которой находится список команд меню. Элемент управления «контекстное меню» может использоваться в следующих целях:

для выполнения контекстных команд с выделенным участком текста («Копировать», «Вырезать», «Вставить», «Проверка орфографии» и

т. д.);

для работы с буфером обмена;

для выполнения пользовательских команд (в соответствии с контекстом выполняемого программного приложения);

для выполнения команд для работы с объектом, который нельзя выбрать или выделить.

Контекстное меню с одной панелью (рис. 76а) рекомендуется применять для относительно короткого списка команд. При этом необходимо использовать разделители для логической группировки команд. Если число команд большое, то может применяться каскадное контекстное меню [54], в котором используются всплывающие элементы и возможность прокрутки. Также могут использоваться разделители для группировки логически связанных команд (рис. 766). В случае использования элемента управления «контекстное меню» в процессе разработки пользовательского интерфейса необходимо придерживаться следующих рекомендаций:

* 1. Название команд в списке контекстного меню должны быть короткими. Длинные команды рекомендуется обрезать. Имя команды должно начинаться с прописной буквы
* 2. Между группами логически связанных команд необходимо использовать разделители.
* 3. Количество команд в списке контекстного меню не должен быть большим. Это необходимо для увеличения скорости работы пользователя.
* 4. Если команда уже есть в пользовательском интерфейсе, то ее нежелательно использовать в контекстном меню (то есть, в списке контекстном меню должны быть расположены команды, которые отсутствуют на экране).



Рисунок 76 Элемент управления «контекстное меню»

Элемент управления «диалоговое окно» пользовательского интерфейса предоставляет пользователю информацию о текущем состоянии программного приложения [55]. В большинстве случаев диалоговые окна являются модальными и блокируют дальнейшую работу с программным приложением (пока не будут закрыты явным образом, например, с помощью нажатия кнопки). Как правило, диалоговые окна требуют от пользователя выполнения некоторого действия. Диалоговые окна предназначены для того чтобы сообщить пользователю некоторую информацию о дальнейшей работе с программным приложением. Пользователь должен прочитать эту информацию, осознать ее перед продолжением работы и подтвердить дальнейшее действие (рис. 77). Элемент управления «диалоговое окно» рекомендуется использовать в следующих случаях [55]:

для вывода сообщений об угрозах безопасности пользователя; для подтверждения намерения изменить ценные данные без возможности отмены действия;

для подтверждения намерения удалить ценные данные; для подтверждения совершения действий, критичных для пользователя (например, списание средств со счета пользователя в случае подтверждения покупки).

для организации вывода сообщений об ошибках программного приложения без прекращения его работы (например, ошибки работы с файлами, ошибки подключения к сети);

для задания пользователю вопроса от имени программного приложения (если приложение не может сделать выбор от имени пользователя, то пользователю должны быть предоставлены понятные варианты выбора действий).

В случае использования элемента управления «диалоговое окно» желательно соблюдать следующие рекомендации [55]:

* 1. В первой строке текста (в заголовке) диалогового окна необходимо изложить, что должен делать пользователь в диалоговом окне. При этом заголовок не обязателен, хоть он и является основной инструкцией для пользователя. Заголовок текста диалогового окна должен быть кратким (заголовки, превышающие одну строку, рекомендуется усекать). Если диалоговое окно используется для простого сообщения или вопроса, то заголовок можно опустить. В этом случае необходимую информацию для пользователя должен передавать текст сообщения. При этом необходимо, чтобы заголовок и текст сообщения соответствовали выбору кнопок в диалоговом окне.
* 2. Сообщение внутри диалогового окна является обязательным. Содержание сообщения должно быть как можно проще. Если есть заголовок диалогового окна, то сообщение внутри диалогового окна должно быть использовано для более подробных сведений или пояснения терминологии.
* 3. В диалоговом окне должна быть хотя бы одна кнопка. Название кнопки должно отражать конкретный вариант ответа на инструкцию, приведенную в содержании сообщения внутри диалогового окна.
* 4. Диалоговые окна после возникновения ошибок отображают сообщение об ошибке в диалоговом окне, а также сведения о причинах ошибки и вариантах дальнейших действий пользователя.



Рисунок 77 Модальное диалоговое окно

Использование фильтрации позволяет пользователю просмотреть элементы списка, удовлетворяющие заданному критерию фильтрации [56]. В результате фильтрации происходит скрытие элементов списка, не удовлетворяющих условию, заданному пользователем. На рис. 78а приведено использование условия «Best rates» для списка категорий. Фильтрация наиболее применима в случаях, если предъявляемый пользователю список состоит из большого количества элементов (более 25 элементов) с применением прокрутки. Использование команды сортировки в отличие от команды фильтрации не скрывает элементы списка, а просто меняет их порядок (рис. 786). Сортировка также лучше всего применима для списков большого размера с прокруткой (более 25 элементов).

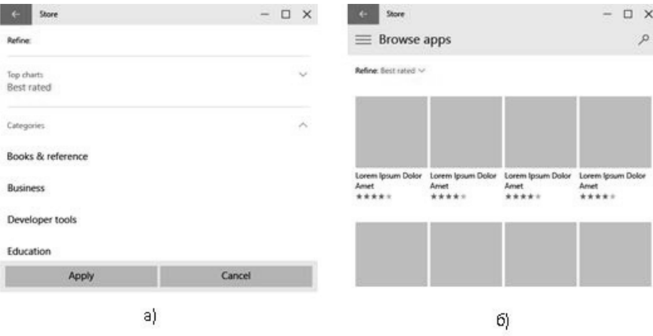


Рисунок 78 Примеры использования фильтрации и сортировки

Существует несколько способов выполнения фильтрации.

* 1. Фильтрация произвольной формы. Для такого вида фильтрации может быть использован элемент управления «поле автозаполнения» для ввода текста. В результате ввода текста в элемент управления «поле автозаполнения» (описан выше в данной главе) отображаются только элементы списка, которые соответствуют тексту фильтра. Фильтрацию произвольной формы лучше всего применять в случае, если пользователь знает содержание списка. При этом реализация такого типа фильтрации на пользовательском интерфейсе не занимает много места на экране.
* 2. Предопределенные параметры фильтрации. Пользователю предоставляется набор параметров фильтрации, которые относятся к элементам в списке. Такой вид фильтрации лучше всего применять, когда пользователь не очень хорошо знаком с содержанием списка. Данный вид фильтрации более сложный по сравнению с фильтрацией произвольной формы. При этом реализация фильтрации такого типа может занимать много места на пользовательском интерфейсе в зависимости от количества параметров фильтрации.

При реализации фильтрации и сортировки в пользовательском интерфейсе необходимо придерживаться следующих рекомендаций:

* 1. Необходимо сообщать пользователю об активации фильтра. Если пользователь не видит активации фильтра, то он может не знать о том, что, кроме отфильтрованных элементов списка, есть и другие (скрытые) элементы списка.
* 2. Должна быть реализована очистка фильтра.
* 3. Пользователи должны знать о возможности добавления или исключения параметров фильтрации.
* 4. Необходимо информировать пользователя о том, каким образом сортируются элементы (для того, чтобы пользователь понял, как просматривать элементы списка).

Элемент управления «элемент пролистывания» (FflipView) [57] предназначен для просмотра коллекции изображений. Элемент управления позволяет отображать только одно изображение. В случае наличия сенсорного экрана для прокрутки коллекции необходимо сдвинуть элемент управления с помощью жеста. В случае манипуляции мышью в пользовательском интерфейсе должны быть кнопки навигации, которые выделяются при наведении на нее указателя мыши (рис. 79а). В случае перемещения по коллекции изображений с использованием клавиатуры необходимо использовать клавиши со стрелками. В случае использования элемента управления «элемент пролистывания» следует придерживаться следующих рекомендаций:

* 1. Элемент пролистывания лучше всего подходит для просмотра содержимого маленьких и средних коллекций изображений (до 25 элементов в коллекции).
* 2. Для удобства работы с коллекцией необходимо добавление индикатора страницы, который обеспечивает наличие удобной точки отсчета. На рис. 796 применен индикатор страницы в виде серии точек, каждая из которых может быть выбрана и соответствует изображению в коллекции. Оптимальным расположением индикатор контекста обычно является расположение по центру и ниже коллекции. Выделенная точка указывает текущий элемент коллекции и обычно имеет белый цвет. Остальные точки имеют серый цвет. Количество точек для удобства пользователя должно быть ограничено (не более 10 точек). В случае большего количества точек пользователь может испытывать затруднения с определением своего положения в коллекции изображений.
* 3. В случае коллекций изображений, состоящей из 10-25 элементов и, если пользователю тяжело ориентироваться в коллекции изображений, вместо индикатора страницы используется индикатор контекста (рис. 79в), который позволяет пользователям переходить к определенным элементам коллекции. Кроме того, индикатор контекста, позволяет пользователям получать информацию о том, насколько велика коллекция и в каком месте коллекции он находится.
* 4. Пользователю должна быть предоставлена возможность быстрого перехода к нужному элементу коллекции изображений, а также возможность контроля своего положения в коллекции изображений.
* 5. Стандартной формой реализации «элемента пролистывания» является горизонтальный просмотр коллекции изображений, начиная с крайнего левого элемента (рис. 84а). Такая форма реализации «элемента пролистывания» может применяться как в книжной, так и в альбомной ориентации на экранах всех устройств.
* 6. В больших коллекциях изображений (более 25 изображений) использование «элемента пролистывания» не рекомендуется, так как длительное перелистывание элементов коллекции может стать утомительным для пользователя. Для таких коллекций изображений рекомендуется использовать списки (рассмотрены ранее в данной главе).



Рисунок 79 Реализация пролистывания в пользовательском интерфейсе

С помощью элемента управления «всплывающий элемент» («FlyOut») [58] производится реализация всплывающих контекстнозависимых диалоговых окон, которые появляются на пользовательском интерфейсе в зависимости от действий пользователя. Элемент управления позволяет показать меню, обнаружить скрытый элемент управления, отобразить дополнительные сведения о каком-либо элементе управления пользовательского интерфейс, а также запросить у пользователя подтверждение его действий. Всплывающее диалоговое окно закрывается

* (исчезает) после нажатия на участок экрана за пределами диалогового окна. Всплывающее диалоговое окно содержит простое сообщение без кнопок. Текст в диалоговом окне должен переноситься по словам. Если же размер текста превышает контейнер диалогового окна, то необходима полоса прокрутки. Элемент управления «всплывающий элемент» может быть использован в следующих случаях:
  + 1. Для показа пользователю сообщений в случае отображения в пользовательском интерфейсе каких-либо ранее скрытых от пользователя элементов управления.
  + 2. Для отображения дополнительных сведений, таких как сведения об элементах управления на пользовательском интерфейсе, не скрытых от пользователя.
  + 3. Для показа пользователю предупреждений и подтверждений, в том числе связанных с потенциально опасными действиями.

Если для построения пользовательского интерфейса используются элементы управления «всплывающий элемент», то необходимо придерживаться следующих рекомендаций:

* 1. Всплывающие диалоговые окна должны появляться рядом с теми точками пользовательского интерфейса, из которых они вызываются.
* 2. Должен быть указан элемент управления пользовательского интерфейса, к которому привязано всплывающее диалоговое окно и сторону элемента управления, над которой появилось всплывающее диалоговое окно.
* 3. Всплывающее диалоговое окно не должно загораживать важные для пользователя элементы пользовательского интерфейса.
* 4. Всплывающее диалоговое окно должно закрываться после того, как пользователь что-то в нем выберет.
* 5. Если с помощью всплывающих диалоговых окон реализуется меню, то предпочтительным является использование одноуровневых всплывающих меню.

Элемент управления «главный раздел» («Hub») позволяет группировать информацию, предъявляемую пользователю в пользовательском интерфейсе программного приложения, в связанные между собой разделы (категории) [59].

Элементы управления «главный раздел» обычно имеют заголовок страницы, а несколько разделов содержимого получают заголовок раздела (рис. 80).

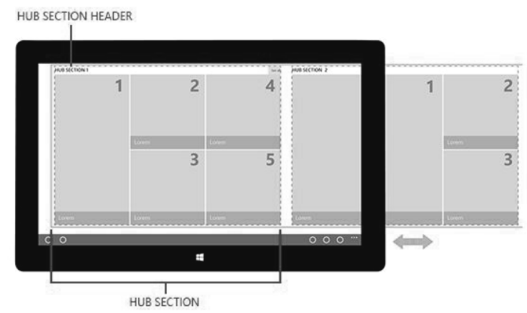


Рисунок 80 Элемент управления «главный раздел»

Разделы (категории) предназначены для того, чтобы пользователь мог перемещаться от одних данных к другим на пользовательском интерфейсе в том порядке, который для него является наиболее предпочтительным. Содержимое элемента управления «главный раздел» может отображаться более подробно (в случае если это необходимо пользователю). Элемент управления «Главный раздел» имеет несколько функций:

* 1. Навигация (элемент управления позволяет визуально отображать информацию в кратком и наглядном виде, что позволяет легко ее просматривать).
* 2. Разделение информации, содержащейся в элементе управления, на разделы (категории), в каждом из которых информация упорядочена определенным образом.
* 3. Возможность использования различного визуального отображения разделов (различные геометрические размеры и соотношения сторон) в случае, если в разделах элемента управления содержится разнородная информация.
* 4. Поддержка различной ширины разделов в случае отображения информации, находящейся на различных уровнях вложенности.

Элемент управления используется для отображения большого объема информации, организованного в форме иерархии. В элементе управления могут отражаться различные категории содержимого. Каждая из категорий соответствует отдельной странице с информацией (рис. 81). Страницы отображаться в той форме, которая наилучшим образом соответствует содержимому категории.

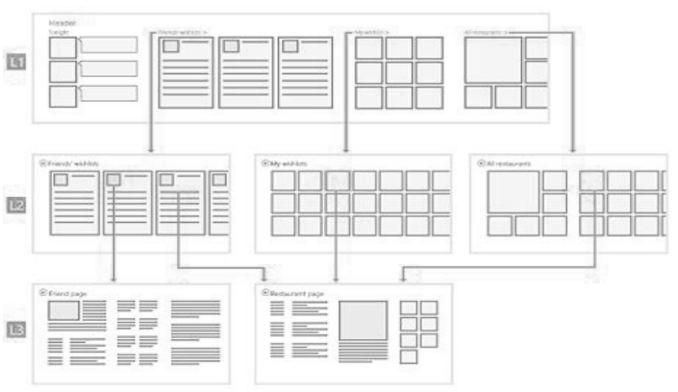


Рисунок 81 Иерархическая организация элемента управления «главный раздел»

Для размещения содержимого элемента управления в пользовательском интерфейсе и перехода к нужной информации может быть использовано несколько способов:

* 1. Горизонтальный сдвиг элемента управления с вертикальной прокруткой списка или сетки.
* 2. Вертикальный сдвиг с горизонтальной прокруткой списка или сетки. При этом списки содержимого в элементе управления «главный раздел» сдвигаются в направлении, перпендикулярном направлению прокрутки элемента управления.

При разработке пользовательского интерфейса с использованием элемента управления «главный раздел» необходимо придерживаться следующих рекомендаций:

* 1. В случае разработки программных приложений для мобильных устройств не могут одновременно отображаться несколько элементов управления «главный раздел».
* 2. Содержимое элемента управления «главный раздел» может быть раскрыто более подробно. Для того чтобы пользователь видел возможность более подробного раскрытия информации, предъявляемое пользователю содержимое элемента управления (верхний «слой» содержимого) должно быть уменьшено таким образом, чтобы из-под него была видна часть более подробного содержимого (часть нижнего «слоя»).
* 3. В элемент управления «главный раздел» может быть добавлено несколько разделов, каждый из которых будет выполнять свою функцию. Переход между разделами пользователь может осуществлять с помощью касаний, поддержка которых встроена в элемент управления.
* 4. Для того чтобы адаптировать содержимое элемента управления под различные размеры экранов устройств, на которых используется программное приложение, необходимо производить динамическое переформатирование содержимого элемента управления в соответствии с размерами экрана устройства.
* 5. Если программное приложение использует несколько элементов управления «главный раздел», то для корректного отражения элементов управления может быть использован элемент управления «контекстное масштабирование» (рассмотрен ранее в данной главе).
* 6. Не рекомендуется использовать в разделах элемента управления «главный раздел» элементы управления для ссылок на другие разделы. Для перехода к другому разделу рекомендуется использовать интерактивные заголовки.
* 7. Элемент управления «главный раздел» должен быть настроен в соответствии с потребностями проектируемого программного приложения. Могут быть настроены следующие параметры:

число разделов;

тип содержимого в каждом из разделов;

размещение и порядок следования разделов;

размер разделов, интервалы между разделами;

интервалы между разделами сверху или снизу от главного раздела;

стиль и размер текста в заголовках и содержимом, цвет фона, разделов, заголовков разделов и содержимого разделов.

При использовании элемента управления «гиперссылка» [60] пользователь либо переходит в другую часть программного приложения, либо к другому программному приложению, либо открывает определенный URI-адрес в отдельном программном приложении-браузере. Используются два типа элемента управления «гиперссылка»: текстовый элемент, который отображается в виде динамического текста (класс HyperLink, рис. 82а), и кнопка, которая отображается в виде размеченного текста (класс HyperLinkButton, рис. 826).

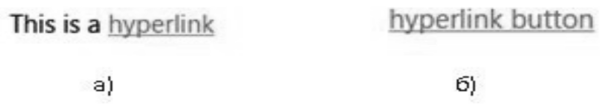


Рисунок 82 Элемент управления «гиперссылка»

В случае использования элемента управления «гиперссылка» при разработке пользовательского интерфейса необходимо придерживаться следующих рекомендаций:

* 1. Элемент управления применяется в случае, если в пользовательском интерфейсе необходимо иметь фрагмент текста, который будет реагировать выбор его пользователем и перенаправлять пользователя по адресу, указанному во фрагменте текста.
* 2. Текстовая гиперссылка используется, если пользователь не нуждается в разрыве фрагмента текста (текстовая гиперссылка занимает мало места и поэтому может быть успешно применена в составе фрагмента текста). При этом текстовые гиперссылки обычно небольшие, и на них бывает трудно попасть, особенно на устройствах с сенсорным экраном.
* 3. Кнопка-гиперссылка используется в том случае, если нужен автоматический разрыв строки или если необходим более крупный (чем текстовая гиперссылка) элемент управления. Высота кнопки- гиперссылки составляет 44 ерх (эффективных пикселя).

Элемент управления «всплывающая подсказка» (класс ToolTip) [61] - это краткое сообщение, привязанное к другому элементу управления (рис. 83).

Такие элементы управления позволяют пользователям понять назначение незнакомых для пользователя объектов в пользовательском интерфейсе программного приложения. Элемент управления «всплывающая подсказка» автоматически появляется в том случае, если пользователь наводит и удерживает палец (на сенсорном экране) или указатель мыши на элементе управления. Элемент управления исчезает, если пользователь передвинет палец (указатель мыши) с элемента управления.

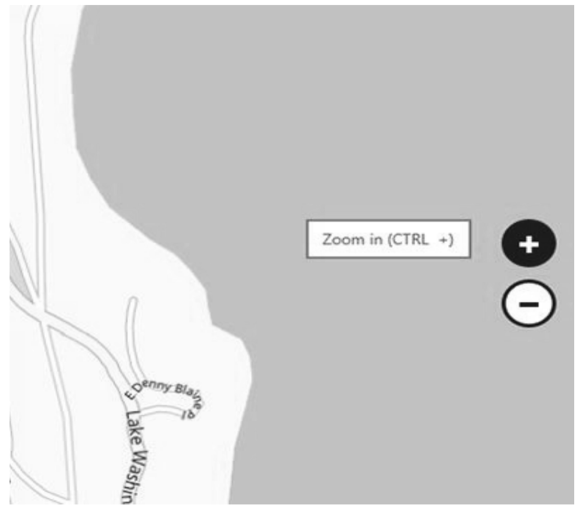


Рисунок 83 Элемент управления «всплывающая подсказка»

Элемент управления «всплывающая подсказка» используется для того чтобы получить дополнительную информацию об элементе управления, с которым пользователь собирается совершить какое-то действие.

Также можно использовать всплывающую подсказку для отображения названия элемента управления для выполнения команды (это позволяет пользователю удостовериться в правильности выбора элемента управления).

При использовании элемента управления «всплывающая подсказка» следует придерживаться следующих рекомендаций:

* 1. Элемент управления следует использовать только тогда, он должен отображаться только в результате действий пользователя. Не допускается самостоятельное отображение всплывающих подсказок.
* 2. Всплывающие подсказки следует использовать для тех элементов управления, для которых нет текстовых обозначений (внутри элементов управления расположены иконки или графические метки).
* 3. Элемент управления необходимо использовать, если необходимо более подробное описание какого-либо элемента управления на пользовательском интерфейсе. При этом может не хватать места на пользовательском интерфейсе для размещения дополнительной информации.
* 4. Элемент управления не следует использовать, если в нем должна выдаваться информация об ошибках, предупреждениях, а также о статусе элементов управления пользовательского интерфейса.
* 5. Если пользователю необходимо будет взаимодействовать с всплывающей подсказкой, то элемент управления «всплывающая подсказка» не рекомендуется использовать.
* 6. Если всплывающие подсказки раздражают или отвлекают пользователя, то элемент управления «всплывающая подсказка» не рекомендуется использовать.
* 7. Текст в элементе управления «всплывающая подсказка» должен быть кратким и информативным. В тексте не должно повторяться текст, который уже есть в пользовательском интерфейсе. Текст должен состоять из дополнительной информации, без которой пользователь может обойтись.
* 8. Использование изображений в подсказках рекомендуется для случаев их применения для перехода со страницы на страницу.
* 9. В элементе управления «всплывающая подсказка» не должны содержаться интерактивные элементы управления.

Шаблон «панель навигации» (класс Split View) [62] позволяет сохранять свободное пространство экрана устройства за счет использования элементов навигации верхнего уровня. В основном такой шаблон используется в программных приложениях для мобильных устройств, но также шаблон применяется и для устройств с большими экранами. Шаблон используется в режиме наложения или в режиме стыковки. При использовании в режиме наложения панель навигации остается свернутой и не открывается до тех пор, пока пользователь не нажмет кнопку (если экранного пространства недостаточно для отражения всех элементов управления пользовательского интерфейса). При использовании в режиме стыковки панель навигации всё время остается в развернутом состоянии (пользователь будет иметь больше возможностей навигации в пользовательском интерфейсе, если экранное пространство позволяет разместить все элементы управления пользовательского интерфейса). Шаблон «панель навигации» состоит из кнопки, панели для категорий навигации и области содержимого (рис. 84) [76].

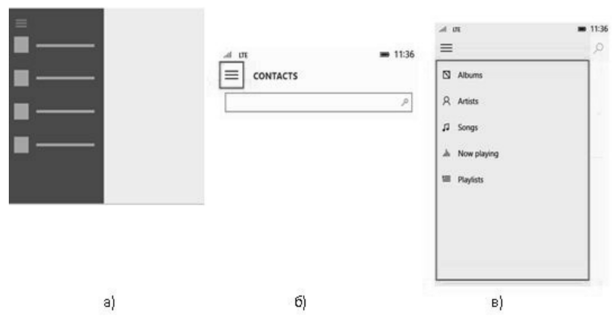


Рисунок 84 Элемент управления «панель навигации»

Кнопка позволяет пользователю открывать и закрывать панель и представляется по умолчанию в виде трех горизонтальных линий («кнопка-гамбургер»: левый верхний угол, рис. 846). Такая кнопка позволяет пользователю открывать и закрывать «панель навигации» по мере необходимости. Кнопка не перемещается вместе с «панелью навигации». Обычно кнопка имеет название в виде текстовой строки. На верхнем уровне навигации пользовательского интерфейса поясняющий текст кнопки располагается рядом с кнопкой. На более низких уровнях навигации поясняющий текст может быть связан со страницей, на которой в данный момент находится пользователь. Если «панель навигации» отображается в режиме стыковки, то кнопка может не использоваться. Панель категорий навигации предназначена для размещения элементов навигации (рис. 84в, 84а, левая часть). В области содержимого (рис. 84а, правая часть) отображаются сведения из выбранного элемента навигации. В области содержимого могут содержаться отдельные элементы управления пользовательского интерфейса, отражающие содержание выбранного элемента навигации. Также в области содержимого могут отражаться или элементы навигации другого подуровня. Шаблон «панель навигации» лучше всего использовать в следующих случаях:

* 1. Для программных приложений, в которых предусмотрены элементы навигации верхнего уровня для представления однородной информации (например, информации по категориям товаров мебельного магазина).
* 2. Для организации навигации пользователя в программном приложении (если в пользовательском интерфейсе используется только элемент управления «панель навигации»).
* 3. В случае если количество категорий навигации верхнего уровня, которые необходимо использовать в пользовательском интерфейсе, больше 10.
* 4. В случае если необходимо экономить место в пользовательском интерфейсе (использование элемента управления в режиме наложения). В этом случае данный элемент управления может быть использован для скрытия тех элементов пользовательского интерфейса, которые редко используются.
* 5. В режиме наложения «панель навигации» может быть использована для экранов любого размера (как в книжной, так и в альбомной ориентации). В режиме наложения «панель навигации» по умолчанию находится в свернутом состоянии. Поэтому в пользовательском интерфейсе отображается только кнопка. Нажатие кнопки открывает и закрывает «панель навигации». Развернутое состояние отменяется, когда пользователь осуществляет выбор, когда нажата кнопка «Назад», когда пользователь касается экрана или кликает левой клавишей мыши за пределами панели. Следует учесть, что «панель навигации» при появлении из режима наложения появляется поверх элементов управления пользовательского интерфейса, не адаптируясь под их расположение. Лучше всего «панель навигации» в режиме наложения использовать при разработке программных приложений для устройств с небольшими экранами (мобильные телефоны и фаблеты).
* 6. В режиме стыковки шаблон «панель навигации» лучше всего использовать для устройств с большими экранами (планшеты) с шириной экрана более 720 ерх (эффективных пикселей). Для альбомной ориентации экрана необходимо применять масштабирование содержимого «панели навигации».
* 7. Работа с шаблоном «панелью навигации» должна поддерживать действия по перетаскиванию объектов в «панель навигации» и из неё.
* 8. Элементы списка в «панели навигации» (те, которые выбраны пользователем) должны быть выделены.
* 9. Если экран устройства слишком узкий для показа «панели навигации» в режиме стыковки, то при повороте устройства из альбомной в книжную ориентацию «панель навигации» должна переходить в режим наложение (свернутое состояние). Если происходит поворот из книжной в альбомную ориентацию, то «панель навигации» может переводиться из режима наложения (свернутого состояния) в режим стыковки.

Элемент управления «ползунок» (элемент управления диапазоном, Slider) [63] позволяет выбрать значение параметра из заданного диапазона. Ползунок состоит из «дорожки» и «бегунка» (рис. 85). Элемент управления «ползунок» может быть расположен по горизонтали или по вертикали.



Рисунок 85 Элемент управления «ползунок»

«Дорожка» представляет собой полоску, на которой может отображаться диапазон значений параметра в виде шкалы. У «бегунка» большая площадь касания. Для обеспечения поддержки сенсорных возможностей пользователя, «бегунок» должен располагаться достаточно далеко от края экрана. Работа с «ползунком» происходит посредством нажатия на «бегунок» и дальнейшего передвижения его вдоль «дорожки» с помощью мыши или касания пользователя к сенсорному экрану (рис. 85).

В случае использования элементов управления «ползунок» при разработке пользовательского интерфейса необходимо придерживаться следующих рекомендаций:

* 1. Элемент управления «ползунок» рекомендуется применять, если пользователь должен видеть мгновенную реакцию программного приложения на изменение значения параметра. Элемент управления «ползунок» не должен использоваться для индикации хода выполнения задачи.
* 2. Если параметр может принимать менее четырех значений, то рекомендуется использовать элемент управления «переключатель». Если значение параметра не может быть представлено в виде относительной величины, то рекомендуется применять элементы управления «переключатель» или «список».
* 3. Размер элемента управления «ползунок» должен позволять пользователю легко выбрать нужное значение параметра с помощью мыши, клавиатуры или сенсорного экрана. Размеры «бегунка» (входит в состав элемента управления «ползунок») не должны меняться в процессе работы программного положения. Частота расположения делений на «дорожке» (входит в состав элемента управления «ползунок») должна позволять пользователям различать их. При отключении «ползунка» необходимо отключать и все связанные с ним метки или визуальные индикаторы.
* 4. Для показа диапазона значений параметра необходимо использовать метки диапазона. Если «ползунок» используется для выбора целочисленных значений, то необходимо установить фиксированное целочисленное значение шага. Последний шаг диапазона значений параметра соответствует максимальному значению.
* 5. Отображение текущего значения «ползунка» производится с помощью метки значения (текста). Также текущее значение «ползунка» может отображаться в ином визуальном представлении (показ рядом с «бегунком» результатов применения выбранного значения параметра).
* 6. Метка (поясняющий текст) для «ползунка» должна указывать, для чего он используется. Метку «ползунка» не рекомендуется размещать под «ползунком» (в этом случае при прикосновении пользователя к «ползунку» палец пользователя может загородить метку). Текст метки «ползунка» не должен содержать знаки препинания и состоять из одного слова. Метки на краях «ползунка» должны носить описательный характер и иметь противоположные значения (например, «Макси- мум/минимум»).
* 7. Необходимо придерживаться естественного расположения «ползунка» в соответствии с аналогами в реальном мире (например, для отображения температуры должна использоваться вертикальная ориентация, а для отображения объектов мультимедиа используется горизонтальная ориентация). Также ориентация «ползунка» должна соответствовать макету страницы. Если «ползунок» расположен по вертикали, то максимальное значение параметра должно быть размещено наверху. Если же «ползунок» расположен горизонтально, то при расположении содержимого пользовательского интерфейса «слева направо» минимальное значение параметра должно быть размещено слева.
* 8. Если элемент управления используется на странице, которая прокручивается только в одном направлении, то необходимо использовать ориентацию «ползунка», отличную от направления прокрутки. В противном случае пользователи могут посчитать, что «ползунок» предназначен для прокрутки страницы.

Элемент управления «метка» (Label) [64] предназначен для отображения имени или названия элемента управления или группы элементов управления (рис. 86).

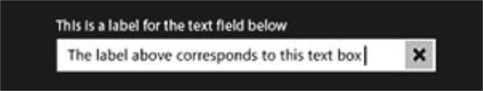


Рисунок 86 Элемент управления «метка» (расположен над текстбоксом)

При использовании элемента управления «метка» для разработки пользовательского интерфейса следует придерживаться следующих рекомендаций:

* 1. Элемент управления «метка» следует использовать для отображения пояснений пользователю о назначении элементов управления в пользовательском интерфейсе или о действиях, который он должен выполнить.
* 2. Текст в метке для элементов управления должен отображаться в виде одного существительного или короткого текста. При этом в тексте не должно быть знаков препинания.

В шаблоне основных и подробных данных [65] отображается главная панель (с представлением списка) и область сведений для содержимого. Если выбрать элемент в главном списке, область сведений обновляется. При реализации шаблона основных и подробных данных рекомендуется использовать стиль «стопкой» или стиль «рядом» в зависимости от доступного места на экране устройства (если доступная ширина окна устройства 320-719 ерх, то рекомендуемый стиль - стопкой, если более 720 ерх, то рекомендуемый стиль - «рядом»). При использовании расположения «стопкой» отображается только одна панель: главная или область сведений (рис. 87). Пользователь начинает работу с главной панели и детализирует информацию вплоть до ее появления в области сведений. При использовании стиля «рядом» главная панель и область сведений отображается одновременно (рис. 88).



Рисунок 87 Реализация шаблона основных и подробных данных (стиль «стопкой»)



Рисунок 88 Шаблон основных и подробных данных

В списке на главной панели есть визуальный элемент выбора для указания выбранного элемента. При выборе нового элемента в главном списке сразу же обновляется область сведений. При разработке пользовательского интерфейса с использованием шаблона основных и подробных данных необходимо придерживаться следующих рекомендаций:

* 1. Шаблон основных и подробных данных хорошо подходит для работы с электронной почтой, адресными книгами, для случаев, когда необходимо выделить главный элемент в большой коллекции информации, а также в случае необходимости быстрого обмена информацией (удаление, перемещение, добавление) между элементами коллекции.
* 2. При разработке шаблона основных и подробных данных используются отдельно друг от друга страница (шаблон) для построения главной панели и отдельно страница (шаблон) для построения области сведений.
* 3. Для построения главной панели шаблона основных и подробных данных, в которой отображается список, который может содержать текст и изображения, лучше всего использовать элемент управления «список» («представление списка», см. выше в данной главе). В области сведений шаблона основных и подробных данных рекомендуется использовать элемент управления содержимым [21], который предназначены для поддержки интерактивной связи с пользователем. Элемент управления содержимым представляет собой специализированный тип элемента управления, который может хранить некоторое содержимое. Все элементы управления содержимым являются наследниками класса ContentControl, который позволяет:

определять содержимое внутри элемента управления;

определять порядок перехода между разделами содержимого;

отображать фон (цвет поверхности), передний план и рамки;

поддерживать форматирование размера текстового содержания и шрифта.

Если в содержимом много разделов, то для размещения содержимого может быть использован элемент управления «список» («представление сетки», см. выше в данной главе).

Элемент управления «проигрыватель мультимедиа» (класс SystemMediaTransportControls) предназначен для просмотра и прослушивания видео, звуков и изображений [67] и содержит стандартный набор кнопок (кнопка воспроизведения/паузы, переход вперед, переход назад), некоторые из которых отображаются, а некоторые остаются скрытыми в зависимости от устройства. Воспроизведение мультимедиа может быть встроенным (внедренным в просматриваемую страницу или в группу других элементов управления) или выполняться в специальном полноэкранном режиме. Элемент управления «проигрыватель мультимедиа» поддерживает однострочный и двухстрочный режимы работы. Однострочный режим предусматривает работу с кнопкой «Воспроизве- дение/пауза», расположенной слева от временной шкалы (рис. 89а). Такой режим работы лучше всего подходит для компактных экранов устройств.

В двухстрочном режиме для элементов управления предоставлено больше места (рис. 896), что упрощает работу пользователя с временной шкалой.



Рисунок 89 Однострочный и двухстрочный режимы работы элемента управления «проигрыватель мультимедиа»

Наиболее предпочтительной для пользователя является использование элемента управления «проигрыватель» с темным фоном, который обеспечивает более высокую контрастность (особенно в условиях слабого освещения).

Наиболее информативным для пользователя является полноэкранный режим работы «проигрывателя мультимедиа» с двухстрочным режимом работы.

Настройка кнопок и других элементов управления внутри «проигрывателя мультимедиа» должна быть ограничена (если были настроены для использования по умолчанию в конкретном пользовательском интерфейсе).

В элементе управления «проигрыватель мультимедиа» могут быть размещены дополнительные средства управления (кнопки и другие элементы управления). При этом панель управления внутри не должна быть перегружена средствами управления.

Временная шкала в элементе управления «проигрыватель мультимедиа» должна быть такая, чтобы пользователь мог без затруднений выбрать на ней нужный ему момент времени.

Элемент управления «индикатор выполнения» (Progress, ProgressBar, ProgressRing) [68] служит для уведомления пользователя о том, что выполняется длительная операция. Определенный «индикатор выполнения» (рис. 90в) показывает степень выполнения операции в процентах. Определенный «индикатор выполнения» представляет собой цветную полоску, увеличивающуюся в размере и стремящуюся заполнить серую фоновую полоску. Пропорция цветной полоски по отношению к фоновой полоске дает приблизительное представление о том, какая часть команды уже выполнена. Неопределенный индикатор выполнения (рис. 90а) и кольцевой индикатор выполнения (рис. 906) показывает, что операция еще выполняется, и неизвестно, сколько времени до конца ее выполнения. Индикатор выполнения доступен только для чтения (с ним нельзя взаимодействовать). Неопределенный «индикатор выполнения» или кольцевой «индикатор выполнения» состоит из непрерывно движущихся цветных точек. Элемент управления «индикатор выполнения» необходимо отображать в пользовательском интерфейсе очень аккуратно (чрезмерное использование «индикатора выполнения» может отвлекать пользователя от работы с программным приложением и значительно снижать скорость работы с приложением).

Если выполнение команды, запущенной пользователем, требует более двух секунд, то необходимо отображать элемент управления «индикатор выполнения». После выполнения операции «индикатор выполнения» необходимо скрыть через 500 миллисекунд.

При использовании элемента управления «индикатор выполнения» следует учесть, что он должен отражать длительность выполнения команды компьютером, а не время работы пользователя. При этом пользователю не должен показываться «индикатор выполнения», если программное приложение само (без команды пользователя) выполняет команду, а пользователь не знает о выполнении этой команды.

Если команда выполняется в фоновом режиме, но ее выполнение важно для пользователя, то «индикатор выполнения» должен выводиться и отражать состояние выполнения такой команды.

Для обозначения незавершенности выполнения команды необходимо использовать многоточия (рис. 90г). Если выполняется несколько команд, то может быть отображено количество оставшихся задач (при завершении задач «индикатор выполнения» должен исчезать).

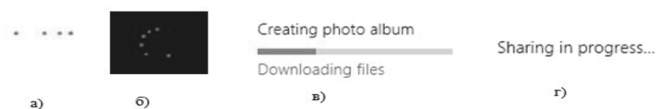


Рисунок 90 Элемент управления «индикатор выполнения»

Если содержимое, над которым выполняются действия, используется для демонстрации хода выполнения команды, то «индикатор выполнения» не используется. В этом случае отображение «индикатора выполнения» в пользовательском интерфейсе будет отвлекать внимание пользователя, а также снижать скорость его работы.

Если при выполнении команды может быть определена часть выполненной работы (завершение команды предсказуемо, а продолжительность выполнения известна), то используется определенный «индикатор выполнения». В противном случае должен быть использован неопределенный «индикатор выполнения» или кольцевой «индикатор выполнения».

Кольцевой «индикатор выполнения» применяется в случае выполнения команд с неизвестным временем выполнения и модальных команд (команд, блокирующих действия пользователя).

Неопределенный «индикатор выполнения» применяется в случае выполнения команд с неизвестным временем выполнения и немодальных команд (команд, не блокирующих действия пользователя).

Если действия пользователя при выполнении команды блокируются менее двух секунд, то такие команды являются немодальными. Также считаются немодальными команды, которые блокируют действия пользователя до тех пор, пока не выполнится какой-то процент задачи, после чего пользователь снова может работать в программном приложении. Для таких команд используется неопределенный «индикатор выполнения» хода выполнения.

Если модальное состояние команды длится более двух секунд, то во время модальной фазы команды должен использоваться кольцевой «индикатор выполнения», а во время немодальной фазы команды происходит переход к неопределенному «индикатору выполнения».

Должна быть предусмотрена возможность отмены или приостановки выполнения команды в случаях, когда действия пользователя блокированы более 10 секунд, предусмотрите способ ее отмены. Возможность отмены должна быть доступна с начала выполнения операции и при этом пользователь знает, сколько времени осталось до окончания выполнения команды.

Использование «индикатора выполнения» не должно происходить одновременно с использованием вида курсора для ожидания «песочные часы» (наличие двух способов для визуализации хода выполнения задачи отвлекает внимание пользователя от работы с программным приложением).

Если выполняются несколько связанных друг с другом команд, то для визуализации их хода выполнения используется один «индикатор выполнения», а не несколько «индикаторов выполнения» для каждой из команд.

Расположение и размер «индикатора выполнения» не должны изменяться в ходе выполнения команды.

Обновление хода выполнения команды должно происходить равномерно. Не должно быть остановок индикации хода выполнения команды. После 100-процентного выполнения команды необходима пауза для окончательного заполнения «индикатора выполнения». После этого «индикатор выполнения» необходимо скрыть.

Если выполнение команды остановлено (самим пользователем или по каким-то другим причинам) и при этом пользователь может возобновить ее выполнение, то необходимо визуально отображать остановку выполнения. В этом случае под «индикатором выполнения» должна быть размещена текстовая строка с соответствующим сообщением.

Если выполнение задачи остановлено, и выполнение команды нельзя возобновить или необходимо начать выполнение команды с начала, то необходимо визуально отобразить сообщение об ошибке. В этом случае под «индикатором выполнения» должна быть размещено текстовое сообщение, объясняющее пользователю, что произошло и что как решить проблему (если это возможно).

Если для определения точного времени выполнения команды необходимо некоторое время (или выполнение какого-либо действия), то необходимо сначала использовать неопределенный «индикатор выполнения», а затем необходимо переключиться на определенный индикатор. После переключения «индикатор выполнения» должен находиться в одном и том же месте пользовательского интерфейса и иметь такой же размер.

Если имеется список элементов, и при этом над каждым из элементов может быть выполнена какая-то определенная команда, то необходимо отобразить определенный «индикатор выполнения» рядом с элементом (рис. 91). При этом название команды необходимо отображать над «индикатором выполнения», а состояние выполнения необходимо отражать под «индикатором выполнения». Текст, поясняющий состояние выполнения команды, не отображается, если состояние является очевидным для пользователя. После завершения выполнения команды необходимо скрыть «индикатор выполнения».

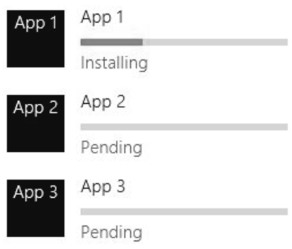


Рисунок 91 Размещение списка «индикаторов выполнения»

При отображении списка выполняемых команд необходимо выровнять содержимое таким образом, чтобы пользователи могли быстро оценить степень выполнения команд (рис. 91). При этом отображаются «индикаторы выполнения» для всех команд, в том числе и для тех, выполнение которых отложено. После завершения выполнения команды соответствующий ей «индикатор выполнения» удаляется из списка.

Если выполняемая команда блокирует выполнение программного приложения, то «индикатор выполнения» должен быть отображен в панели (плитке) этого программного приложения.

Значение «индикатора выполнения» команды не должно уменьшаться (от 100 % до 0 %). Если возникла необходимость вернуться к предыдущему состоянию выполнения команды («откат выполнения команды»), то в пользовательском интерфейсе необходимо отразить новый

«индикатор выполнения», который будет отражать ход выполнения «отката выполнения команды» и сопровождаться поясняющим текстом.

Кольцевой «индикатор выполнения» необходимо отражать в пользовательском интерфейсе рядом с элементом управления, с помощью которого пользователь запустил выполнение команды, или в том месте пользовательского интерфейса, где будут отражаться данные. Поясняющий текст, отображающий состояние выполнения команды, необходимо отражать справа от кольцевого «индикатора выполнения» при этом кольцевой «индикатор выполнения» и поясняющий текст должны иметь одинаковый цвет. Элементы управления в пользовательском интерфейсе, с которыми пользователь не должен взаимодействовать во время выполнения команды, должны быть блокированы до завершения ее выполнения. Если выполнение команды завершено или во время выполнения команды произошла ошибка, необходимо скрыть «индикатор выполнения», а на его месте необходимо отобразить текстовое сообщение о завершении выполнения команды или о произошедшей ошибке.

В диалоговом окне программного приложения, если команда должна завершиться до перехода к другому диалоговому окну, необходимо отобразить кольцевой «индикатор выполнения» над областью элемента управления, с помощью которого пользователь запустил выполнение команды. При этом расположение кольцевого «индикатора выполнения» необходимо выровнять по левому краю в соответствии с расположением содержимого диалогового окна (рис.92а). В диалоговом окне программного приложения с элементами управления, выровненными по правому краю (рис. 926), кольцевой «индикатор выполнения» необходимо размещать слева от элемента управления, с помощью которого пользователь запустил выполнение команды, или над ним. В диалоговом окне программного приложения с элементами управления, выровненными по левому краю (рис. 92в), кольцевой «индикатор выполнения» необходимо размещать справа от элемента управления, с помощью которого пользователь запустил выполнение команды, или под ним. При отображении в пользовательском интерфейсе списка запущенных команд (рис. 92г) необходимо кольцевой «индикатор выполнения» и текст, поясняющий состояние выполнения команды, разместить под названием элемента списка. В случае окончания выполнения команды или возникновения ошибки при выполнении команды необходимо скрыть кольцевой «индикатор выполнения», а на его месте необходимо отобразить текстовое сообщение о завершении выполнения команды или о произошедшей ошибке. Ход выполнения команды во всплывающем элементе отражается с помощью неопределенного «индикатора выполнения», расположенного в верхней части всплывающего элемента (рис. 92).

Ширина неопределенного «индикатора выполнения» настраивается так, чтобы он занимал всю ширину диалогового окна. В этом случае «индикатор выполнения» сообщает пользователю о выполнении команды, но практически не отвлекает внимание пользователя. Для того чтобы не мешать отображению «индикатора выполнения», в верхней части диалогового окна не должно быть поясняющего текста о назначении всплывающего элемента. Рекомендацию по использованию кольцевого «индикатора выполнения (размеры, расположение поясняющих текстов, название, размеры и цвет шрифта поясняющего текста) приведены на рис. 92д.

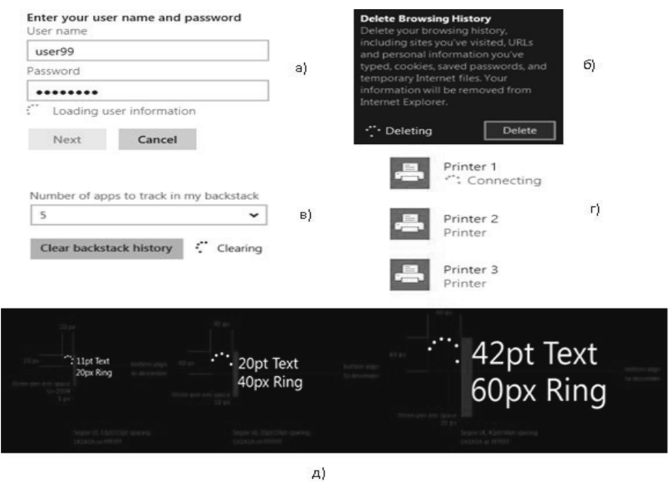
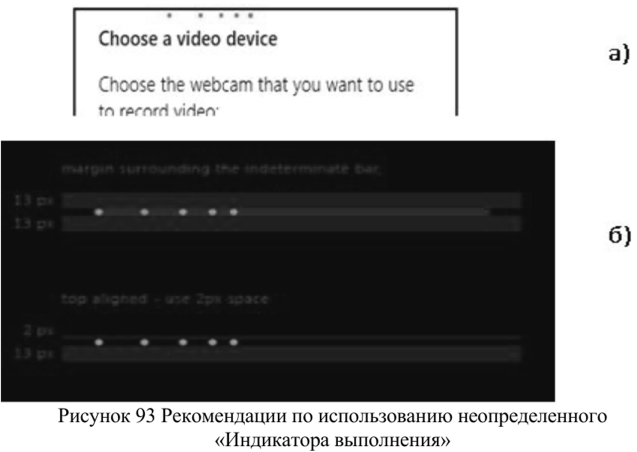


Рисунок 92 Рекомендации по размещению кольцевого «Индикатора выполнения»

Неопределенный индикатор может располагаться по ширине диалогового окна (рис. 93а). Ширина неопределенного «индикатора выполнения» настраивается так, чтобы он занимал всю ширину всплывающего элемента. В этом случае «индикатор выполнения» сообщает пользователю о выполнении команды, но практически не отвлекает внимание пользователя. Для того чтобы не мешать отображению «индикатора выполнения», в верхней части всплывающего элемента не должно быть поясняющего текста о назначении всплывающего элемента. Рекомендации по размещению неопределенного «Индикатора выполнения»: ширина окружающих линий в случае размещения «Индикатора выполнения» внутри диалогового окна (верхняя часть рис. 936) и в верхней части диалогового окна (нижняя часть рис. 936).



Для описания состояния выполнения команды необходимо использовать существительные, образованные от глаголов («Соединение», «Скачивание», «Отправка»). Если выполнение команды приостановлено или при выполнении команды возникла исключительная ситуация, то для описания состояния выполнения команды необходимо использовать причастия («Приостановлено», «Скачивание прервано» или «Выполнение отменено»).

Элемент управления «оценка» (объект Rating) [69] предназначен для демонстрации пользователем степени предпочтения, оценки или удовлетворенности чем-то (информационным сервисом, фильмом, программным приложением, предприятием). Элемент управления «оценка» содержит пять звезд: одна звезда означает самую низкую оценку, а пять звезд означает самую высокую оценку. Элемент управления «оценка» может показывать три вида оценок: среднюю оценку (рис. 94а), предварительную оценку (рис. 946) и пользовательскую оценку (рис. 94в).



Рисунок 94 Элемент управления «оценка»

Элемент управления «оценка» не используется в качестве фильтра и как элемента управления, отображающего оценку пользователя с двумя значениями (повторное нажатие на элемент управления не снимет уже поставленную «звезду»). Для предоставления пользователю дополнительной информации при использовании элемента управления «оценка» необходимо использовать всплывающие подсказки. Всплывающая подсказка позволяет показать пользователю информацию по каждой звезде («отлично», «очень хорошо», «неплохо», рис. 104а).

После выставления оценки пользователем элемент управления «оценка» отключается для того чтобы пользователь не мог добавить или изменить оценку. Если элемент управления «оценка» выключен, то сделанная пользователем оценка продолжает отображаться. При этом пользователи не могут добавлять или изменять оценку. Если элемент управления «оценка» доступен в режиме «только чтение» и его невозможно включить, то его размер должен быть меньше по сравнению с элементами управления «оценка», значение которых может изменить пользователь (рис 956). При использовании элемента управления «оценка» (если необходимо пользователю) одновременно отображается средняя оценка и оценка пользователя (рис. 95в).

При этом может два варианта такого отображения: среднее значение отображается в текстовой строке, прикрепленной к элементу управления «оценка»;

одновременно отображаются два элемента управления «оценка» (для отображения оценки пользователя и для отображения средней оценки).

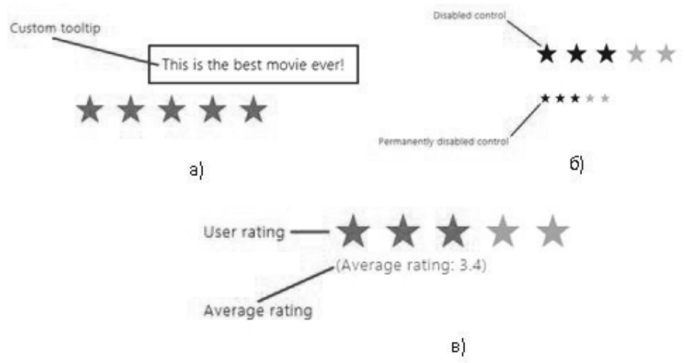


Рисунок 95 Использование элемента управления «оценка»

С помощью элемента управления «поиск» (класс AutoSuggestBox) позволяет обнаружить необходимую для пользователя информацию в пользовательском интерфейсе программного приложения [70]. Ввод текста для поиска информации является самым распространенным режимом ввода данных. Другими способами ввода данных для поиска информации является голосовой ввод и жестовый ввод, которые требуют наличия дополнительного оборудования для взаимодействия с устройством, а также наличия дополнительных элементов управления или специального диалогового окна в пользовательском интерфейсе.

Если пользователь пока еще не ввел текст в поле ввода, то в пользовательском интерфейсе отображается «поле нулевого ввода», которое может отражать недавнюю историю поиска, популярные поисковые запросы, различные предложения поиска, рекомендации и подсказки. Если пользователь начинает вводить информацию, то автозаполнение заменяет информацию в «поле нулевого ввода». Пользователю предлагается непрерывно обновляемый по мере ввода информации набор вариантов запросов. Это позволяет ускорить ввод информации в элемент управления «поиск» и точнее сформулировать запрос. Возможности по предложению пользователю различных вариантов запроса предоставляет элемент управления «поле автозаполнения» (рассмотрен выше в данной главе) при этом внутри элемента управления «поиск» появляется значок «микрофон» или «фиксация» (рис 96а, б).

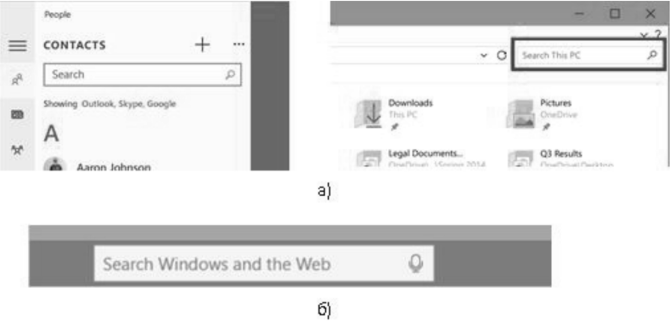


Рисунок 96 Варианты реализации элемента управления «поиск»

Результаты поиска, как правило, отображаются рядом с элементом управления «поиск». Размещение рядом поля ввода и результатов поиска позволяет пользователю осуществить быстрый доступ к результатам запроса или вводу нового запроса.

При повторном использовании элемента управления «поиск» происходит выделение предыдущего запроса (текстовая строка, которая была ранее в поле ввода исчезает, но при этом строка предыдущего запроса остается в поле для того чтобы пользователь мог ее выбрать для редактирования).

Результаты поиска могут отображаться в любой форме, которая лучше всего передает их содержание. Элемент управления «список» типа «представление списка» (рассмотрено ранее в этой главе) может быть использован для большинства поисковых запросов. Элемент управления «список» типа «представление сетки» (рассмотрено ранее в этой главе) может быть использовано для отображения результатов поиска в виде изображений и мультимедиа. Также для отображения результатов поиска может быть использована карта (для отражения пространственного распределения результатов запроса).

Для отображения в элементе управления «поиск» подсказок об области поиска необходимо использовать текстовую строку (рис. 966), в которой сообщаются области поиска (например, «Поиск в Windows», «Поиск в списке контактов», «Поиск в почтовом ящике», «Поиск в настройках», «Поиск места»).

Наличие правильно сформулированной подсказки об области поиска обеспечивает пользователю комфортные условия для работы с элементом управления «поиск».

Для осуществления поиска в элементе управления «поиск» используется глиф увеличительного стекла (рис. 96а), в качестве которого используется символ шрифта Segoe UI размером 15 эффективных пикселей или глиф микрофона (рис. 966).

Элемент управления «комбинированный режим» [71] имеет в составе разворачиваемую и сворачиваемую панель, а также область содержимого. Область содержимого остается всегда видимой. Панель можно развернуть, свернуть, расположить в левой (правой части) диалогового окна программного приложения.

Панель элемента управления «комбинированный режим» может использоваться в трех режимах:

* 1. Наложение (панель скрыта, а после открытия перекрывает содержимое диалогового окна программного приложения).
* 2. Встроенный режим (пользователь постоянно видит панель, которая не перекрывает содержимое диалогового окна программного приложения, при этом пространство диалогового окна делится между областью, занимаемой панелью, и областью, где отражена информация для пользователя, рис. 97а).



Рисунок 97 Панель элемента управления «комбинированный режим»

3. Компактный (пользователь постоянно видит панель, которая имеет размеры, необходимые только для отображения значков размером около 48 ерх в ширину, при этом пространство диалогового окна делится между областью панели и областью содержимого, рис. 976).

Элемент управления «комбинированный режим» представляет собой шаблон, который после добавления кнопки развертывания и свертывания («кнопки-гамбургера») и элемента управления «список» типа «представление списка» может использоваться как меню навигации. Использование панели для навигации на основе элемента управления «комбинированный режим» повышение скорости работы пользователя с программным приложением (обеспечивается быстрый доступ к другим функциям программного приложения).

Элемент управления «представление веб-страницы» (класс Web- View) [72] внедряет в программное приложение возможности по работе с информацией, аналогичные с возможностями браузера Microsoft Edge, который предназначен для замены браузеров Internet Explorer 11 и Internet Explorer Mobile на всех устройствах, которые будут поставляться с операционной системой Windows 10. В Microsoft Edge появились возможности по созданию заметок из веб-страниц. Пользователь может делать пометки или рисунки непосредственно на веб-странице и затем передавать веб-страницу с обновлениями. Элемент управления «представление веб-страницы» предназначен для отображения:

* 1. Содержимого HTML-страниц расширенного формата (программных кодов сценариев, обеспечивающих связь между сценариями и выполняемым программным приложением).
* 2. Динамически генерируемого кода.
* 3. Содержимого файлов программного приложения.

Для использования элемента управления «представление вебстраницы» при разработке пользовательского интерфейса необходимо, чтобы загружаемые веб-страницы использовали цвета, шрифтовое оформление и навигацию, отображение информации, соответствующее отображению информации в выполняемом программном приложении. Если отображение информации в веб-страницах не соответствуют отображению информации в выполняемом программном приложении, то необходимо для загрузки информации или использовать другие элементы управления, или применять преобразование веб-страниц в удобное для пользователя состояние.