

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ

УНИВЕРСИТЕТ»

Кафедра информатики
и информационных технологий

Б1.В.10 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Лабораторные работы. Технология разработки мобильного приложения

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам и самостоятельной работе

Направление подготовки

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Квалификация (степень) выпускника

бакалавр

Уфа 2021

Рекомендовано к изданию методической комиссией экономического
факультета (25 марта 2021 г., протокол № 8)

Составитель: доцент, к.ф.-м.н. Шамсутдинова Т.М.

Рецензент: ст. преподаватель Прокофьева С.В.

Ответственный за выпуск: зав. кафедрой ИИТ, д.т.н.,
Беляева А.С.

г.Уфа, БГАУ, Кафедра информатики и информационных
технологий

Содержание

Лабораторная работа №1 Технология разработки и сборки мобильного приложения

Лабораторная работа №2 Особенности использования механизмов платформы

Лабораторная работа №3 Использование мобильного функционала

Лабораторная работа №4 Особенности разработки интерфейса

Лабораторная работа №5 Программирование логики и работа с базами данных

Лабораторная работа №6 Разработка мобильного клиента для сервиса

Лабораторная работа №7 Разработка облачного сервиса

Список литературы

Лабораторная работа № 1. Технология разработки и сборки мобильного приложения

Цель работы: разработка простого приложения, помогающего понять структуру приложения, освоить основные операторы, освоить среду разработки

Теоретическая часть

Приступая к разработке мобильных приложений, нужно иметь представление о том, какие виды приложений существуют. Дело в том, что если удастся определить к какому типу относится приложение, то становится понятнее на какие моменты в процессе его разработки необходимо обращать основное внимание. Можно выделить следующие виды приложений:

- Приложения переднего плана выполняют свои функции только, когда видимы на экране, в противном же случае их выполнение приостанавливается. Такими приложениями являются, например, игры, текстовые редакторы, видеопроигрыватели. При разработке таких приложений необходимо очень внимательно изучить жизненный цикл активности, чтобы переключения в фоновый режим и обратно проходили гладко (бесшовно), т. е. при возвращении приложения на передний план было незаметно, что оно вообще куда-то пропадало. Для достижения этой гладкости необходимо следить за тем, чтобы при входе в фоновый режим приложение сохраняло свое состояние, а при выходе на передний план восстанавливало его. Еще один важный момент, на который обязательно надо обратить внимание при разработке приложений переднего плана, удобный и интуитивно понятный интерфейс.

- Фоновые приложения после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии. Примерами таких приложений могут служить, службы экранирования звонков, SMSa-ответчики. В большинстве своем фоновые приложения нацелены на отслеживание событий, порождаемых аппаратным обеспечением, системой или другими приложениями, работают незаметно. Можно создавать совершенно невидимые сервисы, но тогда они будут неуправляемыми. Минимум действий, которые необходимо позволить пользователю: санкционирование запуска сервиса, настройка, приостановка и прерывание его работы при необходимости.

- Смешанные приложения большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки. Обычно взаимодействие с пользователем сводится к уведомлению о каких-либо событиях. Примерами таких приложений могут служить мультимедиа-проигрыватели, программы для обмена текстовыми сообщениями (чаты), почтовые клиенты. Возможность реагировать на пользовательский ввод и при этом не терять работоспособности в фоновом режиме является характерной особенностью смешанных приложений. Такие приложения обычно содержат как видимые активности, так и скрытые (фоновые) сервисы, и при взаимодействии с пользователем должны учитывать свое текущее состояние.

Возможно потребуется обновлять графический интерфейс, если приложение находится на переднем плане, или же посылать пользователю уведомления из фонового режима, чтобы держать его в курсе происходящего. И эти особенности необходимо учитывать при разработке подобных приложений.

– Виджеты - небольшие приложения, отображаемые в виде графического объекта на рабочем столе. Примерами могут служить, приложения для отображения динамической информации, такой как заряд батареи, прогноз погоды, дата и время. Разумеется, сложные приложения могут содержать элементы каждого из рассмотренных видов. Планируя разработку приложения, необходимо определить способ его использования, только после этого приступать к проектированию и непосредственно разработке.

Обратим внимание на организацию исполнения приложений в ОС Android. Как уже было отмечено приложения под Android разрабатываются на языке программирования Java, компилируется в файл с расширением .apk, после этот файл используется для установки приложения на устройства, работающие под управлением Android. После установки каждое Android приложение "живет" в своей собственной безопасной "песочнице", рассмотрим, как это выглядит:

– операционная система Android является многопользовательской ОС, в которой каждое приложение рассматривается как отдельный пользователь;

– по умолчанию, система назначает каждому приложению уникальный пользовательский ID, который используется только системой и неизвестен приложению;

– система устанавливает права доступа ко всем файлам приложения следующим образом: доступ к элементам приложения имеет только пользователь с соответствующим ID;

– каждому приложению соответствует отдельный Linux процесс, который запускается, как только это необходимо хотя бы одному компоненту приложения, процесс прекращает работу, когда ни один компонент приложения не использует его или же системе требуется освободить память для других (возможно, более важных) приложений;

– каждому процессу соответствует отдельный экземпляр виртуальной машины Dalvik, в связи с этим код приложения исполняется изолировано от других приложений.

Перечисленные идеи функционирования приложения в ОС Android реализуют принцип минимальных привилегий, т. е. каждому приложению, по умолчанию, разрешен доступ только к компонентам, необходимым для его работы и никаким больше. Таким образом обеспечивается очень безопасная среда функционирования приложений.

Однако, в случае необходимости приложения могут получить доступ к данным других приложений и системным сервисам (услугам). В случае, когда двум приложениям необходимо иметь доступ к файлам друг друга, им присваивается один и тот же пользовательский ID. Для экономии системных ресурсов такие приложения запускаются в одном Linux процессе и делят

между собой один и тот же экземпляр виртуальной машины, в этом случае приложения также должны быть подписаны одним сертификатом. В случае же, когда приложению требуется доступ к системным данным, например, контактам, SMS сообщениям, картам памяти, камере, Bluetooth и т. д., пользователю необходимо дать приложению такие полномочия во время установки его на устройство.

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и Android SDK.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Создать новое приложение и изучить его структуру.
3. Настроить интерфейс приложения;
4. Реализовать логику приложения.
5. Ответить на контрольные вопросы.
6. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Каково устройство платформы Android?
2. Что представляет собой Android SDK?
3. Назовите основные средства разработки под Android.
4. Перечислите достоинства и недостатки эмуляторов Android.
5. Выясните объем продаж мобильных устройств с ОС Android.
6. Какая версия платформы наиболее популярна в настоящее время?

Лабораторная работа № 2. Особенности использования механизмов платформы

Цель работы: разработка приложения в 1С: Предприятие

Теоретическая часть

Для создания конфигурации в 1С: Предприятие под мобильное приложение необходимо выполнить настройку конфигуратора. Для этого нужно зайти в меню «Сервис», «Параметры» (рисунок 1).

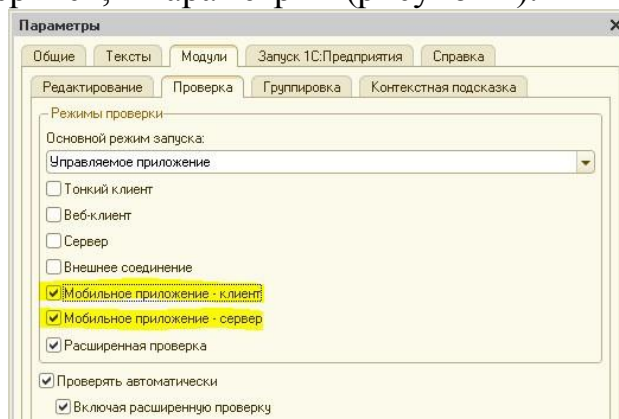


Рисунок 1 — Окно «Параметры»

После этого необходимо указать Конфигуратору, что данная конфигурация разрабатывается только для мобильного приложения.

Нужно вызывать контекстное меню корневого узла конфигурации, в свойстве «Назначение использования», поставить флаг «Мобильное устройство», и снять флаг «Персональный компьютер» (рисунок 2).

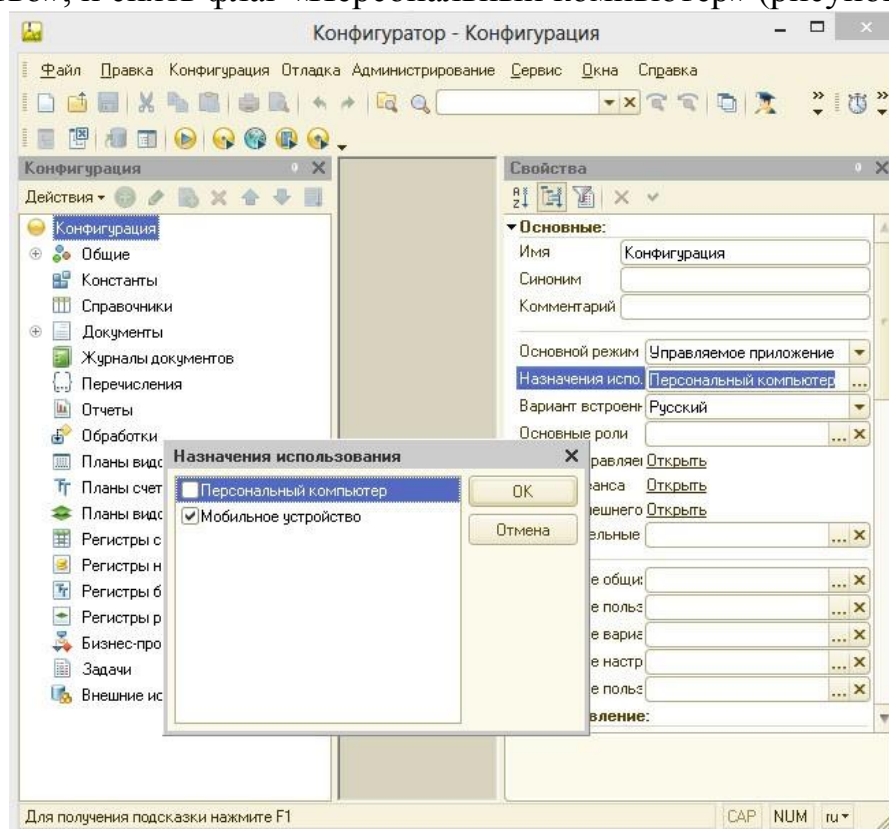


Рисунок 2 — Окно «Назначение использования»

После нажатия «ОК» дерево метаданных изменит свой вид. В нем появятся недоступные объекты. Это объекты, с которыми мобильное приложение работать не может.

Необходимо подготовить две конфигурации: стационарную и мобильную.

Объекты конфигурации – это составные элементы, из которых складывается любое прикладное решение.

Они представляют собой настраиваемые объекты (объекты с настраиваемыми свойствами), поддерживаемые на уровне технологической платформы. По большому счету задача разработчика заключается в том, чтобы собрать из этих объектов, как из конструктора, необходимую структуру прикладного решения и затем описать специфические алгоритмы функционирования и взаимодействия этих объектов, отличающиеся от их типового поведения.

Подсистема. С помощью подсистем осуществляется визуальное разделение прикладного решения на крупные, и мелкие функциональные блоки. Подсистемы составляют основу для формирования командного интерфейса программы.

Команда. Команды – это действия, которые может выполнить пользователь. Помимо команд, формируемых платформой, разработчик может создавать собственные команды, принадлежащие всему прикладному решению или отдельным объектам конфигурации.

Справочник. Справочники служат для описания таких сущностей как товары, контрагенты, валюты, склады и пр. Все эти сущности имеют общие свойства: внутренняя идентификация объекта в системе, необходимость поддержки иерархии и группировки элементов, необходимость поддержки вложенных таблиц и т.д.

Документ, Журнал документов. Служат для описания таких сущностей как счета, накладные, заказы и пр. Эти сущности фиксируют различные события, происходящие в жизни организации, они привязаны ко времени, содержат вложенные таблицы, должны отражаться в учетных механизмах и т.д.

План обмена. Планы обмена предназначены для описания структуры распределенной информационной системы и задания перечня данных, которыми будет производиться обмен в пределах этой распределенной системы. Позволяет создавать территориально распределенные информационные системы как на основе информационных баз «1С: Предприятия», так и с использованием произвольных информационных систем, не основанных на «1С: Предприятии».

Стационарная и мобильная конфигурации используют план обмена «Обмен данными с мобильным устройством» для синхронизации данных. Он обеспечивает обмен справочниками и документами между конфигурациями, а также изменение данных уже существующих записей.

Для передачи вложений создан периодический регистр сведений «При

крепленные данные», с периодичностью – в течение секунды, и создано измерение «Ссылка на объект», которое имеет ссылочный тип документа списания, и также ресурс «Данные», с типом Хранилище значений.

После передачи данных в регистре сведений «Прикрепленные данные» появляется запись. Чтобы просмотреть фото нужно открыть запись и нажать на кнопку «Открыть вложение», которая находится на командной панели.

Взаимодействие между информационной базой и мобильным приложением осуществляется с помощью веб-сервиса.

Для нашего мобильного приложения понадобится веб-сервер для обмена данными.

Веб-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

В нашем случае, все это будем делать через 1С, которая, в свою очередь, будет настраивать этот сервер. Так как 1С сама по себе не может выступать в роли веб сервиса, то нам необходимо настроить или IIS или Apache.

Будем использовать вариант с Apache (рисунок 3), так как он является легким в настройке и кроссплатформенным.

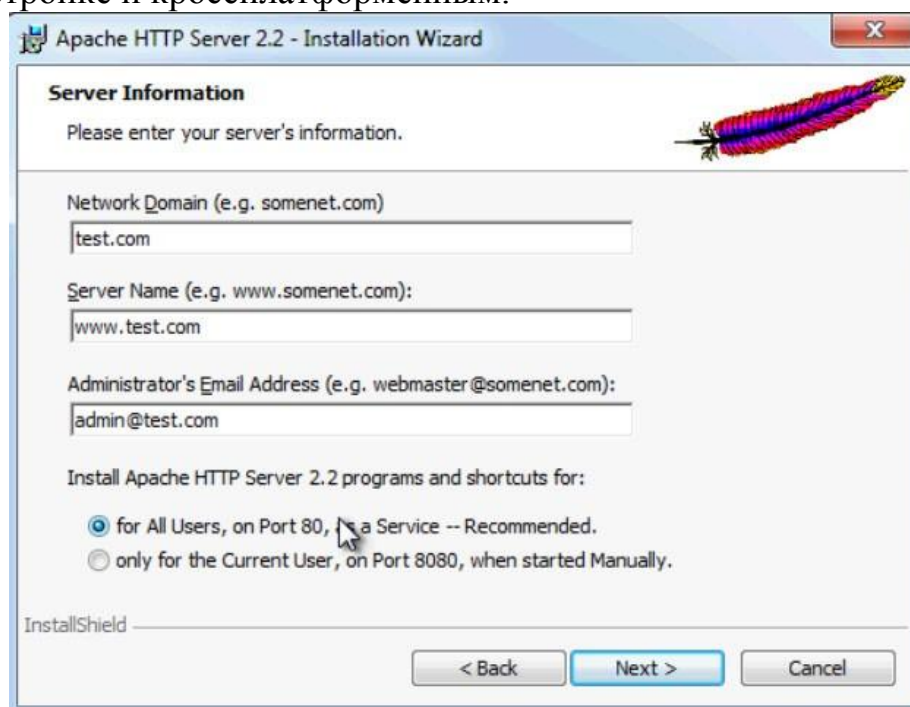


Рисунок 3 — Окно установки «Apache»

Программа запрашивает порт, который она будет прослушивать данные сервера.

Проверить работоспособность apache можно, введя в адресной строке браузера адрес 127.0.0.1 или localhost.

Для конфигурации, разрабатываемой как мобильное приложение, следует устанавливать значение «Мобильное устройство» в свойство «Назначение использования» (рисунок 4). В этом случае система автомати-

чески скроет возможности, недоступные для мобильной платформы, а рабочие инструменты (проверка синтаксиса, проверка конфигурации и пр.) будут настроены на работу именно с тем контекстом встроенного языка, который доступен в мобильной платформе.

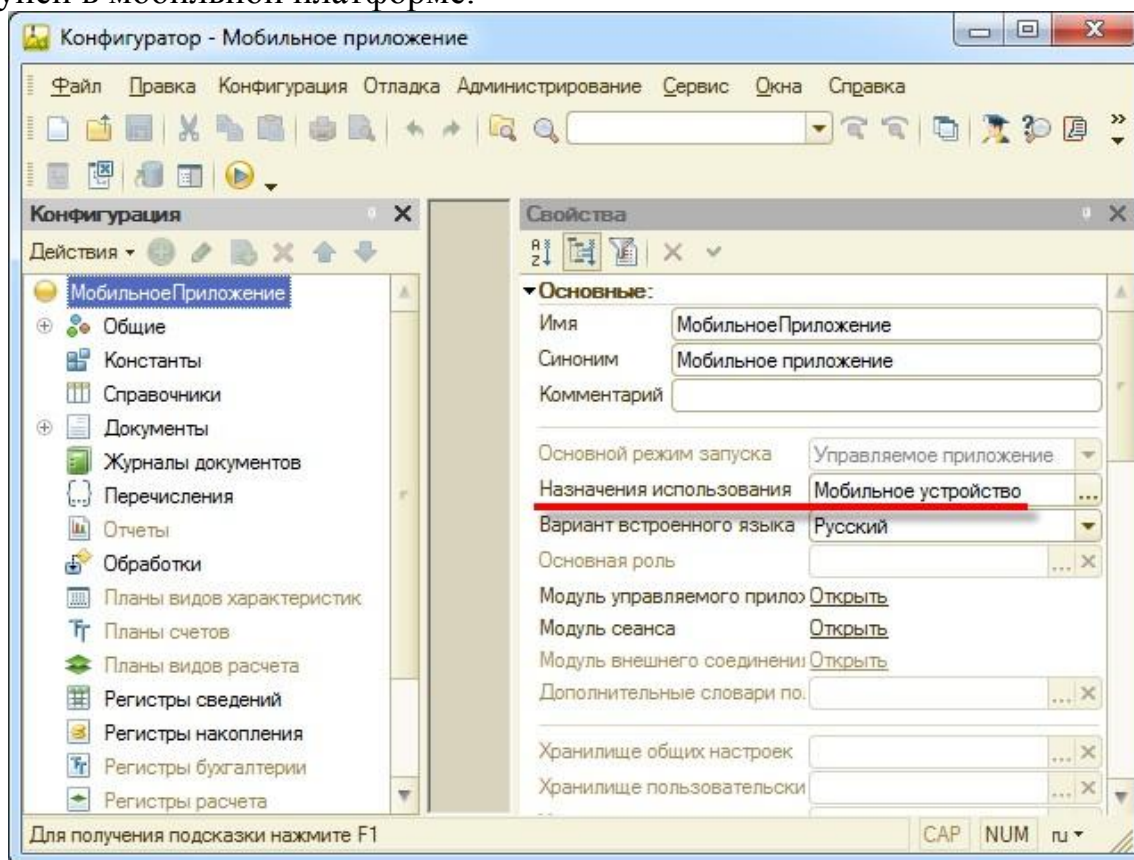


Рисунок 4 — Окно свойств конфигурации

Чтобы была возможность переноса конфигурации на телефон, нужно опубликовать конфигурацию. В данном случае мобильное приложение подключается к серверу и забирает конфигурацию. Этот вариант предполагает наличие доступа к компьютеру из мобильного приложения через сеть.

Для того чтобы опубликовать конфигурацию нужно зайти в раздел «Конфигурация» в главном меню, выбрать «Публиковать» в пункте «Мо-бильное приложение». Затем заполнить данные и нажать кнопку «Опубликовать» (рисунок 5).

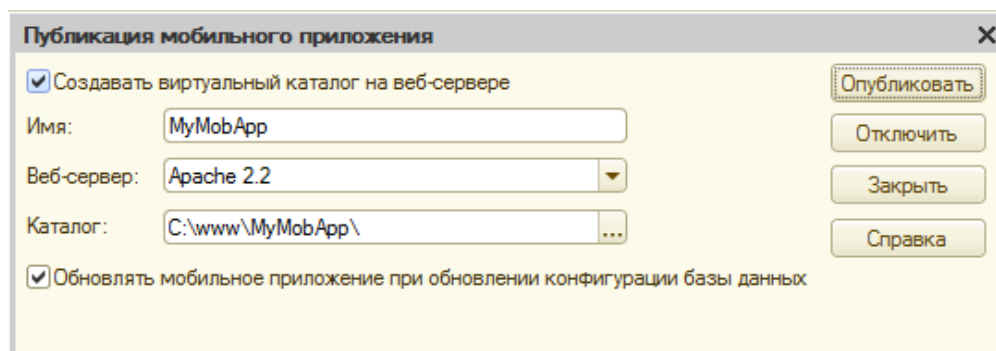


Рисунок 5 — Окно «Публикация мобильного приложения»

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной платформой 1С: Предприятие.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Разработать мобильное приложение для установленной платформы 1С: Предприятие.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Опишите этапы создания конфигурации в 1С: Предприятие под мобильное приложение .
2. Как указать в Конфигураторе, что данная конфигурация разрабатывается только для мобильного приложения.
3. Как опубликовать мобильное приложение.

Лабораторная работа № 3. Использование мобильного функционала

Цель работы: научиться создавать приложения, состоящие из нескольких активностей, и диалоговые окна, а также познакомиться с элементами тач-интерфейса.

Теоретическая часть

Для мобильных приложений главным ограничением является размер экрана устройства. Очень часто невозможно разместить все элементы полнофункционального приложения так, чтобы их можно было увидеть одновременно. Очевидным решением этой проблемы является разделение интерфейса на части по какому-либо принципу. Основные пути решения этой проблемы:

- Использовать различные сообщения (диалоговые окна, уведомления, всплывающие подсказки). Этот способ наиболее прост и не требует редактирования файла манифеста, однако очевидно, что так можно решить только часть задач.
- Использовать в одном приложении несколько активностей. Способ универсальный и подходит для любых приложений, однако прежде чем его реализовывать, необходимо очень хорошо продумать структуру будущего приложения. Здесь требуется редактировать манифест и организовать переключение между различными активностями удобным для пользователя способом.
- Разместить компоненты на активности таким образом, что в нужный момент можно будет легко переключиться на работу с другой частью интерфейса.

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и Android SDK.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Подумайте над собственным приложением, сочетающим различные возможности проектирования многооконных приложений, рассмотренные выше. Создайте прототип этого приложения и настройте его пользовательский интерфейс.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть

выполнен в редакторе MS Word и оформлен согласно требованиям

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Использование класса Dialog
2. Варианты отображения уведомлений.
3. Назначение всплывающих подсказок.
4. Особенности разработки приложения, содержащего несколько активностей

Лабораторная работа № 4. Особенности разработки интерфейса

Цель работы: изучение основ разработки интерфейсов мобильных приложений.

Теоретическая часть

1. Визуальный дизайн интерфейсов.

Силы, вложенные в разработку модели поведения программного продукта, будут потрачены впустую, если вы не сумеете должным образом донести до пользователей принципы этого поведения. В случае мобильных продуктов это делается визуальными средствами путем отображения объектов на дисплее (в некоторых случаях целесообразно использовать тактильные ощущения от нажатия).

Визуальный дизайн интерфейсов очень нужная и уникальная дисциплина, которую следует применять в сочетании с проектированием взаимодействия и промышленным дизайном. Она способна серьезно повлиять на эффективность и привлекательность продукта, но для полной реализации этого потенциала нужно не откладывать визуальный дизайн на потом, а сделать его одним из основных инструментов удовлетворения потребностей пользователей и бизнеса.

1.1. Изобразительное искусство, визуальный дизайн интерфейсов и прочие дисциплины дизайна

Художники и визуальные дизайнеры работают с одними и теми же изобразительными средствами, однако их деятельность служит различным целям. Цель художника создать объект, взгляд на который вызывает эстетический отклик. Изобразительное искусство способ самовыражения художника. Художник не связан почти никакими ограничениями. Чем необычнее и своеобразнее продукт его усилий, тем выше он ценится.

Дизайнеры создают объекты, которыми будут пользоваться другие люди. Если говорить о дизайнерах визуальных интерфейсов, то они ищут наилучшее представление, доносящее информацию о поведении программы, в проектировании которой они принимают участие. Придерживаясь целеориентированного подхода, они должны стремиться представлять поведение и информацию в понятном и полезном виде, который поддерживает

маркетинговые цели организации и эмоциональные цели персонажей. Разумеется, визуальный дизайн пользовательских интерфейсов не исключает эстетических соображений, но такие соображения не должны выходить за рамки функционального каркаса.

1.2. Графический дизайн и пользовательские интерфейсы

Графические дизайнеры обычно очень хорошо разбираются в визуальных аспектах и хуже представляют себе понятия, лежащие в основе поведения программного продукта и взаимодействия с ним. Они способны создавать красивую и адекватную внешность интерфейсов, а кроме того приносить фирменный стиль во внешний вид и поведение программного

продукта. Для таких специалистов дизайн или проектирование интерфейса есть в первую очередь тон, стиль, композиция, которые являются атрибутами бренда, во вторую очередь прозрачность и понятность информации и лишь затем передача информации о поведении посредством ожидаемого назначения.

Дизайнерам визуальной части интерфейса необходимы некоторые навыки, которые присущи графическим дизайнерам, но они должны еще обладать глубоким пониманием и правильным восприятием роли поведения. Их усилия в значительной степени сосредоточены на организационных аспектах проектирования. В центре их внимания находится соответствие между визуальной структурой интерфейса с одной стороны и логической структурой пользовательской ментальной модели и поведения программы с другой. Кроме того, их заботит вопрос о том, как сообщать пользователю о состояниях программы и что делать с когнитивными аспектами пользовательского восприятия функций.

1.3. Визуальный информационный дизайн

Информационные дизайнеры работают над визуализацией данных, содержимого и средств навигации. Усилия информационного дизайнера направлены на то, чтобы представить данные в форме, способствующей их верному истолкованию. Результат достигается через управление визуальной иерархией при помощи таких средств, как цвет, форма, расположение и масштаб. Распространенными объектами информационного дизайна являются всевозможные графики, диаграммы и прочие способы отображения количественной информации.

Чтобы создавать привлекательные и удобные пользовательские интерфейсы, дизайнер интерфейсов должен владеть базовыми визуальными навыками пониманием цвета, типографики, формы и композиции и знать, как их можно эффективно применять для передачи поведения и представления информации, для создания настроения и стимулирования физиологических реакций. Дизайнеру интерфейса также требуется глубокое понимание принципов взаимодействия и идиом интерфейса, определяющих поведение продукта.

2. Строительные блоки визуального дизайна интерфейсов

Дизайн интерфейсов сводится к вопросу о том, как оформить и расположить визуальные элементы таким образом, чтобы внятно отразить поведение и представить информацию. Каждый элемент 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 элемента или группы

элементов. Чтобы создать полезный и привлекательный пользовательский интерфейс, следует тщательно поработать с каждым из этих свойств.

2.1 Форма

Форма главный признак сущности объекта для человека. Мы узнаем объекты по контурам. 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.

2.2 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.

2.3 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.

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.

2.4 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.

2.5 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.

2.6 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.

2.7 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).

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и Android SDK.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

3. Изучить рекомендуемую литературу.
4. Изучить элементы интерфейса.
5. Практическим путём научиться размещать элементы и менять их свойства.
6. Разработать прототип интерфейса собственного приложения.
7. Ответить на контрольные вопросы.
8. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В

конце отчеты приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Особенности визуального дизайна интерфейсов, строительных блоках и элементах управления.
2. Особенности проектирования GUI под Android.
3. Принципы разработки удобных пользовательских интерфейсов для мобильных приложений.

Лабораторная работа № 5. Программирование логики и работа с базами данных

Цель работы: разработка Android приложения, демонстрирующего возможности работы с базой данных SQLite.

Теоретическая часть

SQLite небольшая и при этом мощная система управления базами данных. Эта система создана в 2000 году, ее разработчик доктор Ричард Хипп (Dr. Richard Hipp). В настоящее время является одной из самых распространенных SQL-систем управления базами данных в мире. Можно выделить несколько причин такой популярности SQLite: она бесплатная; она маленькая, примерно 150 Кбайт; не требует установки и администрирования. Подробнее см. <http://www.sqlite.org>.

База данных SQLite это обычный файл, его можно перемещать и копировать на другую систему (например, с телефона на рабочий компьютер) и она будет отлично работать.

В Android имеется встроенная поддержка одной из распространенных систем управления базами данных - SQLite. Для этого в пакете `android.database.sqlite` определен набор классов, которые позволяют работать с базами данных SQLite. И каждое приложение может создать свою базу данных.

Чтобы использовать SQLite в Android, надо создать базу данных с помощью выражение на языке SQL. После этого база данных будет храниться в каталоге приложения по пути:

DATA/data/[Название_приложения]/databases/[Название_файла_базы_данных]

ОС Android по умолчанию уже содержит ряд встроенных баз SQLite, которые используются стандартными программами - для списка контактов, для хранения фотографий с камеры, музыкальных альбомов и т.д.

Основную функциональность по работе с базами данных предоставляет пакет `android.database`. Функциональность непосредственно для работы с SQLite находится в пакете `android.database.sqlite`.

База данных в SQLite представлена классом `android.database.sqlite.SQLiteDatabase`. Он позволяет выполнять запросы к бд, выполнять с ней различные манипуляции.

Класс `android.database.sqlite.SQLiteCursor` предоставляет запрос и позволяет возвращать набор строк, которые соответствуют этому запросу.

Класс `android.database.sqlite.SQLiteQueryBuilder` позволяет создавать SQL-запросы.

Сами sql-выражения представлены классом `android.database.sqlite.SQLiteStatement`, которые позволяют с помощью плейсхолдеров вставлять в выражения динамические данные.

Класс `android.database.sqlite.SQLiteOpenHelper` позволяет создать базу данных со всеми таблицами, если их еще не существует.

В SQLite применяется следующая система типов данных:

- INTEGER: представляет целое число, аналог типу int в java
- REAL: представляет число с плавающей точкой, аналог float и double в java
- TEXT: представляет набор символов, аналог String и char в java
- BLOB: представляет массив бинарных данных, например, изображение, аналог типу int в java

Сохраняемые данные должны представлять соответствующие типы в java.

Создание и открытие базы данных

Для создания или открытия новой базы данных из кода Activity в Android мы можем вызвать метод `openOrCreateDatabase()`. Этот метод может принимать три параметра:

- название для базы данных
- числовое значение, которое определяет режим работы (как правило, в виде константы `MODE_PRIVATE`)
- необязательный параметр в виде объекта `SQLiteDatabase.CursorFactory`, который представляет фабрику создания курсора для работы с бд

Например, создание базы данных `app.db`:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
```

Для выполнения запроса к базе данных можно использовать метод `execSQL` класса `SQLiteDatabase`. В этот метод передается SQL-выражение. Например, создание в базе данных таблицы `users`:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS users (name TEXT, age
INTEGER);");
```

Если нам надо не просто выполнить выражение, но и получить из бд какие-либо данные, то используется метод `rawQuery()`. Этот метод в качестве параметра принимает SQL-выражение, а также набор значений для выражения `sql`. Например, получение всех объектов из базы данных:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS users (name TEXT, age
INTEGER);");
Cursor query = db.rawQuery("SELECT * FROM users;", null);
if(query.moveToFirst){

    String name = query.getString(0);
    int age = query.getInt(1);
}
```

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и Android SDK.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Разработать приложение, демонстрирующее возможности работы с базой данных: создание, добавление записей, просмотр записей, удаление базы данных.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним.

Контрольные вопросы:

1. Основы работы с базами данных, SQLite.
2. Каким методом можно открыть базу данных для использования в мобильном приложении?
3. Основные принципы разработки приложений для смартфонов с использованием базы данных.

Лабораторная работа № 6. Разработка мобильного клиента для сервиса

Цель работы: освоить технологию работы с форматами JSON и XML и организации мобильного клиента для сервиса.

Теоретическая часть

Данные, которые передаются между веб-сервисом и клиентским приложением, кодируют с использованием определенных форматов. Существует множество таких форматов. Например, в ответ на запрос загрузки веб-страницы с веб-сервера могут поступить данные в формате HTML. При работе с API веб-сервисов обычно используются другие форматы, среди них наиболее распространены JSON (Java Script Object Notation) и XML (eXtensible Markup Language). Эти форматы могут использоваться и в задачах, которые не связаны с веб-службами. Например, в XML или JSON могут быть сериализованы объекты (при условии возможности сериализации), состояние которых нужно сохранить в постоянной памяти и загрузить, десериализовав, при очередном запуске приложения.

Как мы увидим ниже, JSON и XML имеют определенные особенности, которые можно учитывать, принимая решение об использовании того или иного формата. С их помощью можно закодировать одни и те же данные, существенная разница между ними будет заключаться в размере полученных данных и в удобстве восприятия данных человеком (нужно обычно при отладке). Ниже мы рассмотрим эти характеристики на практике.

С вопросами передачи и хранения данных тесно связана тема сжатия данных. Сжатие позволяет особым образом обработать данные, получив их представление, имеющее меньший размер, чем исходные данные, но содержащее ту же информацию. В данном случае речь идёт о так называемом сжатии информации без потерь, такое сжатие применимо, например, к документам (или к любым другим данным), когда критически важно точное соответствие данных, хранящихся в исходном документе тем данным, которые получены из сжатой копии документа. Среди алгоритмов сжатия без потерь можно отметить, например, алгоритм .ZIP.

Существует и так называемое сжатие с потерями – оно используется в тех случаях, когда отбрасывание некоторой части информации не ухудшает (или ухудшает незначительно) возможности по работе с информацией. Взамен потери некоторой части информации мы получаем очень большие уровни сжатия. Такое сжатие используется для кодирования изображений (.JPG), звуковых файлов (.MP3), видеофайлов.

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Подумайте, как ваше приложение может воспользоваться возможностями сериализации и десериализации в организации его работы. Оцените возможную выгоду от использования возможностей сжатия данных при организации хранения данных вашего приложения. На основе примера, приведенного в лабораторной работе, создайте приложение, которое позволяет оценить скорость сериализации объекта, объем данных которого можно регулировать в диапазоне 10 – 50 Мб, и десериализации данных в такой объект. Используйте в эксперименте форматы JSON и XML. Для оценки времени можно воспользоваться возможностями получения текущего времени в начале и в конце операции.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Работа с JSON, XML.
2. Технология сжатия данных.
3. Выбор формата передачи данных позволяющий создать приложение, эффективно использующее системные ресурсы.
4. Затраты вычислительных ресурсов при сжатии данных.

Лабораторная работа № 7. Разработка облачного сервиса

Цель работы: освоить технологию работы с классами `webClient` и `HttpWebRequest`.

Теоретическая часть

Для работы с веб-сервисами из приложений обычно используют классы `System.Net.WebClient`

(<http://msdn.microsoft.com/en-us/library/windowsphone/develop/system.net.webclient%28v=vs.105%29.aspx>)

и `System.Net.HttpWebRequest`

(<http://msdn.microsoft.com/en-us/library/windowsphone/develop/system.net.httpwebrequest%28v=vs.105%29.aspx>).

И тот и другой класс позволяют достигать схожих целей, однако, между ними есть некоторые отличия, которые и определяют выбор того или иного класса для использования в конкретном приложении.

Так, обычно `WebClient` применяют тогда, когда нужно получить какие-либо данные из Интернета, он обеспечивает простую и удобную работу с GET/POST запросами. То есть, если наша задача – это получение, например, RSS-ленты с сервиса, или отправка файла с использованием POST-запроса, возможностей `WebClient` для этого вполне хватит. Кроме того, его использование позволит упростить код. Его возможностей хватит и для выполнения многих других действий, но более сложные сценарии взаимодействия с веб-службами обычно реализуют с использованием `HttpWebRequest`. Этот класс, в частности, нужен там, где предполагается использование PUT/DELETE запросов, он предоставляет больший уровень контроля над параметрами запроса. Например, если речь идёт об отправке файла на веб-сервер, то обычно для этого нам понадобится `HttpWebRequest`. При работе над конкретным проектом стоит ознакомиться с наборами инструментов, доступных в пространстве имен `System.Net` (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/btdf6a7e%28v=vs.105%29.aspx>).

Можно сказать, в итоге, что для выполнения простых задач использование `WebClient` позволяет упростить их решение, при прочих равных условиях `HttpWebRequest` потребует более сложных программных конструкций, больше настроек. А при выполнении задач более сложных, требующих более полного контроля над процессом взаимодействия с веб-службой, некоторая усложнённость использования `HttpWebRequest` вполне оправдана.

Оборудование и материалы: для выполнения данной лабораторной работы необходим компьютер с установленной операционной системой Windows 7/10 и программными продуктами: MS Word, Adobe Reader, и Visual Studio 2012.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.
2. Если приложение, над которым вы работаете, подразумевает работу с каким-либо веб-сервисом, получение данных из Интернета, подумайте над тем, какие из рассмотренных механизмов вы сможете в них использовать. В частности, исходя из круга задач, которые ваше приложение будет решать с использованием интернет-сервисов аргументируйте использование в нём таких средств, как классы `HttpWebRequest` или `WebClient`, задача вызова веб-браузера из приложения, элемент управления, который позволяет встраивать веб-браузер в страницу приложения.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям..

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись и ссылку на него в тексте.

Контрольные вопросы:

1. Назначение `HttpWebRequest` и `WebClient`.
2. Методы асинхронного выполнения задач.
3. Средства выполнения асинхронных вызовов.

Список литературы

- 1) Хрусталева, Е. Ю. Знакомство с разработкой мобильных приложений на платформе "1С:Предприятие 8" [Текст] / Е. Ю. Хрусталева издание 2 – Москва: ООО «1С-Публишинг», 2015
- 2) Майер, Р. Android 4. Программирование приложений для планшетных компьютеров и смартфонов. / Пер. с англ. – М.: Эксмо, 2013. – 816 с.
- 3) Харди, Б. Программирование под Android / Харди Б., Филлипс Б. / Пер. с англ. –СПб.: Питер, 2014. – 592 с.
- 4) Лутай С. Windows 8 для C# разработчиков/ С. Лутай, С. Байдачный. – Издательство: Самоиздат, 2012. – 278 с.
- 5) Пугачев, С. Разработка приложений для Windows 8 на языке C# / С. Пугачев, А.Шериев, К. Кичинский. – БХВ-Петербург, 2013. – 416 с.