

6-ЛЕКЦИЯ. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

6.1 Использование инструментов для создания приложений в системе Андроид

Дизайн интерфейсов сводится к вопросу о том, как оформить и расположить инструменты таким образом, чтобы внятно отразить поведение и представить информацию. Каждый инструмент визуальной композиции имеет ряд свойств, и сочетание этих свойств придает элементу смысл. Пользователь получает возможность разобраться в интерфейсе благодаря различным способам приложения этих свойств к каждому из элементов интерфейса. В тех случаях, когда два объекта обладают общими свойствами, пользователь предположит, что эти объекты связаны или похожи. Когда пользователи видят, что свойства отличаются, они предполагают, что объекты не связаны.

Создавая пользовательский интерфейс, проанализируйте перечисленные ниже визуальные свойства каждого инструмента или группы инструментов. Чтобы создать полезный и привлекательный пользовательский интерфейс, следует тщательно поработать с каждым из этих свойств.

Форма

Форма - главный признак сущности объекта для человека. Мы узнаем объекты по контурам. Если мы увидим на картинке синий ананас, мы его сразу опознаем, потому что мы помним его форму. И лишь потом мы удивимся странному цвету (см. рис. 6.1). При этом различение форм требует большей концентрации внимания, чем анализ цвета или размера. Поэтому форма - не лучшее свойство для создания контраста, если требуется привлечь внимание пользователя.



Рис. 6.1. В первую очередь мы видим ананас, а уже потом начинаем задумываться, почему он синий

Размер

Более крупные элементы привлекают больше внимания, особенно если они значительно превосходят размерами окружающие элементы. Люди автоматически упорядочивают объекты по размеру и склонны оценивать их по размеру; если у нас есть текст в четырех размерах, предполагается, что относительная важность текста растет вместе с размером и что полужирный текст более важен, чем текст с нормальным начертанием. Таким образом, размер - полезное свойство для обозначения информационных иерархий.

Цвет

Цветовые различия быстро привлекают внимание. В некоторых профессиональных областях цвета имеют конкретные значения, и этим можно пользоваться. Так, для бухгалтера красный цвет - отрицательные результаты, а черный - положительные.

Цвета приобретают смыслы и благодаря социальным контекстам, в которых проходит наше взросление. Например, белый цвет на Западе ассоциируется с чистотой и миром, а в Азии и арабских странах - с похоронами и смертью. При этом цвет изначально не обладает свойством упорядоченности и не выражается количественно, поэтому далеко не идеален для передачи информации такого рода. Кроме того, не следует делать цвет единственным способом передачи информации, поскольку цветовая слепота встречается довольно часто.

Применяйте цвет с умом. Чтобы создать эффективную визуальную систему, позволяющую пользователю выявлять сходства и различия объектов, используйте ограниченный набор цветов - эффект радуги перегружает восприятие пользователя и ограничивает возможности по передаче ему информации.

Выбор цветовой палитры для программы необходимо проводить очень осторожно. По разным данным, той или иной формой цветовой слепоты страдают до 10% мужчин, и использование, например, красного и зеленого цветов для указания контраста затрудняет работу с приложением для этих людей.

Яркость

Понятия темного и светлого обретают смысл преимущественно в контексте яркости фона. На темном фоне темный текст почти не виден, тогда как на светлом он будет резко выделяться. Контрастность люди воспринимают легко и быстро, так что значение яркости может стать хорошим инструментом привлечения внимания к тем элементам, которые требуется подчеркнуть. Значение яркости - также упорядоченная переменная, например, более темные (с более низкой яркостью) цвета на карте легко интерпретируются: они обозначают большие глубины или большие значения других параметров.

Направление

Направление полезно, когда требуется передавать информацию об ориентации (вверх или вниз, вперед или назад). Помните, что восприятие направления может быть затруднено в случае некоторых форм и при малых размерах объектов, поэтому ее лучше использовать в качестве вторичного признака. Так, если требуется показать, что рынок акций пошел вниз, можно использовать направленную вниз стрелку красного цвета.

Текстура

Разумеется, изображенные на экране элементы не обладают настоящей текстурой, но способны создавать ее видимость. Текстура редко бывает полезна для передачи различий или привлечения внимания, поскольку требует значительной концентрации на деталях. И тем не менее текстура может быть важной подсказкой. Засечки и выпуклости на элементах пользовательского интерфейса обычно указывают, что элемент можно перетаскивать, а фаски или тени у кнопки усиливают ощущение, что ее можно нажать.

Расположение

Расположение - это переменная, упорядоченная и выражаемая количественно, а значит, полезная для передачи иерархии. Расположение также может служить средством создания пространственных отношений между объектами на экране и объектами реального мира (например, небо в верхней половине, земля в нижней).

Расположение элементов мобильного приложения очень сильно влияет на удобство использования и зависит от того, как пользователь будет держать устройство (см. рис. 5.2). Подробнее об этом будет рассказано в продолжении данного курса.



Рис. 6.2. Различные варианты удержания смартфона

6.2. Размещение кнопок, текстовых полей и других инструментов

Элементы управления - это доступные для манипулирования самостоятельные экранные объекты, посредством которых люди взаимодействуют с цифровыми продуктами. Элементы управления (controls, другое название - widgets - сокращение от windows gadgets - оконные приспособления) - это базовые строительные блоки графического пользовательского интерфейса.

Рассматривая элементы управления с учетом целей пользователя, их можно разбить на четыре основные категории:

- командные элементы управления, применяемые для выполнения функций;
- элементы выбора, позволяющие выбирать данные или настройки;
- элементы ввода, применяемые для ввода данных;
- элементы отображения, используемые для наглядного непосредственного манипулирования.

Некоторые элементы управления сочетают в себе свойства более чем одной категории.

Командные элементы управления

Командные элементы управления выполняют действия, причем делают это немедленно. Главным и по сути единственным командным элементом является кнопка, которая обладает множеством вариантов отображения. Элементы меню также являются командными идиомами.

Кнопки

Кнопки обычно легко опознаются благодаря их псевдотрехмерности (рис. 5.3). Действие выполняется сразу после нажатия на кнопку. Часто особым образом выделяется кнопка по умолчанию, соответствующая наиболее часто используемому действию.



Рис. 6.3 Скриншоты популярной игры "Cut the Rope". Кнопки кажутся выпуклыми, а нажатая кнопка меняет цвет

Кнопка - удобный и привлекательный с визуальной точки зрения элемент управления. Весь ее облик подсказывает, что на нее можно нажать, и это характеризует ее ожидаемое назначение. Рекомендуется изменять внешний вид нажатой кнопки, так как это облегчает понимание работы программы пользователем.

Кнопки-значки

Кнопки, помещенные на панель инструментов, обычно становятся квадратными, теряют текстовую надпись и обзаводятся пиктограммой - пояснением в виде графического значка (рис. 6.4).

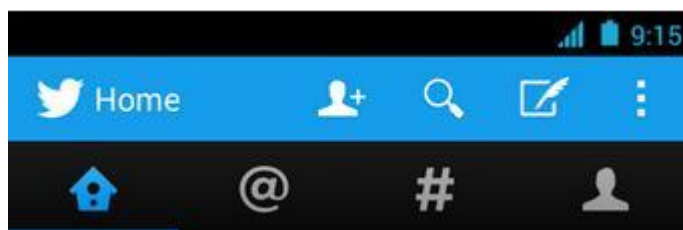


Рис. 6.4. Кнопки-значки в Android-приложении Twitter

Считается, что кнопки-значки очень удобны: они постоянно на виду и взаимодействовать с ними проще, чем с элементами раскрывающегося меню. Поскольку они постоянно видны, то легко запоминаются. У большинства пользователей не возникает вопросов относительно ожидаемого назначения кнопки. Проблема в том, что изображение на кнопке иногда бывает непонятным. Например, пиктограмма с изображением дискеты,

традиционно обозначающая сохранение, часто непонятна молодым пользователям, которые никогда не работали с реальными дискетами.

Гиперссылки

Текстовые гиперссылки - распространенный способ навигации в сети и веб-приложениях, однако при программировании для мобильных устройств их следует избегать. Дело в том, что попасть по ссылке пальцем с первого раза часто затруднительно и пользователей раздражает необходимость повторения этих действий.

Элементы управления выбором

Элементы выбора позволяют пользователю выбрать из группы допустимых объектов тот, с которым будет совершено действие. Элементы выбора применяются также для действий по настройке. Распространенными элементами выбора являются флажки и списки.

Раньше использование элементов управления выбором не приводило к немедленному выполнению действий - требовалась еще и активация командного элемента. Сейчас возможны оба варианта. Если желательно дать пользователю возможность несколько раз осуществить выбор перед выполнением действия, следует создать явный командный элемент управления (кнопку). Если же пользователю полезно сразу видеть результат своих действий, и эти действия легко отменить, разумно сделать так, чтобы элемент выбора играл также и роль командного элемента.

Флажки

Назначение флажка очевидно. Щелкнув по флажку, пользователь немедленно увидит появившуюся галочку. Флажок прост, нагляден и изящен, однако основан на тексте. Качественный текст может исключить возможность неоднозначного толкования флажка. Однако этот же поясняющий текст вынуждает пользователя замедляться для прочтения, а также занимает значительное экранное пространство.

Традиционно флажки имеют квадратную форму. Не забывайте, что пользователи распознают визуальные объекты по форме, и квадратная форма флажков - важный стандарт.

Выключатели

Существует возможность сделать флажок более наглядным, применив в качестве основы кнопку-значок, которая может фиксироваться в нажатом состоянии. Такой элемент называется **выключателем** (рис. 6.5).



Рис. 6.5. Один из самых распространенных выключателей - кнопка "Pause" в играх

Состояние выключателя остается неизменным до следующего щелчка. Выключатели экономно расходуют экранное пространство: они занимают меньше места, потому что их назначение описывается не с помощью текста, а посредством визуальных средств. Разумеется, это означает, что им присущ тот же недостаток обычных кнопок-значков - неоднозначность пиктограмм.

Триггеры

Кнопки-триггеры - это разновидность элементов управления. Они призваны экономить экранное пространство, к сожалению, ценой значительной дезориентации пользователя. Классический пример - размещение на одной кнопке функций воспроизведения и паузы для музыкального проигрывателя. Подводным камнем такого подхода является то, что элемент управления можно ошибочно посчитать индикатором состояния проигрывателя ("на паузе" или "идет воспроизведение"). Элемент управления может служить либо индикатором состояния, либо кнопкой переключения состояний, но не тем и другим одновременно (рис. 6.6).



Рис. 6.6. Если кнопка говорит "ВКЛЮЧЕНО", когда находится в состоянии "выключено", то непонятно, в каком же состоянии находится кнопка. Если кнопка говорит "ВЫКЛЮЧЕНО", когда находится в состоянии "выключено", тогда возникает вопрос, где искать кнопку "ВКЛЮЧЕНО"?

Радиокнопки

Радиокнопки внешне похожи на флажки (рис. 6.7), но являются взаимоисключающими, то есть выбор одного из вариантов автоматически аннулирует предыдущий выбор. В каждый момент времени может быть выбрана только одна кнопка. Радиокнопки всегда объединяются в группы из двух или более радиокнопок, причем в каждой группе одна радиокнопка всегда выбрана. Радиокнопки всегда круглые по той же причине, по которой флажки всегда имеют квадратную форму: именно такими они были изначально.

Радиокнопки занимают даже больше места, чем флажки, однако в некоторых случаях такой расход экранного пространства оправдан.

Кнопка-значок преобразовала радиокнопки так же, как флажки, заменив их в основном интерфейсе приложения. Если два или более выключателя объединены схемой взаимного исключения - так, чтобы в каждый момент мог быть включен лишь один из них, - они ведут себя точно так же, как радиокнопки. Так образуются **радиокнопки со значками**. Элементы управления цветом в Adobe Photoshop - хороший пример радиокнопок со значками (рис. 6.8).

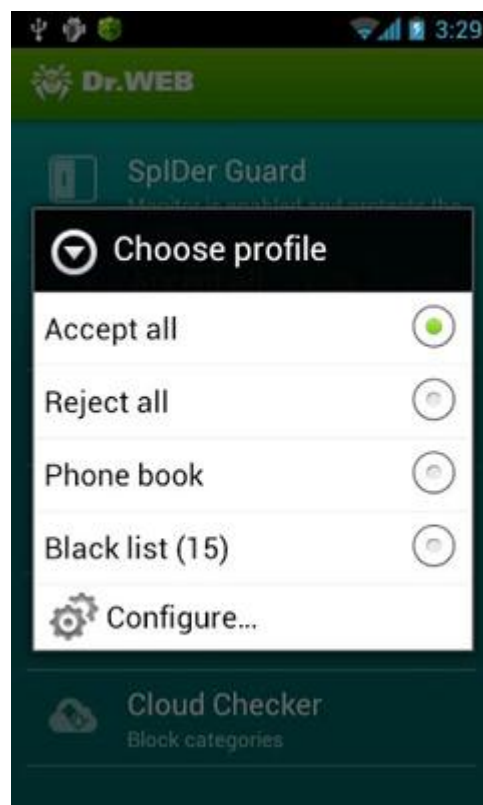


Рис. 6.7. Радиокнопки Android-приложения Dr.Web

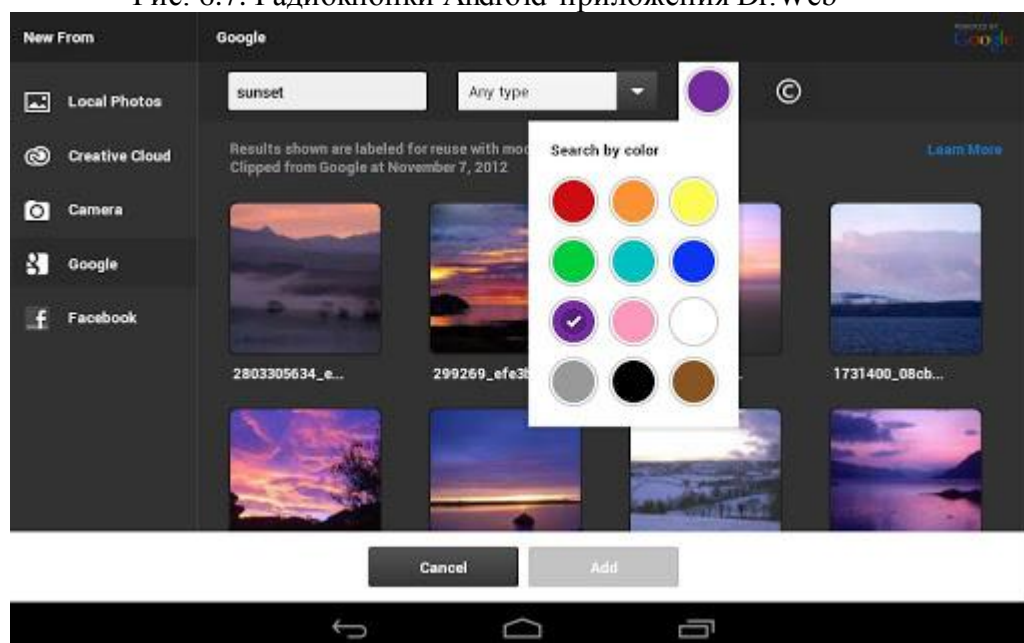


Рис. 6.8. Элементы управления цветом в Android-приложении Adobe Photoshop представляют собой группу радиокнопок со значками

Списки

Элементы управления типа "список" позволяют осуществлять выбор из конечного множества текстовых строк, каждая из которых представляет команду, объект или признак. Подобно радиокнопкам, списки - мощный инструмент, упрощающий взаимодействие за счет устранения возможности неправильного выбора. Списки - это небольшие текстовые области с полосой прокрутки, автоматически подключаемой при

необходимости (рис. 6.9). Пользователь может выбрать единственную строку текста, нажав на нее.



Рис. 6.9. Стандартный список в Android

Раскрывающийся список - повсеместно встречающийся вариант обычного списка. Он показывает лишь выбранный элемент в одну строку, но если нажать на стрелку, открываются другие варианты выбора.

Элемент управления **представление в виде списка** предоставляет возможность сопровождать каждую строку текста пиктограммой. Такая возможность весьма полезна - существует множество ситуаций, когда можно упростить работу пользователя, располагая графические идентификаторы рядом со строками важных вариантов выбора (рис. 6.10).



Рис. 6.10. Элемент управления типа "список" с пиктограммами в Android-приложении, позволяющий визуально оценивать погоду в различных городах

Комбо-списки и комбо-кнопки

Комбо-элементы представляют собой сочетание элементов. Комбо-кнопка - разновидность радиокнопки со значком. Обычно она выглядит как кнопка-значок с небольшой стрелкой, но если нажать на стрелку и удерживать ее в нажатом состоянии, разворачивается меню.

Комбо-список представляет собой сочетание списка и поля редактирования (рис. 6.11).

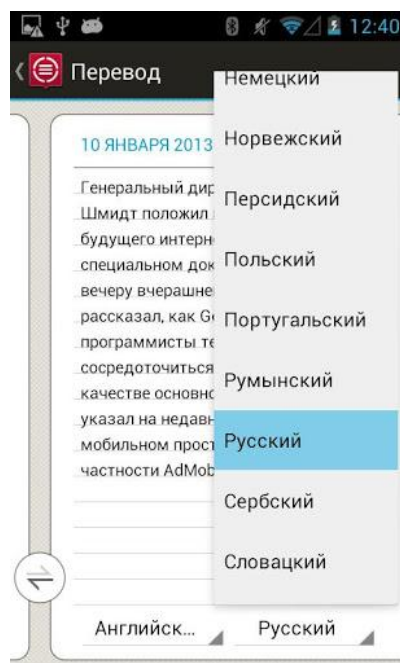


Рис. 6.11. Раскрывающееся поле со списком в Android-приложении ABBYY Translator позволяет выбрать нужный язык из списка

Вариант с раскрывающимся списком значительно экономит экранное пространство. Комбо-список хорошо подходит для тех случаев, когда необходимо организовать выбор единственного объекта.

Элементы ввода

Элементы ввода дают пользователю возможность не только выбирать существующие сведения, но и вводить новую информацию. Самый простой элемент - поле редактирования текста (поле ввода). В эту категорию попадают также такие элементы управления, как счетчики и ползунки.

Ограничивающие элементы ввода

Любой элемент управления, ограничивающий набор значений, доступных для ввода пользователем, является ограничивающим элементом ввода. Так, например, ползунок со шкалой значений от 0 до 100 является ограничивающим элементом ввода. Независимо от действий пользователя не может быть введено число, выходящее за диапазон определенных программой значений. Проще говоря, ограничивающие элементы ввода должны использоваться везде, где необходимо ограничить множество допустимых значений.

Ограничивающий элемент ввода должен четко информировать пользователя о допустимых границах. Текстовое поле, которое отвергает ввод пользователя после того, как он выполнил ввод, не может считаться ограничивающим элементом управления. Если пользователь должен выразить выбор числовым значением в определенных границах, предоставьте ему элемент управления, сообщающий об этих границах и предотвращающий ввод недопустимых значений. Такую возможность дает ползунок. Ползунок позволяет пользователю определять числовые значения в относительных терминах, а не в результате непосредственного ввода с клавиатуры. Но для ввода точных значений лучше подходят счетчики.

Счетчики

Счетчик состоит из небольшого поля ввода и двух прикрепленных к нему кнопок (рис.6.12). Благодаря счетчикам грань между ограничивающими и неограничивающими

элементами ввода данных становится размытой. Маленькие кнопки со стрелками позволяют пользователю изменять значение в поле редактирования небольшими шагами. Эти шаги могут выполняться до определенного предела: значение не может превысить максимум, установленный программой, или стать меньше установленного минимума. Если пользователь пожелает ввести определенное число, он может сделать это за счет прямого ввода числа в поле редактирования.

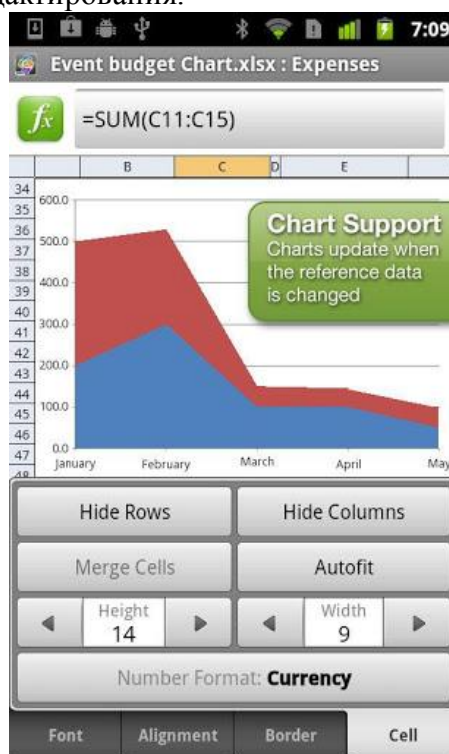


Рис. 6.12. Реализация счетчиков в Android-приложении Quickoffice

Рукоятки и ползунки

Рукоятки и ползунки очень эффективно расходуют экранное пространство, и оба этих элемента управления замечательно справляются с задачей обеспечения визуальной обратной связи по настройкам (рис. 6.13). Ползунки и рукоятки применяются в основном в качестве ограничивающих элементов управления ввода. Например, ползунки - превосходное средство для действий, связанных с масштабированием.



Рис. 6.13. Ползунок в Android-приложении Winamp осуществляет перемотку воспроизводимой композиции

Неограничивающие элементы ввода

Пожалуй, главный неограничивающий элемент ввода - поле ввода текста. Этот простейший элемент управления позволяет пользователям набирать любые алфавитно-цифровые строки. Как правило, поля ввода - это небольшие области, внутри которых можно набрать одно-два слова, но они могут быть реализованы и в виде довольно сложных текстовых редакторов.

Когда пользователю предложено неограничивающее текстовое поле ввода, которое при этом принимает лишь строки определенного формата, вероятно, имеется необходимость помочь пользователям вводить "допустимые" строки. Имеется множество стандартных форматов вводимых данных - даты, телефонные номера, почтовые индексы, номера социального страхования. Ключ к успешному проектированию элемента ввода с проверкой данных - в хорошо развитой обратной связи с пользователем.

Элементы управления отображением

Элементы управления отображением используются для управления визуальным представлением информации на экране. Типичными примерами элементов отображения являются разделители и полосы прокрутки. Сюда же входят разделители страниц, линейки, направляющие, сетки и рамки.

Текстовые элементы

Вероятно, самый простой элемент управления отображением - элемент вывода текстовой информации, который отображает текстовое сообщение в некоторой позиции на экране. Он предоставляет текстовые метки для других элементов управления и выводит данные, которые не могут или не должны быть изменены пользователем. Единственная серьезная проблема этого элемента состоит в том, что он зачастую используется там, где должны присутствовать элементы ввода (и наоборот).

Полосы прокрутки

Полосы прокрутки служат важной цели - они позволяют осмысленным образом помещать большие объемы информации внутри рамок окон и панелей. К сожалению, они

расходуют экранное пространство и ими сложно манипулировать. Однако замечательное преимущество полосы прокрутки состоит в создании контекста текущего положения в окне. Бегунок полосы прокрутки указывает текущее положение и нередко масштаб "территории", доступной для прокрутки.

Разделители

Разделители - удобный инструмент для деления главного окна приложения на несколько связанных между собой панелей, в каждой из которых можно просматривать, изменять или переносить ту или иную информацию. Подвижные разделители всегда должны сообщать о своей подвижности посредством изменения формы курсора. Однако следует проявлять осторожность, выбирая, какие именно разделители должны стать подвижными. В общем случае разделитель не должен перемещаться таким образом, чтобы содержимое панели становилось непригодным к использованию.

Выдвижные панели

Выдвижные панели - это панели приложения, которые можно открывать и закрывать в одно действие. Выдвижные панели - замечательное место для элементов управления и функций, которые используются совместно с основной рабочей областью приложения, но не столь часто. Выдвижные панели более удобны, чем диалоговые окна, так как не закрывают основное окно (рис. 5.14).



Рис. 6.14. Скриншоты популярной игры "Cut the Rope". Новостная панель появляется, если потянуть за кольцо

6.3. Применение событий к кнопкам

Когда с кнопкой ничего не происходит, она находится в не нажатом состоянии. Если на кнопке находится фокус ввода, она переходит в состояние `state_focused`. Если

нажать на кнопку пальцем, она будет находиться в состоянии `state_pressed`, пока мы не отпустим палец. Обработчик события `OnClickListener` определяет нажата ли кнопка. Существует небольшое отличие между обработкой событий нажатий на кнопки Yes (Да) и No (Нет). Давайте рассмотрим метод `OnClickListener.onClick()` для элемента `Button`, представляющего кнопку Yes (Да):

```
Button yesButton = (Button) findViewById(R.id.Button_Yes);
yesButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        handleAnswerAndShowNextQuestion(true);
    }
});
```

Метод `View.OnClickListener()` для элемента `Button`, представляющего кнопку No (Нет), практически идентичен обработчику событий нажатий на кнопки Yes (Да), показанному выше. Единственное отличие заключается в том, что в метод `handleAnswerAndShowNextQuestion()` передается значение `false`. Это пользовательский метод, который вы реализуете для обработки промежуточных результатов игры и управления любой логикой, связанной с элементами `ViewSwitcher`.

Контрольные вопросы:

1. Какие инструменты используются для создания приложений в системе Андроид?
2. Какие способы размещения кнопок, текстовых полей и других инструментов существуют?
3. Применение событий к кнопкам.