

Мобильное приложение – специально разработанное программное обеспечение, предназначенное для использования на мобильных устройствах.

За короткое время мобильные приложения стали очень востребованы. Выполнение программ в мобильных устройствах осуществляется с помощью программного обеспечения, куда входят операционная система и ключевые приложения. Программное обеспечение для работы мобильных устройств называют мобильной платформой.

Наиболее распространенные платформы – Android и iOS. Android-платформа основана на ядре ОС Linux, поэтому имеет открытый исходный код, который можно использовать для создания собственных приложений. Платформа поддерживается и развивается компанией Google.

iOS –платформа использует программный код, разработанный компанией Apple для своих планшетных компьютеров и смартфонов.

В настоящее время наибольшей функциональностью обладают мобильные приложения, использующие ресурсы операционной системы Android. Поэтому рассмотрим процесс разработки мобильного приложения на базе этой операционной системы.



Процесс разработки мобильных приложений состоит из следующих шагов:

1. Определение назначения мобильного приложения;
2. Определение платформы, на котором оно будет использоваться;
3. Разработка прототипа пользовательского интерфейса;

Пользовательский интерфейс – форма взаимодействия пользователя с приложением. Он должен быть удобным и легко понимаемым для пользователя.

4. Разработка дизайна мобильного приложения;
Отрисовываются различные состояния для всех сценариев использования
5. Разработка первой версии приложения;
6. Тестирование мобильного приложения и его доработка;
Мобильное приложение проходит тщательную проверку, в ходе которой выявляются все ошибки и несоответствия и дорабатывается приложение
7. Разработка логотипа
Графический элемент, который станет «лицом» приложения
8. Публикация мобильного приложения
Мобильное приложение размещается в App Store или Google Play

Наблюдение и изучение поведения пользователей говорит о том, что люди плохо воспринимают медленные и перегруженные сложным дизайном программы. Пользователи не хотят ждать полной загрузки данных, а также терять время на длительное изучение навигации по разделам. Пользователи хотят начать работать сразу после установки приложения и понять его функциональность в течение первых двух секунд после запуска. Если же этого не происходит, они просто скачивают другой, более удобный инструмент. Чтобы избежать этого еще на этапе разработки мобильного приложения необходимо помнить о простых правилах позволяющих создать успешное мобильное приложение.

Правила успешности мобильного приложения:

- **Эффективность использования.** Определяется скоростью и удобством посетителей при использовании вашего приложения для выполнения задач.
- **Обработка ошибок.** Насколько часто пользователи совершают ошибки при работе с мобильным приложением, уровень серьезности этих ошибок и возможность их исправления.
- **Простота обучения.** Насколько быстро пользователи смогут освоиться с выполнением простых задач и действий, если видят интерфейс мобильного приложения впервые.
- **Запоминаемость.** Определяется степенью, в которой пользователи смогут запомнить принципы интерфейса и навигации по разделам приложения, и не придется ли им при повторном запуске изучать его заново.
- **Субъективное восприятие.** Насколько пользователям «нравится» пребывание и работа с приложением.

Виды мобильных приложений:

Нативные приложения – это приложения, разработанные на “родном” для платформы языке. Они пишутся на языках программирования, утвержденных разработчиками программного обеспечения, под каждую конкретную платформу. Приложения загружаются через магазины приложений и устанавливаются в ПО смартфона.

Веб-приложения — это мобильная версия сайта, только с расширенным интерактивом, которое создается с использованием технологии HTML5. Такие приложения не требуют установки, а работают в браузере телефона. Поэтому их написание не привязано к конкретной платформе, а само приложение не может использовать ПО смартфона.

Гибридные мобильные приложения — это кроссплатформенные приложения, которые имеют возможность работать с ПО телефона. По сути, это сочетание некоторых функций нативного и веб-приложения. Качество и возможности гибридных приложений зависят от ПО, которым пользовался разработчик.

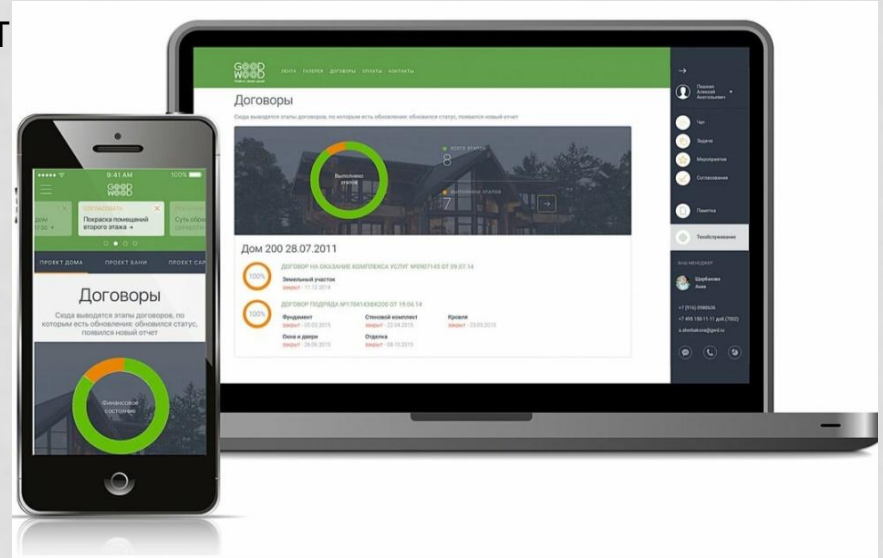
Преимущества и недостатки мобильных приложений:

Вид	Преимущества	Недостатки
Нативное	<ul style="list-style-type: none">• максимальная функциональность и скорость работы• не требуется интернет-соединение для использования• имеет доступ к ПО смартфона (gps, плеер, камера)• распространение через магазины приложений	<ul style="list-style-type: none">• выше стоимость и длиннее сроки разработки• требует от разработчика знаний определенной среды программирования• работает с одной платформой• при косметических изменениях необходимо обновление
Веб (HTML5)	<ul style="list-style-type: none">• кросс-платформенность• не требует загрузки из магазина мобильных приложений• можно легко адаптировать обычный сайт• легче найти веб-разработчика, чем разработчика под определенную платформу• простота создания и поддержки	<ul style="list-style-type: none">• требует подключения к интернету• не имеет доступа к ПО• не может отправлять push-уведомления• должен быть запущен интернет-браузер• при продаже требуется использование своей платежной системы
Гибридное	<ul style="list-style-type: none">• функциональность нативного приложения на независимой платформе• запускается не из браузера, в отличие от веб-приложения• возможность независимого обновления• распространение через магазины приложений	<ul style="list-style-type: none">• загружается из магазина мобильных приложений (необходимо соответствовать требованиям)• разработчик должен быть знаком с разными API

По способу создания приложений они делятся на два вида:

- Генераторы.

Такие платформы создают мобильное приложение на основе вашей существующей web-страницы. Вы даете генератору URL своего сайта, и он автоматически создает мобильное приложение с теми же разделами и контентом, что и у вас на сайте.



- Конструкторы.

Современные конструкторы, не используя языки программирования, на основе выбранных шаблонов и стилей оформления позволяют создавать приложения, мало отличающиеся от разработанных профессионалами



Для создания мобильных приложений используют **конструктор мобильных приложений** – ПО, с помощью которого можно создать свое приложение. Это может быть специальный сайт, на котором вы сами можете собрать свое приложение из готовых элементов. Также может использоваться интегрированная среда разработки.

Конструктор мобильных приложений

Конструкторы мобильных приложений появились относительно недавно. Разработчики ПО поняли, что программы для смартфонов очень нужны предпринимателям, чтобы продавать свои товары и услуги, но не все из них могут сделать приложение сами. Так появились платформы-конструкторы, которые дают любому желающему возможность самостоятельно «склеить» мобильное приложение, как коллаж из фотографий.

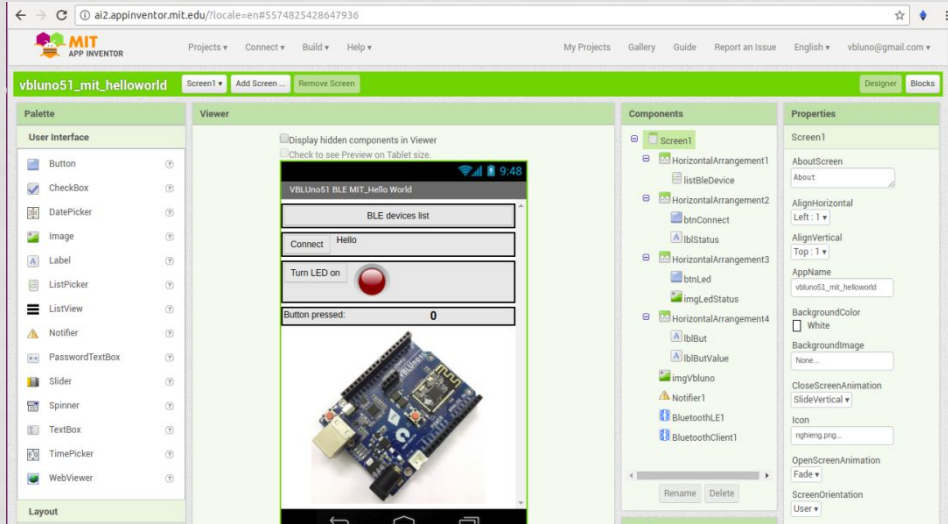
Платформы для создания мобильных приложений различаются между собой набором функций, ценами, способами разработки, то есть тем, как с их помощью можно сделать приложение.

Перед тем, как приступить к созданию Android-приложения, нужно выбрать инструментарий.

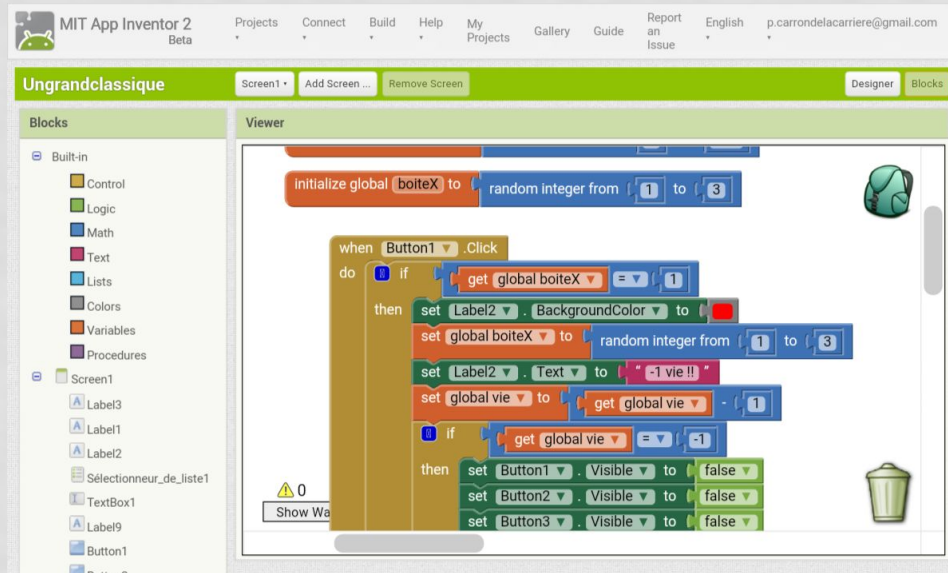
Одним из доступных конструкторов для создания Android-приложений является **MIT App Inventor** – облачная среда визуальной разработки. Для работы в среде необходимы знания основ алгоритмизации и наличие Google-аккаунта.

The image shows a web browser window with several tabs open, including 'Welcome to App Inventor 2!' and 'MIT App Inventor'. The address bar shows 'ai2.appinventor.mit.edu'. The page header includes the MIT App Inventor logo and navigation links like 'Projects', 'Connect', 'Build', 'Settings', and 'Help'. Below the header is a green bar with buttons for 'Start new project', 'Move To Trash', 'View Trash', 'Login to Gallery', and 'Publish to Gallery'. The main content area features a 'Projects' table with columns for 'Name', 'Date Created', and 'Date Modified'. A large green and white modal window is displayed in the center, titled 'Welcome to MIT App Inventor'. It contains a graphic of the Android robot with a gear, and a code block showing a 'join' block connected to three text blocks: 'MIT App Inventor', 'New', and 'Gallery!'. Below the graphic, it says 'Release nb186 is out. This includes our new MIT App Inventor Gallery.' with a link to 'Release Notes'. At the bottom of the modal, there is a link to 'Set up and connect an Android device' and a 'Continue' button (highlighted with a red box) and a checkbox for 'Do Not Show Again'.

Разработка мобильного приложения происходит в два этапа:



1 этап: Проектирование интерфейса пользователя, т.е. «Как это будет выглядеть»



2 этап: Программирование компонентов приложения, т.е. «Как компоненты будут себя вести?»

1-й этап: создание интерфейса приложения.

В режиме **Дизайнер** разместить на модели экрана мобильного устройства нужные компоненты



Настроить свойства компонентов с помощью панели свойств



2-й этап: создание программного кода.

В режиме **Блоки** создать код для каждого компонента, т.е. его реакцию на события, происходящие действия



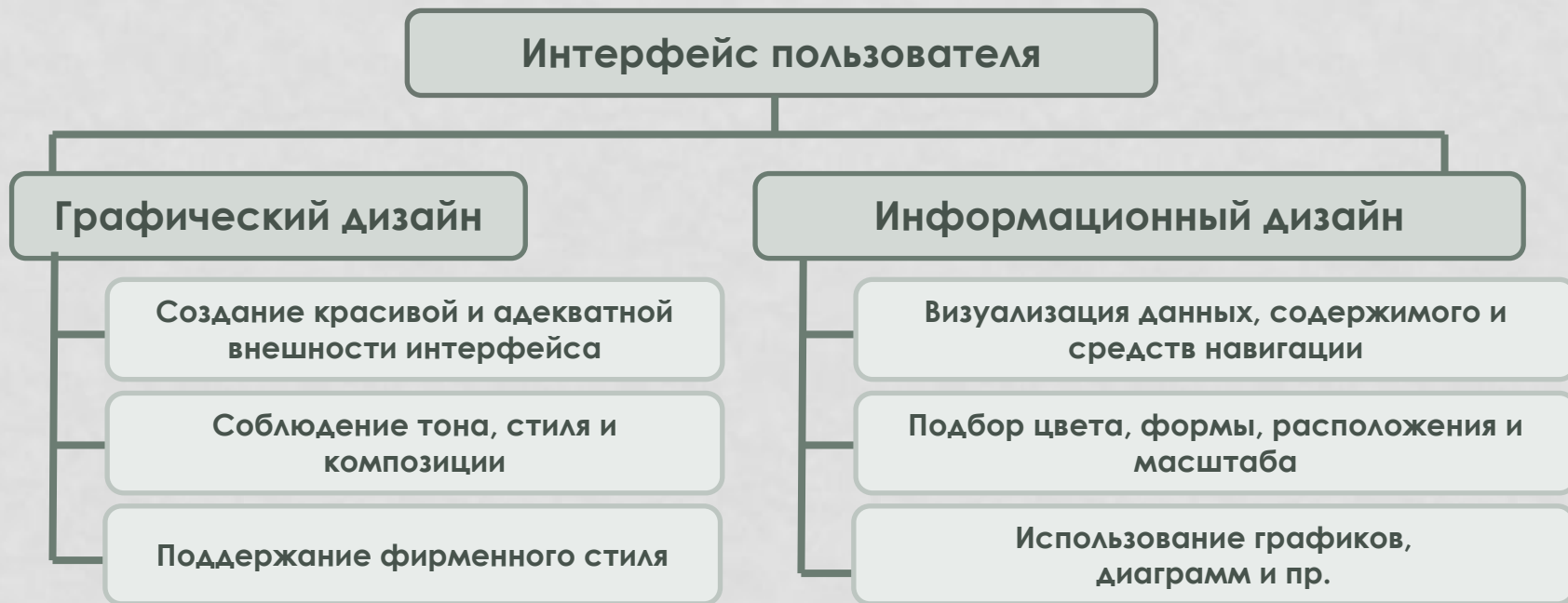
Каждый компонент имеет свои команды, для активизации нужных блоков команд необходимо выделить компонент

Этапы реализуются в отдельных окнах среды MIT App Inventor .
Освоение программы мы начнем с проектирования и создания интерфейса приложения

1 этап: Проектирование интерфейса пользователя,

Разработка удобного интерфейса для мобильных приложений является довольно сложным процессом основой которого являются:

1. Визуальный дизайн интерфейсов – очень нужная и уникальная часть создания приложения, это создание интерфейса способного серьезно повлиять на эффективность и привлекательность продукта, на наилучшее представление, доносящее информацию о поведении программы. Состоит он из 2-х видов:



Чтобы создать полезный и привлекательный пользовательский интерфейс, следует тщательно проработать визуальные свойства всех используемых в приложении объектов (компонентов).

Строительные блоки визуального дизайна интерфейсов: форма, размер, цвет, яркость, направление, текстура, расположение.

2. Элементы управления – это доступные для манипулирования самодостаточные экранные объекты, посредством которых люди взаимодействуют с цифровыми продуктами. Они являются базовыми строительными блоками графического пользовательского интерфейса.

Классификация элементов управления:

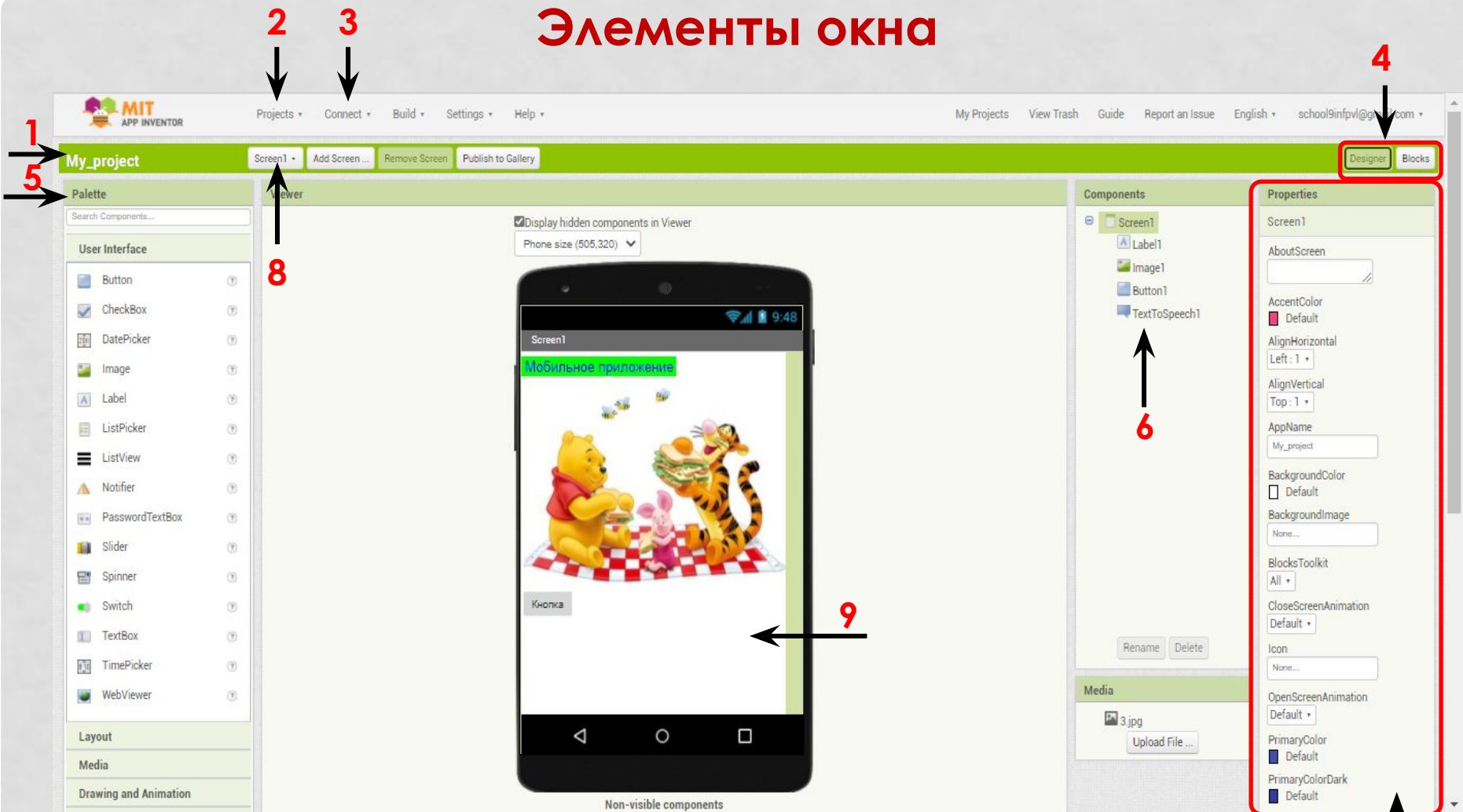
- командные элементы управления, применяемые для выполнения функций;
Командные элементы управления выполняют действия, причем делают это немедленно. Главным командным элементом является кнопка, которая обладает множеством вариантов отображения.
- элементы выбора, позволяющие выбирать данные или настройки;
Элементы выбора позволяют пользователю выбрать из группы допустимых объектов тот, с которым будет совершено действие. Они применяются также для действий по настройке компонентов. Распространенными элементами выбора являются флажки и списки.
- элементы ввода, применяемые для ввода данных;
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.
- элементы отображения, используемые для наглядного непосредственного манипулирования.
Элементы управления отображением используются для управления визуальным

Когда платформа Android только появилась, не было никаких рекомендаций по разработке дизайна, поэтому все разработчики проектировали внешний вид приложений по своему вкусу. Отсутствие единого стиля сказалось на интерфейсах не лучшим образом. В настоящее время существует стандарт Android Design.

Рекомендации по дизайну:

- Реальные объекты гораздо веселее, чем кнопки и меню. Позвольте людям манипулировать знакомыми вещами! Тогда работа будет эффективнее.
- Картинки работают быстрее, чем слова.
- Используйте короткие фразы, состоящие из простых слов.
- Никогда не теряйте пользовательскую информацию. Велика вероятность того, что он откажется использовать ваше приложение.
- Если объекты похожи, они должны выполнять сходные действия.
- Показывайте только то, что необходимо пользователю именно в этот момент.
- Выводите пользователю сообщения, только если вопрос действительно важен.
- Делайте важные вещи быстро.
- Разбивайте сложные задачи на несколько простых шагов.
- Будьте вежливы и корректны в общении с пользователем.
- Пользователь всегда должен быть уверен в том, что он знает, где сейчас находится. На любом шаге он должен иметь возможность вернуться назад.
- Используйте интерфейсные элементы, которые будут работать в любой ситуации.
- Самый главный принцип - НЕ УСЛОЖНЯЙТЕ пользователю жизнь!

Элементы окна



1. Имя проекта;

2. Project - меню работы с проектами;

3. Connect - выбор способа запуска созданного приложения;

4. Designer/Blocks - переключение режимов создания проекта;

5. Palette - палитра

6. Components - компоненты

7. Properties - панель свойств компонента

8. Screen1 - переключение между экранами

9. Экран приложения (просмотр)

Режим «**Designer**» – режим, в котором создается интерфейс приложения. Данный режим используют для выбора и размещения различных компонентов приложения, которые отображаются на экране вашего устройства при запуске приложения. Интерфейс для разработки дизайна проекта состоит из следующих основных элементов:

Palette (Палитра) включает наборы (группы) компонентов будущего приложения

Components (Компоненты) – список компонентов, которые вы уже используете в своем проекте. Компонентами называются функциональные элементы приложения, такие как кнопки, изображения, текст, надпись, флажок, список и др. Некоторые компоненты являются невидимы на экране.

Properties (Панель свойств) – в этой части экрана устанавливаются значения свойств компонентов вашего приложения, например, цвет, размер шрифта, источники изображений и звуков, надписи, первоначальное значение.

Screen1 (Переключение между экранами). Например, на первом экране у вас инструкция к приложению, а на втором экране, собственно, приложение – его функциональная часть.

Просмотр – экран вашего приложения. В приложении можно использовать несколько экранов, где будут производиться различные действия.

Назначение некоторых компонентов:



Button

Компонент Button (Кнопка) – командная кнопка, с помощью которой пользователь может вызывать выполнение какого-либо действия.



CheckBox

Компонент CheckBox (Флажок) предоставляет пользователю два варианта выбора – его можно установить или снять. Установленный флажок отмечен галочкой.



Label

Компонент Label (Надпись) используется для вывода на форму текста, который пользователь не может изменить во время выполнения программы.



Image

Компонент Image (Рисунок) предназначен для вывода содержимого графических файлов в форме



TextBox

Компонент управления TextBox (текст) предназначен для ввода текста в режиме выполнения программы (одной или нескольких строк).



TextToSpeech

Компонент TextToSpeech (Текст в речь) предназначен для преобразования текста в речь.