

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Гарбар Олег Викторович

Должность: Заместитель директора по учебно-воспитательной работе

Дата подписания: 29.10.2021 14:01:28

Уникальный программный ключ:

5769a34aba1fca5ccbf44edc23bf8f457c6d4fb4

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

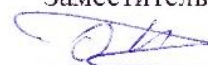
Индустриальный институт (филиал)

**Федерального государственного бюджетного образовательного учреждения
высшего образования «Югорский государственный университет»**

(ИндИ (филиал) ФГБОУ ВО «ЮГУ»)

УТВЕРЖДАЮ

Заместитель директора по УВР



Гарбар О.В.

«09» сентября 2021 г.


**Методические указания
по выполнению практических работ
ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ
МДК 01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ
для специальности 09.02.07 Информационные системы и программирование**

РАССМОТРЕНО:

Предметной цикловой

Комиссией специальных технических
дисциплин

Протокол №1 от 09.09.2021

 Шарипова И.А.

СОГЛАСОВАНО:

заседанием Методсовета

протокол №1 от 16.09.2021

Председатель Методсовета

 Н.И. Савватеева

Методические указания по выполнению практических работ по МДК 01.03. Разработка мобильных приложений разработаны в соответствии с рабочей программой профессионального модуля ПМ.01. Разработка модулей программного обеспечения для компьютерных систем по специальности 09.02.07 Информационные системы и программирование.

Разработчик: Игнатенко Е.С., преподаватель ИнДИ (филиала) ФГБОУ ВО «ЮГУ».

СОДЕРЖАНИЕ

Пояснительная записка.....	4
Перечень практических работ.....	6
Практическая работа № 3.1. Платформа Android.....	8
Практическая работа № 3.2. Java 2 MicroEdition.....	14
Практическая работа № 3.3. Протокол Bluetooth.....	17
Практическая работа № 3.4. Установка инструментария для разработки мобильных приложений	21
Практическая работа № 3.5. Настройка среды для разработки мобильных приложений.....	21
Практическая работа № 3.6. Установка среды разработки мобильных приложений с применением виртуальной машины.....	33
Практическая работа № 3.7. Создание эмуляторов. Подключение устройств. Настройка режима терминала.....	40
Практическая работа № 3.8. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна.....	42
Практическая работа № 3.9. Обработка событий: подсказки. Обработка событий: цветовая индикация. Подготовка стандартных модулей. Обработка событий: переключение между экранами.....	48
Практическая работа № 3.10. Передача данных между модулями. Тестирование мобильного приложения. Оптимизация мобильного приложения пошаговый алгоритм тестирования мобильных приложений.....	52
Практическая работа № 3.11. Создание тем для упрощения работы с элементами. Применение DDMS для отладки приложения. Создание лога. Списки. Работа с ориентацией экрана, применение различных layouts. Анимация. Рисование. Меню. Кнопки. Диалоговые окна. Сообщения. Мультимедиа. Shared preferences. БД SQLite. Интернет-соединение. Content providers. Работа с картами и GPS. Виджеты. Публикация приложения.....	55
Практическая работа № 3.12. Создание приложения, которое состоит из нескольких activities. Написание приложения, работающее с разными темами/стилями. Создание приложения, содержащее анимированные интерфейсные элементы.....	57
Практическая работа № 3.13. Создание приложения, отображающее после запуска карты Google или какие ни будь другие карты. Создание собственнй виджет с настройками. Создание приложение, использующее опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента.....	58
Практическая работа № 3.14. Создание первого приложения под Android. Основы разработки интерфейсов мобильных приложений	64
Практическая работа № 3.15. Создание многоэкранного приложения.....	67
Практическая работа № 3.16. Демонстрации распознавания стандартных жестов. Принципы работы с жестами вводимыми пользователями	68
Практическая работа № 3.17. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек	70
Практическая работа № 3.18. Работа с базами данных в Android. Основные приемы работы с инструментами разработки	71
Windows Phone SDK 8.0.....	71
Практическая работа № 3.19. Шаблоны проектов, структура проектов. Элементы управления. Разработка пользовательского элемента управления.....	72
Практическая работа № 3.20. Навигация в приложении. Обмен данными внутри приложения	74
Практическая работа № 3.21. Использование шаблона проектирования MVVM. Работа с JSON, XML, сжатие данных	76
Практическая работа № 3.22. Работа WebClient и HttpWebRequest.....	77
Практическая работа № 3.23. Работа с API веб-сервисов	78
Практическая работа № 3.24. Хранение данных на устройстве. Локальные базы данных.....	80
Практическая работа № 3.25. Разработка для Windows Azure	81
Практическая работа № 3.26. Сервисы Live Connect: SkyDrive	81
Практическая работа № 3.27. Многопоточное программирование.....	82
Практическая работа № 3.28. Сенсорный пользовательский интерфейс	83
Практическая работа № 3.29. Работа с датчиками, определение местоположения.....	84
Практическая работа № 3.30. Распознавание и синтез речи, работа с камерой.....	85
Список литературы	86

Пояснительная записка

Методические указания по выполнению лабораторных и практических работ по МДК 01.03. Разработка мобильных приложений разработаны в соответствии с рабочей программой профессионального модуля и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении профессионального модуля, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

МДК 01.03. Разработка мобильных приложений относится к профессиональному циклу, изучается на 4 курсе и при его изучении отводится значительное место выполнению практических работ.

Освоение содержания МДК 01.03. Разработка мобильных приложений во время выполнения практических работ обеспечивает достижение обучающимися следующих **результатов:**

Код	Наименование общих компетенций
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 4	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 5	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарта антикоррупционного поведения.
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 11	Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3.	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4.	Выполнять тестирование программных модулей
ПК 1.5.	Осуществлять рефакторинг и оптимизацию программного кода

В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	В разработке кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений
-------------------------	---

Уметь	осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства
Знать	основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов

В соответствии с рабочей программой по ПМ 01. Разработка модулей программного обеспечения для компьютерных систем, практические работы по МДК 01.03. Разработка мобильных приложений проводятся в седьмом семестре. Целесообразность данной группировки обусловлена необходимостью обобщения и систематизации знаний перед экзаменом.

Рабочая программа профессионального модуля предусматривает проведение практических работ МДК 01.03. Разработка мобильных приложений в объеме 48 часов.

Порядок выполнения практической и лабораторной работы

- записать название работы, ее цель в тетрадь;
- выполнить основные задания в соответствии с ходом работы;
- выполнить индивидуальные задания.

Рекомендации по оформлению практической и лабораторной работы

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе

Результаты выполнения практических работ необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

Критерии оценки практической и лабораторной работы

- наличие Цель выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие Цель выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие Цель выполняемой работы, выполнение всех основных и индивидуальных заданий (отлично).

Перечень практических работ

№	Наименование разделов и тем профессионального модуля (ПМ)	Наименование лабораторных работ и практических занятий	Объем часов
	Раздел 3. Разработка мобильных приложений		2
	МДК.01.03 Разработка мобильных приложений		2
1.	Тема 3.1. Мобильные приложения и технологии	Практическая работа № 3.1. Платформа Android	2
2.		Практическая работа № 3.2. Java 2 MicroEdition	2
3.		Практическая работа № 3.3. Протокол Bluetooth	2
4.		Практическая работа № 3.4. Установка инструментария для разработки мобильных приложений	2
5.		Практическая работа № 3.5. Настройка среды для разработки мобильных приложений	2
6.		Практическая работа № 3.6. Установка среды разработки мобильных приложений с применением виртуальной машины	2
7.	Тема 3.2 Создание и тестирование модулей для мобильных приложений	Практическая работа № 3.7. Создание эмуляторов. Подключение устройств. Настройка режима терминала	2
8.		Практическая работа № 3.8. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна	2
9.		Практическая работа № 3.9. Обработка событий: подсказки. Обработка событий: цветовая индикация. Подготовка стандартных модулей. Обработка событий: переключение между экранами	2
10.		Практическая работа № 3.10. Передача данных между модулями. Тестирование мобильного приложения. Оптимизация мобильного приложения	2
11.	Тема 3.3. Разработка мобильных приложений под Android в среде eclipse и другими способами.	Практическая работа № 3.11. Создание тем для упрощения работы с элементами. Применение DDMS для отладки приложения. Создание лога. Списки. Работа с ориентацией экрана, применение различных layouts. Анимация. Рисование. Меню. Кнопки. Диалоговые окна. Сообщения. Мультимедиа. Shared preferences. БД SQLite. Интернет-соединение. Content providers. Работа с картами и GPS. Виджеты. Публикация приложения.	2
12.	Тема 3.4. Разработка мобильных приложений под iOS.	Практическая работа № 3.12. Создание приложения, которое состоит из нескольких activities. Написание приложения, работающее с разными темами/стилями. Создание приложения, содержащего анимированные интерфейсные элементы	2
13.	Тема 3.5. Разработка мобильных приложений под Windows Phone.	Практическая работа № 3.13. Создание приложения, отображающее после запуска карты Google или какие-нибудь другие карты. Создание собственного виджета с настройками. Создание приложения, использующего опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента	2
14.		Практическая работа № 3.14. Создание первого приложения под Android. Основы разработки интерфейсов мобильных приложений	2

15.	Практическая работа № 3.15. Создание многоэкранного приложения	2
16.	Практическая работа № 3.16. Демонстрации распознавания стандартных жестов. Принципы работы с жестами вводимыми пользователями	2
17.	Практическая работа № 3.17. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек	2
18.	Практическая работа № 3.18. Работа с базами данных в Android. Основные приемы работы с инструментами разработки	2
19.	Практическая работа № 3.19. Шаблоны проектов, структура проектов. Элементы управления. Разработка пользовательского элемента управления	2
20.	Практическая работа № 3.20. Навигация в приложении. Обмен данными внутри приложения	2
21.	Практическая работа № 3.21. Использование шаблона проектирования MVVM. Работа с JSON, XML, сжатие данных	2
22.	Практическая работа № 3.22. Работа WebClient и HttpRequest	2
23.	Практическая работа № 3.23. Работа с API веб-сервисов	2
24.	Практическая работа № 3.24. Хранение данных на устройстве. Локальные базы данных	2
25.	Практическая работа № 3.25. Разработка для Windows Azure	2
26.	Практическая работа № 3.26. Сервисы Live Connect: SkyDrive	2
27.	Практическая работа № 3.27. Многопоточное программирование	2
28.	Практическая работа № 3.28. Сенсорный пользовательский интерфейс	2
29.	Практическая работа № 3.29. Работа с датчиками, определение местоположения	2
30.	Практическая работа № 3.30. Распознавание и синтез речи, работа с камерой	2
Итого		60

Практическая работа № 3.1. Платформа Android

Цель работы: Изучить основы работы с платформой Android

Начало работы с Android

Бурное развитие информационных технологий в последнее время привело к тому, что появилось много новых устройств и технологий, таких, как планшеты, смартфоны, нетбуки, другие гаджеты. Они все более прочно входят в нашу жизнь и становятся привычным делом. Лидирующей платформой среди подобных гаджетов на сегодняшний день является ОС Андроид.

Android используется на самых разных устройствах. Это и смартфоны, и планшеты, и телевизоры, и смарт-часы и ряд других гаджетов. По разным подсчетам за 2020 год этой операционной системой пользуются около 85% владельцев смартфонов, а общее количество пользователей смартфонов на ОС Android оценивается в более чем 2,5 млрд. человек по всему миру.

ОС Андроид была создана разработчиком Энди Рубином (Andy Rubin) в качестве операционной системы для мобильных телефонов и поначалу развивалась в рамках компании Android Inc. Но в 2005 году Google покупает Android Inc. и начинает развивать операционную систему с новой силой. Android постоянно эволюционирует, и вместе с операционной системой эволюционируют средства и инструменты для разработки. На данный момент (ноябрь 2020 года) последней версией является Android 11.0, которая вышла в сентябре 2020 года:

Таблица 1

Версия	Кодовое имя	Дата выпуска	Уровень API
11.0	11	8 сентября 2020	30
10.0	10	3 сентября 2019	29
9.0	Pie	6 августа 2018	28
8.1	Oreo	5 декабря 2017	27
8.0	Oreo	21 августа 2017	26
7.1	Nougat	4 октября 2016	25
7.0	Nougat	22 августа 2016	24
6.0	Marshmallow	5 октября 2015	23
5.1	Lollipop	9 марта 2015	22
5.0	Lollipop	3 ноября 2014	21
4.4	KitKat	31 октября 2013	19
4.3	Jelly Bean	24 июля 2013	18
4.2	Jelly Bean	13 ноября 2012	17
4.1	Jelly Bean	9 июля 2012	16
4.0	Ice Cream Sandwich	16 декабря 2011	15
2.3	Gingerbread	6 декабря 2010	10

Что нужно для разработки?

Стоит отметить, что разрабатывать приложения под Android можно с помощью различных фреймворков и языков программирования. Так, в качестве языков программирования могут применяться Java, Kotlin, Dart (фреймворк Flutter), C++, Python, C# (платформа Xamarin) и т.д. В данном руководстве мы будем использовать именно язык Java, как наиболее распространенный и используемый. Поэтому прежде чем приступить к освоению программирования под Android по данному руководству, необходимо освоить хотя бы базовые моменты языка Java.

Установка средств разработки

Существуют разные среды разработки для Android. Рекомендуемой средой разработки является Android Studio, которая создана специально для разработки под ОС Android. Поэтому мы ее и будем использовать. Загрузить файл установщика можно с официального сайта: <https://developer.android.com/studio>:

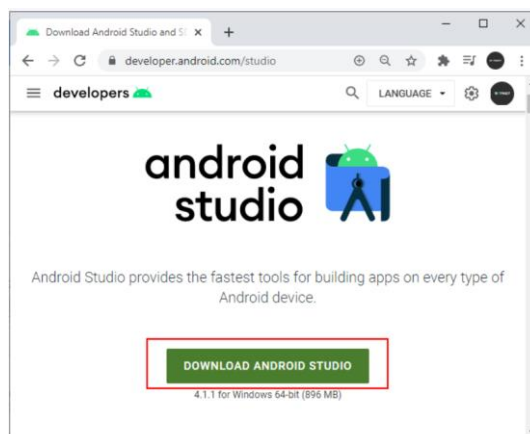


Рисунок 1

Кроме самой среды Android Studio для разработки также потребуется набор инструментов, который называется Android SDK. Например, если ранее Android SDK еще не было установлено, то при первом обращении к Android Studio она сообщит, что Android SDK отсутствует.

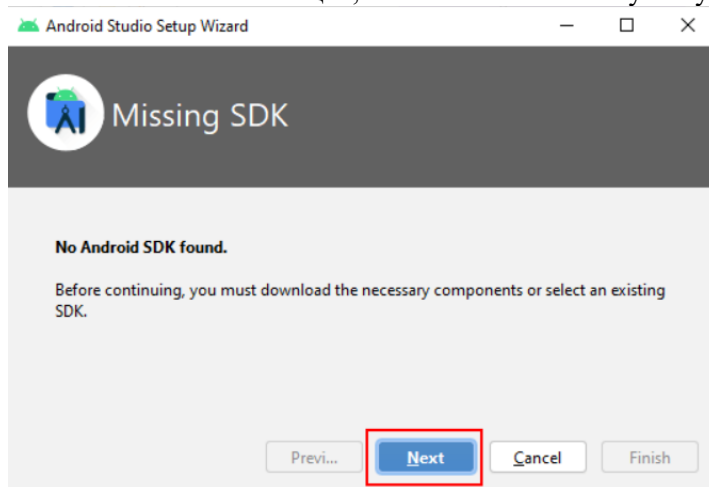


Рисунок 2

Мы можем отдельно вручную загрузить Android SDK с официального сайта и установить его. Либо мы можем сделать это непосредственно из Android Studio. Так, нажмем на кнопку Next. И на следующем экране нам будет предложено загрузить Android SDK для последней версии API (в данном случае для Android 11):

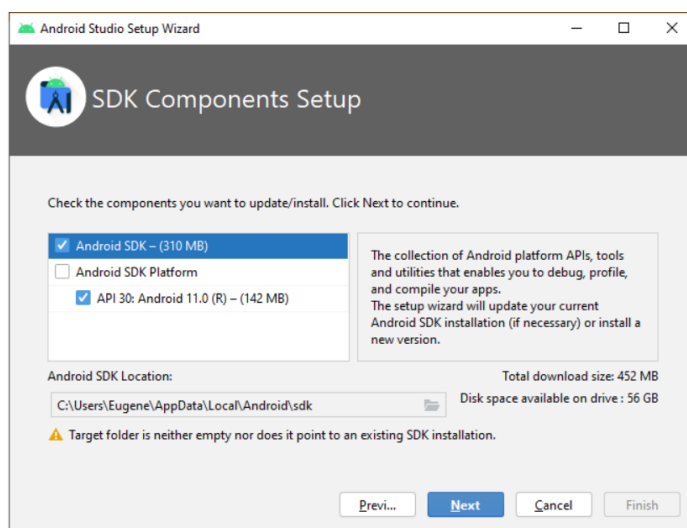


Рисунок 3

Здесь же мы можем указать место для установки Android SDK, если путь по умолчанию нас не устраивает.

Нажмем на кнопку Next, и далее нам отобразится окно со сводкой того, что именно будет установлено:

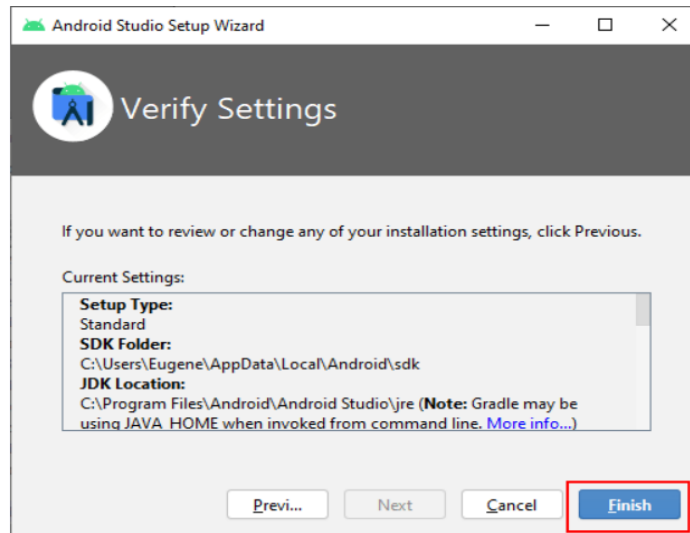


Рисунок 4

Нажмем на кнопку Finish, чтобы, наконец, все это установить.

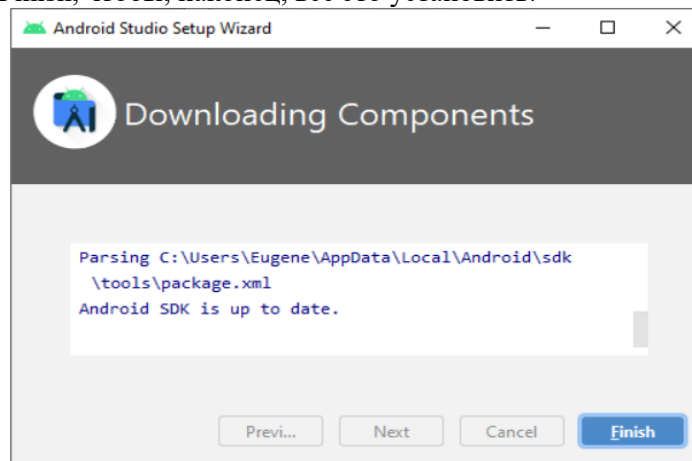


Рисунок 5

И после завершения установки нажмем на кнопку Finish. И мы можем приступить к созданию приложений.

Первый проект в Android Studio

Теперь создадим первое приложение в среде Android Studio для операционной системы Android. Откроем Android Studio и на начальном экране выберем пункт Create New Project:

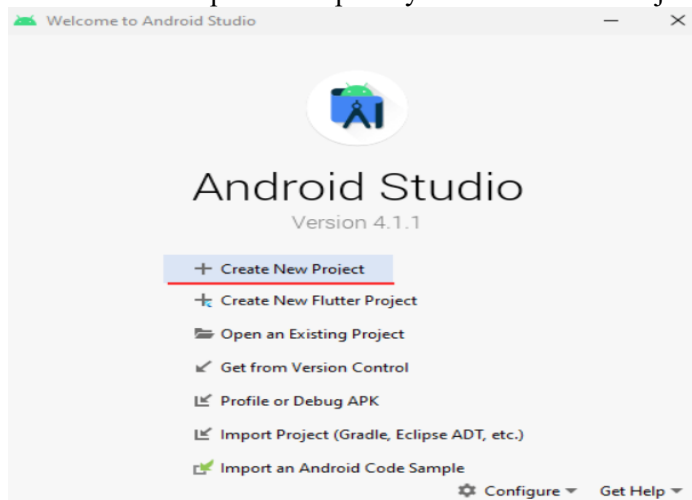


Рисунок 6

При создании проекта Android Studio вначале предложит нам выбрать шаблон проекта:

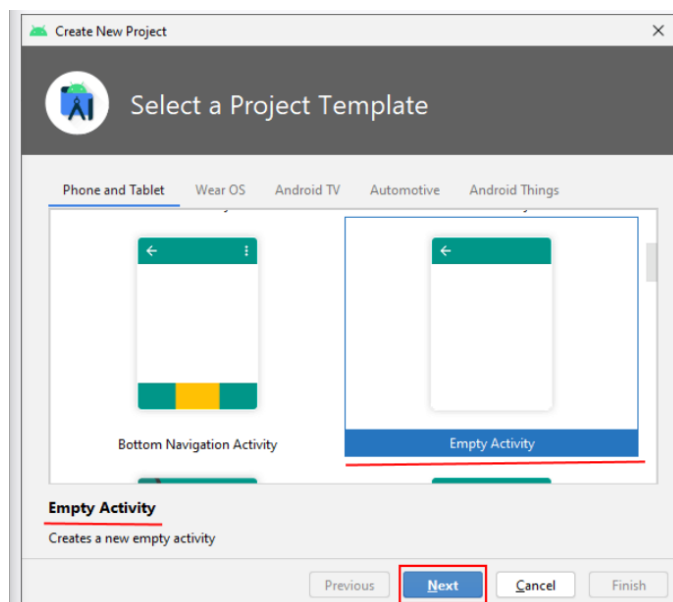


Рисунок 7

Android Studio предоставляет ряд шаблонов для различных ситуаций, но самыми распространенными являются Basic Activity и Empty Activity. Это самые удобные шаблоны для старта для создания большинства приложений. И по умолчанию выбран шаблон Empty Activity (если он не выбран, выберем его) и нажмем на кнопку Next.

После этого отобразится окно настроек нового проекта:

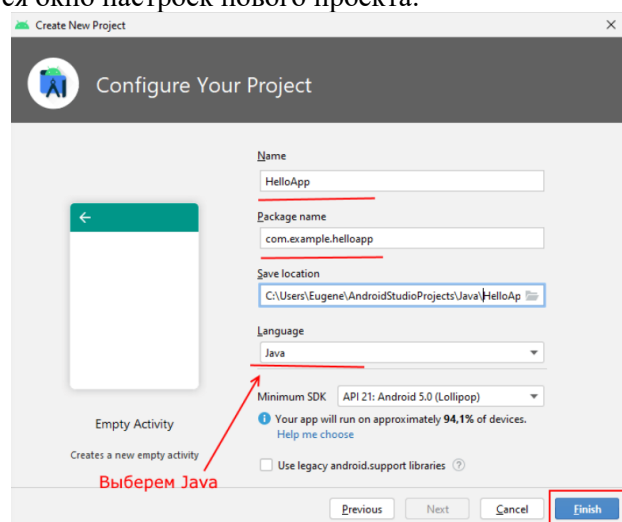


Рисунок 8

В окне создания нового проекта мы можем установить его начальные настройки:

В поле Name вводится название приложения. Укажем в качестве имени название HelloApp

В поле Package Name указывается имя пакета, где будет размещаться главный класс приложения. В данном случае для тестовых проектов это значение не играет большого значения, поэтому установим com.example.helloapp.

В поле Save Location устанавливается расположение файлов проекта на жестком диске. Можно оставить значение по умолчанию.

В поле Language в качестве языка программирования укажем Java (будьте внимательны, так как по умолчанию в этом поле стоит Kotlin)

В поле Minimum SDK указывается самая минимальная поддерживаемая версия SDK. Оставим значение по умолчанию - API 21: Android 5.0 (Lollipop), которая означает, что наше приложение можно будет запустить начиная с Android 5.0, а это 94% устройств. На более старых устройствах запустить будет нельзя.

Стоит учитывать, что чем выше версия SDK, тем меньше диапазон поддерживаемых устройств. Далее нажмем на кнопку Finish, и Android Studio создаст новый проект:

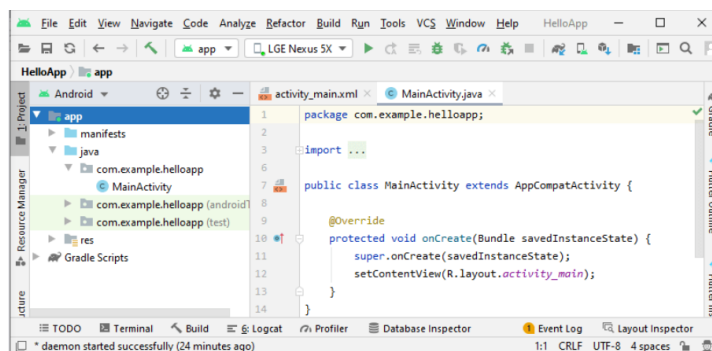


Рисунок 9

Вначале вкратце рассмотрим структуру проекта, что он уже имеет по умолчанию

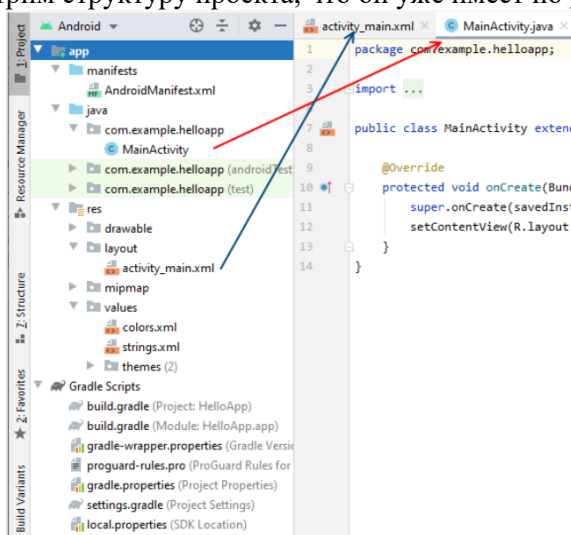


Рисунок 10

Проект Android может состоять из различных модулей. По умолчанию, когда мы создаем проект, создается один модуль - app. Модуль имеет три подпапки:

- manifests: хранит файл манифеста AndroidManifest.xml, который описывает конфигурацию приложения и определяет каждый из компонентов данного приложения.
- java: хранит файлы кода на языке java, которые структурированы по отдельным пакетам. Так, в папке com.example.helloapp (название которого было указано на этапе создания проекта) имеется по умолчанию файл MainActivity.java с кодом на языке Java, который представляет класс MainActivity, запускаемый по умолчанию при старте приложения
- res: содержит используемые в приложении ресурсы. Все ресурсы разбиты на подпапки.
- папка drawable предназначена для хранения изображений, используемых в приложении
- папка layout предназначена для хранения файлов, определяющих графический интерфейс. По умолчанию здесь есть файл activity_main.xml, который определяет интерфейс для класса MainActivity в виде xml
- папки mipmap содержат файлы изображений, которые предназначены для создания иконки приложения при различных разрешениях экрана.
- папка values хранит различные xml-файлы, содержащие коллекции ресурсов - различных данных, которые применяются в приложении.

По умолчанию здесь есть два файла и одна папка:

- файл colors.xml хранит описание цветов, используемых в приложении
- файл strings.xml содержит строковые ресурсы, используемые в приложении
- папки themes хранит две темы приложения - для светлую (дневную) и темную (ночную)

Отдельный элемент Gradle Scripts содержит ряд скриптов, которые используются при построении приложения.

Во всей этой структуре следует выделить файл MainActivity.java, который открыт в Android Studio и который содержит логику приложения и собственно с него начинается выполнение приложения. И также выделим файл activity_main.xml, который определяет графический интерфейс - по сути то, что увидит пользователь на своем смартфоне после загрузки приложения.

Запуск проекта

Созданный выше проект уже содержит некоторый примитивный функционал. Правда, этот функционал почти ничего не делает, только выводит на экран строку "Hello world!". Тем не менее это уже фактически приложение, которое мы можем запустить.

Для запуска и тестирования приложения мы можем использовать эмуляторы или реальные устройства. Но в идеале лучше тестировать на реальных устройствах. К тому же эмуляторы требуют больших аппаратных ресурсов, и не каждый компьютер может потянуть требования эмуляторов. А для использования мобильного устройства для тестирования может потребоваться разве что установить необходимый драйвер.

Режим разработчика на телефоне

По умолчанию опции разработчика на смартфонах скрыты. Чтобы сделать их доступными, надо зайти в Settings > About phone (Настройки > О телефоне) (в Android 8 это в Settings > System > About phone (Настройки > Система > О телефоне)) и семь раз нажать Build Number (Номер сборки).

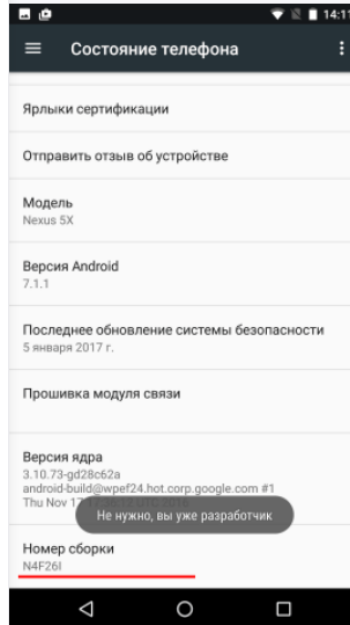


Рисунок 11

Вернитесь к предыдущему экрану и там вы увидите доступный пункт Developer options (Для разработчика).

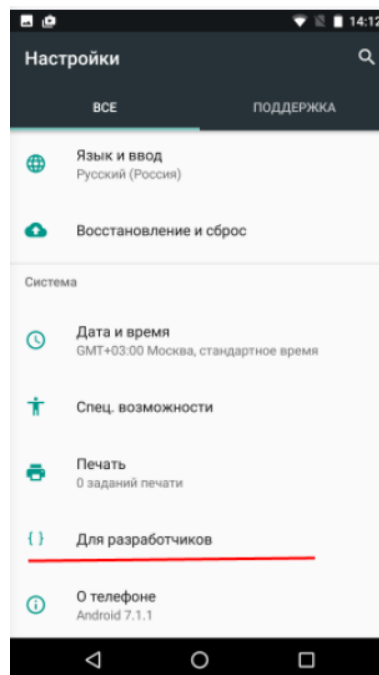


Рисунок 12

Перейдем к пункту Для разработчиков и включим возможность отладки по USB:

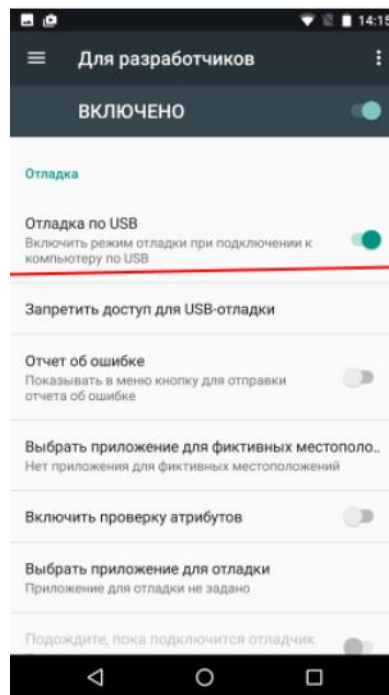


Рисунок 13

Запуск приложения

Подключим устройство с ОС Android (если мы тестируем на реальном устройстве) и запустим проект, нажав на зеленую стрелочку на панели инструментов.

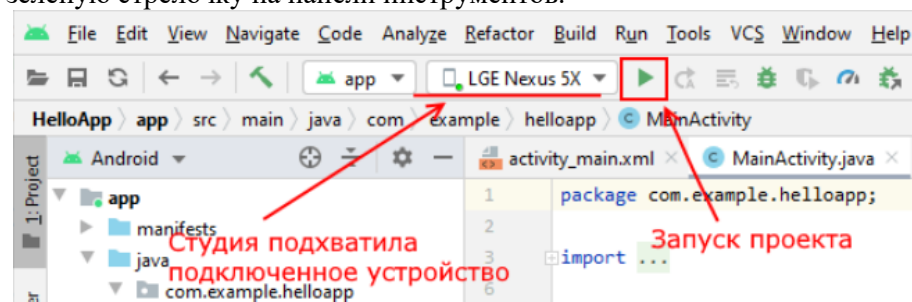


Рисунок 14

Выберем устройство и нажмем на кнопку ОК. И после запуска мы увидим наше приложение на экране устройства:

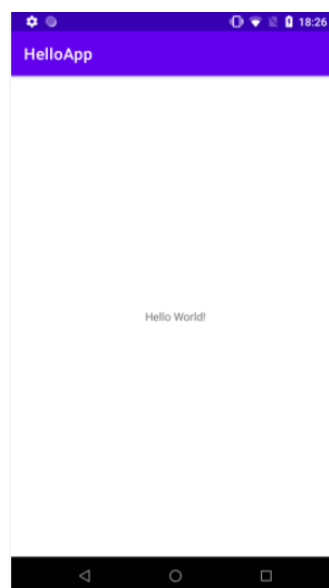


Рисунок 15

Практическая работа № 3.2. Java 2 MicroEdition

Цель работы: получения практических навыков работы с инструментальными средствами J2ME.

Установка платформы J2SE

Для установки Sun ONE Studio IDE требуется наличие в системе платформы Java SE версии не ниже 1.4.0. Для проверки версии в окне команд выполните

```
C:\>java -version
```

Результат должен выглядеть примерно так

```
C:\>java -version java version "1.4.0" Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0-b92) Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)
```

Для установки платформы J2SE v. 1.4.0 в ОС Microsoft Windows, выполните следующие шаги:

1. Поместите файл j2sdk-1_4_0-win.exe (его можно скачать с сайта <http://www.sun.com>) в каталог, отведенный вами для этой цели, например C:\Java2.
2. Деинсталлируйте предыдущую версию J2SE, если она присутствует в системе.
3. Запустите инсталлятор j2sdk-1_4_0-win.exe.
4. Следуйте инструкциям на экране до завершения процесса установки.

Установка IDE

1. Поместите файл ffj_me_win32.exe (его можно получить с <http://www.sun.com/>) в каталог, отведенный для Sun ONE Studio IDE, например C:\Java2\studio.
2. Запустите инсталлятор ffj_me_win32.exe.
3. Следуйте инструкциям.
4. Укажите каталог с ПО Java 2 SDK.
5. Укажите каталог для инсталляции. Имя каталога не должно содержать пробелов и каталог должен быть пустым или новым.
6. Нажмите Next для подтверждения списка модулей. Должны быть установлены Core Platform and Modules и J2ME Wireless Toolkit
7. Подтвердите настройки. Начнется процесс инсталляции.
8. Можно установить связь расширений .java и .nbm с Sun ONE Studio IDE.
9. Завершите инсталляцию.

Настройка IDE

1. Запустите Sun ONE Studio IDE.
2. При первом запуске программа запросит ваш пользовательский каталог. Введите имя каталога, в котором хотите хранить свои настройки, информацию о проектах и нажмите OK. Этот каталог должен отличаться от каталога, в который установлено IDE.
3. Выберите режим окна.
4. Выберите активные модули (по умолчанию - все).
5. Выберите режим автоматического обновления.
6. Выберите метод регистрации.

Проверка установленного Sun ONE Studio IDE

1. Установленный Sun ONE Studio IDE можно проверить на одном из тестовых приложений.
2. Откройте каталог примеров (ffj-user-dir/sampledир) в окне Explorer IDE.
3. Кликните комплект мидлетов правой клавишей мыши и выберите Execute.
4. Если Sun ONE Studio IDE установлен правильно, запустится эмулятор мобильного устройства, установленный по умолчанию, с выбранным мидлетом на экране.

Монтирование файловой системы

1. Начните работу с монтирования файловой системы или каталога, в котором предполагается размещать код приложений. Монтирование позволит вам просматривать файлы и работать с ними в рамках Sun ONE Studio IDE. Монтированная файловая система включается в путь к классам Java, что необходимо для компиляции, запуска и отладки.
2. Чтобы монтировать файловую систему из меню File выберите Mount Filesystem. Это откроет окно выбора шаблона файловой системы. Выберите Local Directory и нажмите Next. Укажите требуемый каталог и нажмите Finish.
3. Выбранный каталог появится на вкладке Filesystems панели Explorer. Каталог смонтирован и готов к работе.

Установка эмулятора по умолчанию

Эмулятор позволяет симулировать исполнение приложения на целевом устройстве. Обычно эмулятор включает примеры устройств, которые он эмулирует. Такие образцы называются "skins". Эмулятор J2ME Wireless Toolkit Emulator входит в поставку Sun ONE Studio и содержит несколько примеров устройств, например, телефон Motorola i85.

Можно создать несколько экземпляров эмулятора, что позволит быстрее и проще отлаживать приложение для различных устройств.

Эмулятор по умолчанию - один из установленных эмуляторов, который используется для запуска приложений, которые не являются частью комплектов мидлетов и для новых комплектов мидлетов. Для установки эмулятора по умолчанию:

1. На вкладке Runtime панели Explorer раскройте узел Installed Emulators и кликните правой кнопкой мыши на эмуляторе. Выберите Set As Default.
2. В панели Explorer кликните правой кнопкой мыши на первом экземпляре J2ME Wireless Toolkit и выберите Properties из меню.
3. Установите свойства для этого эмулятора.

Для установки одного из примеров устройств в качестве устройства по умолчанию для эмулятора:

1. На вкладке Runtime панели Explorer разверните Installed Emulators, затем разверните J2ME Wireless Toolkit. Вы увидите список устройств (skins).
2. Кликните правой клавишей мыши на одном из устройств и выберите Set as Default.

Создание пакетов для мидлета

Для создания пакета:

1. В панели Explorer на вкладке Filesystems кликните правой клавишей мыши каталог и выберите New \ Java Package. Новый пакет можно разместить в любом каталоге файловой системы.
2. В окне New Wizard введите имя нового пакета и нажмите Finish. Созданный пакет будет отображаться в панели Explorer с иконкой в виде папки.

Создание мидлета

1. В панели Explorer кликните правой клавишей мыши на имени пакета. В контекстном меню выберите New \ MIDP \ MIDlet.
2. Выбор шаблона мидлета запустит wizard для создания мидлета.
3. Введите имя мидлета. нажмите Next.
4. Нажмите Finish для создания мидлета.
5. Созданный мидлет появится на вкладке Filesystem панели Explorer, его код появится в окне Source Editor. Иконка мидлета представляет сетку нулей и единиц, это показывает, что мидлет еще не скомпилирован.
6. Разверните узел мидлета для просмотра его классов, а также их полей, конструкторов и методов.

Программирование мидлета

Вы можете писать код мидлета двумя способами: непосредственно вводя исходный код в окне Source Editor или используя инструменты для создания методов, полей, конструкторов, инициализаторов, классов и интерфейсов.

Например, добавление нового поля к классу можно выполнить так:

1. На вкладке Filesystem панели Explorer кликните правой клавишей мыши на имени класса и выберите Add \ Field.
2. Заполните диалоговое окно Add New Field.
 - a. Введите имя нового поля, выберите его тип.
 - b. Установите модификаторы поля и спецификаторы доступа.
 - c. Установите исходное значение поля.
 - d. Нажмите OK.
3. Двойной клик на имени класса откроет его код в окне Source Editor.
4. Введите необходимые описания и комментарии.

Компиляция мидлета

Процесс компиляции включает в себя собственно компиляцию исходного кода и создание бинарных файлов .class, а также предварительную проверку созданных классов и оптимизацию кода для виртуальной машины CLDC.

Для компиляции мидлета:

1. В панели Explorer кликните правой клавишей мыши на имени мидлета и выберите Compile. Также можно использовать пункты меню Compile или Build для компиляции класса мидлета и всех связанных с ним классов.
2. Просмотрите сообщения компилятора в окне Output.

3. Если в процессе компиляции произошли ошибки, класс в панели Explorer будет помечен иконкой с буквой "X". В окне Output отображается строка(и) кода, содержащая ошибку и эта строка подсвечивается в окне Source Editor.

Тестирование мидлета

Для тестирования мидлета после успешной компиляции используйте пункт меню Execute, который запустит мидлет на исполнение с использованием эмулятора по умолчанию.

Создание комплекта мидлетов (MIDlet Suite)

Комплект мидлетов организует исходные файлы и файлы атрибутов приложения MIDP. При создании комплекта мидлетов автоматически создается необходимый файл JAR, который содержит файлы приложения. Также Sun ONE Studio IDE создает файл описателя приложения JAD.

Для создания комплекта мидлетов:

1. Создайте новый каталог, в котором будет размещаться комплект мидлетов.
2. В панели Explorer в контекстном меню созданного каталога выберите New \ MIDP \ MIDlet Suite.
3. В окне MIDletSuite wizard введите имя нового комплекта мидлетов. нажмите Next.
4. Выберите опцию Use Existing MIDlet и введите имя класса мидлета для включения его в комплект.
5. Установите свойства мидлета.
6. Выберите эмулятор по умолчанию для мидлета. нажмите Finish.
7. Разверните узел комплекта и проверьте его содержимое.

Наполнение комплекта мидлетов

1. В контекстном меню комплекта выберите Edit Suite, появится соответствующее окно. Выберите вкладку Jar Contents для просмотра содержимого файла JAR.
2. Выберите классы, которые надо добавить к комплекту и нажмите Add. Используйте кнопки Remove и Remove AP для удаления содержимого из JAR.
3. Используйте пункт Properties контекстного меню комплекта для настройки его свойств.
4. Выберите пункт Update JAR контекстного меню комплекта для активизации внесенных изменений.
5. Можно просмотреть содержимое файла JAD и декларации JAR с помощью пункта View Manifest/JAD контекстного меню комплекта.

Тестирование комплекта мидлетов

1. В контекстном меню комплекта мидлетов выберите пункт Compile или Compile AP (или Build или Build AP) для компиляции приложений в комплекте.
2. В контекстном меню комплекта мидлетов выберите пункт Execute для запуска комплекта.

Обратите внимание, что приложение запускается на эмуляторе и устройстве, выбранном по умолчанию для данного комплекта.

Вместо запуска конкретного приложения на эмуляторе будет отображен список приложений комплекта для выбора пользователем.

Задания

1. Установите среду разработки Sun ONE Studio IDE.
2. Проведите сборку одного из комплектов мидлетов в каталоге примеров, запустите его на выполнение на эмуляторе различных устройств.
3. Создайте простой мидлет, отображающий список из нескольких элементов и имеющий команд}' завершения работы.
4. Сформируйте комплект мидлетов из 1-2 мидлетов, обратите внимание на различное поведение эмулятора при обработке комплекта и отдельных мидлетов.

Контрольные вопросы по теории

1. Дайте определение пакета, мидлета и комплекта мидлетов.
2. Опишите их назначение
3. Опишите их взаимосвязь.

Практическая работа № 3.3. Протокол Bluetooth

Цель работы: научиться применять устройства Bluetooth для передачи информации: физически устанавливать соединение и программно реализовывать передачу данных.

Bluetooth – это технология передачи данных по радиоканалам на короткие расстояния, позволяющая осуществлять связь беспроводных телефонов, компьютеров и различной периферии даже в тех случаях, когда нарушается требование прямой видимости.

Первый стандарт Bluetooth 1.0. был выпущен в декабре 1999 г. Технология названа в честь датского короля Гарольда Блатана (пер. с датского «синий зуб» на английский «blue tooth»), объединившего в X веке земли Дании и Норвегии и прославившийся тем, что умел находить общий язык с князьями-вассалами.

Первоначально технология задумывалась как средство простого соединения ПК и телекоммуникационных устройств мобильных телефонов. Но оказалась настолько удачной, что ее развитие видится весьма перспективным.

Специалисты предполагают два направления использования Bluetooth. Первое — домашние сети, включающие в себя различную электронную технику, в частности компьютеры, телевизоры и т. п. Второе — локальные сети офисов небольших фирм, где стандарт Bluetooth может прийти на смену традиционным проводным технологиям.

Расстояние, на которое может быть установлено соединение Bluetooth – невелико и составляет от 10 до 30 м. В настоящее время разработчики пытаются его увеличить хотя бы до 100 м. Зато для Bluetooth не требуется прямой видимости или направленной антенны, соединение может быть установлено через стену, при условии, что она не экранирована.

Главной особенностью интерфейса Bluetooth является то, что поддерживающие его устройства соединяются друг с другом автоматически, стоит им только оказаться в пределах досягаемости.

Основу архитектуры Bluetooth составляет пикосеть (piconet), состоящая из одного главного узла (M) и нескольких (до семи) подчиненных узлов (S), расположенных в радиусе 10 м. В одной и той же комнате, если она достаточно большая, могут располагаться несколько пикосетей.

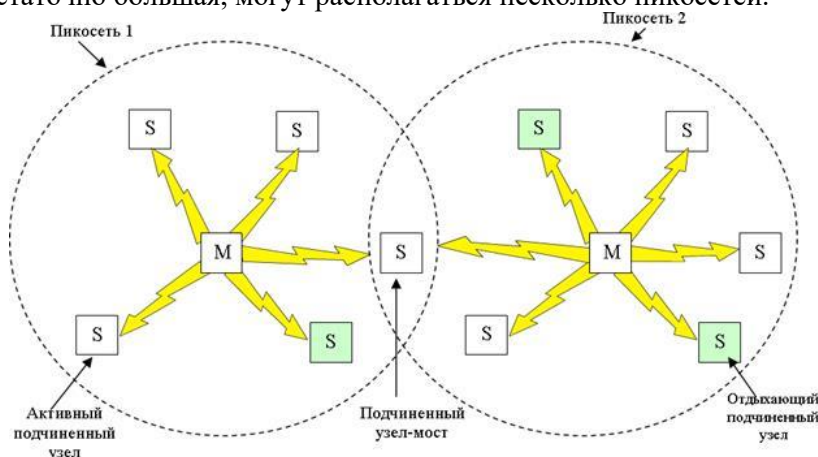


Рисунок 16

Несколько объединенных вместе пикосетей составляют рассеянную сеть (scatternet). Помимо семи активных подчиненных узлов, один главный узел может поддерживать до 255 так называемых отдыхающих узлов. Это устройства, которые главный узел перевел в режим пониженного энергопотребления (за счет чего продлевается ресурс их источников питания).

В таком режиме узел может только отвечать на запросы активации или на сигнальные последовательности от главного узла. Существуют еще два промежуточных режима энергопотребления – приостановленный и анализирующий.

Такое решение с главным и подчиненными узлами оказалось очень простым и дешевым в реализации, несмотря на то, что подчиненные узлы получились неразговорчивыми — они лишь выполняют то, что им прикажет главный узел.

В основе пикосетей лежит принцип централизованной системы с временным уплотнением. Главный узел контролирует временные интервалы и распределяет очередность передачи данных каждым из подчиненных узлов. Связь существует только между подчиненным и главным узлами. Прямой связи между подчиненными узлами нет.

Большинство сетевых протоколов просто предоставляют каналы связи между коммуникационными единицами и оставляют прикладное использование этих каналов на усмотрение разработчиков. В противоположность этому, например, спецификация Bluetooth V1.1, называет 13 поддерживаемых приложений и для каждого из них предоставляет свой набор протоколов, называемые профилями.

Профиль – основа, на которой строятся реальные приложения. Его главная задача состоит в создании канала между главным и подчиненным узлами.

Довольно общим является профиль определения сервиса, используемый для определения того, какие сервисы могут быть предоставлены другими устройствами. Вся аппаратура системы Bluetooth должна поддерживать два описанных ниже профиля. Все прочие являются необязательными.

Профиль последовательного порта — это транспортный протокол, который используется большинством других профилей. Он эмулирует последовательный канал и полезен при работе с приложениями, которым требуется этот канал.

Профиль общего объектного обмена определяет клиент-серверные взаимоотношения, возникающие при обмене данными. Клиенты иницируют операции, но подчиненная станция может выступать либо в роли клиента, либо в роли сервера.

Следующая группа профилей имеет отношение к сетям. Профиль доступа к ЛВС позволяет устройству Bluetooth подсоединиться к стационарной вычислительной сети. Он позволяет ноутбуку соединиться с мобильным телефоном, имеющим встроенный модем.

Профиль «Факс» позволяет беспроводным факс-машинам отсылать/получать факсы при помощи мобильного телефона.

Три профиля относятся к телефонии. Профиль беспроводной телефонии обеспечивает связь телефонной трубки с базой. Сейчас домашний телефон не может использоваться в качестве мобильного, даже если он не имеет совсем никаких проводов, однако в будущем, эти два устройства обязательно будут объединены.

Профиль Intercom позволяет двум телефонам соединяться друг с другом наподобие раций. Профиль Гарнитура представляет собой приложение, позволяющее устройствам hands-free держать связь с базой (телефоном).

Последние три профиля (Передача объектов, Передача файлов, Синхронизация) предназначены для организации обмена данными между беспроводными устройствами. Объекты могут представлять собой электронные визитные карточки, изображения или файлы с данными.


Пока недостатком технологии Bluetooth является сравнительно низкая скорость передачи данных (720 кбит/с), поэтому она не способна обеспечить передачу видеосигнала.

Начиная с 2001 г. вышло несколько версий Bluetooth. Наиболее существенным изменением в технологии стало введение аутентификации и, как следствие, шифрования передаваемых файлов с целью усиления системы защиты беспроводной сети.

Bluetooth-устройство включает в себя радиоприемник и радиопередатчик, которые работают в диапазоне частот от 2400 до 2483,5 МГц (этот диапазон в большинстве стран мира является открытым и свободным от лицензирования). Используемые частоты определяют возможности Bluetooth по передаче данных. Ширина канала для устройств Bluetooth составляет 723,2 кбит/с в асинхронном режиме.



Прежде чем установить сеанс связи, устройства Bluetooth должны обмениваться криптографическими ключами, подтверждающими идентичность устройства и его владельца. Дальнейшие операции разрешаются только при совпадении ключей обеих сторон. При этом используется упрощенный механизм аутентификации, предполагающий, что каждое устройство выполняет в этом процессе роль либо ведущего (master), генерирующего ключ, либо ведомого (slave), осуществляющего проверку ключа.

Задание 1. Подключите и настройте устройства Bluetooth

1. Физически подключите устройства Bluetooth к USB-разъемам компьютеров. Если ранее это устройство не устанавливалось на компьютере, то будет произведен поиск драйверов и их установка. В итоге в области уведомления появится значок .
2. Откройте диалоговое окно Устройства Bluetooth (Пуск/Панель управления/Устройства Bluetooth).
3. Установите Bluetooth-имя для первого компьютера:
 - перейдите на вкладку Оборудование;
 - дважды щелкните по элементу USB Bluetooth Wireless Adapter;
 - на вкладке Дополнительно, введите имя этого компьютера (Comp_1);
 - примените параметры кнопкой ОК.
4. Аналогично установите Bluetooth-имя Comp_2 для второго компьютера.
5. Разрешите обнаружение компьютеров другими устройствами Bluetooth, установив на вкладке Параметры флажок Включить обнаружение.
6. Выполните обнаружение Bluetooth-устройств:

- перейдите на вкладку Устройства;
- щелкните по кнопке Добавить;
- установите флажок Устройство установлено и готово к обнаружению и закройте окно кнопкой Далее.
После этого появится список обнаруженных устройств. Там должен отображаться другой компьютер.
- выберите другой компьютер (Comp_1) и щелкните по кнопке Далее;
- выберите Выбрать ключ доступа самостоятельно и введите в поле любое значение (например, 123). Закройте окно кнопкой Далее.
На другом компьютере будет выведено сообщение о необходимости ввести ключ доступа.
- Введите тот же ключ, что и на первом компьютере - (123) и закройте текущее окно кнопкой Далее.
- сбросьте флажок Выключить обнаружение;
- примените параметры кнопкой Готово.

Задание 2. Передайте файл с одного компьютера на другой:

1. Подготовьте на первом компьютере файл в формате RTF, содержащий сведения о технологии Bluetooth.
2. Подготовьте второй компьютер для приема файлов. Для этого щелкните по значку  и выберите команду Принять файл.
3. Передайте файл:
 - вызовите диалоговое окно Мастера передачи файлов на первом компьютере (щелкните по значку  и выберите Отправить файл);
 - выберите получателя вашего файла, для чего щелкните по кнопке Обзор и выберите имя другого компьютера, щелкните Далее;
 - выберите файл для передачи, для этого щелкните по кнопке Обзор и перейдите в папку с файлом и дважды щелкните по нужному файлу;
 - инициализируйте передачу кнопкой Далее;
 - завершите передачу файла кнопкой Готово.
4. Примите полученный файл на другом компьютере:
 - щелкните по кнопке Обзор и перейдите в вашу папку, щелкните Далее;
 - завершите приеме файла кнопкой Готово.
5. Проверьте полученный файл, открыв его в текстовом процессоре.

Задание 3. Установите и настройте Личную сеть (PAN):

1. Откройте окно свойств Bluetooth:
 - на первом компьютере откройте окно Сетевые подключения;
 - вызовите контекстное меню элемента Сетевое подключение Bluetooth и выберите команду Свойства.
2. Перейдите на вкладку Общие.
3. Установите адрес этого компьютера вручную:
 - откройте диалоговое окно свойств элемента Протокол Интернета (TCP/IP);
 - выберите Использовать следующий IP-адрес;
 - введите в поле IP-адрес – 192.168.0.1;
 - введите в поле Маска подсети – 255.255.255.0;
 - примените параметры кнопкой ОК.
4. Закройте текущее окно кнопкой ОК.
5. Аналогично установите адрес другого компьютера. Используйте следующие данные:
 - IP-адрес – 192.168.0.2;
 - Маска подсети – 255.255.255.0.
6. Подключите Личную сеть Bluetooth:
 - вызовите контекстное меню индикатора Bluetooth в области уведомления и выберите команду Присоединиться к личной сети (PAN);
 - выберите другой компьютер и щелкните Подключить.

Задание 4. Выполните самостоятельные задания:

1. Определите время, необходимое для передачи файла размером 3 Мб, и оформите результат в виде таблицы:

Таблица 2

Способ передачи	Объем файла (Мбайт)	Время затраченное на передачу (в секундах)	Теоретическая скорость передачи (бит/с)	Расчетная скорость передачи (бит/с)
Инфракрасная связь				
Bluetooth				

2. Настройте личную сеть Bluetooth для работы по динамически формируемым адресам.
3. Организуйте обмен данными между двумя мобильными телефонами посредством Bluetooth.
4. Организуйте передачу информации с мобильного телефона на ПК и обратно.

Вопросы для проверки

1. В чем суть технологии Bluetooth?
2. Опишите направления использования Bluetooth.
3. В чем состоит главная особенность интерфейса Bluetooth?
4. Опишите пикосеть и рассейную сеть.
5. Что такое профиль? Опишите основные профили.

Практическая работа № 3.4. Установка инструментария для разработки мобильных приложений

Практическая работа № 3.5. Настройка среды для разработки мобильных приложений

Цель работы: выполнить установку среды разработки и настроить ее для работы

Установка и настройка компонентов среды разработки

Приложения для Android, как и большинство приложений для коммуникаторов, разрабатываются на стандартном ПК, где ресурсы не ограничены (по сравнению с мобильным устройством) и загружаются на целевой коммуникатор для отладки, тестирования и последующего использования. Приложения можно отлаживать и тестировать на реальном устройстве под управлением Android или на эмуляторе. Для первоначальной разработки и отладки удобнее использовать эмулятор, а затем выполнять окончательное тестирование на реальных устройствах.

Разработчик предоставляется возможность использовать средства разработки приложений для Android на ПК под управлением любой из распространенных операционных систем: Windows, Linux и Mac OS X. Ниже будет детально описан процесс установки компонентов среды разработки под Windows XP, для прочих операционных систем отличия весьма незначительны.

Установка JDK

Для Android SDK требуется JDK версии не ниже 5 (на момент написания данного методического руководства была актуальна 7-я версия Sun JDK и 6-я версия Open-JDK). Для Windows традиционно используется Sun JDK, который можно бесплатно загрузить на сайте разработчика:

Страница загрузки Sun JDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



Рисунок 17



Рисунок 18

Принятия условий лицензионного соглашения Oracle Binary Code License Agreement for Java SE (1) можно выбрать подходящую для вашей платформы ссылку(2)
Установка загруженного JDK достаточно проста:

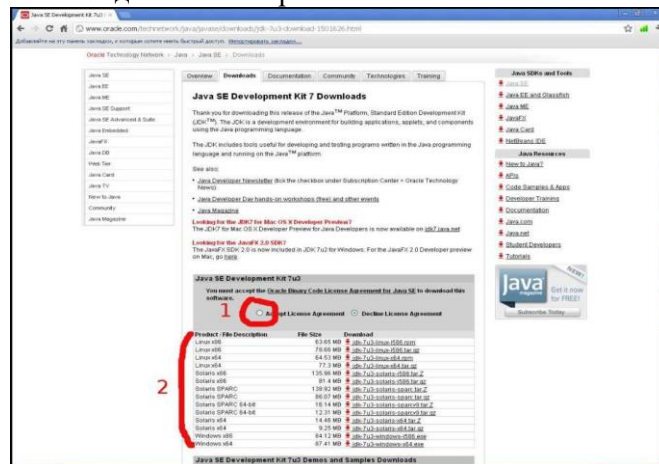


Рисунок 19

каталог для установки по умолчанию (1) не подходит, его можно изменить (2), а затем продолжить установку:

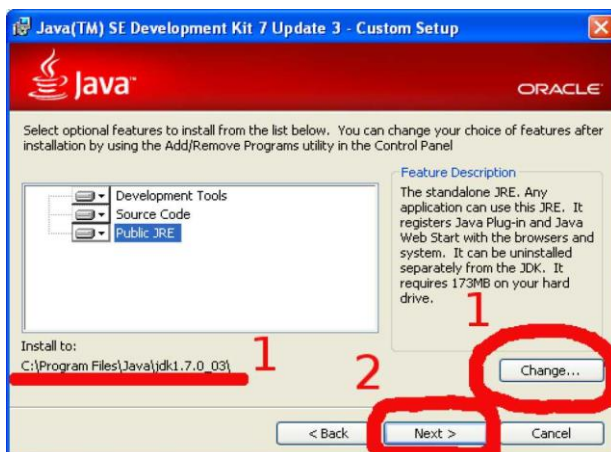


Рисунок 20

При установке Java Runtime так же можно изменить каталог для установки:



Рисунок 21

и продолжить установку:



Рисунок 22

Далее при желании можно зарегистрировать установленный продукт:



Рисунок 23

И установить JavaFX SDK, если не жалко 50 МБ дискового пространства:



Рисунок 24



Рисунок 25



Рисунок 26

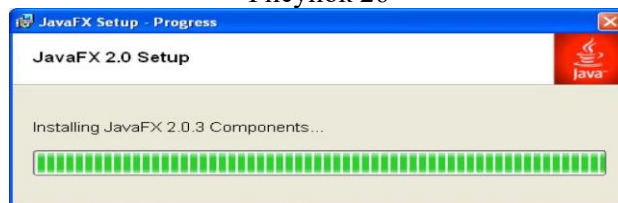


Рисунок 27

И, наконец, установка JDK закончена:



Рисунок 28

Установка Android SDK

На странице <http://developer.android.com/sdk> выбираем installer_r16-windows.exe

Здесь же имеются дополнительные инструкции по установке, а также ссылка на пошаговую инструкцию по установке SDK на все поддерживаемые платформы:

<http://developer.android.com/sdk/installing.html>

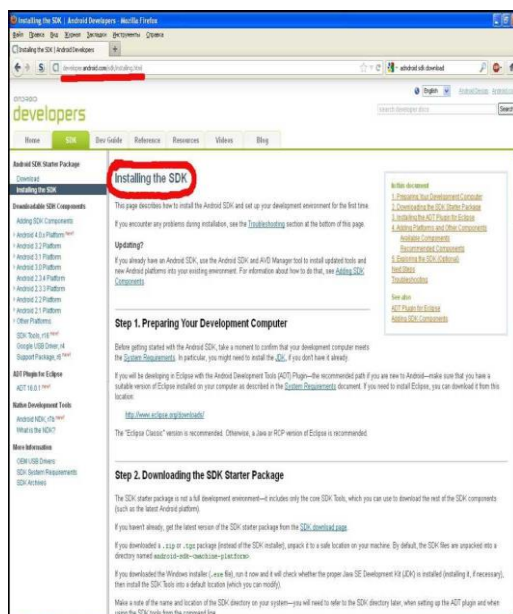


Рисунок 29

Ресурс <http://developer.android.com> является основным источником информации о платформе Android «из первых рук». В нем содержится огромное количество самых разнообразных сведений, необходимых разработчику, начиная с описания API и кончая такими вещами, как рекомендации по дизайну приложений и повышению производительности приложений. В данном методическом пособии в дальнейшем будут встречаться ссылки на конкретные страницы, посвященные изучаемым темам.

Установка загруженного Android SDK также не отличается особой сложностью:



Рисунок 30

Мастер установки обнаружит (если сможет) установленную версию JDK (1), после чего установку можно продолжить (2):

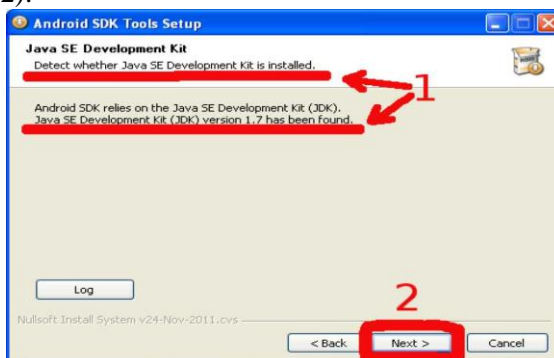


Рисунок 31



Рисунок 32

Установка начинается

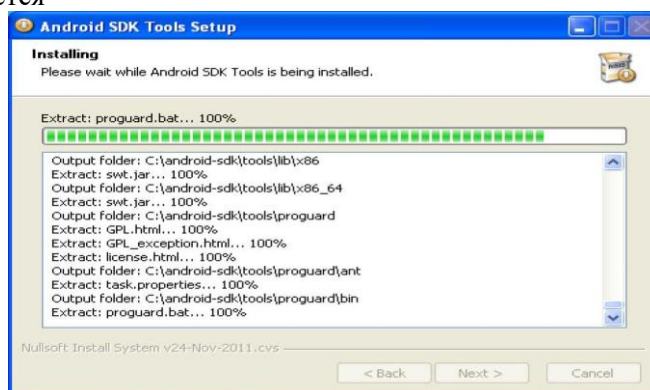


Рисунок 33

и заканчивается:

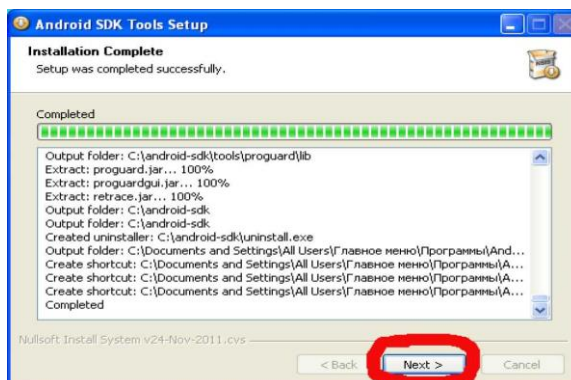


Рисунок 34

Можно сразу поставить галку(1) и запустить SDK Manager для окончательной настройки(2):



Рисунок 35

Требуется загрузить нужные версии API и другие полезные инструменты:

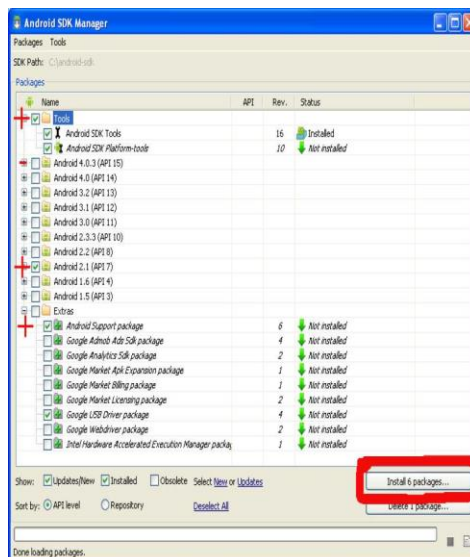


Рисунок 36

Далее мы запустим менеджер AVD (Android Virtual Devices) и сконфигурируем эмулятор на использование Android версии 2.1 с расширением Google API (GAPI). GAPI нужен, как правило, для приложений, использующих картографические возможности Android.

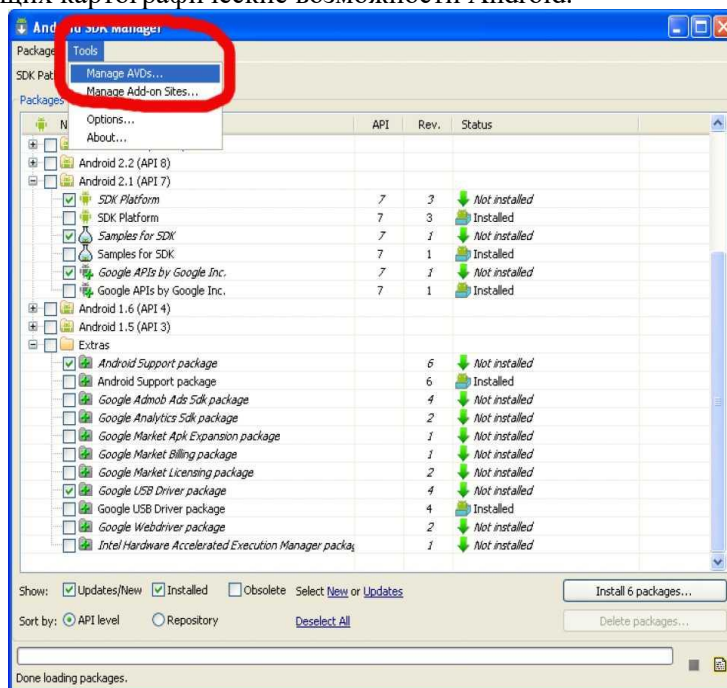


Рисунок 37

Создадим новое виртуальное устройство:

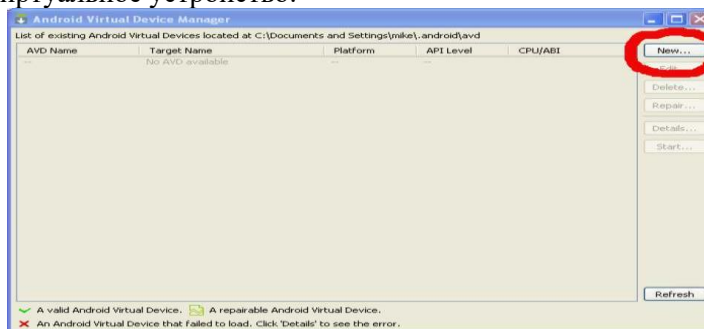


Рисунок 38

При создании нового виртуального устройства выбираем его название(1), целевой API (2), размер (или файл с образом) виртуальной SD-карты (3), одно из стандартных или собственное разрешение экрана (4) и, наконец, создаем устройство(5).

При необходимости можно указать плотность пикселей на экране, максимальный размер кучи для приложений внутри виртуальной машины, а также другие параметры:

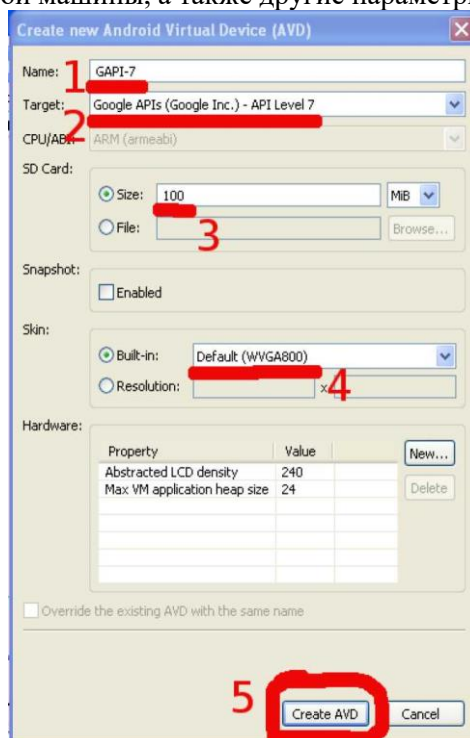


Рисунок 39

Убедимся, что все получилось, как хотели (1) и продолжим работу (2):

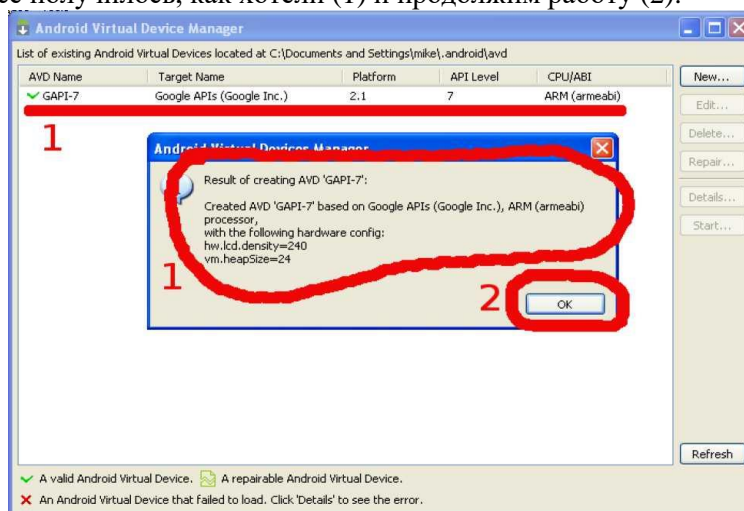


Рисунок 40

Пришло время запустить эмулятор:

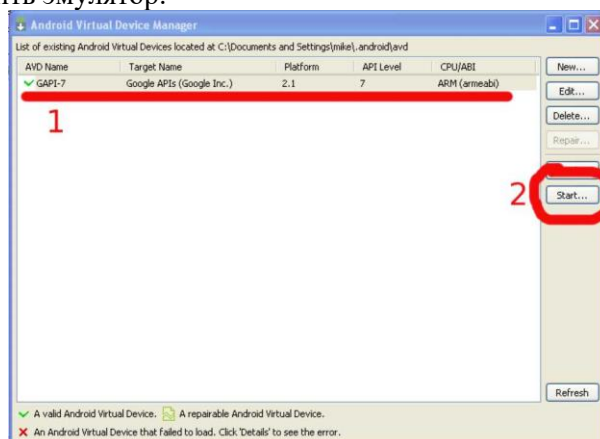


Рисунок 41

Важно: при использовании русскоязычных версий ОС Windows путь к рабочему каталогу AVD может включать русские буквы, что приведет к невозможности запуска эмуляторов. В этом случае можно создать каталог для AVD в корневом каталоге диска (например, C:\AVD) и присвоить переменной окружения ANDROID_SDK_HOME значение этого пути.

При запуске эмулятора из менеджера AVD можно дополнительно указать некоторые параметры, особенно полезно управление размерами или плотностью пикселей экрана виртуального устройства: поставив галку (1), можно указать желаемый размер экрана (2) в дюймах, вычислить, при необходимости, плотность пикселей (3) на используемом мониторе ПК:

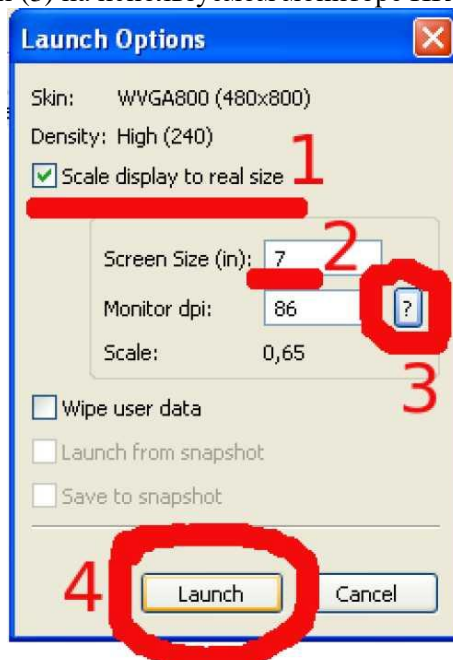


Рисунок 42

При первом запуске вы можете указать, следует ли отправлять анонимную статистику использования SDK в Google:

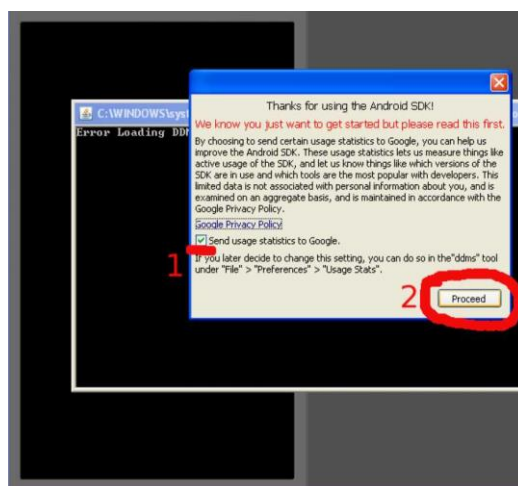


Рисунок 43

Более того, «пощупать» самые новые версии ОС Android и оценить их функциональные возможности можно безвозмездно, то есть даром просто выбрав нужную версию API при создании виртуального устройства.

Одна из неприятных особенностей платформы Android с точки зрения разработчика заключается в большом количестве выпущенных версий, значительно различающихся по возможностям. В настоящий момент Open Handset Alliance (т. е. Google) несколько затормозила этот процесс, и на данный момент флагманской версией является 4.0.x. Тем не менее, при выборе уровня API (API level) для нового ПО следует учитывать наличие у пользователей большого количество относительно старых устройств, так что для охвата максимального сегмента рынка оптимальным выбором является Android 2.1 (поддерживается 97% процентов устройств с Android).

Так же, как и в реальном, в виртуальном устройстве существует опасность разрушить файловую систему при неправильном прекращении работы (сбое питания и т. п.).
Установка IDE Eclipse

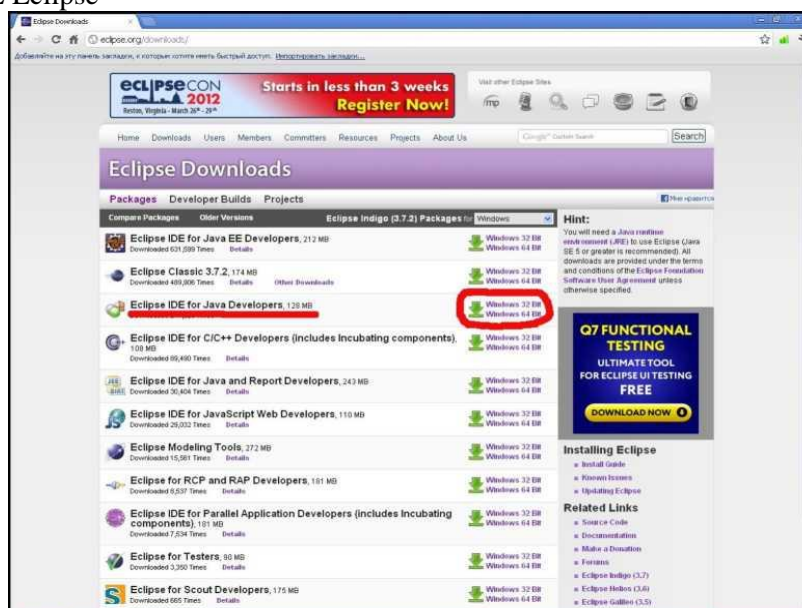


Рисунок 44

Интегрированная среда разработки Eclipse (довольно странное название, по-английски означает, в том числе помутнение рассудка) доступна для загрузки на официальном сайте по адресу <http://eclipse.org/downloads/> и распространяется в виде ZIP-архива:

В нашем случае подходящим вариантом является Eclipse IDE for Java Developers.

После выбора подходящей для используемой на ПК операционной системы версии IDE, для уменьшения времени загрузки имеет смысл выбрать европейское зеркало сайта:

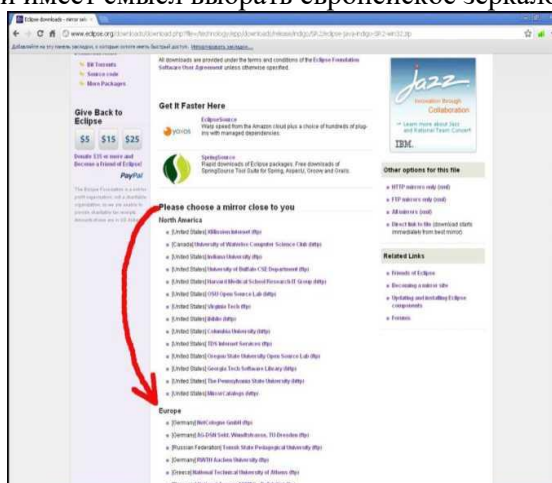


Рисунок 45

Получив нужный архив, выбираем подходящий путь для установки и распаковываем архив:

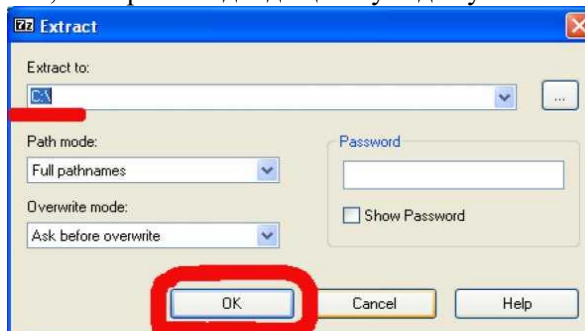


Рисунок 46

Установка плагина ADT

Для установки плагина ADT не требуется ручная загрузка, он будет установлен системой управления плагинами Eclipse. Тем не менее, мы рассмотрим установку подробно.

Запустим Eclipse

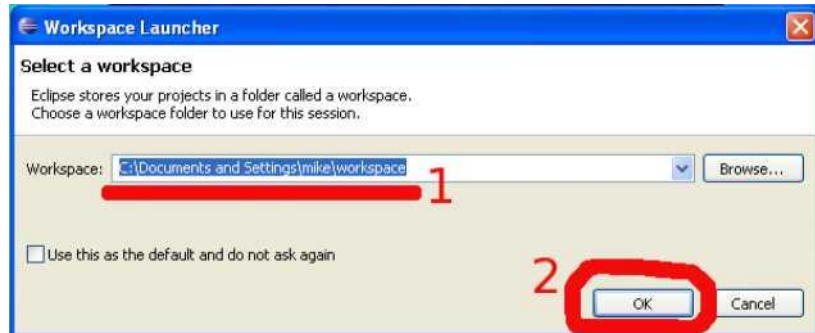


Рисунок 47

Выберем нужный пункт меню

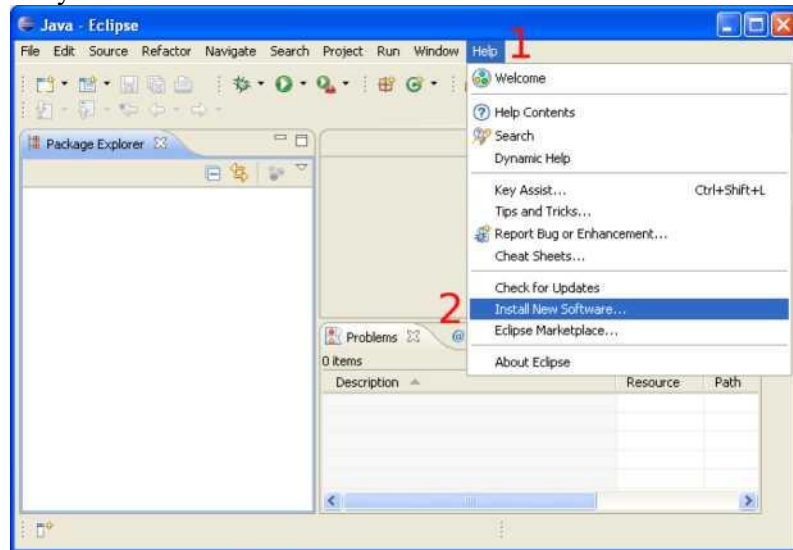


Рисунок 48

После установки выбранных компонентов перезапустим Eclipse
И настроим плагин ADT на использование уже установленного Android SDK.

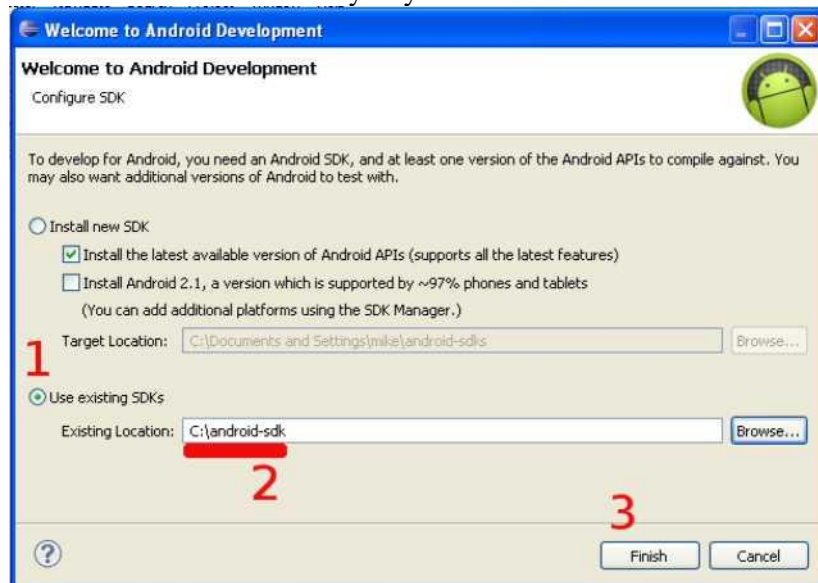


Рисунок 49

На снимке экрана (выше) можно увидеть, что ADT в состоянии сам загрузить и установить Android SDK. Мы использовали ручную установку для ускорения процесса (не тратили время на ожидание загрузки SDK).

Среда разработки установлена, пришло время создать первое приложение для платформы Android и запустить его в эмуляторе.

Создание первого приложения под Android

Для создания приложений в установленной нами среде разработки удобно использовать

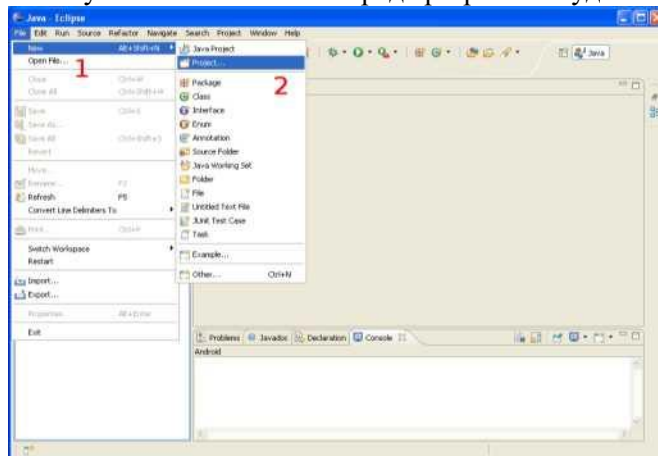


Рисунок 50

«Мастер создания нового проекта»:
Выбираем «Android Project»

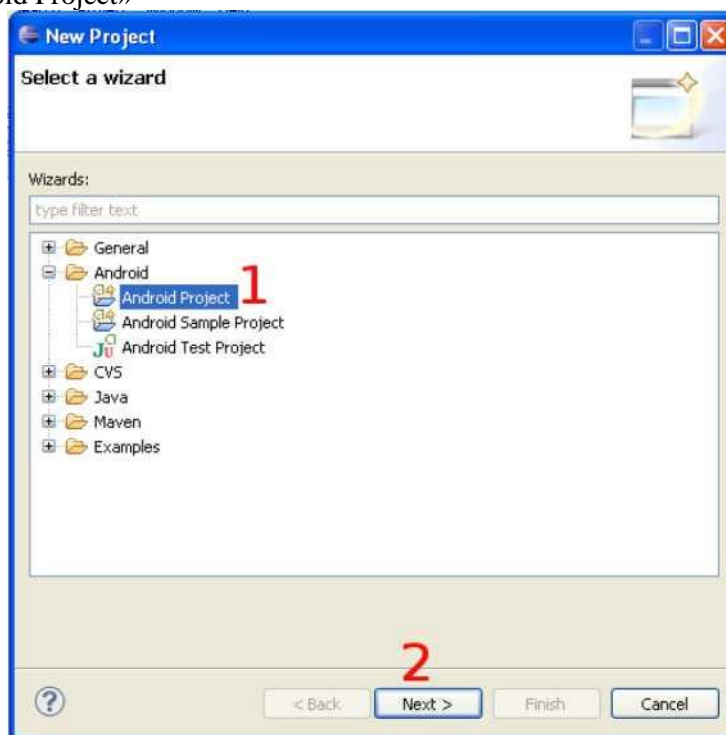


Рисунок 51

Вводим имя пакета (1) и имя класса Активности(2), который будет для нас автоматически создан мастером: Созданный проект сразу пока мы не внесли в него несовместимы с жизнью изменения является работоспособным приложением, так что мы его можем запустить:

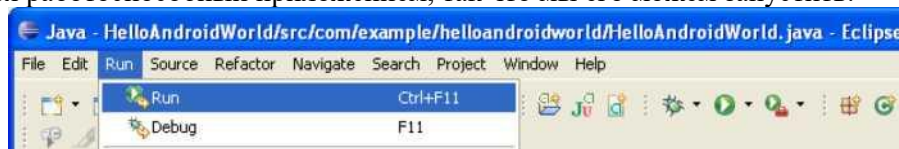


Рисунок 52

Даем понять Eclipse, как именно мы хотим запустить на выполнение наш проект:



Рисунок 53

Практическая работа № 3.6. Установка среды разработки мобильных приложений с применением виртуальной машины

Цель работы: научиться устанавливать программное обеспечение на компьютер, получить практические навыки администрирования ОС.

Установка и настройка платформы виртуализации Oracle VM VirtualBox

VirtualBox - программа абсолютно бесплатная и полностью на русском языке, что делает её очень привлекательной для использования как на домашнем, так и на рабочем компьютере. Впервые система была предоставлена в 2007 г. компанией InnoTek в двух вариантах – с открытым и закрытым исходными кодами, причем обе были бесплатны при условии некоммерческого использования. В 2008 г. платформа была перекуплена компанией Sun Microsystems, которая и занимается её разработкой в настоящее время.

Платформа представляет собой систему виртуализации для host-систем Windows, Linux и Mac OS и обеспечивает взаимодействие с гостевыми операционными системами Windows (2000/XP/2003/Vista/Seven), Linux (Ubuntu/Debian/ OpenSUSE/ Mandriva и пр.), OpenBSD, FreeBSD, OS/2 Warp.

Установка платформы Oracle VM VirtualBox

Скачать платформу, подходящую под Вашу систему, Вы можете по ссылке: <http://www.virtualbox.org/wiki/Downloads>

После того как установочный пакет оказался у Вас на жестком диске можно приступить к установке программы.

После запуска инсталлятора Вы увидите приветственное окно. Нажмите кнопку «Next» и в новом окне согласитесь с условиями лицензионного соглашения, поставив флажок «I accept the terms in the License Agreement». В следующем окне Вам будет предложено выбрать компоненты для установки и задать расположение исполняемых файлов. По умолчанию все компоненты устанавливаются на жесткий диск (а нам нужны все), а сама программа устанавливается в папку «Program Files» на системном диске. Если же вы хотите задать другое расположение, нажмите кнопку Browse и выберите новую папку для установки приложения.

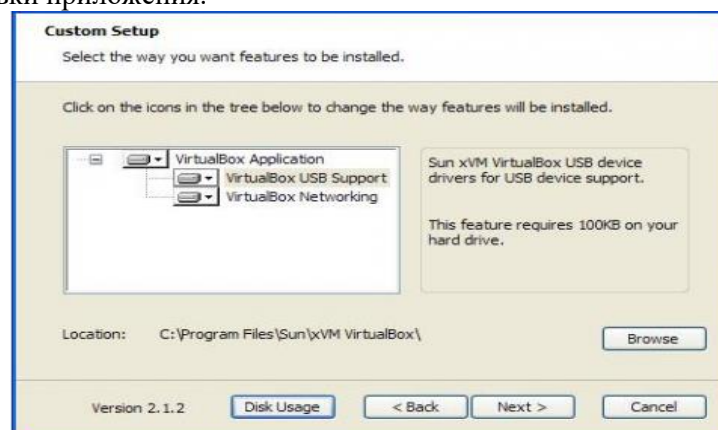


Рисунок 54

Далее процесс установки не потребует от Вас никаких вмешательств, кроме разрешения на создания ярлыков на рабочем столе и в меню «Пуск». По окончании установки программа запуститься автоматически.

Создание и первичная настройка виртуальной машины

Запустим приложение Oracle VM VirtualBox (при установке платформы на рабочем столе создается ярлык, которым Вы можете воспользоваться). Для создания Вашей первой виртуальной машины щелкните кнопку «Создать»:

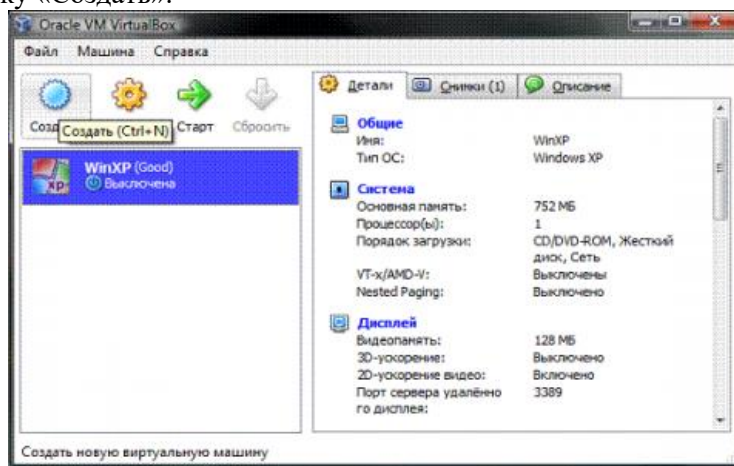


Рисунок 55

Примечание. В рассматриваемой платформе виртуализации уже существуют созданные виртуальные машины, и поэтому при первом создании своей Вы не увидите списка виртуальных операционных систем.

Перед Вами откроется новое окно, в котором будет сообщение о запуске мастера создания виртуальной машины. Нажимаем кнопку «Next» и видим новое окно, предлагающее выбрать имя операционной системы, её семейство и версию. На рис. приводится выбор Windows XP, но Вы может выбрать любую из доступных систем по своему вкусу.

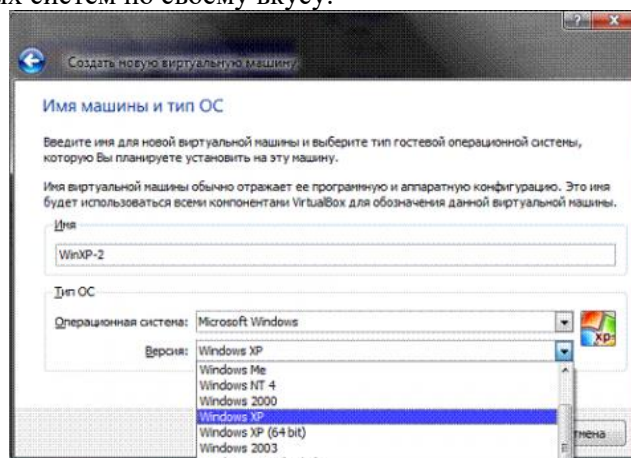


Рисунок 56

После нажатия кнопки «Next » Вам будет предложено определить размер оперативной памяти, выделяемой виртуальной машин. Здесь выбрано 1024мб, но для стабильной работы с виртуальной системой Windows XP достаточно будет и 512мб.

Далее потребуются создать виртуальный жесткий диск. Если Вы уже создавали виртуальные диски, можете использовать их, но мы рассмотрим именно процесс создания нового диска. Подтвердим, что создаваемый нами жесткий диск загрузочный, поставим флажок «Создать новый жесткий диск» и нажмем кнопку «Next».

Далее появится новое окно, которое сообщит Вам, что запущенный мастер поможет в создании виртуального диска, нажимаем кнопку «Next» для продолжения работы. В новом окне Вам будет предложено выбрать тип создаваемого диска – «динамически расширяющийся образ» или «образ фиксированного размера». Разница объясняется в справке данного окна, а от себя замечу, что непосредственно загрузочный диск удобнее создать фиксированного размера – это позволит Вам автоматически ограничить его размер, упростить и ускорить хранение, восстановление и создание резервных копий диска. К тому же, Вы можете создать для Вашей системы несколько жестких дисков

и вот уже те, которые не будут являться загрузочными, удобнее создавать динамически расширяющимися.

В следующем окне от Вас потребуется выбрать расположение создаваемого виртуального жесткого диска и его размер. Для загрузочного жесткого диска с системой Windows XP достаточно размера установленного по умолчанию (10 Гб), а вот расположить его лучше вне Вашего системного раздела, т.к. не стоит перегружать Ваш реальный загрузочный диск и создавать на нем файлы такого размера.

После этого появится окно «Итог», в котором будет указан тип, расположение и размер создаваемого Вами жесткого диска. Если Вы согласны создать диск с такими параметрами, нажмите «Финиш» и наблюдайте за процессом создания жесткого диска.

По завершения создания жесткого диска появится новое окно «Итог», в котором будут указаны параметры создаваемой Вами виртуальной машины. Если Вы не передумали ни по одному из описанных пунктов, нажимайте «Финиш» и переходите к настройке аппаратной части Вашей виртуальной машины.

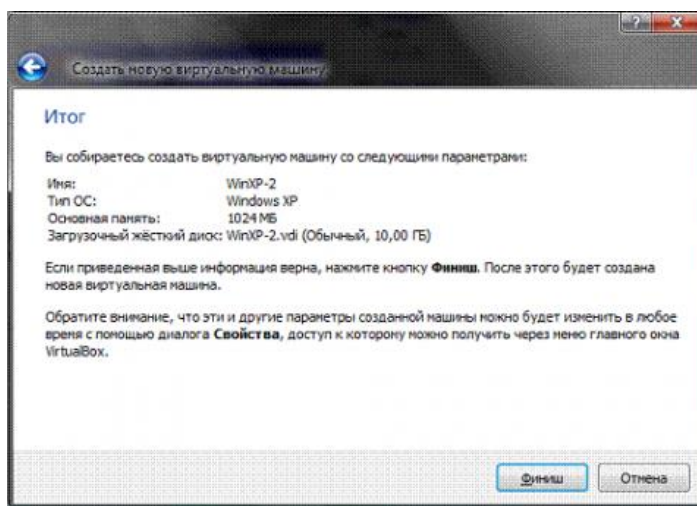


Рисунок 57

Настройка аппаратной части виртуальной машины

Итак, Вы создали виртуальный жесткий диск, теперь настала очередь собрать наш виртуальный компьютер полностью. Для этого снова вернитесь к главному окну VirtualBox, в нем Вы уже можете увидеть только что созданную виртуальную машину WinXP-2, а в поле с правой стороны представлено её описание, которое еще не похоже на описание полноценного ПК.

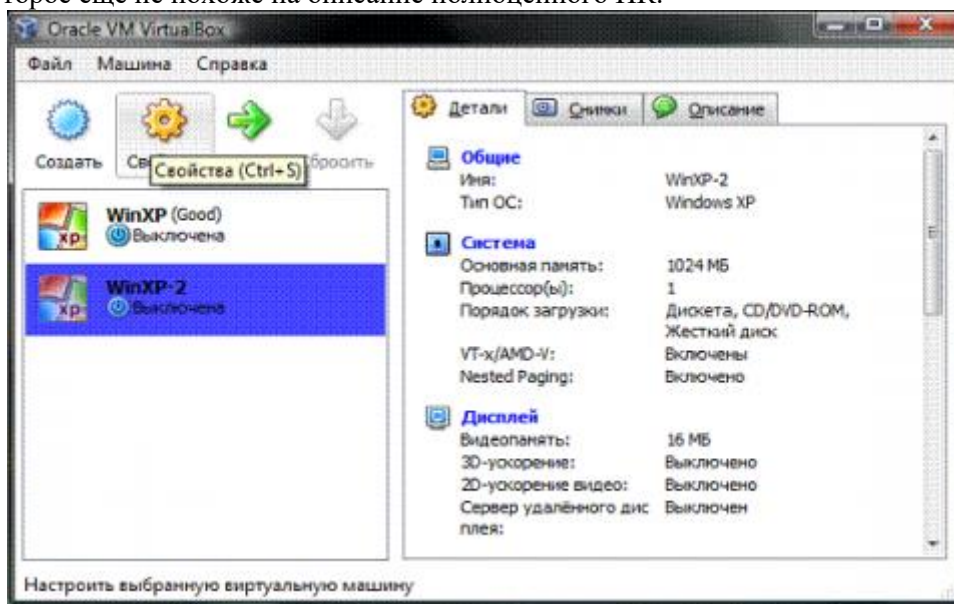


Рисунок 58

В колонке слева выберем нашу WinXP-2 и откроем её свойства, где колонка с левой стороны напоминает диспетчер устройств. На первой вкладке раздела «Общие» мы видим основные параметры нашей виртуальной машины:

Перейдем на вкладку дополнительно и посмотрим, какие настройки системы мы можем произвести:

- «Папка для снимков». Если Вы разместили Ваш жесткий в собственном расположении, то лучше и эту папку перенести туда же, т.к. снимки имеют большой вес и, опять-таки, не стоит перегружать Ваш системный диск. Моя рекомендация – создавать снимки перед каждым значительным изменением, которые Вы хотите произвести в виртуальной системе, причем даже на одну виртуальную машину Вы можете создать несколько снимков, содержащих отличные друг от друга настройки и установленные приложения;
- «Общий буфер обмена» – определение того, как будет работать буфер обмена между Вашей host-системой и виртуальной машиной. Вариантов работы буфера предоставлено четыре – «выключено», «только из гостевой ОС в основную», «только из основной ОС в гостевую», «двунаправленный». Мы выберем последний вариант, т.к. это обеспечит нам максимальное удобство в работе;
- «Сменные носители информации запоминать изменения в процессе работы», тут мы ставим флажок в знак согласия, т.к. данная опция позволит системе запомнить состояние CD/DVD-приводов;
- «Мини тулбар» – это небольшая консоль, содержащая элементы управления виртуальной машиной. Её лучше применять только в полноэкранном режиме, т.к. она полностью дублируется главным меню рабочего окна виртуальной машины. Располагать её действительно лучше сверху просто потому, что можно случайно нажать на какой-нибудь элемент управления, пытаясь, например, развернуть окно из панели задач виртуальной машины.

Перейдем к разделу система и на первой вкладке материнская плата произведем следующие настройки:

- если нужно, откорректируем размер оперативной памяти Вашей виртуальной машины, хотя окончательно убедиться в правильности выбранного объема Вы сможете только после запуска виртуальной машины. Выбирать размер Вы можете, исходя из объема доступной физической памяти, установленной на Вашем ПК. Например, при наличии 4ГБ ОЗУ оптимальным будет выделение 1ГБ, т.е. одной четвертой части, что позволит Вашей виртуальной машине работать без малейших зависаний;
- откорректируем порядок загрузки - дисковод гибких дисков («дискета») можно вообще отключить, а первым обязательно поставьте CD/DVD-ROM, чтобы обеспечить возможность установки ОС с загрузочного диска. При этом в роли загрузочного диска может выступать как и компакт-диск, так и образ ISO;
- все остальные настройки описаны в динамической справке снизу, и их применение зависит от аппаратной части вашего реального ПК, причем если Вы выставите настройки неприменимые к Вашему ПК система виртуальной машины просто не запустится;

Перейдем к вкладке «Процессор», тут Вы можете выбрать количество процессоров, установленных на Вашу виртуальную материнскую плату. Обратите внимание, что это опция будет доступна только при условии поддержки аппаратной виртуализации AMD-V или VT-x, а также включенной опцииOI APIC на предыдущей вкладке.

Здесь обратите Ваше внимание на настройки аппаратной визуализации AMD-V или VT-x. Перед включением этих настроек, нужно выяснить, поддерживает ли эти возможности Ваш процессор и включены ли они по умолчанию в BIOS (нередко они отключены).

Перейдем к разделу «Дисплей». В данном разделе на вкладке «Видео» Вы можете установить размер памяти виртуальной видео карты, а также включить 2D и 3D ускорение, причем включение 2D ускорения желательно, а 3D необязательно. На вкладке «Удаленный дисплей» Вы можете включить опцию, при которой Ваша виртуальная машина будет работать как сервер удаленного рабочего стола (RDP).

Переходим к разделу носители. Тут Вы можете увидеть созданной ранее виртуальный жесткий диск и позицию с надписью пусто. Выделяем эту позицию и осуществляем настройку.

Для настройки виртуального привода компакт-дисков можно пойти двумя путями:

- первый вариант - в раскрывающемся меню «Привод» выбираем Ваш реальный или виртуальный CD/DVD-ROM (существующие в реальной системе) и загружаем в него физический диск с дистрибутивом Windows XP или ISO-образ, если это эмулятор;
- второй вариант - щелкаем значок так, как показано на рисунке ниже и в отрывшемся окне добавляем ISO-образ загрузочного диска Windows XP, этим путем мы и пойдем.

Примечание. В данном пункте Вы уже не можете выбрать дистрибутив другой операционной системы, т.к. версия ОС уже была определена в самом начале процесса настройки виртуальной машины.

На рисунке ниже представлена процедура добавления ISO-образов в менеджер виртуальных носителей. В него Вы можете внести любое количество образов различного назначения, например, игры, дистрибутивы приложений, базы данных и пр., которые Вы сможете потом быстро переключать через главное меню окна виртуализации VirtualBox.

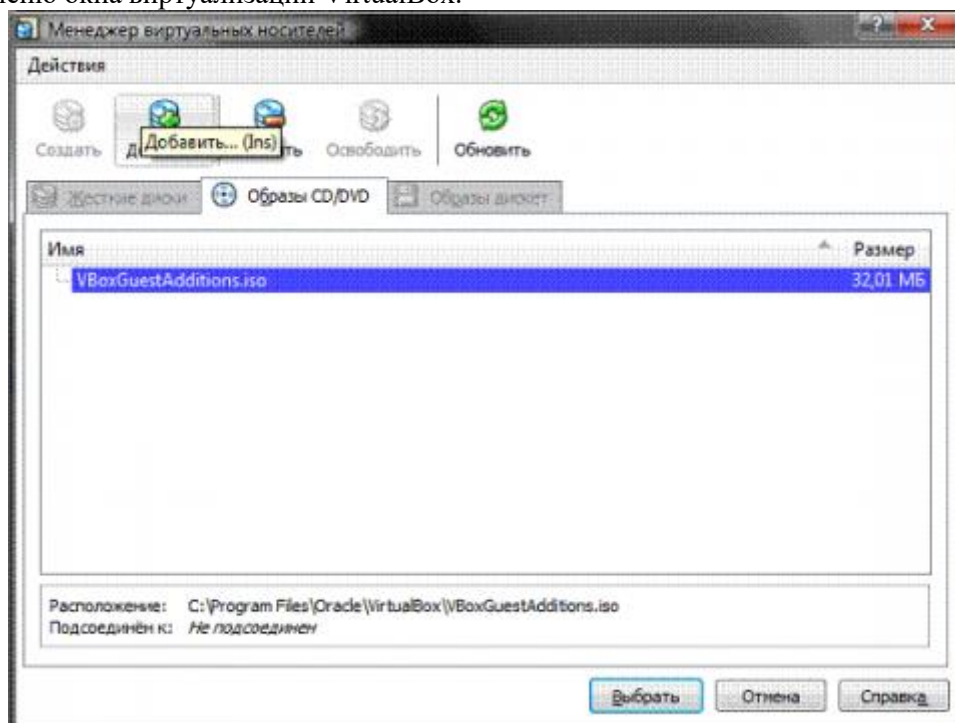


Рисунок 59

Далее Вы можете настроить слоты подключения накопителей, для упрощения описания привожу скриншоты, по которым Вы можете произвести действия по настройке. По привычке, я устанавливаю привод компакт-дисков как «Первичный мастер IDE», жесткий диск, содержащий загрузочный раздел, как «Вторичный мастер IDE», а дополнительный виртуальный жесткий диск «Первичный слейв IDE».

Настройка сети и сетевого взаимодействия не рассматривается в рамках данной статьи, поэтому замечу лишь то, что сетевой адаптер типа NAT включен по умолчанию, а этого уже достаточно для предоставления Вашей виртуальной машине доступа в Интернет. Тип выбираемого адаптера должен быть «Pcnet-Fast III (Am79C973)», т.к. только для этого адаптера присутствуют драйверы в ОС Windows XP.

Раздел COM я подробно не описываю, т.к. подключать к портам данного типа уже нечего. В случае если Вам все же потребуется подключить устройство с интерфейсом RS-232C, то наиболее удобным будет включить COM-port виртуальной машины в режиме «хост-устройство», а в качестве «пути к порту» использовать имя реально порта Вашего ПК, которое Вы можете посмотреть в диспетчере устройств.

Переходим к разделу USB, здесь ставим оба доступных флажка, а затем, используя кнопку с изображением «вилки» USB и «плюса», добавляем все доступные контроллеры.

Переходим к разделу «Общие папки» и выбираем папки, которые нужно сделать доступными для виртуальной машины.

Примечание. Обратите внимание на динамическую справку снизу – именно таким образом, через окно командной строки, Вы сможете подключить общие папки к Вашей виртуальной машине.

На этом настройка аппаратной части Вашей виртуальной машины закончена, и можно перейти к установке операционной системы.

Настройка операционной системы виртуальной машины

Описание установки операционной системы в статье не описывается, т.к. на сайте представлено достаточно информации о методах и тонкостях данной операции, поэтому укажу первый шаг – возвращаемся к главному окну VirtualBox и нажимаем кнопку «Старт».

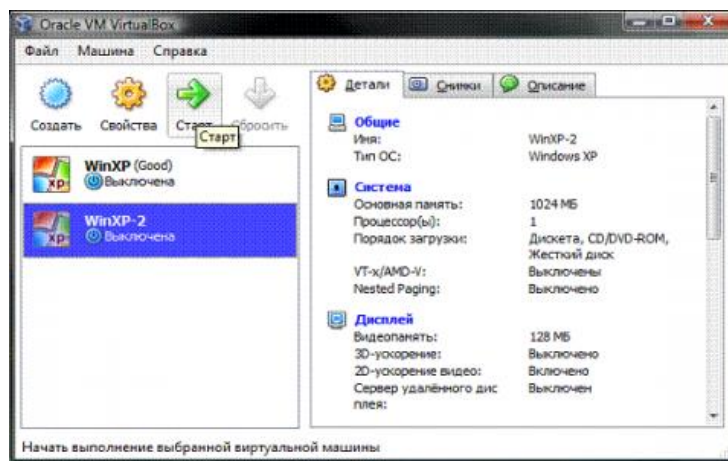


Рисунок 60

Начнется установка операционной системы Windows 7 Professional. После завершения установки Вы увидите следующее окно:

Для начала нам нужно установить драйверы для всех виртуальных аппаратных компонентов нашего виртуального ПК. Для этого в главном меню выбираем пункт «Устройства» - «Приводы оптических дисков» - «VboxGuestAdditions.iso». Впоследствии таким же образом Вы сможете подключить к своей виртуальной машине физический CD-ROM или загрузить ISO-образ.

После подключения образа «VboxGuestAdditions.iso» в папке Мой компьютер Вы увидите, что в привод компакт-дисков загружен данный виртуальный диск – остается его только запустить двойным щелчком левой кнопки мыши.

Сам процесс установки происходит практически без Вашего участия и только в случае, если Вы включили ранее 3D-ускорение, то следует выбрать соответствующий компонент для дополнительной установки.

В последнем окне процесса инсталляции Вам будет предложено перезагрузить виртуальную машину - соглашайтесь, после перезагрузки Вы увидите, что изображение стало четче, размер окна виртуальной машины изменяется динамически, включена функция интеграции мыши и есть доступ в Интернет.

Давайте теперь подключим общие папки, чтобы получить возможность перенести в созданную виртуальную машину нужные для работы файлы и установить приложения. Это можно сделать с помощью командной строки, следуя справке VirtualBox, но я приведу способ с использованием проводника Windows. Для этого откройте папку Мой компьютер, в главном меню выберите «Сервис» - «Подключить сетевой диск» и открывшемся окне в поле папка введите \\vboxsrv\имя_общей_папки, т.е. в нашем случае: \\vboxsrv\Share

После этих действий в папке «Мой компьютер» появится Ваша общая папка, доступная в качестве сетевого диска.

Давайте теперь проверим, есть ли у Вас доступ в Интернет. Для этого откройте: меню «Пуск» - «Программы» - «Стандартные» - «CMD-командная строка» и в открывшемся окне (рис. 31) введите следующую команду: Ping ya.ru

Если в результате отработки команды Вы видите, что пакеты отправлены и получены, пусть даже частично, то у Вас все получилось, и доступ в Интернет виртуальной машине обеспечен.

Быстрый доступ и комфортная работа

Далее хотелось бы сказать несколько слов о двух возможностях VirtualBox, которые помогут ускорить Вашу работу с виртуальными машинами и сделать ее комфортнее.

Ярлык для быстрого запуска виртуальной машины

Для более быстрого и удобного запуска Вашей виртуальной машины создадим ярлык именно для нее. Сделать это можно следующим образом:

- Щелчком правой кнопки мыши на рабочем столе вызовем контекстное меню и выберем пункт «Создать ярлык»;
- В открывшемся окне в поле «Укажите размещение объекта» введите "C:\Program Files\Oracle\VirtualBox\VBXManage.exe"
- startvm Win7Pro;
- В следующем окне введите имя ярлыка, например, «Win7Pro» и нажмите кнопку «Готово»;
- На Вашем рабочем столе появился созданный ярлык «Win7Pro», щелкните на нем правой кнопкой;

- f. В открывшемся контекстном меню выберите «Свойства»;
- г. В появившемся окне выберите сменить значок и поле выбора значков выберите любой значок, который посчитаете нужным.

Режим интеграции дисплеев

В режиме интеграции дисплеев Вы легко можете со своего рабочего стола организовать доступ к рабочему столу и элементам управления виртуальной машины, т.е. все окна, открываемые Вами в виртуальной машине, будут отображаться уже на Вашем рабочем столе, а не в отдельном окне VirtualBox. Данная функция значительно облегчает доступ к виртуальной машине, её элементам управления и, установленным на ней приложениям. Чтобы включить этот режим Вам нужно в главном меню окна визуализации VirtualBox выбрать пункт с соответствующим названием или нажать сочетание клавиш «HOST + L», где «HOST» клавиша – левый «Ctrl» (по умолчанию).

Программа выполнения работы

1. Скачайте платформу, подходящую под Вашу систему по ссылке: <http://www.virtualbox.org/wiki/Downloads>
2. Создайте на диске D: папку DISTRIB. С указанного преподавателем сетевого ресурса скачайте в эту папку ISO-образ операционной системы Windows 7 Pro (32 bit).
3. Запустите инсталлятор и выполните установку платформы Oracle VM VirtualBox.
4. Создайте новую виртуальную машину Win7Pro. Выберите для новой виртуальной машины объем оперативной памяти 1024 MB и создайте новый загрузочный жесткий диск фиксированного размера емкостью 25 GB на несистемном диске (диске D:) вашего компьютера.
5. Выполните настройку аппаратной части созданной виртуальной машины.
6. На несистемном диске (диске D:) вашего компьютера создайте папку для снимков.
7. Настройте буфер обмена между хостовой и виртуальной машинами как «двунаправленный».
8. Откорректируйте порядок загрузки. Первым обязательно поставьте CD/DVD-ROM, чтобы обеспечить возможность установки ОС с загрузочного диска. При этом в роли загрузочного диска может выступать как и компакт-диск, так и образ ISO.
9. На вкладке «Процессор» выберите количество процессоров, установленных на Вашу виртуальную материнскую плату. Обратите внимание, что это опция будет доступна только при условии поддержки аппаратной виртуализации AMD-V или VT-x, а также включенной опцииOI APIC на предыдущей вкладке. Здесь обратите Ваше внимание на настройки аппаратной визуализации AMD-V или VT-x. Перед включением этих настроек, нужно выяснить, поддерживает ли эти возможности Ваш процессор и включены ли они по умолчанию в BIOS (нередко они отключены).
10. В разделе «Дисплей» на вкладке «Видео» установите размер памяти виртуальной видеокарты, а также включите 2D и 3D ускорение, причем включение 2D ускорения желательно, а 3D необязательно.
11. В разделе «Носители» выделите позицию с надписью «Пусто» и добавьте ISO-образ загрузочного диска Windows 7.
12. В разделе USB поставьте оба доступных флажка, а затем, используя кнопку с изображением «вилки» USB и «плюса», добавьте все доступные контроллеры.
13. Запустите установку и установите операционную систему Windows 7 Pro.
14. Установите драйверы для всех виртуальных аппаратных компонентов нашего виртуального ПК. Для этого в главном меню выберите пункт «Устройства» - «Приводы оптических дисков» - «VboxGuestAdditions.iso». Впоследствии таким же образом Вы сможете подключить к своей виртуальной машине физический CD-ROM или загрузить ISO-образ. После подключения образа «VboxGuestAdditions.iso» в папке Мой компьютер Вы увидите, что в привод компакт-дисков загружен данный виртуальный диск – остается его только запустить двойным щелчком левой кнопки мыши.
15. Создайте ярлык для более быстрого и удобного запуска Вашей виртуальной машины.
16. Щелчком правой кнопки мыши на рабочем столе вызовите контекстное меню и выберите пункт «Создать ярлык»;
17. В открывшемся окне в поле «Укажите размещение объекта» введите "C:\Program Files\Oracle\VirtualBox\VBxManage.exe"
18. startvm Win7Pro;
19. В следующем окне введите имя ярлыка, например, «Win7Pro» и нажмите кнопку «Готово»;

20. На Вашем рабочем столе появился созданный ярлык «Win7Pro», щелкните на нем правой кнопкой;
21. В открывшемся контекстном меню выберите «Свойства»;
22. В появившемся окне выберите значок? Который посчитаете нужным.
23. Нажмите кнопку «ОК», а затем «Применить».
24. Включите режим интеграции дисплеев, выбрав в Главном меню пункт с соответствующим названием.
25. Оформите отчет по практической работе.

Практическая работа № 3.7. Создание эмуляторов. Подключение устройств. Настройка режима терминала

Цель работы: Знакомство с интерфейсом среды программирования. Изучение структуры проекта. Рассмотрите основные понятия Android проекта.

Структура проекта

- src—«исходный код» приложения (java-классы)
- assets —пустая директория. Может использоваться для сохранения raw-файлов.
- gen—хранилище генерируемых системных файлов. В частности, здесь располагается файл R.java, в котором хранятся идентификаторы всех ресурсов, создаваемых в проекте (строковые ресурсы и т.п.).
- libs—различные библиотеки, используемые приложением
- res—ресурсы проекта.
- AndroidManifest.xml—файл описания проекта (поддерживаемые версии SDK, версия приложения и т.п.)
- project.properties—файл, включающий настройки проекта, такие как build target.

Ресурсы проекта

- anim/ Содержит XML файлы, компилируемые в анимационные объекты.
- color/ Содержит XML файлы описывающие цвета.
- drawable/ Содержит растровые файлы (PNG, JPEG, orGIF), 9-Patch
- файлы, и XML файлы, описывающие Drawableshapes или Drawableobjects включающие
- множественные состояния(нормальное, нажатое, состояние фокуса).
- layout/ Содержит XML файлы описывающие макеты экрана
- menu/ Содержит XML файлы, определяющие меню приложения.
- raw/ Для хранения произвольных файлов.
- values/ Содержит XML файлы, компилируемые во множество видов ресурсов (strings.xml и т.д).
- xml/ СодержитXML файлы конфигурирующие компоненты приложения. Например, XML файл определяющий экран настроек.

Пример простейшего файла AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.untitled"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="19"/>
    <application
        android:label="@string/app_name"
        android:icon="@drawable/ic_launcher">
        <activity android:name="MyActivity"
            android:label="@string/app_name">
            <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```


Для начала работы требуется:

- Java Development Kit
- Android Software Development Kit

Android Virtual Device Manager

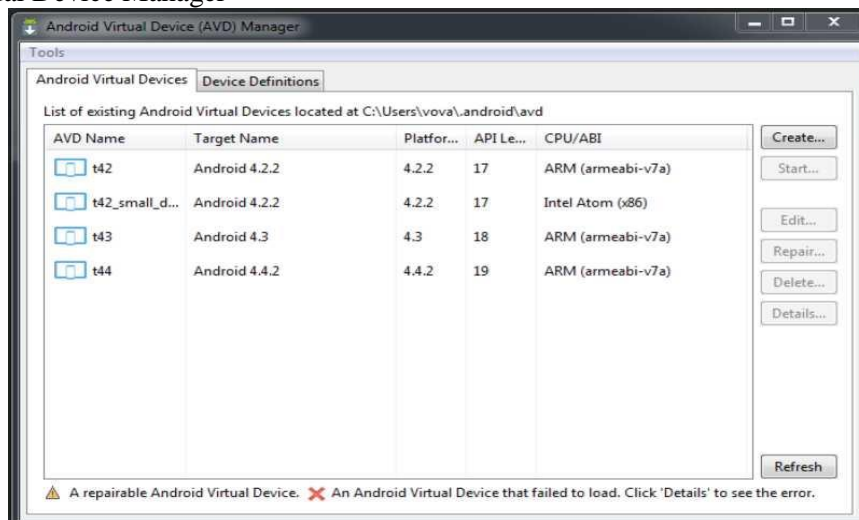


Рисунок 61

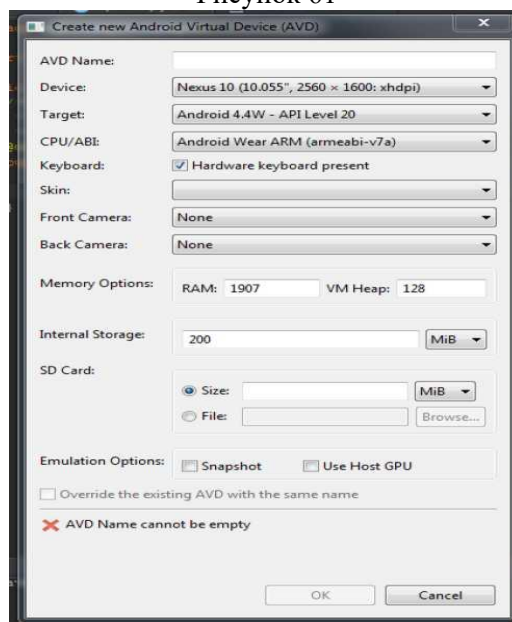


Рисунок 62

Задание 1. Создание проекта приложения

Запустите среду программирования в IDE IntelliJ Idea

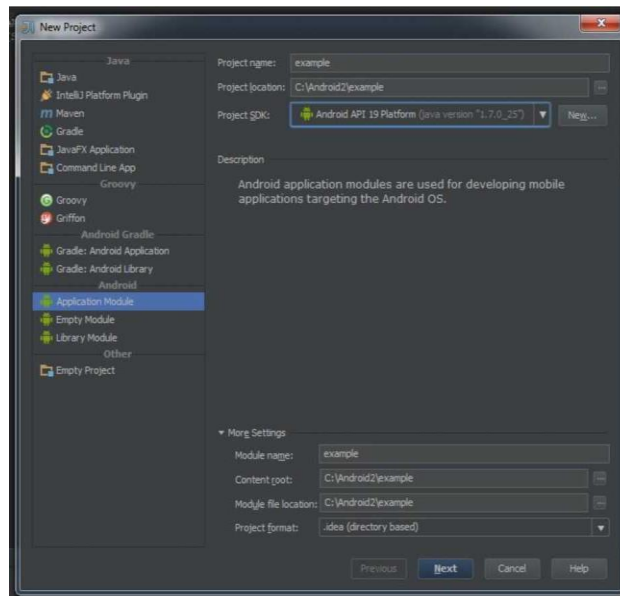


Рисунок 63

Задание 2. Создание приложений с одним экраном (Activity) Необходимо создать два activity и организовать переход между ними. Содержимое Activity 1 кнопка с именем btn1

Содержимое Activity 2: TextView с текстом «Переданный параметр: значение_параметра». значение параметра - принятый параметр из Activity 1.

При запуске приложения пользователь должен попадать на экран с Activity 1. После нажатия на кнопку btn1 необходимо осуществить переход к Activity 2 и передавать параметр из Activity 1. В качестве значения передаваемого параметра использовать свою фамилию.

Ключевые классы: Activity, Intent, Button, TextView.

Практическая работа № 3.8. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна

Цель работы: Изучить работу Android приложения с базой данных.

SQLite

- SQLite—компактная встраиваемая реляционная база данных с открытым исходным кодом. Поддерживает динамическое типизирование данных.
- Возможные типы полей: INTEGER, REAL, TEXT, BLOB.

SQLiteOpenHelper

Абстрактные методы

- onCreate() -вызывается при первом создании базы данных;
- onUpgrade() -вызывается при модификации базы данных

Реализация метода onCreate

```
public void onCreate( SQLiteDatabase db) {
    db.execSQL( "CREATE TABLE"+ TABLE_NAME
    + "( _id INTEGER PRIMARY KEY AUTOINCREMENT , "
    + COL NAME + " TEXT , "+ COL PHONE + " TEXT ) ; " );
}
```

Реализация метода onUpgrade

@ Override

```
public void onUpgrade( SQLiteDatabase db, int oldVersion, int newVersion)
{
    db.execSQL( "DROP TABLE IF EXISTS"+ TABLE_NAME );
    onCreate( db );
}
```

Чтение из базы данных

```
Cursor query( String table, String [ ] columns, String
selection, String [ ] selectionArgs,
String groupBy, String having , String sortOrder)
```

Позиционирование курсора

1. moveToFirst() -перемещает курсор в первую запись в выборке;

2. `moveToLast()` -перемещает курсор в последнюю запись в выборке;
3. `moveToNext()` -перемещает курсор в следующую запись и одновременноопределяет, существует ли эта запись. Метод `moveToNext()` возвращает `True`, если курсор указывает на другую строку после перемещения, и `false` если текущая запись была последней в выборке;
4. `moveToPrevious()` -перемещает курсор в предыдущую запись;
5. `moveToPosition()` -перемещает курсор в указанную позицию;
6. `getPosition()` -возвращает текущий индекс позиции курсора.
7. `isFirst()` ;
8. `isLast()` ;
9. `isBeforeFirst()` ;
10. `isAfterLast()` .

Обновление и удаление записей

11. `intupdate(String table, ContentValuesvalues, String whereClause, String [] whereArgs)`
12. `intdelete (String table, String whereClause, String [] whereArgs)`

Задание 1. Необходимо создать приложение, взаимодействующее с базой данных. Первое активности должно содержать три кнопки. При нажатии на первую кнопку должно открываться новое активности, выводящее информацию из таблицы «Одногруппники» в удобном для восприятия формате.

При запуске приложения необходимо:

1. Создать БД, если ее не существует.
2. Создать таблицу «Одногруппники», содержащую поля:
3. ID;
4. ФИО;
5. Время добавления записи.
6. Удалять все записи из БД, а затем вносить 5 записей об одногруппниках.

При нажатии на вторую кнопку необходимо внести еще одну запись в таблицу.

При нажатии на третью кнопку необходимо заменить ФИО в последней внесенной записи на Иванов Иван Иванович.

Задание 2. Создать новое отдельное приложение на основе приложения, созданного в части 1.

Переопределить функцию `onUpgrade`. При изменении версии БД необходимо удалить таблицу «Одногруппники», создать таблицу «Одногруппники» содержащую следующие поля:

1. ID;
2. Фамилия;
3. Имя;
4. Отчество;
5. Время добавления записи.
6. Изменить версию базы данных.

Задание 3.

1. Создайте новый проект `MetroPicker`.
2. Добавьте вспомогательную Активность `ListViewActivity` для отображения и выбора станций метро, в качестве заготовки используйте результаты лабораторной работы «Использование `ListView`» .
3. Отредактируйте файл разметки `res/layout/main.xml`: добавьте кнопку выбора станции метро, присвоив идентификаторы виджетам `TextView` и `Button` для того, чтобы на них можно было ссылаться в коде.
4. Установите обработчик нажатия на кнопку в главной Активности для вызова списка станции и выбора нужной станции.
5. Напишите нужный обработчик для установки выбранной станции метро в виджет `TextView` родительской Активности (метод `setText` виджета `TextView` позволяет установить отображаемый текст). Не забудьте обработать ситуацию, когда пользователь нажимает кнопку «Назад» (в этом случае «никакой станции не выбрано» и главная Активность должна известить об этом пользователя).
6. Убедитесь в работоспособности созданного приложения, проверив реакцию различные действия потенциальных пользователей.
7. Неявные намерения

8. Неявные намерения используются для запуска Активностей для выполнения заказанных действий в условиях, когда неизвестно (или безразлично), какая именно Активность (и из какого приложения) будет использоваться.
 9. При создании Намерения, которое в дальнейшем будет передано методу `startActivity`, необходимо назначить действие (action), которое нужно выполнить, и, возможно, указать URI данных, которые нужно обработать. Также можно передать дополнительную информацию с помощью свойства `extras` Намерения. Android сам найдет подходящую Активность (основываясь на характеристиках Намерения) и запустит ее. Пример неявного вызова телефонной «звонилки»:
 10. `Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:(495)502-99-11"));`
 11. `startActivity(intent);`
 12. Для определения того, какой именно компонент должен быть запущен для выполнения действий, указанных в Намерениях, Android использует Фильтры Намерений (Intent Filters'). Используя Фильтры Намерений, приложения сообщают системе, что они могут выполнять определенные действия (action) с определенными данными (data) при определенных условиях (category) по заказу других компонентов системы.
 13. Для регистрации компонента приложения (Активности или Сервиса) в качестве потенциального обработчика Намерений, требуется добавить элемент `<intent-filter>` в качестве дочернего элемента для нужного компонента в Манифесте приложения. У элемента `<intent-filter>` могут быть указаны следующие дочерние элементы (и соответствующие атрибуты у них):
 14. `<action>`. Атрибут `android:name` данного элемента используется для указания названия действия, которое может обслуживаться. Каждый Фильтр Намерений должен содержать не менее одного вложенного элемента `<action>`. Если не указать действие, ни одно Намерение не будет «проходить» через этот Фильтр. У главной Активности приложения в Манифесте должен быть указан Фильтр Намерений с действием `android.intent.action.MAIN`
 15. `<category>`. Сообщает системе, при каких обстоятельствах должно обслуживаться действие (с помощью атрибута `android:name`). Внутри `<intent-filter>` может быть указано несколько категорий. Категория `android.intent.category.LAUNCHER` требуется Активности, которая желает иметь «иконку» для запуска. Активности, запускаемые с помощью метода `startActivity`, обязаны иметь категорию `android.intent.category.DEFAULT`
 16. `<data>`. Дает возможность указать тип данных, которые может обрабатывать компонент. `<intent-filter>` может содержать несколько элементов `<data>`. В этом элементе могут использоваться следующие атрибуты:
 17. `android:host` : имя хоста (например, `www.specialist.ru`)
 18. `android:mimeType` : обрабатываемый тип данных (например, `text/html`)
 19. `android:path` : «путь» внутри URI (например, `/course/android`)
 20. `android:port` : порт сервера (например, 80)
 21. `android:scheme` : схема URI (например, `http`)
- Пример указания Фильтра Намерений:
- ```
<activity android:name=".MyActivity" android:label="@string/app_name" > <intent-filter>
<action android:name="android.intent.action.SEND" />
<category android:name="android.intent.category.DEFAULT" />
<data android:mimeType="text/plain" />
</intent-filter>
</activity>
```
22. При запуске Активности с помощью метода `startActivity` неявное Намерение обычно подходит только одной Активности. Если для данного Намерения подходят несколько Активностей, пользователю предлагается список вариантов.
  23. Определение того, какие Активности подходят для Намерения, называется Intent Resolution. Его задача - определить наиболее подходящие Фильтры Намерений, принадлежащие компонентам установленных приложений. Для этого используются следующие проверки в указанном порядке:
  24. Проверка действий. После этого шага остаются только компоненты приложений, у которых в Фильтрах Намерений указано действие Намерения. В случае, если действие в Намерении отсутствует, совпадение происходит для всех Фильтров Намерений, у которых указано хотя бы одно действие.

25. Проверка категорий. Все категории, имеющиеся у Намерения, должны
26. присутствовать в Фильтре Намерений. Если у Намерения нет категорий, то на данном этапе ему соответствуют все Фильтры Намерений, за одним исключением, упоминавшимся выше: Активности, запускаемые с помощью метода `startActivity`, обязаны иметь категорию `android.intent.category.DEFAULT`, так как Намерению, использованному в этом случае, по умолчанию присваивается данная категория, даже если разработчик не указал ничего явно. Из этого исключения, в свою очередь, есть исключение: если у Активности присутствуют действие
27. `android.intent.action.MAIN` и категория `android.intent.category.LAUNCHER`, ему не требуется иметь категорию `android.intent.category.DEFAULT`.
28. Проверка данных. Здесь применяются следующие правила:
29. Намерение, не содержащее ни URI, ни типа данных, проходит через Фильтр, если он тоже ничего перечисленного не содержит.
30. Намерение, которое имеет URI, но не содержит тип данных (и тип данных невозможно определить по URI), проходит через Фильтр, если URI Намерения совпадает с URI Фильтра. Это справедливо только в случае таких URI, как `mailto:` или `tel:`, которые не ссылаются на реальные данные.
31. Намерение, содержащее тип данных, но не содержащее URI подходит только для аналогичных Фильтров Намерений.
32. Намерение, содержащее и тип данных, и URI (или если тип данных может быть вычислен из URI), проходит этот этап проверки, только если его тип данных присутствует в Фильтре. В этом случае URI должен совпадать, либо(!) у Намерения указан URI вида `content:` или `file:`, а у Фильтра URI не указан. То есть, предполагается, что если у компонента в Фильтре указан только тип данных, то он поддерживает URI вида `content:` или `file:`. В случае, если после всех проверок остается несколько приложений, пользователю предлагается выбрать приложение самому. Если подходящих приложений не найдено, в выпустившей Намерение Активности возникает Исключение.

Задание 5.

Модифицируйте проект `MetroPicker` следующим образом:

1. Добавьте главное меню в Активность, отображающую список станций метро. В меню должен быть один пункт: «вернуться». Меню создайте динамически в коде, без использования строковых ресурсов.
2. Динамически создайте контекстное меню для Представления `TextView`, отображающего выбранную станцию метро главной Активности. Выбор пункта меню должен сбрасывать выбранную станцию.
3. Для главной Активности создайте основное меню из двух пунктов: «сбросить» и «выйти». Реализуйте нужные функции при выборе этих пунктов.
4. Повторите реализацию п.п. 1, 2 и 3 с помощью ресурсов, описывающих меню.

Задание 6.

1. Создайте новый Android проект `ListViewSample`.
2. В каталоге `res/values` создайте файл `arrays.xml` с данными полученными у преподавателя.
3. В каталоге `res/layout` создайте файл `list_item.xml` с данными полученными у преподавателя.
4. Модифицируйте метод `onCreate` вашей Активности:

```
@Override public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 Resources r = getResources();
 String[] stationsArray = r.getStringArray(R.array.stations);
 ArrayAdapter<String> aa = new ArrayAdapter<String>(this, R.layout.list_item, stationsArray);
 setListAdapter(aa);
 ListView lv = getListView();
}
```
5. Измените базовый класс Активности с `Activity` на `ListActivity`.
6. Запустите приложение.
7. Для реакции на клики по элементам списка требуется добавить обработчик такого события, с помощью метода `setOnClickListener`. В качестве обработчика будет использоваться анонимный объект класса `OnClickListener`. Добавьте следующий код

в нужное место:

```
lv.setOnItemClickListener(new OnItemClickListener() {
 public void onItemClick(AdapterView<?> parent, View v,
 int position, long id) {
 CharSequence text = ((TextView) v).getText();
 int duration = Toast.LENGTH_LONG;
 Context context = getApplicationContext();
 Toast.makeText(context, text, duration).show();
 }
});
```

8. Запустите приложение и «покликайте» по станциям метро.

Задание 7. «Использование управляющих элементов в пользовательском интерфейсе»

Подготовка

1. Создайте новый проект ControlsSample.

2. Отредактируйте файл res/layout/main.xml так, чтобы остался только корневой элемент LinearLayout. В него в дальнейшем будут добавляться необходимые дочерние элементы:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout
width="fill parent" android:layout height="fill parent" android:orientation="vertical" >
</LinearLayout>
```

Использование графической кнопки

Для использования изображения вместо текста на кнопке потребуются три изображения для трех состояний кнопки: обычное, выбранное («в фокусе») и нажатое. Все эти три изображения с соответствующими состояниями описываются в одном XML файле, который используется для создания такой кнопки.

1. Скопируйте нужные изображения кнопки в каталог res/drawable-mdpi, для обновления списка содержимого каталога в Eclipse можно использовать кнопку F5.

2. В этом же каталоге создайте файл smile\_button.xml, описывающий, какие изображения в каких состояниях кнопки нужно использовать:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item android:drawable="@drawable/smile_pressed" android:state_pressed="true"/>
<item android:drawable="@drawable/smile_focused" android:state_focused="true"/>
<item android:drawable="@drawable/smile_normal" />
</selector>
```

3. Добавьте элемент Button внутри LinearLayout в файле разметки res/layout/main.xml:

```
<Button
 android:id="@+id/button"
 android:layout_width="wrap_content" android:layout_height="wrap_content"
 android:background="@drawable/smile_button" android:onClick="onButtonClicked"
 android:padding="10dp" />
```

4. Обратите внимание на атрибут android:onClick= "onButtonClicked", указывающий, какой метод из Активности будет использоваться как обработчик нажатия на данную кнопку. Добавьте этот метод в Активность:

```
public void onButtonClicked(View v) {
 Toast.makeText(this, "Кнопка нажата", Toast.LENGTH_SHORT).show();
}
```

5. Запустите приложение и посмотрите, как изменяется изображение кнопки в разных состояниях, а также как функционирует обработчик нажатия на кнопку.

Использование виджета CheckBox

1. Добавьте элемент CheckBox внутри LinearLayout в файле разметки res/layout/main.xml:

```
<CheckBox
 android:id="@+id/checkbox"
 android:layout_width="wrap_content" android:layout_height="wrap_content"
 android:onClick="onCheckboxClicked"
 android:text="Выбери меня" />
```

2. Атрибут `android:onClick= "onCheckboxClicked"` определяет, какой метод из Активности будет использоваться как обработчик нажатия на виджет. Добавьте этот метод в Активность:

```
public void onCheckboxClicked(View v) { if (((CheckBox) v).isChecked()) {
 Toast.makeText(this, "Отмечено", Toast.LENGTH_SHORT).show();
} else {
 Toast.makeText(this, "Не отмечено", Toast.LENGTH_SHORT).show();
}
}
```

3. Запустите приложение и посмотрите на поведение чекбокса в разных ситуациях.

Использование виджета `ToggleButton`

Данный виджет хорошо подходит в качестве альтернативы радиокнопкам и чекбоксам, когда требуется переключаться между двумя взаимоисключающими состояниями, например, Включено/Выключено.

1. Добавьте элемент `ToggleButton` внутри `LinearLayout` в файле разметки `res/layout/main.xml`:

```
<ToggleButton android:id= "@+id/togglebutton" android:layout_width= "wrap_content"
android:layout_height="wrap_content" android:textOn="Звонок включен" android:textOff="Звонок
выключен" android:onClick= "onToggleClicked"/>
```

2. Атрибут `android:onClick= "onToggleClicked"` определяет, какой метод ИЗ Активности будет использоваться как обработчик нажатия на виджет. Добавьте этот метод в Активность:

```
public void onToggleClicked(View v) {
 if (((ToggleButton) v).isChecked()) {
 Toast.makeText(this, "Включено", Toast.LENGTH_SHORT).show();
 } else {
 Toast.makeText(this, "Выключено", Toast.LENGTH_SHORT).show();
 }
}
```

3. Запустите приложение и проверьте его функционирование.

Использование виджета `RadioButton`

Радиокнопки используются для выбора между различными взаимоисключающими вариантами. Для создания группы радиокнопок используется элемент `RadioGroup`, внутри которого располагаются элементы `RadioButton`.

1. Добавьте следующие элементы разметки внутри `LinearLayout` в файле `res/layout/main.xml`:

```
<RadioGroup
 android:layout_width="fill_parent"
 android:layout_height="wrap_content"
 android:orientation="vertical" >
 <RadioButton
 android:id= "@+id/radio_dog"
 android:layout_width= "wrap_content" android:layout_height= "wrap_content"
 android:onClick="onRadioButtonClicked" android:text="Собачка" />
 <RadioButton
 android:id="@+id/radio_cat"
 android:layout_width="wrap_content" android:layout_height="wrap_content"
 android:onClick="onRadioButtonClicked" android:text="Кошечка" />
 <RadioButton
 android:id= "@+id./radio_rabbit" android:layout_width="wrap_content" android:layout
 height="wrap content" android:onClick="onRadioButtonClicked
 android:text="Кролик" />
</RadioGroup>
```

2. Добавьте метод `onRadioButtonClicked` в Активность:

```
public void onRadioButtonClicked(View v) { RadioButton rb = (RadioButton) v;
 Toast. makeText(this, "Выбрано животное: " + rb.getText(),
 Toast.LENGTH_SHORT).show();
}
```

3. Проверьте работу приложения.

Использование виджета `EditText`



Виджет EditText используется для ввода текста пользователем. Установленный для этого виджета обработчик нажатий на кнопки будет показывать введенный текст с помощью Toast.

1. Добавьте элемент EditText внутри LinearLayout в файле разметки

res/layout/main.xml:

```
<EditText
 android:id="@+id/user_name"
 android:layout_width="fill_parent"
 android:layout_height="wrap_content"
 android:hint="Введите имя"/>
```

2. Для обработки введенного пользователем текста добавьте следующий код в конце метода onCreate. Обратите внимание, этот обработчик, в отличие от предыдущих, использованных нами, возвращает значение true или false. Семантика этих значений традиционна: true означает, что событие (event) обработано и больше никаких действий не требуется, false означает, что событие не обработано этим обработчиком и будет передано следующим обработчикам в цепочке. В нашем случае реагирование происходит только на нажатие (ACTION\_DOWN) кнопки Enter (KEYCODE\_ENTER):

```
final EditText userName = (EditText) findViewById(R.id.user_name);
userName.setOnKeyListener(new View.OnKeyListener() {
 @Override
 public boolean onKeyDown(View v, int keyCode, KeyEvent event) {
 if ((event.getAction() == KeyEvent.ACTION_DOWN)
 && (keyCode == KeyEvent.KEYCODE_ENTER)) {
 Toast.makeText(getApplicationContext(), userName.getText(),
 Toast.LENGTH_SHORT).show();
 return true;
 }
 return false;
 }
});
```

3. Запустите приложение и проверьте его работу.

4. Добавьте кнопку «Очистить» в разметку и напишите обработчик, очищающий текстовое поле (используйте метод setText виджета EditText)

5. Проверьте работу приложения.

### **Практическая работа № 3.9. Обработка событий: подсказки. Обработка событий: цветовая индикация. Подготовка стандартных модулей. Обработка событий: переключение между экранами**

Цель работы: Создание обработчиков событий. Изучить работу с потоками. Научиться работать с мультимедиа файлами. Изучить работу с классом AsyncTask

Главный и обычный потоки

Асинхронные потоки в Android

AsyncTask-это специальный абстрактный класс, предоставляющий набор методов для реализации:

- onPreExecute-для размещения инициализирующего кода (UI поток)
- doInBackground-для размещения тяжелого кода, который будет выполняться в другом потоке
- onProgressUpdate-для информирования о прогрессе (UI поток)
- onPostExecute-для обработки результата, возвращенного doInBackground(UI поток) и вспомогательных методов
- isCancelled-для получения информации об отмене задачи
- publishProgress-для перевода сообщения о прогрессе в UI поток с последующим вызовом onProgressUpdate

Последовательность выполнения методов AsyncTask onPreExecute doInBackground onPostExecute publishProgress onProgressUpdate

publishProgress

onProgressUpdate

Правила использования AsyncTask:

Объект AsyncTask должен быть создан в UI-потоке

- Метод execute должен быть вызван в UI-потоке
- Метод execute может быть запущен только один раз
- Не вызывайте методы onPreExecute, doInBackground, onPostExecute и onProgressUpdate

Передача данных в AsyncTask

Объявляем класс

```
Class MyAsyncTask extends AsyncTask<String, Integer, Long>{
}
```

• Первый параметр используется методом doInBackground protectedLong doInBackground(String... urls)

• Второй параметр используется методом onProgressUpdate protectedvoid onProgressUpdate(Integer... progress)

• Третий параметр используется методом onPostExecute

```
protectedvoid onPostExecute(Longresult)
```

Промежуточные данные

Последовательность действий для передачи промежуточных данных в основной поток программы:

- В методе doInBackground вызываем метод publishProgress

- В методе onProgressUpdate обрабатываем переданный в publishProgress параметр и выводим прогресс

Метод get

- Возвращает результат выполнения метода doInBackground

- Вызывается из UI потока

```
MyAsyncTask at = new MyAsyncTask();
```

```
result = at.get();
```

Задание 1. Разработать мобильное приложение, состоящее из четырех activity.

После запуска приложения пользователь должен попадать на экран с activity1. На данном экране должно быть представлено меню, состоящее из четырех кнопок. Высота кнопок должна составлять 20% от высоты экрана. Расстояние между кнопками - 2%. Первая и последняя кнопка должны быть на равном расстоянии от краев экрана. Ширина кнопок 75%, выравнивание по середине.

После нажатия на первую кнопку пользователь должен переходить к activity2, его внешний вид представлен на рисунке 64. Верстка должна осуществляться с использованием LinearLayout, ширина кнопок должна задаваться в процентах от ширины экрана.



Рисунок 64

После нажатия на вторую кнопку в activity1 пользователь должен переходить к activity3, его внешний вид представлен на рисунке 65. Верстка должна осуществляться с использованием RelativeLayout (не использовать LinearLayout).

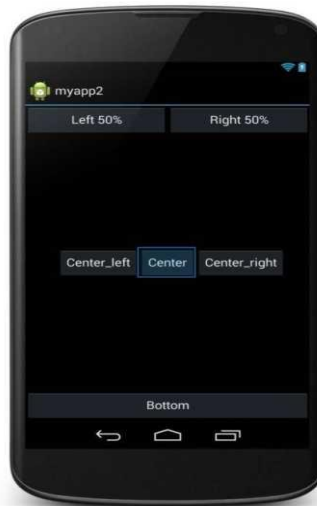


Рисунок 65

Третья кнопка в activity1 должна создавать activity3. Внешний вид activity3 представлен на рис.66



Рисунок 66

Кнопка должна быть выровнена по центру экрана. Цвет обводки кнопки #505050. Толщина обводки в соответствии с месяцем вашего рождения (от 1 до 12). Радиус скругления 24dp. Цвет фона экрана #FFFFFF. При нажатии на кнопку ее цвет должен изменяться на светло-зеленый.

Нажатие на четвертую кнопку в activity1 должно приводить к закрытию приложения.

Задание 2. Рассмотрите пример передачи данных. `MyAsyncTaskat = newMyAsyncTask();`

`at.execute("url1", "url2");`

`doInBackground(String... urls)`

Задание 3. Рассмотрите пример вывода промежуточных данных.

`@Override`

`Protected void doInBackground(String... urls) { try{`

`int cnt= 0;`

`for(Stringurl: urls) {`

`// обрабатываем первый параметр`

`// выводим промежуточные результаты`

`cnt++;`

`publishProgress(cnt);`

`}`

`TimeUnit.SECONDS.sleep(1);`

`} catch(InterruptedExceptione) {`

`e.printStackTrace();`

`}`

`return null;`

`}`

```
@Override
protected void onProgressUpdate(Integer... values) {
 super.onProgressUpdate(values); tv.setText("обработано " + values[0] + " параметров");
}
```

Задание 4. Проверьте пример создания простой асинхронной задачи

```
public class MainActivity extends Activity {
 MyAsyncTask at;
 TextView tv;
 public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.main);
 tv = (TextView) findViewById(R.id.tv);
 MyAsyncTask at = new MyAsyncTask();
 at.execute();
 }
}
```

Задание 5. На основании изученных примеров разработать приложение сохраняющее статистику проигрываемых песен на радио Мегабайт. Для сохранения песни и названия необходимо создать базу данных, содержащую таблицу со следующими полями:

- ID
- Исполнитель
- Название трека
- Время внесения записи

При включении приложения необходимо производить проверку подключения к Интернету. В случае если подключение отсутствует - выводить всплывающее сообщение (Toast) с предупреждением о запуске в автономном режиме (доступен только просмотр внесенных ранее записей).

После включения приложение должно производить асинхронный опрос сервера с интервалом 20 секунд. Если название трека не совпадает с последней записью в таблице необходимо произвести запись в БД.

URL адрес, по которому можно получить информацию о текущем треке и исполнителе:

[http://media.ifmo.ru/api\\_get\\_current\\_song.php](http://media.ifmo.ru/api_get_current_song.php)

Формат возвращаемых данных - JSON.

В случае успешного выполнения запроса результат будет иметь вид:

```
{ "result": "success", "info" : "Исполнитель - Название трека" }
```

В случае ошибки API вернет следующую строку:

```
{ "result": "error", "info" : "Информация об ошибке" }
```

Для успешного взаимодействия с API при обращении к странице необходимо передавать логин (login) и пароль (password) как POST-параметры.

login: 4707login

password: 4707pass

Приложение должно содержать активности, позволяющие просматривать внесенные в базу данных записи.

Задание 6. Работа с внешними файлами.

Разработать мобильное приложение, позволяющее пользователю асинхронно скачивать файлы журнала Научно-технический вестник. Файлы хранятся на сервере в формате PDF и расположены по адресу: [http://ntv.ifmo.ru/file/journal/идентификатор\\_журнала.файл](http://ntv.ifmo.ru/file/journal/идентификатор_журнала.файл)

Не для всех ID имеются журналы, поэтому необходимо предусмотреть сообщение об отсутствии файла. В случае если файл не найден, ответ от сервера будет содержать главную страницу сайта.

Определить существует ли файл можно по возвращаемому сервером заголовку (параметр content-type).

Примеры ссылок:

<http://ntv.ifmo.ru/file/journal/1.pdf> - возвращен PDF файл

<http://ntv.ifmo.ru/file/journal/2.pdf> - файл не найден, возвращена главная страница сайта

Файлы должны храниться на устройстве в папке, создаваемой при первом запуске приложения (путь до папки и ее название определите самостоятельно).

После окончания загрузки файла должна становиться доступной кнопка «Смотреть» и кнопка «Удалить».

При нажатии на кнопку «Смотреть» должно происходить открытие сохраненного на устройстве файла. Предусмотреть ошибку, если на устройстве не установлено приложение, открывающее PDF файлы. При нажатии на кнопку «Удалить» загруженный файл должен удаляться с устройства.

### **Практическая работа № 3.10. Передача данных между модулями. Тестирование мобильного приложения. Оптимизация мобильного приложения пошаговый алгоритм тестирования мобильных приложений**

Цель работы: Провести тестирование созданного приложения и выполнить его оптимизацию

Обеспечение качества (QA, от английского - Quality Assurance) является неотъемлемой частью жизненного цикла разработки любых приложений, включая мобильные. К сожалению, многие упускают из виду критические особенности тестирования мобильных приложений, которые часто приводят к сбоям, ошибкам в работе приложения и плохому качеству обслуживания клиентов.

Чтобы обеспечить успешную разработку любого приложения, специалист-тестировщик должен принимать участие во всех этапах разработки - от создания концепции и анализа требований, до создания спецификаций тестирования и выпуска готового продукта. Обеспечение качества также является ключевым элементом в последующих, после прохождения этапов разработки, обзорах программного продукта.

Однако часто бывает сложно определить, с чего начать организацию процесса тестирования мобильного приложения. Для беспрепятственного тестирования мы рекомендуем просто выполнить девять указанных ниже шагов.

Давайте рассмотрим особенности тестирования мобильных приложений.

Цикл жизни спринтов

Этап 1: Планирование

Когда этап разработки приложения почти завершен, вы должны снова поставить перед собой вопрос - чего вы пытаетесь достичь разработкой данного приложения и какие у вас есть ограничения.

Вы должны определить следующее:

- Взаимодействует ли ваше приложение с другими приложениями?
- Насколько функциональны все возможности приложения?
- Является ли тестируемое мобильное приложение нативным, Mobile- web или гибридным?
- Ограничена ли задача тестирования приложения тестированием только внешнего интерфейса?
- Стоят ли задачи на тестирование бэкенда?
- Какова должна быть совместимость с различными беспроводными сетями?
- Как сильно данные приложения и свободное пространство, занимаемое им, зависят от особенностей использования приложения?
- Насколько быстро загружается ваше приложение, насколько быстро происходит серфинг по меню приложения и его функциям?
- Как будет обрабатываться возможное увеличение нагрузки на приложение?
- Влияют ли различные изменения в статусе и состоянии телефона на работу мобильного приложения?

Убедитесь, что вы договорились с командой тестировщиков о роли каждого из них и о ваших ожиданиях от процесса тестирования. В конце концов, общение является ключом к поддержанию правильной рабочей среды в команде.

Правильное понимание ролей и задач также относится и к моменту прописывания списка тест кейсов. Вся команда QA должна поддерживать и обновлять этот документ с отчетами по тестированию всех функций, реализованных на протяжении всего процесса разработки.

Этап 2. Определение необходимых типов тестирования мобильных приложений

Перед тестированием любых мобильных приложений определите, что именно в данном мобильном приложении вы хотите протестировать: набор функциональности, удобство использования, совместимость, производительность, безопасность и т. д. На этом же этапе имеет смысл выбрать методы тестирования мобильного приложения.

Определите, на какие целевые устройства направлено данное приложение, и какие требования к функционалу следует проверить.

Вы также должны определить, какие целевые устройства нужно включить в список тестирования.

Вы можете сделать это следующим образом:

- Выяснить, какие устройства будет поддерживать приложение;
- Определить, какая версия операционной системы будет самой ранней из тех, что поддерживаются приложением;
- Выявить наиболее популярные модели мобильных устройств у целевой аудитории;
- Определить набор не основных (дополнительных) устройств с экранами разных размеров, потенциально поддерживаемых приложением;
- Решить, будете ли вы использовать для тестирования физические устройства или их эмуляторы.

#### Этап 3: Тестовые случаи и разработка сценариев тестирования приложения

Подготовьте документ, описывающий тестовые случаи (test cases) для каждой тестируемой функции и функциональности.

Также перед началом тестирования важно определиться, какое сочетание ручного и автоматического тестирования вы будете применять.

При необходимости подготовьте отдельные наборы ручных тестовых случаев и сценариев для автоматического тестирования и адаптируйте их согласно требованиям проекта.

#### Этап 4: Ручное и автоматическое тестирование

Теперь пришло время для выполнения ручных и автоматизированных тестов. Ранее, на предыдущих этапах, вы уже определили, какие тесты и скрипты использовать и подготовили их. Теперь, на текущем этапе, вы выполняете запуск тестов для проверки механизмов основной функциональности, чтобы убедиться в отсутствии поломок.

Автоматизированное тестирование мобильных приложений хорошо экономит время и другие ресурсы тестировщиков.

#### Этап 5: Тестирование юзабилити и бета-тестирование

После того, как базовый функционал протестирован, настало время убедиться, что мобильное приложение является достаточно простым в использовании и обеспечивает удовлетворительный пользовательский опыт. На этом этапе необходимо поддерживать соответствие матрице кроссплатформенности, чтобы обеспечить охват пользователей различных платформ, достигнутый бета-тестерами.

##### Пример матрицы поддержки разных версий платформы iOS

После того, как приложение будет протестировано внутри компании, вы сможете выпустить бета-версию приложения на рынок.

##### Тестирование совместимости

Мобильные устройства различаются в зависимости от платформы, модели и версии их операционной системы. Важно выбрать такое подмножество устройств, которое будет соответствовать вашему приложению.

##### Тестирование пользовательского интерфейса

Пользовательский опыт является ключевым элементом, при тестировании приложения. Ведь наше приложение разрабатывается именно для конечных пользователей. Вам следует качественно проверить удобство использования приложения, навигацию по его элементам и контент. Тестируйте меню, опции, кнопки, закладки, историю, настройки и навигацию приложения.

##### Тестирование интерфейса

Тестирование пунктов меню, кнопок, закладок, истории, настроек и навигации по приложению.

##### Тестирование внешних факторов

Приложения для мобильных устройств не будут единственными приложениями на устройстве пользователя. Вместе с вашим приложением будут установлены приложения от сторонних разработчиков. Возможно десятки таких приложений. Следовательно, вашему приложению придется взаимодействовать с этими сторонними приложениями и прерывать работу различных функций устройства, таких как различные типы сетевых подключений, обращение к SD-карте, телефонные звонки и другие функции устройства.

##### Тестирование доступности

Мобильными устройствами могут пользоваться различные люди с ограниченными возможностями. По этой причине важно протестировать возможность работы с приложением людей с дальтонизмом, нарушениями слуха, проблемами пожилого возраста и другими возможными проблемами. Такое тестирование является важной частью общего тестирования юзабилити.

#### Этап 6: Тестирование производительности

Мобильные устройства предоставляют для приложений меньший объем памяти и меньшую доступную мощность процессора, чем стационарные компьютеры и ноутбуки. По этой причине в

работе мобильных приложений очень важна эффективность использования предоставляемых ресурсов. Вам следует проверить работоспособность тестируемого приложения, изменив соединение с 2G, 3G на WIFI, проверить скорость отклика, потребление заряда батареи, стабильность работы и т. д.

Рекомендуется проверять приложение на предмет масштабируемости применения и наличие возможных проблем с производительностью.

В рамках этого этапа важно пройти и нагрузочное тестирование мобильного приложения.

**Функциональное тестирование**

Функциональность приложения должна быть полностью протестирована. Особое внимание следует уделить установке, обновлениям, регистрации и входу в систему, обеспечению, работе со специфическими функциями устройства и сообщениям об ошибках.

Функциональное тестирование мобильного приложения, по большей части, может быть выполнено так же, как вы выполнили бы его для любого другого типа приложения. По этой причине мы не будем вдаваться в подробности этого типа тестирования. Однако следует указать области, которые имеют особое значение для мобильных приложений.

Имейте в виду, что функциональное тестирование должно включать в себя тестирование всех функций приложения и не должно быть излишне сосредоточено на какой-то одной функции.

**Этап 7: Аттестационное тестирование и тестирование безопасности приложения**

Безопасность и конфиденциальность данных имеют огромное значение в наше время. Пользователи требуют, чтобы вся их информация хранилась безопасно и конфиденциально.

Убедитесь, что тестируемое приложение надежно защищено. Выполните проверку на возможность внедрения SQL инъекций, на возможность перехвата сеансов, анализа дампов данных, анализа пакетов и SSL трафика.

Очень важно проверить безопасность хранилища конфиденциальных данных вашего мобильного приложения и его поведение в соответствии с различными схемами разрешений для устройств.

Помимо проверки безусловного шифрования имен пользователей и паролей, задайте себе следующие вопросы:

- Есть ли у приложения сертификаты безопасности?
- Использует ли приложение безопасные сетевые протоколы?
- Существуют ли какие-либо ограничения, например количество попыток входа в систему до блокировки пользователей?

**Этап 8: Тестирование устройства**

Выполните тесты по тем алгоритмам, которые вы ранее прописали в тестовых случаях и сценариях тестирования на всех определенных для тестирования устройствах, в облаке и / или на физических устройствах.

**Этап 9: контрольный этап и резюме**

Этот этап включает в себя подробное и полное тестирование - от ранних итеративных этапов тестирования до регрессионных тестов, которые все еще могут потребоваться для стабилизации работы приложения и выявления незначительных дефектов.

На этом этапе тестирования вы можете добавить для проверки новые функции и изменить настройки на те, которых не будет в финальной версии.

После завершения тестирования приложения, дополнительные параметры и функции, добавленные для проверки на этом этапе, удаляются, и окончательная версия становится готовой для представления общественности.

**Итоговый отчет о тестировании**

Весь процесс тестирования мобильных приложений должен быть тщательно задокументирован. Проверьте дважды, сделаны ли нужные записи, и после этого сформируйте свой окончательный отчет о тестировании (test summary report).

Этот отчет должен включать:

- Важную информацию, выявленную в результате проведенных испытаний;
- Информацию о качестве проводимого тестирования;
- Сводную информацию о качестве тестируемого мобильного приложения;
- Статистику, полученную из отчетов об различных инцидентах;
- Информацию о видах тестирования и времени, затраченном на каждый из них.
- Следует также указать в отчете, что:



- данное мобильное приложение пригодно для использования в том
- качестве, в котором заявлено;
- соответствует всем критериям приемлемости функционала и качества работы.

Задание. Провести тестирование, выполнить оптимизацию системы и составить отчет

**Практическая работа № 3.11. Создание тем для упрощения работы с элементами. Применение DDMS для отладки приложения. Создание лога. Списки. Работа с ориентацией экрана, применение различных layouts. Анимация. Рисование. Меню. Кнопки. Диалоговые окна. Сообщения. Мультимедиа. Shared preferences. БД SQLite. Интернет-соединение. Content providers. Работа с картами и GPS. Виджеты. Публикация приложения.**

Цель работы: Изучить основы верстки. Научиться управлять интерфейсом мобильного устройства при разработке программного приложения.

Просмотрите основные сведения о классах, которые понадобятся при разработке приложения.

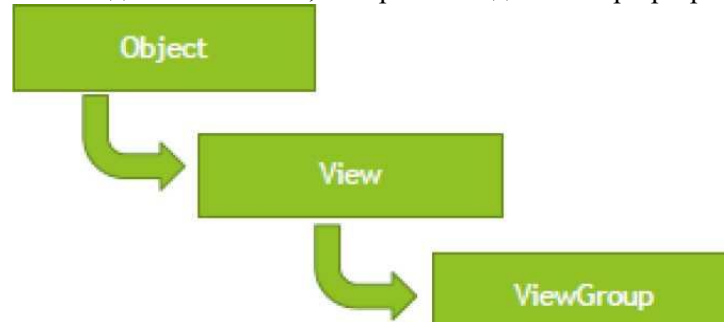


Рисунок 67

Файл разметки имеет следующую структуру

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<LinearLayout android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="0.25" android:padding="5dp">
<Button android:layout_width="0dp"
android:layout_height="0.33"
android:layout_height="wrap_content"
android:text="Button1"
android:id="@+id/button3" android:layout_gravity="right"/>
<Button android:layout_width="0dp"
android:layout_height="0.33"
android:layout_height="wrap_content"
android:text="Button2" android:id="@+id/button"/>
<Button android:layout_width="0dp"
android:layout_height="0.33"
android:layout_height="wrap_content"
android:text="Button3"
android:id="@+id/button2" android:layout_gravity="center_horizontal"/>
</LinearLayout>
<LinearLayout android:layout_width="match_parent"
android:layout_height="0dp"
android:paddingLeft="40dp"
android:paddingRight="40dp"
android:layout_weight="0.5"
android:gravity="center_vertical">
<Button android:layout_width="0dp"
android:layout_height="0.33"

```

```

android:layout_height="wrap_content"
android:text="Button4" android:id="@+id/button3"/>
<Button android:layout_width="0dp"
android:layout_height="0.33"
android:layout_height="wrap_content"
android:text="Button5" android:id="@+id/button"/>
</LinearLayout>
<LinearLayout android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="0.25" android:padding="5dp"
android:gravity="bottom">
</LinearLayout>
</LinearLayout>

```

Распространенные виды макетов

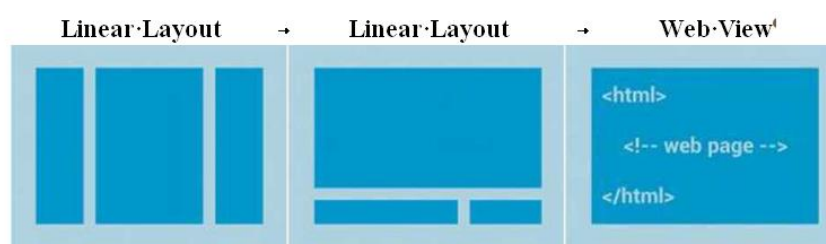


Рисунок 68

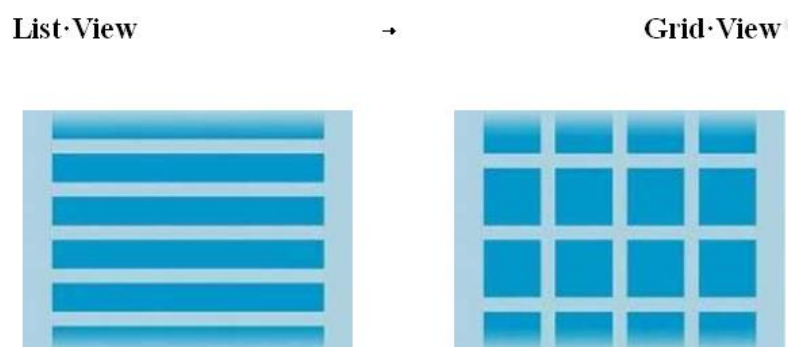


Рисунок 69

Атрибуты LinearLayout

Таблица 3

Attribute Name	Related Method	Description
android:baselineAligned	setBaselineAligned(boolean)	When set to false, prevents the layout from aligning its children's baselines.
android:baselineAlignedChildIndex	setBaselineAlignedChildIndex(int)	When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).
android:divider	setDividerDrawable(Drawable)	Drawable to use as a vertical divider between buttons.
android:gravity	setGravity(int)	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
android:measureWithLargestChild	setMeasureWithLargestChildEnabled(boolean)	When set to true, all children with a weight will be considered having the minimum size of the largest child.

android:orientation	setOrientation(int)	Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
android:weightSum	Defines the maximum weight sum.	

Задание 1. Разработать мобильное приложение, состоящее из четырех activity.

После запуска приложения пользователь должен попадать на экран с activity1. На данном экране должно быть представлено меню, состоящее из четырех кнопок. Высота кнопок должна составлять 20% от высоты экрана. Расстояние между кнопками - 2%. Первая и последняя кнопка должны быть на равном расстоянии от краев экрана. Ширина кнопок 75%, выравнивание посередине.

После нажатия на первую кнопку пользователь должен переходить к activity2. Верстка должна осуществляться с использованием LinearLayout, ширина кнопок должна задаваться в процентах от ширины экрана.

После нажатия на вторую кнопку в activity1 пользователь должен переходить к activity3. Верстка должна осуществляться с использованием RelativeLayout (не использовать LinearLayout).

Третья кнопка в activity1 должна создавать activity3.

Кнопка должна быть выровнена по центру экрана. Цвет обводки кнопки #505050. Толщина обводки в соответствии с месяцем вашего рождения (от 1 до 12). Радиус скругления 24dp. Цвет фона экрана #FFFFFF. При нажатии на кнопку ее цвет должен изменяться на светло-зеленый.

Нажатие на четвертую кнопку в activity1 должно приводить к закрытию приложения.

### **Практическая работа № 3.12. Создание приложения, которое состоит из нескольких activities. Написание приложения, работающее с разными темами/стилями. Создание приложения, содержащее анимированные интерфейсные элементы**

Цель работы: Знакомство с интерфейсом среды программирования. Изучение структуры проекта. Рассмотрите основные понятия Android проекта.

Структура проекта

- src—«исходный код» приложения (java-классы)
- assets —пустая директория. Может использоваться для сохранения raw-файлов.
- gen—хранилище генерируемых системных файлов. В частности, здесь располагается файл R.java, в котором хранятся идентификаторы всех ресурсов, создаваемых в проекте (строковые ресурсы и т.п.).
- libs—различные библиотеки, используемые приложением
- res—ресурсы проекта.
- AndroidManifest.xml—файл описания проекта (поддерживаемые версии SDK, версия приложения и т.п.)
- project.properties—файл, включающий настройки проекта, такие как build target.

Ресурсы проекта

- anim/ Содержит XML файлы, компилируемые в анимационные объекты.
- color/ Содержит XML файлы описывающие цвета.
- drawable/ Содержит растровые файлы (PNG, JPEG, orGIF), 9-Patch файлы, и XML файлы, описывающие Drawables или Drawableobjects включающие множественные состояния(нормальное, нажатое, состояние фокуса).
- layout/ Содержит XML файлы описывающие макеты экрана
- menu/ Содержит XML файлы, определяющие меню приложения.
- raw/ Для хранения произвольных файлов.
- values/ Содержит XML файлы, компилируемые во множество видов ресурсов (strings.xml и т.д).
- xml/ СодержитXML файлы конфигурирующие компоненты приложения. Например, XML файл определяющий экран настроек.

Пример простейшего файла AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.untitled"
```

```

 android:versionCode="1"
 android:versionName="1.0">
 <uses-sdk android:minSdkVersion="19"/>
 <application android:label="@string/app_name" android:icon="@drawable/ic_launcher">
 <activity android:name="MyActivity"
 android:label="@string/app_name">
 <intent-filter>
 <action android:name="android.intent.action.MAIN"/>
 <category android:name="android.intent.category.LAUNCHER"/>
 </intent-filter>
 </activity>
 </application>
 </manifest>

```

Для начала работы требуется:

- Java Development Kit
- Android Software Development Kit

Задание 1. Создание проекта приложения

Запустите среду программирования в IDE IntelliJ Idea

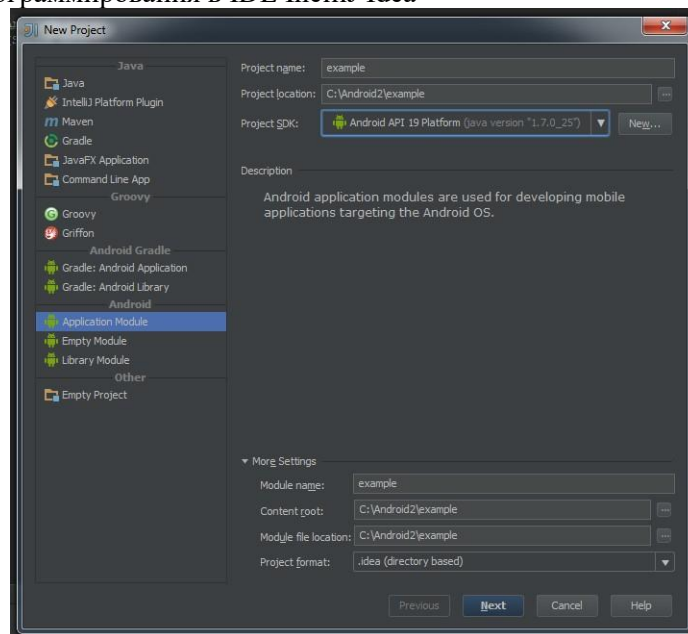


Рисунок 70

Введите данные проекта. Сохраните проект.

Задание 2. Создание приложений с одним экраном (Activity) Необходимо создать два activity и организовать переход между ними. Содержимое Activity 1 кнопка с именем btn1

Содержимое Activity 2: TextView с текстом «Переданный параметр: значение\_параметра». значение\_параметра – принятый параметр из Activity 1.

При запуске приложения пользователь должен попадать на экран с Activity 1. После нажатия на кнопку btn1 необходимо осуществить переход к Activity 2 и передавать параметр из Activity 1. В качестве значения передаваемого параметра использовать свою фамилию.

Ключевые классы: Activity, Intent, Button, TextView.

**Практическая работа № 3.13. Создание приложения, отображающее после запуска карты Google или какие ни будь другие карты. Создание собственный виджет с настройками. Создание приложение, использующее опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента**

Цель работы: создать приложение с картами и виджет меню

Создание андроид-приложения с картами Google Maps с использованием Google Services и Android Studio

Как зарегистрировать андроид-приложение в Google Maps API и как добавить карты Гугла в ваше приложение на android, как добавить метку на карту и как отобразить на карте текущее

местоположение устройства с помощью Android Studio и службы Google Framework для создания простых картографических приложений на Android, вы узнаете в этом уроке. Нужно выполнить следующие шаги:

1. Настроить Android Studio для Maps API
2. Получить ключ API key для Google Maps Android API
3. Добавить карту на экран приложения, используя MapFragment
4. Как добавить метку на карту Google Maps
5. Как добавить текущее местоположение на карту Google Maps

#### Настройка Android Studio

Для того чтобы использовать новый Maps API, мы должны предоставить Google имя пакета нашего приложения. Поэтому нам необходимо создать новое приложение в Android Studio и настроить некоторые зависимости для успешного подключения к Maps API.

Откройте Android Studio и выберите создание нового проекта. На первом экране настройки нужно ввести данные, такие как имя проекта — здесь пишем MapApp, и домен компании — я укажу адрес своего сайта [fandroid.info](http://fandroid.info). Имя пакета вашего приложения формируется по умолчанию из перевернутого доменного имени и имени проекта. Запомните его, оно нам еще понадобится.

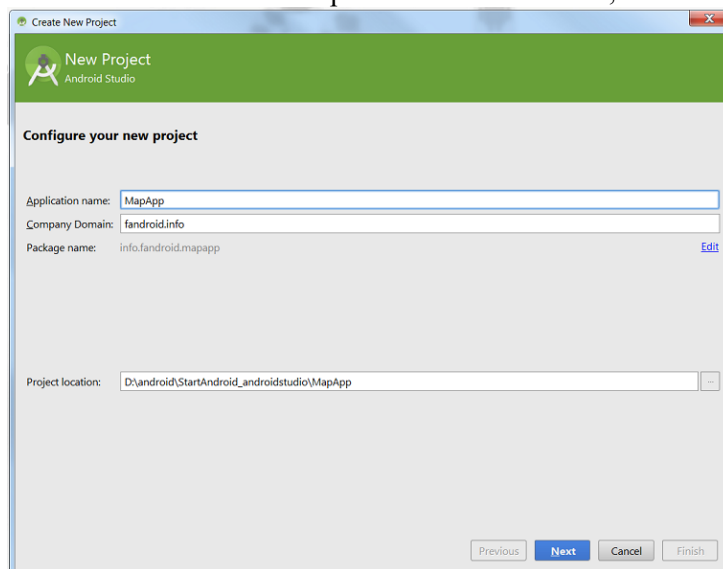



Рисунок 71

При выборе Minimum required SDK, вы можете выбрать версию ниже 4.0, однако для этого потребуются загрузка дополнительной библиотеки поддержки в Android SDK. В этом уроке мы будем устанавливать Minimum required SDK версии API 14 Android 4.0. Около 94% android-устройств будут совместимы с нашим приложением (по данным Google Play на момент публикации).

Идите вперед и жмите кнопку Далее, ничего не меняя, до конца. Если вы впервые создаете проект, используя Android Studio, среда может также скачать файлы gradle сборки. Эти файлы весят около 50мб, так что процесс создания проекта займет немного больше времени.

После создания проекта Android Studio будет индексировать его, это может занять некоторое время в зависимости от мощности вашей машины. Вы можете увидеть, что индексация закончилась, когда кнопки в верхней части экрана станут активными.

Теперь, когда проект создан, мы должны убедиться, что у нас есть необходимые компоненты SDK, для подключения нашего приложения к сервисам Google. В верхней панели инструментов, выберите кнопку SDK менеджера . Откройте папку Extras в SDK Manager и убедитесь, что у вас установлены следующие пакеты:

Еще нужно добавить в файл build.gradle нашего проекта зависимости для Google Play Services. По умолчанию есть два файла build.gradle внутри структуры проекта; один внутри папки модуля (обозначен цифрой 1 на рисунке ниже) и один в папке проекта, на верхнем уровне (с цифрой 2 на рисунке ниже).

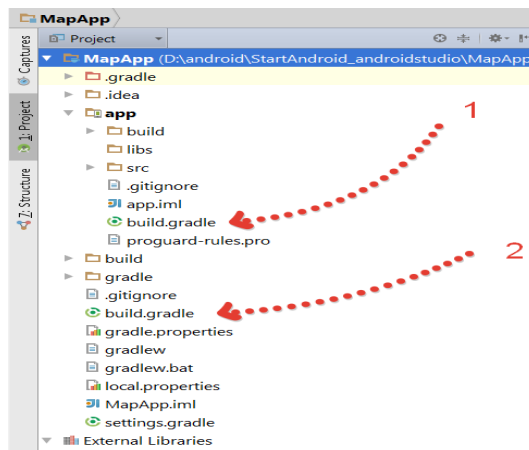


Рисунок 72

Откройте файл в папке модуля, под цифрой 1 на рисунке выше. В раздел «dependencies» нужно добавить последнюю версию библиотеки Google Play Services. По состоянию на момент написания, это 7.5.0. Вы можете узнать номер последней версии этой, а также многих других популярных библиотек, по ссылке на странице полезных ссылок, в разделе «Разное».

После добавления строки нажмите в правом верхнем углу ссылку «Sync Now» для обновления файла сборки. Обновление займет некоторое время.

Создание API ключа для приложения

Теперь мы должны зарегистрировать наше приложение в консоли разработчика Google Developer Console, чтобы позволить ему доступ к Play Services API.

Перейдите по ссылке в Google Developer Console

Если потребуется, создайте новый проект. После создания этого проекта, щелкните по нему и выберите “APIs & auth” в меню слева. Из списка интерфейсов выберите Google Maps Android API, и нажмите кнопку «Enable API».

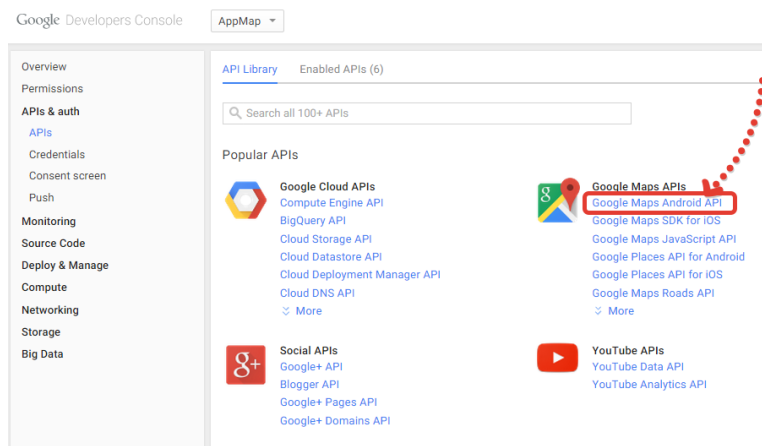


Рисунок 73

После этого, выберите “Credentials” в меню слева. В разделе «Public API access» щелкните Create New Key, а затем Android Key.

Выполните следующую команду в командной строке для генерации сертификатов разработчика SHA1 fingerprint key: `keytool -list -v -keystore mystore.keystore`

Если вы получаете ошибку с сообщением: `mystore.keystore doesn't exist`, выполните следующую команду (заменяя {имя пользователя} вашим именем пользователя): `keytool -list -alias androiddebugkey -keystore C:\Users\{Username}\.android\debug.keystore -storepass android -keypass android`

В результате получаем SHA1 fingerprint key, который имеет вид набора пар чисел и символов, разделенных двоеточиями.

Возьмите ваш ключ SHA1, и добавьте имя пакета приложения в конец ключа, после точки с запятой. Результат будет выглядеть примерно так:

AA:95:81:D7:A7:B8:C9:79:AF:C5:7A:A6:8F:89:96:94:9A:BF:68:AB;info.fandroid.mapapp

Вставьте результат в текстовое поле и нажмите кнопку Создать. Автоматически сгенерируется API key, скопируйте его в буфер обмена.

Теперь нам нужно добавить API key в наше приложение. Переходим в Android Studio.

В проекте, откроем файл манифеста AndroidManifest.xml. Просто перед тегом </application> необходимо объявить две декларации Мета-данных . Первый Мета-тег встраивает данные версии библиотеки Google Play Services, добавленной в приложение. Второй содержит наш API ключ, который мы ранее скопировали из консоли разработчика.

```
<meta-data
 android:name="com.google.android.gms.version"
 android:value="@integer/google_play_services_version" />

<meta-data
 android:name="com.google.android.maps.v2.API_KEY"
 android:value="AIzaSyB9oARLFPe4wRGyeuWJq0IqZ0g84TjmjVI" />
```

Теперь нам нужно указать разрешения для нашего приложения. Мы должны добавить следующие разрешения:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Наконец, чтобы карта занимала весь экран устройства, мы объявим стиль, который удаляет строку заголовка в верхней части приложения. Для этого изменим значение атрибута android:theme:

```
android:theme="@android:style/Theme.NoTitleBar"
[wpanchor id=»3"]
```

Добавление карты на экран приложения

Теперь, когда наши библиотеки, API-ключи и разрешения настроены, пришло время добавить на экран нашего приложения карту. Открываем файл макета — activity\_main.xml , если вы не переименовали его во время создания проекта. Мы будем добавлять map view как фрагмент, а затем настроим его для загрузки содержимого на карту при старте приложения. Измените ваш макет следующим образом:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent" >

 <fragment
 android:id="@+id/mapView"
 android:name="com.google.android.gms.maps.MapFragment"
 android:layout_width="match_parent"
 android:layout_height="match_parent"/>
```

В файле MainActivity.java, нам нужно инициализировать mapView. Для начала нам нужно объявить локальную переменную, которая ссылается на mapView. Мы также должны импортировать Google Maps library.

```
import com.google.android.gms.maps.*;
public class MainActivity extends Activity {
 /** Local variables */
 GoogleMap googleMap;
```

Мы создадим закрытый метод, которая проверяет, имеется ли карта, попытается инициализировать ее, если это возможно, или в противном случае отображает ошибки пользователю. Обязательно добавьте следующие импорты в верхней части файла:

```
import com.google.android.gms.maps.*;
/**
 * Initialises the mapView
 */
private void createMapView(){
 /**
 * Catch the null pointer exception that
 * may be thrown when initialising the map
```



```

 */
 try {
 if(null == googleMap){
 googleMap = ((MapFragment) getFragmentManager().findFragmentById(
 R.id.mapView)).getMap();
 /**
 * If the map is still null after attempted initialisation,
 * show an error to the user
 */
 if(null == googleMap) {
 Toast.makeText(getApplicationContext(),
 "Error creating map", Toast.LENGTH_SHORT).show();
 }
 }
 } catch (NullPointerException exception){
 Log.e("mapApp", exception.toString());
 }
 }
}

```

Далее мы создадим метод, который будет добавлять маркер на карту. Добавьте следующие операторы импорта в верхней части файла:

```

import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
Метод добавления маркера:
/**
 * Adds a marker to the map
 */
private void addMarker(){

```

```

 /** Make sure that the map has been initialised */
 if(null != googleMap){
 googleMap.addMarker(new MarkerOptions()
 .position(new LatLng(0, 0))
 .title("Marker")
 .draggable(true)
);
 }
}

```

Осталось теперь добавить методы createMapView и addMarker в метод onCreate:

```

@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 createMapView();
 addMarker();
}

```

Приложение теперь при запуске загружает карту и ставит маркер прямо в центре её. Удерживая маркер, можно перетащить его по карте.

[wpanchor id=>4"]

Весь код класса для добавления метки с определенными координатами и указанием своего местоположения на карту Google maps используйте такой код:

```

package ...
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;

```

```

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
public class MapActivity extends AppCompatActivity {
 GoogleMap googleMap;
 //координаты для маркера
 private static final double TARGET_LATITUDE = 17.893366;
 private static final double TARGET_LONGITUDE = 19.511868;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_map);
 createMapView();
 addMarker();
 //добавляем на карту свое местоположение
 googleMap.setMyLocationEnabled(true);
 }
 //создаем карту
 private void createMapView(){
 try {
 if(null == googleMap){
 googleMap = ((MapFragment) getFragmentManager().findFragmentById(
 R.id.mapView)).getMap();
 }
 if(null == googleMap) {
 Toast.makeText(getApplicationContext(),
 "Error creating map",Toast.LENGTH_SHORT).show();
 }
 } catch (NullPointerException exception){
 Log.e("mapApp", exception.toString());
 }
 }
 //добавляем маркер на карту
 private void addMarker(){
 double lat = TARGET_LATITUDE;
 double lng = TARGET_LONGITUDE;
 //устанавливаем позицию и масштаб отображения карты
 CameraPosition cameraPosition = new CameraPosition.Builder()
 .target(new LatLng(lat, lng))
 .zoom(15)
 .build();
 CameraUpdate cameraUpdate = CameraUpdateFactory.newCameraPosition(cameraPosition);
 googleMap.animateCamera(cameraUpdate);
 if(null != googleMap){
 googleMap.addMarker(new MarkerOptions()
 .position(new LatLng(lat, lng))
 .title("Mark")
 .draggable(false)
);
 }
 }
}

```

### **Практическая работа № 3.14. Создание первого приложения под Android. Основы разработки интерфейсов мобильных приложений**

Цель работы: разработка простого приложения, помогающего понять структуру приложения, освоить основные операторы, привыкнуть к среде разработки, основ разработки интерфейсов мобильных приложений.

Приступая к разработке мобильных приложений хорошо бы 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, SMS- автоответчики. 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.

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.

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.

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 - привносить 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.

## 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.

### 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.

#### 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. 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.

#### 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).

#### Задания:

1. Изучить рекомендуемую литературу.
2. Создать новое приложение и изучить его структуру.
3. Настроить интерфейс приложения;
4. Реализовать логику приложения.
5. Изучить элементы интерфейса.
6. Практическим путём научиться размещать элементы и менять их свойства.
7. Разработать прототип интерфейса собственного приложения.
8. Ответить на контрольные вопросы.
9. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) 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?
7. Особенности визуального дизайна интерфейсов, строительных блоках и элементах управления.
8. Особенности проектирования GUI под Android.
9. Принципы разработки удобных пользовательских интерфейсов для мобильных приложений.

### Практическая работа № 3.15. Создание многоэкранного приложения

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

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.

Задания:

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) 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

### **Практическая работа № 3.16. Демонстрации распознавания стандартных жестов. Принципы работы с жестами вводимыми пользователями**

Цель работы: разработать простейшие приложения для демонстрации распознавания стандартных жестов, разработка приложения, помогающего понять принципы работы с жестами вводимыми пользователями.

Известно, что смартфон является "умным телефоном": предполагает обязательное наличие операционной системы и возможность установки дополнительных приложений, существенно расширяющих функционал устройства. С одной стороны, смартфон выполняет все привычные функции мобильного телефона и, благодаря компактным размерам, всегда под рукой. С другой стороны, благодаря наличию процессора и операционной системы, позволяет выполнять многие функции полноценного компьютера. Дополнительно ко всему, смартфоны обладают рядом интересных особенностей, не характерных для телефонов и компьютеров.

Для начала обратим внимание на экран смартфона. В современных смартфонах экран занимает практически всю площадь передней панели устройства, имеет высокое разрешение и является чувствительным к прикосновениям. Благодаря такой чувствительности, для взаимодействия с устройством и его приложениями можно использовать виртуальные элементы управления, чаще всего кнопки, отображаемые на экране. В связи с чем отпадает необходимость в физических кнопках. В смартфонах реализуется, так называемый, touch-интерфейс - интерфейс, основанный на виртуальных элементах управления, выбор которых выполняется простым касанием, а также на использовании



жестов (gestures). Если точек касания несколько (т. е. используется несколько пальцев), такой интерфейс, уже называется multi-touch.

Еще одна особенность смартфонов состоит в том, что для большинства их владельцев не последнюю роль играет возможность использования этого "умного телефона" в качестве аудио или видеоплеера, поэтому современные устройства становятся все более и более мультимедийными. В первой лекции обсуждалось, что в состав платформы Android входит набор библиотек для обработки мультимедиа Media Framework, в котором реализована поддержка большинства общих медиа-форматов. В связи с чем, в приложения, разрабатываемые для смартфонов под управлением Android, можно интегрировать запись и воспроизведение аудио и видео, а также работу с изображениями.

Важной и часто используемой особенностью смартфонов является наличие камеры, которая позволяет снимать все самое интересное: от первых шагов ребенка до падения метеорита. Телефон всегда под рукой и готов к работе, в связи с этим количество фотографий и небольших видеороликов резко увеличилось, и любое интересное событие в жизни индивидуума может быть запечатлено и сохранено для потомков. С ростом возможностей получения фото и видео материалов увеличивается потребность в приложениях, способных работать с этими материалами. Платформа Android позволяет разрабатывать такие приложения, которые предоставляют пользователям возможности делать фотоснимки или записывать видео, каким-то образом обрабатывать полученные материалы и использовать их далее.

Большинство смартфонов оснащены GPS-модулем, а некоторые даже комбинированным модулем GPS/ГЛОНАСС, что позволяет использовать такое устройство в качестве инструмента для ориентирования на местности. Во многих случаях смартфон с установленным соответствующим программным обеспечением вполне может заменить GPS навигатор. В разрабатываемых приложениях иногда бывает очень полезно добавить возможность получения координат устройства и хозяина, если оба находятся в одном месте, и использовать эти координаты для каких-либо целей. Например, уже существуют приложения, которые позволяют отслеживать параметры человека (спортсмена) во время преодоления некоторых расстояний бегом, на велосипеде, на лыжах и т. д. Такое приложение работает во время тренировки (устройство должно перемещаться вместе со спортсменом), по окончании можно получить полную статистику маршрута: точное время в пути, расстояние, подъемы/спуски, среднюю скорость, потраченные калории и т. д. Заметим, что большая часть информации опирается на данные, полученные со спутников GPS.

Рассмотрение особенностей смартфонов будет неполным, если оставить без внимания датчики и сенсоры, которыми оснащены большинство устройств. Эти микроустройства обеспечивают связь смартфона с окружающей средой и добавляют новые удивительные функции. С помощью датчика приближения, например, можно отключать подсветку экрана при приближении телефона к уху пользователя во время разговора, блокировать экран, чтобы не было возможности случайно нажать на отбой. Акселерометр может использоваться для смены ориентации экрана, для управления в играх, особенно симуляторах, а также в качестве шагомера. Датчик освещенности позволяет регулировать яркость экрана. Гироскоп может применяться для определения более точного позиционирования устройства в пространстве.

Начиная с версии 1.6, Android предоставляет API для работы с жестами, который располагается в пакете `android.gesture` и позволяет сохранять, загружать, создавать и распознавать жесты.

Задания:

1. Изучить рекомендуемую литературу.
2. Разработать приложение, в котором демонстрируется распознавание всех поддерживаемых жестов.
3. Разработать приложение, в котором демонстрируется распознавание только некоторой части поддерживаемых жестов по выбору программиста.
4. Реализовать жестами ввод чисел в приложении "Угадайка", разработанном в лабораторной работе второй темы. Создать жесты "0", "1", "2", "3", "4", "5", "6", "7", "g", "9" для ввода цифр и жест "S" для остановки ввода числа. В приложение добавить распознавание этих жестов, преобразование их в число и сравнение полученного числа с загаданным.
5. Разработать простой калькулятор с жестовым вводом чисел и операций.
6. Разработать блокнотик для заметок с рукописным вводом текста.
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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выравнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Возможности добавления сенсорного управления в мобильные приложения под Android.
2. Процесс распознавания жеста.
3. Назначение мультимедиа библиотеки Android.
4. Использование встроенной камеры.
5. Технология создания набора жестов.
6. Способы импортировать жесты в проект.
7. Использование созданных жестов в приложении.

### **Практическая работа № 3.17. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек**

Цель работы: разработка многооконного приложения, предоставляющего возможности: воспроизведения аудио и видео файлов, создания и отображения фотоснимков, разработка приложения, демонстрирующего геолокационные возможности.

В современных смартфонах экран занимает практически всю площадь передней панели устройства, имеет высокое разрешение и является чувствительным к прикосновениям. Благодаря такой чувствительности, для взаимодействия с устройством и его приложениями можно использовать виртуальные элементы управления, чаще всего кнопки, отображаемые на экране. В связи с чем отпадает необходимость в физических кнопках. В смартфонах реализуется, так называемый, touch-интерфейс - интерфейс, основанный на виртуальных элементах управления, выбор которых выполняется простым касанием, а также на использовании жестов (gestures). Если точек касания несколько (т. е. используется несколько пальцев), такой интерфейс, уже называется multi-touch.

Еще одна особенность смартфонов состоит в том, что для большинства их владельцев не последнюю роль играет возможность использования этого "умного телефона" в качестве аудио или видеоплеера, поэтому современные устройства становятся все более и более мультимедийными. В первой лекции обсуждалось, что в состав платформы Android входит набор библиотек для обработки мультимедиа Media Framework, в котором реализована поддержка большинства общих медиа-форматов. В связи с чем, в приложения, разрабатываемые для смартфонов под управлением Android, можно интегрировать запись и воспроизведение аудио и видео, а также работу с изображениями.

Важной и часто используемой особенностью смартфонов является наличие камеры, которая позволяет снимать все самое интересное: от первых шагов ребенка до падения метеорита. Телефон всегда под рукой и готов к работе, в связи с этим количество фотографий и небольших видеороликов резко увеличилось, и любое интересное событие в жизни индивидуума может быть запечатлено и сохранено для потомков. С ростом возможностей получения фото и видео материалов увеличивается потребность в приложениях, способных работать с этими материалами. Платформа Android позволяет разрабатывать такие приложения, которые предоставляют пользователям возможности делать фотоснимки или записывать видео, каким-то образом обрабатывать полученные материалы и использовать их далее.

Большинство смартфонов оснащены GPS-модулем, а некоторые даже комбинированным модулем GPS/ГЛОНАСС, что позволяет использовать такое устройство в качестве инструмента для ориентирования на местности. Во многих случаях смартфон с установленным соответствующим программным обеспечением вполне может заменить GPS навигатор. В разрабатываемых приложениях иногда бывает очень полезно добавить возможность получения координат устройства и хозяина, если оба находятся в одном месте, и использовать эти координаты для каких-либо целей. Например, уже существуют приложения, которые позволяют отслеживать параметры человека (спортсмена) во время преодоления некоторых расстояний бегом, на велосипеде, на лыжах и т. д. Такое приложение работает во время тренировки (устройство должно перемещаться вместе со спортсменом), по окончании можно получить полную статистику маршрута: точное время в пути, расстояние, подъемы/спуски, среднюю

скорость, потраченные калории и т. д. Заметим, что большая часть информации опирается на данные, полученные со спутников GPS.

Задания:

1. Изучить рекомендуемую литературу.
2. Разработать приложение, которое предоставляет пользователю возможность выбора рода деятельности: работа с камерой для создания снимков; воспроизведение аудио и видео; просмотр изображений.
3. Разработать приложение, в котором реализовать четыре активности: главная активность, предназначена для выбора рода деятельности, содержит три кнопки, нажатие на каждую кнопку вызывает к жизни соответствующую активность; активность для работы с камерой и создания снимков; активность для воспроизведения аудио и видео; активность для просмотра изображений.
4. Разработать приложения, получающего координаты устройства и отслеживающего их изменение
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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Как настроить интерфейс и реализовать логику активности для работы с камерой.
2. Как настроить интерфейс и реализовать логику активности для воспроизведения аудио и видео.
3. Как настроить интерфейс и реализовать логику активности для просмотра изображений.
4. Как настроить интерфейс и реализовать логику главной активности приложения.
5. Технология разработки приложения, получающего координаты устройства и отслеживающего их изменение
6. Каким образом имитировать передачу данных о местоположении на эмулятор.

### **Практическая работа № 3.18. Работа с базами данных в Android. Основные приемы работы с инструментами разработки**

Цель работы: разработка Android приложения, демонстрирующего возможности работы с базой данных SQLite, познакомиться со средой разработки приложений для Windows Phone 8.

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.

Для того чтобы начать разработку приложений для Windows Phone, нужно установить соответствующие инструментальные средства.

Windows Phone SDK 8.0.

Данный комплект средств разработчика поддерживает создание приложений для Windows Phone 8 и Windows Phone 7.5. При этом с его помощью, при установке соответствующего обновления, о котором читайте ниже, можно создавать и приложения, рассчитанные на Windows Phone 7.8. Скачать SDK можно по данной ссылке: <http://www.microsoft.com/ru-ru/download/details.aspx?id=35471>. Здесь можно загрузить веб-инсталлятор, в некоторых случаях, например, когда установку нужно произвести на несколько компьютеров, удобнее пользоваться ISO-образом, который можно записать на DVD-диск и устанавливать с него. Ссылку на загрузку образа

(<http://go.microsoft.com/fwlink/?LinkID=257234&clcid=0x419>) можно найти в нижней части страницы, в разделе примечаний.

SDK предъявляет следующие требования к операционной системе компьютера и аппаратному обеспечению:

Поддерживаемые операционные системы:

- Windows 8; Windows 8 Pro, клиентские 64-разрядные (x64) версии Windows 8.
- Аппаратное обеспечение:
- 6,5 ГБ свободного места на жестком диске;
- 4 ГБ ОЗУ;
- 64-разрядный (x64) процессор.

Эмулятор Windows Phone 8:

- выпуск Windows 8 Профессиональная или более полнофункциональный
- процессор, поддерживающий трансляцию адресов второго уровня (SLAT)

Задания:

1. Изучить рекомендуемую литературу.
2. Разработать приложение, демонстрирующее возможности работы с базой данных: создание, добавление записей, просмотр записей, удаление базы данных.
3. Создать новый проект, запустить в эмуляторе и сохранить проект приложения.
4. Ответить на контрольные вопросы.
5. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Основы работы с базами данных, SQLite.
2. Системы анимации платформа Android.
3. Требования к 2D и 3D графике.
4. Основные принципы разработки игровых приложений для смартфонов
5. Особенности установки средств разработки для платформы Windows Phone, настройки этих средств.
6. Назначение учетной записи разработчика.
7. Разблокировка устройств, позволяющих отлаживать приложения на смартфонах, работающих под управлением Windows Phone.

### **Практическая работа № 3.19. Шаблоны проектов, структура проектов. Элементы управления. Разработка пользовательского элемента управления**

Цель работы: научиться работать с проектами приложений для Windows Phone в Visual Studio

Можно отметить, что состав доступных проектов на Visual C# и Visual Basic совпадает (за исключением проекта Приложение модульного тестирования Windows Phone, который доступен только на C#). Это - проекты, на основе которых можно создать основную часть Windows Phone-приложений, и, в том числе - игровые приложения, использующие Direct 3D, XNA-игры, которые всё еще можно разрабатывать в расчёте на Windows Phone 7, и, хотя Windows Phone 8-устройства могут исполнять такие приложения, их создание специально для платформы Windows Phone 8 не предусмотрено.

Шаблоны проектов, предусматривающие использование Visual C++ ориентированы на разработку Direct3D-приложений, обычно это игры, и на создание высокопроизводительных компонентов среды выполнения Windows Phone.

Правильный выбор шаблона приложения позволяет ускорить процесс разработки за счёт наличия в созданном по нему проекте приложения некоего стартового набора элементов. В то же время, нельзя сказать, что, например, создавая проект на основе одного шаблона, разработчик принципиально не может реализовать в таком приложении ту же функциональность, которая

предусмотрена шаблоном другими шаблонами. В общем случае проект, созданный по некоему шаблону - это лишь стартовая точка разработки, дающая простейшее работающее приложение, которое в ходе разработки претерпевает множество изменений и дополнений, превращаясь в готовый программный продукт.

В стандартной поставке среды разработки для Windows Phone имеется набор элементов управления. Просмотреть доступные элементы можно, открыв Панель элементов нажатием на её ярлык, расположенный в левой части экрана. При необходимости Панель элементов можно закрепить в видимом состоянии, для её закрепления или переключения в режим автоматического скрытия служит значок с изображением канцелярской кнопки в заголовке её окна.

Состав Панели элементов можно расширить, для этого нужно щёлкнуть по ней правой кнопкой мыши и выбрать в появившемся контекстном меню пункт «Выбрать элементы». На рис. 74. показано окно Visual Studio с открытым окном Выбор элементов панели элементов. Если вы полагаете, что какого-то стандартного элемента управления "нет на месте", вероятнее всего, его отображение просто отключено. С помощью данного окна можно включить отображение нужного элемента. Панель элементов можно вернуть к виду по умолчанию, для этого служит команда её контекстного меню Сброс панели элементов

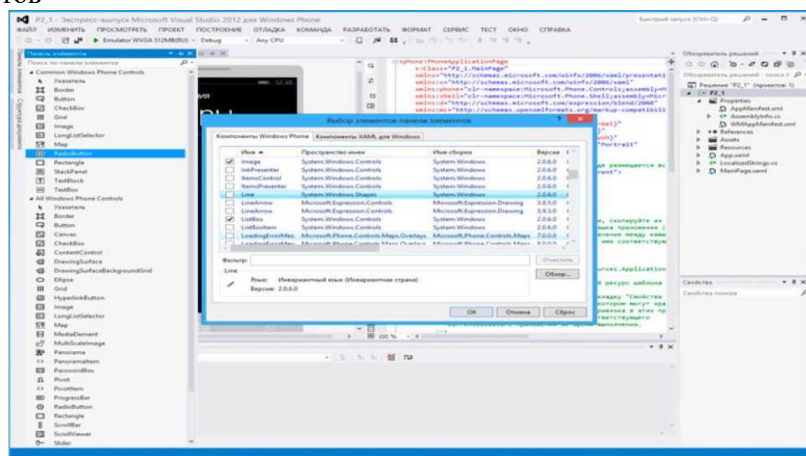


Рисунок 74

В ходе разработки приложений для Windows Phone иногда возникает необходимость в элементах управления, которых не найти ни среди стандартных элементов управления, ни среди тех, которые предлагают сторонние разработчики. Из подобной ситуации есть выход, который заключается в создании собственного элемента управления. Особенно это оправдано в тех случаях, когда подобный элемент управления нужен на нескольких страницах приложения или тогда, когда его планируется использовать в нескольких приложениях.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Изучить рекомендуемую литературу.

2. Создайте проект приложения по шаблону Приложение Windows Phone 8, рассчитанный на платформу Windows Phone 8, добавьте в него новую страницу и настройте проект так, чтобы именно эта страница, а не MainPage.xaml, вызывалась при запуске приложения. Используя несколько графических файлов, подходящих для формирования экрана-заставки на экранах разрешений 480x800 (WVGA), 768x1280 (WXGA) и 720x1280 (720p), добейтесь того, чтобы приложение отображало различные экраны-заставки при запуске на эмуляторах с различным разрешением экрана. Изучите раздел Возможности файла-манифеста приложения для Windows Phone, выясните назначение различных возможностей, которые можно устанавливать для приложения, пользуясь пояснениями и ссылками на документацию, которые приводятся в редакторе манифеста. Выберите 5 любых возможностей и подготовьте развёрнутое сообщение с их описанием.

3. Создайте учебное приложение, которое использует 5-8 элементов управления из пакета Coding4Fun, исследуйте их возможности и подготовьте отчёт о проделанной работе с приложением к нему копий экранов страниц приложений и описанием особенностей настройки и использования каждого из выбранных элементов управления.

4. Поищите другие свободно распространяемые пакеты элементов управления, дайте описание одного из них в отчёте.

5. Создайте проект приложения, добавьте в него пользовательский элемент управления, содержащий анимацию, которая запускается при касании этого элемента управления. Добавьте элемент управления на страницу MainPage.xaml и разместите на ней анимированную с помощью

раскадровки геометрическую фигуру. Настройте поведение приложения так, чтобы раскадровка, описанная на странице MainPage.xaml запускалась после завершения раскадровки, описанной в пользовательском элементе управления

6. Ответить на контрольные вопросы.

7. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Основные особенности шаблонов проектов приложений для Windows Phone 8.

2. Особенности устройства проекта приложения, созданного по шаблону Приложение Windows Phone, рассчитанного на платформу Windows Phone OS 8.0.

3. Структура и назначение основных файлов проекта.

4. Роль и особенности файла-манифеста приложения для Windows Phone.

5. Назначение стандартных элементов управления.

6. Особенности работы со стандартными элементами управления.

7. Элементы управления сторонних разработчиков.

8. Особенности работы с пакетами элементов управления сторонних разработчиков

9. Методика создания пользовательского элемента управления.

10. Методика создания уникальных элементов управления, которые можно использовать на различных страницах приложения или в разных приложениях.

### **Практическая работа № 3.20. Навигация в приложении. Обмен данными внутри приложения**

Цель работы: освоить методику организации навигации в приложениях, освоить методику организации обмена данными в приложениях

С точки зрения пользователя перемещения по страницам приложения происходят в ответ на взаимодействие пользователя с некими элементами управления, либо - как реакция на некоторые события, соответствующие сценарию работы приложения.

Приложение для Windows Phone содержит базовый элемент Microsoft.Phone.Controls.PhoneApplicationFrame (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.controls.phoneapplicationframe%28v=vs.105%29.aspx>), так называемую рамку, которая, в свою очередь, служит контейнером для элементов Microsoft.Phone.Controls.PhoneApplicationPage (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.controls.phoneapplicationpage%28v=vs.105%29.aspx>), то есть - для страниц приложения. Страницы, в свою очередь, могут содержать другие элементы управления, некоторые из которых служат для организации содержимого страницы, некоторые служат для отображения содержимого и организации непосредственного взаимодействия с пользователем. Элементы управления, которые отвечают за организацию содержимого страницы, это, например, System.Windows.Controls.Grid, то есть - сетка для содержимого, Microsoft.Phone.Controls.Panorama - то есть элемент управления для организации панорамного просмотра данных, с общим заголовком, располагающийся на странице и содержащий элементы Microsoft.Phone.Controls.PanoramaItem. Это - элемент управления

Microsoft.Phone.Controls.Pivot, который содержит элементы Microsoft.Phone.Controls.PivotItem и позволяет организовывать данные на странице, снабжая их отдельными заголовками.

Корневая рамка приложения, PhoneApplicationFrame, определяется в файле App.xaml.cs, она носит имя RootFrame, в приложении может быть только одна рамка. В свою очередь, XAML-страницы приложения - это объекты PhoneApplicationPage - их в приложении может быть столько, сколько нужно для решения задач, возложенных на него разработчиком.

Помимо обычных элементов управления, на странице можно описать так называемую панель приложения, Microsoft.Phone.Shell.ApplicationBar (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.shell.applicationbar%28v=vs.105%29.aspx>). Уже из

пространства имён панели приложения ясно, что перед нами системный объект. Поэтому и работа с ним из кода выглядит не так, как работа с обычными элементами управления. Панель приложения располагается в нижней части страницы. Фактически, это - комбинация панели инструментов и меню. "Панель инструментов" может содержать до четырех кнопок с пиктограммами для организации быстрого доступа к наиболее востребованным командам, а в меню, при необходимости, выносятся другие команды. Если на странице нужна функциональность, подобная функциональности панели инструментов и меню, желательно реализовать её именно с помощью панели приложения. Пользователи платформы Windows Phone привыкли к подобному средству взаимодействия с приложением, знают, чего от него можно ожидать, в большинстве случаев панель приложения - это наилучший выбор. Однако, не стоит забывать о том, что если в приложении, например, в графическом редакторе, нужна особая функциональность меню и панелей инструментов, не запрещено реализовывать её средствами, избранными разработчиком. Помимо команд, которые отвечают за выполнение некоторых действий, имеющих отношение к текущей странице приложения, панель приложения можно использовать и для организации перехода на другие страницы, то есть - она является одним из инструментов, который применим в навигации по страницам.

Разрабатывая систему навигации для приложения, стоит помнить о роли кнопки Назад, которая позволяет пользователю вернуться на предыдущую страницу. Не нужно создавать элементы управления, дублирующие её функциональность. Кроме того, стоит постоянно помнить о том, что приложения для Windows Phone - это приложения для мобильных устройств. Чем проще и понятнее будет система навигации по приложению, чем меньше действий пользователю придётся совершить для того, чтобы выполнить то, ради чего он запустил приложение, тем выше шансы приложения на успех. Если без некоторого экрана в вашем приложении можно обойтись - значит избавьтесь от него, сосредоточьтесь на основной задаче приложения, помните о том, что приложением могут пользоваться в условиях, когда на то, чтобы разбираться в сложной структуре его страниц, просто нет времени.

Приложения для Windows Phone, как и любые другие приложения, нуждаются во внутренних механизмах обмена данными. То есть, в простейшем виде, пользователь может ввести некоторые данные на странице №1, эти данные могут понадобиться на странице №2. При этом возможен как сценарий, когда со страницы №1 осуществляется прямой переход на страницу №2, а возможно, что "страница №1" - это страница настроек, данные, введенные на которой должны быть доступны всем остальным страницам. Более того, если расширить эту идею, обмен данными внутри приложения - это не только обмен информацией между страницами, и не только обмен информацией, которую вводит пользователь. Например, в приложении может быть описан некий объект, не имеющий визуального представления, но работающий с данными, которые должны поступать в него из других частей приложения. Приложение может, например, загружать некоторые данные из Интернета, или, скажем, получать ключ доступа к учетной записи пользователя после успешной авторизации в интернет-сервисе. Подобные данные или ключи могут никогда не выводиться в интерфейс, но они могут быть нужны для обеспечения работоспособности различных механизмов.

Задания:

1. Изучить рекомендуемую литературу.

2. Продумайте систему навигации по приложению, идею которого вы выработали после проведения предыдущего семинарского занятия. Подумайте над тем, нужна ли в вашем приложении панель приложения и если нужна - определите примерный состав её кнопок и команд. Создайте новый проект, добавьте в него страницы, которые будут содержаться в вашем приложении, при необходимости - настройте панели приложения и организуйте навигацию между ними в соответствии с особенностями приложения. Убедитесь в том, что навигация по приложению интуитивно понятна, предложите кому-нибудь поработать с системой навигации и оценить - не вызывает ли работа с ней затруднений.

3. Продумайте систему обмена данными между страницами приложения, над которым вы работаете. Подумайте, где достаточно будет передачи данных с использованием параметров, передаваемых странице при её вызове, а где понадобится использовать общедоступное поле, объявленное в классе App. В учебных целях реализуйте оба механизма.

4. Ответить на контрольные вопросы.

5. Оформить отчет.

Содержание отчета: отчет по лабораторной работе должен быть выполнен в редакторе MS Word и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.



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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Технология организации навигации по приложению с использованием кнопок, кнопок-гиперссылок, элементов управления панели приложения.
2. Технология организации обработки событий переходов страниц.
3. Технология управления стеком переходов для организации особых сценариев навигации.
4. Механизмы, которые можно использовать при передаче данных внутри приложения.
5. Передача параметров при вызове страницы и использование общедоступного поля класса App.
6. Построение строки параметров вызывающей страницей и обработка этих параметров вызываемой страницей.

### **Практическая работа № 3.21. Использование шаблона проектирования MVVM. Работа с JSON, XML, сжатие данных**

Цель работы: освоить методику создания приложений с использованием MVVM, освоить технологию работы с форматами JSON и XML и работу со сжатием данных.

Простейшее приложение для Windows Phone, всеми аспектами которого занимается один разработчик, вполне можно написать, не задумываясь о том, чтобы разделять код, имеющий отношение к визуализации данных, к их обработке и получению из некоторых источников. Приложение, в котором большая часть кода, ответственного за всю его функциональность, собрана в программном файле к его начальной странице (особенно, когда эта функциональность требует пары десятков строк кода), будет работать. Но небольшой рост сложности приложения приводит к тому, что таким кодом становится очень неудобно управлять. Его неудобно расширять, тестировать, неудобно привлекать других программистов к работе над приложением. Небольшое изменение в коде может повлечь за собой множество неожиданных, неприятных последствий. Идея разделения кода, который ответственен за работоспособность различных подсистем приложения, это одна из идей, которая легла в основу создания шаблонов проектирования приложений. Один из таких шаблонов, получивший популярность в последние годы, называется MVVM, Model-ViewViewModel, Модель-Представление-Модель представления. Этот шаблон используют на разных платформах, он универсален и позволяет создавать крупные программные проекты, обладающие четкой и понятной структурой. Для того чтобы приступить к использованию MVVM, нужно понять сущность некоторых основных понятий, которые имеют отношение к этому шаблону проектирования.

Очень кратко сущность MVVM можно выразить так: "Модель содержит исходные данные, которые готовятся к выводу с помощью модели представления и отображаются пользователю с помощью представления".

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 идет 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.

Задания:

1. Изучить рекомендуемую литературу.
2. Проработать пример, приведенный в лабораторной работе, чтобы лучше понять работу механизмов приложения, построенного по шаблону MVVM.

3. Самостоятельно ознакомиться со следующими примерами: "Windows Phone Starter Kit for RSS - WP8" ("Стартовый проект для разработки RSS-приложений для Windows Phone 8", <http://code.msdn.microsoft.com/Windows-Phone-Starter-Kit-390ee0ef> ; "Sharing Code between Windows Store and Windows Phone App (PCL + MVVM + OData)" ("Совместное использование кода в приложениях для Магазина Windows и для Windows Phone (PCL + MVVM + OData)", <http://code.msdn.microsoft.com/Sharing-Code-between-411c999b>

4. 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

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Методика реализации приложения, использующего шаблон MVVM.
2. Работа механизмов приложения, построенного по шаблону MVVM
3. Работа с JSON, XML.
4. Технология сжатия данных.
5. Выбор формата передачи данных позволяющий создать приложение, эффективно использующее системные ресурсы.
6. Затраты вычислительных ресурсов при сжатии данных.

### **Практическая работа № 3.22. Работа WebClient и HttpWebRequest**

Цель работы: освоить технологию работы с классами WebClient и HttpWebRequest.

Для работы с веб-сервисами из приложений для Windows Phone 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.

Задания:

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

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

### **Практическая работа № 3.23. Работа с API веб-сервисов**

Цель работы: освоить технологию работы с API веб-сервисов.

Если разработчик хочет создать приложение, взаимодействующее с неким веб-сервисом, сначала следует изучить интерфейс программирования (API), который реализован на данном сервисе и позволяет организовывать взаимодействие с ним приложений. Подавляющее большинство сервисов поддерживают те или иные API, иногда речь идёт о нескольких интерфейсах, построенных с использованием различных технологий. На начальном этапе подготовки к разработке следует выяснить особенности их использования. Обычно владельцы сервисов положительно относятся к разработчикам приложений, так как такие приложения, фактически, расширяют аудиторию сервиса, расширяют его присутствие на различных платформах. В данном случае речь идёт о платформе Windows Phone.

Поэтому на веб-сайтах сервисов обычно имеется документация по их API, которая содержит много полезной информации. Для того, чтобы найти такие разделы, обычно нужно приложить некоторые усилия для их поисков, например, выполнить поисковый запрос по названию сервиса с добавлением ключевых слов "API" или "Development". Например, поиск по ключевым словам "twitter API" позволяет найти ссылку на портал разработчиков Twitter: <https://dev.twitter.com/>, рис. 75. где в разделе Documentation (Документация) можно найти интересные нас материалы. Изучение справочного раздела сервиса Twitter позволяет узнать о том, что этот сервис поддерживает REST-API, для авторизации в сервисе используется протокол OAuth.

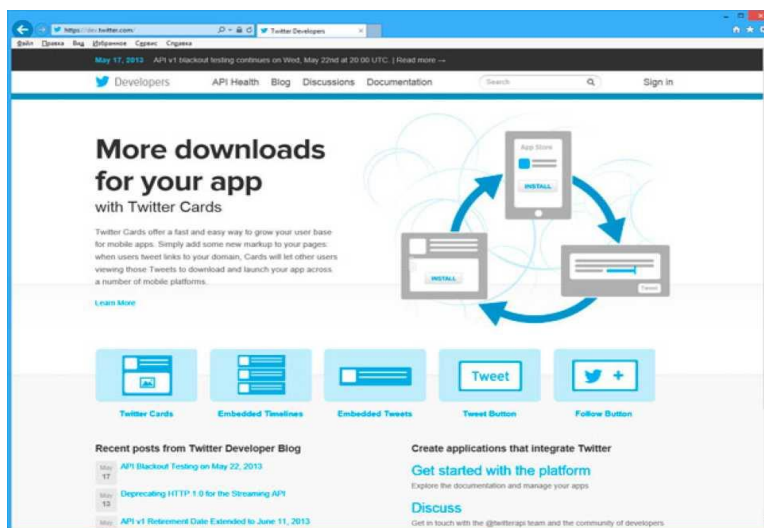


Рисунок 75

Перед началом работы приложения и в Twitter, и во многих других сервисах, сначала следует зарегистрировать приложение, в частности, такая регистрация может быть автоматической, она может подразумевать модерацию, требовать ввода каких-либо сведений о приложении. Регистрация приложения в сервисах преследует цель идентификации приложения. В данном случае процедура регистрации выполняется в разделе Manage & create your application (Управление приложением и создание приложений), для запуска процесса создания приложения служит кнопка Create a new application (Создать новое приложение). После успешного завершения регистрации и получения данных, которые приложение будет использовать для авторизации в сервисе (в частности, это - Consumer key и Consumer secret), можно приступать к изучению API.

Не все вызовы веб-сервисов требуют авторизации пользователя. Например, к некоторым механизмам Twitter можно обращаться напрямую, без авторизации.

Для того, чтобы приступить к созданию приложения для Facebook, нужно начать со страницы <https://developers.facebook.com>. Зарегистрировавшись в этой службе и создав новое приложение, вы получите данные, необходимые для дальнейшей работы, в частности, это AppID/API key и App Secret.

У сервиса Вконтакте так же есть страница документации для разработчиков, <http://vk.com/developers.php#devstep1>. Здесь используется та же процедура регистрации приложения.

Интерфейс для программного доступа к службе имеется и у проекта Wikipedia, [http://www.mediawiki.org/wiki/API:Main\\_page/ru](http://www.mediawiki.org/wiki/API:Main_page/ru).

Приступая к исследованию API веб-служб, следует учесть, что к разработке можно подходить двумя путями. Во-первых, можно самостоятельно изучить подробности реализации API, подготовить вспомогательные классы для работы с ним - для организации вызова API с нужными параметрами, организации разбора ответа сервера и обработки ошибок. Подобная работа может оказаться весьма трудоёмкой и в некоторых случаях, когда существуют доступные библиотеки сторонних разработчиков, предназначенные для работы с тем или иным сервисом, можно воспользоваться такими библиотеками.

Задания:

1. Изучить рекомендуемую литературу.

2. Подумайте, какие выгоды приложение, разработкой которого вы занимаетесь, может извлечь из работы с веб-сервисами, предоставляющими программный доступ к своим возможностям. Существует огромное количество таких сервисов, если известные вам сервисы не представляется возможным использовать в приложении, поработайте со списком веб-служб по этому адресу: <http://www.programmableweb.com/apis>. Здесь сервисы разделены по категориям, присутствуют краткие описания их API. Выберите как минимум 5 служб, которые могли бы представлять интерес для использования в вашем приложении и подготовьте обзорное сообщение по ним. Вы можете выбрать и большее количество служб - возможно, кому-нибудь из группы пригодится один из сервисов, рассмотренных вами.

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Методика разработки приложений, которые могут взаимодействовать с веб-службами, используя API этих служб.
2. Назначение общедоступных библиотек, разработанных для некоторых API служб.

### **Практическая работа № 3.24. Хранение данных на устройстве. Локальные базы данных**

Цель работы: освоить технологию хранения данных на устройстве, освоить технологию работы с локальными базами данных на Windows Phone.

Так как операции, подразумевающие работу с устройствами постоянного хранения данных, сравнительно медленны, для такой работы используют асинхронные механизмы. Кроме того, если в нескольких местах приложения нужно пользоваться данными, которые могут быть считаны, например, из изолированного хранилища настроек, имеет смысл, для ускорения работы приложения, сократить число обращений к хранилищу до минимума. Это можно сделать, воспользовавшись общедоступными переменными, в которые считываются эти данные.

Приложения для Windows Phone 8 могут хранить реляционные данные в локальных базах данных, с помощью механизмов Linq to SQL обеспечивается объектное представление баз данных в приложениях, что позволяет организовать удобную работу с ними.

Задания:

1. Изучить рекомендуемую литературу.
2. Использование файлов, хранящихся в локальных папках приложения, позволяет сохранять сериализованные представления объектов и, соответственно, загружать их. Это позволяет сохранять состояние приложения, например, при выходе из него, и загружать его при запуске приложения, то есть, если это оправдано сценарием приложения, создать у пользователя впечатление о том, что приложение "работает" непрерывно, позволяя ему, не беспокоясь о потере данных, в любой момент работы выходить из приложения, выполнять другие задачи на телефоне, а потом снова запускать его и продолжать работу. Проанализируйте приложение, которое вы создаёте, с учетом возможности сохранения его состояния в перерывах между работой с ним пользователя.
3. Если подобная модель взаимодействия с пользователем подходит вашему приложению, подумайте, как сократить до минимума объем данных, которые нужно хранить между запусками приложения, а так же, если объем данных достаточно велик, о так называемом инкрементном сохранении данных приложения во время работы, а не только при вызове обработчиков жизненного цикла приложения. То же самое касается загрузки приложения - чем меньший объем данных ему приходится считывать и обрабатывать при запуске - тем лучше. Подготовьте отчет о проделанной работе с описанием особенностей сохранения и восстановления данных приложения между сеансами работы.
4. Если приложение, созданием которого вы занимаетесь, подразумевает работу с данными, имеющими сложную структуру, рассмотрите возможность использования локальной базы данных для работы с ними. Кроме того, если приложение рассчитано на работу с локальной базой данных, это упростит, при необходимости, возможность перехода на использование облачной базы данных, работающей на платформе Windows Azure с использованием локальной базы данных в качестве локального кэша. Подготовьте отчет по данному этапу работы, приведите в нём анализ данных, которые использует приложение и сделайте выводы о том, поможет ли использование базы данных улучшить структуру вашего 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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Основные приемы работы с локальными папками приложений, с файлами, и с изолированным хранилищем настроек.
2. Назначение асинхронных механизмов.
3. Назначение общедоступных переменных.
4. Методика разработки приложений, использующих базы данных.
5. Назначение реляционных данных в локальных базах данных.
6. Назначение механизмов Linq to SQL.

### **Практическая работа № 3.25. Разработка для Windows Azure**

Цель работы: освоить технологию работы с Windows Azure.

Основа Windows Azure - дата-центры, размещенные по всему миру. Платформа Azure появилась в 2008-м году, она постоянно развивается, увеличивается количество и мощность дата-центров, появляются новые услуги, существующие услуги улучшаются. Фактически, используя платформу Azure разработчик может создавать широкий спектр приложений, может использовать большой набор программного обеспечения (такого, как операционные системы, средства разработки, системы управления базами данных). На самом деле, список возможностей Azure весьма широк, рекомендуется самостоятельно ознакомиться с их составом и особенностями на <http://www.windowsazure.com>. Служба официально доступна в России, существует немало русскоязычных описаний её функциональных возможностей. Интерфейс управления Azure так же русифицирован.

Для регистрации в службе, <https://account.windowsazure.com>, вам понадобится учетная запись Microsoft, мобильный телефон (для того, чтобы принять код подтверждения), и кредитная (или дебетовая) карта, необходимая для подтверждения личности. При регистрации можно выбрать вариант пробной бесплатной подписки. Бесплатное пробное использование рассчитано на 90 дней, каждый месяц бесплатного пробного периода предоставляется некоторый объем услуг. В частности, во время пробного бесплатна работа с мобильными службами (mobile services), которые представляют особый интерес для мобильных разработчиков, так как позволяют быстро создавать облачные серверные части для мобильных приложений.

Задания:

1. Изучить рекомендуемую литературу.
2. Подумайте над тем, как вы можете использовать мобильные службы Azure в приложении, разработкой которого вы занимаетесь, рассмотрите возможности использования облачной базы данных, работы с Push-уведомлениями, аутентификации пользователей с использованием различных провайдеров аутентификации в применении к вашему приложению и подготовьте отчет о проделанной работе.
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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Технологии реализации работы с мобильными службами Windows Azure в приложениях для Windows Phone 8.
2. Назначение взаимодействия со службой, используя класс MobileServiceClient.

### **Практическая работа № 3.26. Сервисы Live Connect: SkyDrive**

Цель работы: освоить технологию работы с сервисом хранения данных SkyDrive и получить сведения, необходимые для дальнейшей работы со службами Live Connect.

Набор служб Live Connect предоставляет разработчику следующие инструменты:

- API для взаимодействия с хранилищем данных SkyDrive, в котором можно работать с папками, и файлами, такими, как фотоснимки, видеоролики, и документы.

Файлы и папки можно создавать, удалять, перемещать, отправлять в облачное хранилище с устройства и загружать на устройство из облачного хранилища. API SkyDrive, кроме того, позволяет работать с комментариями и тегами.

- Доступ к Hotmail, который позволяет работать с контактами и календарями.

- Работу с Windows Live Messenger, который предназначен для обмена мгновенными сообщениями.

Для работы с этими службами, которая подразумевает предварительную проверку подлинности пользователя, используется сервис Live. Кроме того, этот сервис предоставляет сведения о профиле пользователя. Подробности о службах Live Connect можно узнать из документации: <http://msdn.microsoft.com/ru-ru/library/live/ff621314.aspx>.

Для разработки с использованием Live API следует загрузить и установить соответствующий SDK, <http://msdn.microsoft.com/ru-ru/live/ff621310.aspx>. Он предоставляет набор элементов управления и API, которые позволяют интегрировать приложения со службой единого входа (SSO, Single Sign On) учетных записей Microsoft (в общем случае это позволяет производить автоматическую аутентификацию пользователя в приложениях, которые поддерживают одного и того же провайдера аутентификации), работать со SkyDrive, Hotmail и Windows Live Messenger из приложений для Windows Phone и Windows 8.

Задания:

1. Изучить рекомендуемую литературу.

2. Проанализируйте сценарии работы с приложением, созданием которого вы занимаетесь. Выделите те из них, в реализации которых можно использовать SkyDrive. Подготовьте отчет о проделанной работе.

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Назначение сервисов Live Connect.

2. Назначение службы облачного хранения данных пользователей SkyDrive.

3. Технология работы с файлами, хранящимися в SkyDrive.

### **Практическая работа № 3.27. Многопоточное программирование**

Цель работы: освоить технологию работы с классом BackgroundWorker.

Создавая простое приложение для Windows Phone, не выполняющее сложных вычислений, не обращающееся к веб-сервисам, не выполняющее работу с файлами, о выполнении каких-либо задач вне потока пользовательского интерфейса обычно не задумываются. Если мы, например, в обработчике события нажатия на кнопку захотим выполнить какое-то простое вычисление, или проверку какого-либо условия, вычислительная нагрузка от таких операций ляжет на поток пользовательского интерфейса. До тех пор, пока эта нагрузка незначительна, подобное не сказывается на скорости отклика приложения. Но если нужно выполнить какую-либо ресурсоемкую операцию, это можно сделать и без дополнительных усилий по использованию для неё другого потока, однако, выполнение её в потоке пользовательского интерфейса приведет к заметным задержкам. Как вы увидите в примере, который мы разберем в этой лабораторной работе, интерфейс может быть заблокирован. То есть, приложение не реагирует на взаимодействие пользователя с элементами управления, анимация может либо приостановиться, либо работать рывками.

В предыдущих лабораторных работах мы сталкивались с вызовом асинхронных методов для выполнения задач, которые могут занимать потенциально достаточно большое время. При объявлении метода, в котором мы вызывали асинхронные операции, используется модификатор `async`, операция



вызывается с использованием оператора `await`. Это приводит к тому, что выполнение метода приостанавливается до тех пор, пока не будет получен ответ от вызываемой процедуры. Например - данные, загруженные с вебсервера. При этом использование `async` и `await` упрощает написание асинхронного кода, так как до появления этого механизма (это произошло в Visual Studio 2012) при вызове асинхронного метода нужно было предусматривать использование функции, которая вызывается при завершении работы метода и обрабатывает полученные результаты. Такой подход применим и сейчас, но он усложняет структуру кода, в то время как асинхронный код, написанный с использованием `async` и `await`, хотя и решает задачи асинхронной обработки, выглядит как синхронный.

До сих пор мы вызывали асинхронные методы каких-либо объектов, но нередко возникают задачи, для успешного решения которых нужно асинхронно выполнить произвольный код. То есть, выполнить этот код не в потоке пользовательского интерфейса, а в другом потоке. При таком подходе интерфейс продолжает реагировать на воздействия пользователя. Решить подобную задачу можно, например, с использованием класса `BackgroundWorker` (<http://msdn.microsoft.com/en-us/library/system.componentmodel.backgroundworker%28v=vs.95%29.aspx>).

Задания:

1. Изучить рекомендуемую литературу.

2. Проанализируйте код приложения, которое вы разрабатываете, испытайте различные варианты работы с ним и рассмотрите возможность асинхронного исполнения участков кода, которые выполняют в потоке пользовательского интерфейса какие-либо ресурсоёмкие операции (например, обрабатывают большие массивы данных). Подготовьте отчёт, содержащий данные анализа кода приложения на предмет асинхронного выполнения, и, если это применимо к вашему приложению, примеры модифицированного кода.

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Особенности работы с асинхронным кодом.

2. Назначение использования асинхронного кода.

### **Практическая работа № 3.28. Сенсорный пользовательский интерфейс**

Цель работы: освоить методы работы с сенсорным экраном.

Устройства, работающие под управлением Windows Phone не оснащают аппаратной клавиатурой. Такая тенденция наблюдается сейчас повсеместно, поэтому особую важность во взаимодействии с пользователем приобретает экранная клавиатура устройства. Такая клавиатура вызывается автоматически, когда пользователь входит в режим редактирования полей, предусматривающих текстовый ввод. Кроме того, сейчас разработчик для платформы Windows Phone не может создавать собственные варианты клавиатуры, однако, это ограничение компенсируется большим набором вариантов встроенной в систему экранной клавиатуры.

Для того, чтобы задать тип клавиатуры, которая отображается при попытке пользователя редактировать содержимое элемента управления, поддерживающего ввод текста, нужно настроить свойство этого элемента `InputScope` (Область вводимых данных). Сделать это можно как в XAML-разметке, так и в программном коде (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/gg521152%28v=vs.105%29.aspx>). Существует много вариантов такой клавиатуры. При заполнении этого свойства, например, для текстового поля, можно воспользоваться либо окном свойств, где представлен список доступных вариантов, либо, при заполнении его в XAML-коде - подсказкой IntelliSense.

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

Указания по технике безопасности: к выполнению лабораторных работ допускаются студенты, ознакомившиеся с правилами работы в лаборатории, прошедшие инструктаж безопасности.

Задания:

1. Изучить рекомендуемую литературу.
2. Проанализируйте пользовательский интерфейс приложения, разработкой которого вы занимаетесь, обращая внимание на то, какие именно данные пользователь вводит в текстовые поля, присутствующие в интерфейсе. Подберите такие значения свойства InputScope полей ввода, которые позволят пользователю наиболее удобно и быстро вводить данные. Подумайте, как использование жестов позволит улучшить опыт взаимодействия пользователя и приложения. Подготовьте отчет.
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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Способы настройки экранной клавиатуры с помощью установки свойства InputScope текстового блока.
2. Назначение использования экранной клавиатуры.
3. Методика обработки жестов.

### **Практическая работа № 3.29. Работа с датчиками, определение местоположения**

Цель работы: освоить методы работы со встроенными датчиками.

При разработке для Windows Phone можно использовать показания встроенных датчиков устройства, сведения о них можно найти в обзорном материале по пространству имен Windows.Devices.Sensors (<http://msdn.microsoft.com/en-us/library/windowsphone/develop/br206408.aspx>). Один из доступных датчиков - акселерометр, он позволяет организовывать управление программами путем перемещения телефона в пространстве.

При работе с акселерометром данные, которые можно от него получить, представляют сведения об ускорении устройства в гравитационных единицах. Например, (1,0,0). Если сопоставить эти данные с телефоном, то, если расположить телефон кнопками, расположенными под экраном, вниз, лицевой стороной к наблюдателю, окажется, что ось Y проходит вдоль длинной стороны экрана (положительное направление оси - вверх), X - вдоль короткой (положительное направление оси - вправо), ось Z располагается перпендикулярно экрану (положительное направление - в сторону наблюдателя). Это остаётся справедливым и при поворотах телефона.

Когда телефон неподвижен, акселерометр, регистрирует силу земного притяжения, что отражается в его показаниях (они, по соответствующей оси, близки к 1), перемещения телефона в пространстве приводят к изменению показаний.

Таким образом, оказывается, например, если телефон лежит на горизонтальной поверхности неподвижно, экраном вверх, показания акселерометра выглядят как (0, 0, -1). Если расположить телефон вертикально, кнопками вниз, мы имеем в показаниях (0, -1, 0), перевернув телефон "вверх ногами" - (0, 1, 0). Если повернуть телефон на 90° против часовой стрелки - мы получим (-1,0,0). Акселерометр очень чувствителен, даже при неподвижном устройстве показания не бывают в точности равными идеальным, к тому же, они колеблются в небольших пределах.

Задания:

1. Изучить рекомендуемую литературу.
2. Подумайте, как использование встроенных датчиков устройства может расширить возможности пользователя по управлению приложением, которое вы разрабатываете. Кроме того, проанализируйте возможности использования в приложении сервиса определения местоположения.

Подготовьте отчет о проделанной работе.

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Методика работы с акселерометром и системной определения местоположения в Windows Phone.

2. Назначение акселерометра.

3. Назначение системы определения местоположения.

### **Практическая работа № 3.30. Распознавание и синтез речи, работа с камерой**

Цель работы: освоить технологию хранения данных на устройстве.

Возможности платформы Windows Phone 8 по распознаванию и синтезу речи можно использовать для организации взаимодействия приложения с пользователем. Так, пользователь может отдавать приложению голосовые команды, которые приложение может воспринять благодаря подсистеме распознавания речи. Приложение, в свою очередь, используя функцию синтеза речи, преобразования текста в речь - информировать пользователя о каких-либо событиях.

Задания:

1. Изучить рекомендуемую литературу.

2. Рассмотрите возможность интеграции в приложение, которое вы разрабатываете, функций распознавания и синтеза речи, проанализируйте сценарии работы с приложением, в которых эти возможности наиболее полезны. Если вы разрабатываете приложение для работы с фотографиями, рассмотрите его реализацию с использованием функциональности фотоприложения. Подготовьте отчет.

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. Требования по форматированию: Шрифт TimesNewRoman; интервал - полуторный, Абзацный отступ - 1,25. Текст должен быть выровнен по ширине.

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 (Рисунок 1 - Создание подсистемы) и ссылку на него в тексте

Контрольные вопросы:

1. Методика работы с системой распознавания и синтеза речи в Windows Phone.

2. Возможности приложений по взаимодействию с пользователем.

3. Концепция фотоприложений.

## Список литературы

### Основные источники

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФОРУМ : ИНФРА-М, 2021. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1189951> (дата обращения: 27.05.2021).
2. Маркин, А. В. Программирование на SQL: учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва: Издательство Юрайт, 2021. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/476040> (дата обращения: 27.05.2021).
3. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке Object Pascal: учеб. пособие / Т.И. Немцова, С.Ю. Голова, И.В. Абрамова; под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2018. — 496 с. + Доп. материалы [Электронный ресурс; Режим доступа: <https://new.znanium.com>]. — (Профессиональное образование). - ISBN 978-5-8199-0753-5. - Текст: электронный. - URL: <https://znanium.com/catalog/product/944326> (дата обращения: 27.05.2021).
4. Огнева, М. В. Программирование на языке C++: практический курс: учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. — Москва: Издательство Юрайт, 2021. — 335 с. — (Профессиональное образование). — ISBN 978-5-534-05780-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473118> (дата обращения: 27.05.2021).
5. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2021. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/475892> (дата обращения: 27.05.2021).

### Дополнительные источники

1. Голицына, О. Л. Языки программирования: учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — Москва: ФОРУМ: ИНФРА-М, 2021. - 399 с. - (Среднее профессиональное образование). - ISBN 978-5-00091-613-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1209231> (дата обращения: 27.05.2021).
2. Гуров, В. В. Микропроцессорные системы: учебник / В.В. Гуров. — Москва: ИНФРА-М, 2021. — 336 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-015323-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1514901> (дата обращения: 27.05.2021).
3. Дорогов, В. Г. Основы программирования на языке С: учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2020. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1082440> (дата обращения: 27.05.2021).
4. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C#: учебное пособие / П.Б. Хорев. — Москва: ФОРУМ: ИНФРА-М, 2021. — 200 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-713-8. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1195623> (дата обращения: 27.05.2021).
5. Чернышев, С. А. Основы программирования на Python: учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва: Издательство Юрайт, 2021. — 286 с. — (Профессиональное образование). — ISBN 978-5-534-15160-2. — Текст:

электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/487638> (дата обращения: 27.05.2021).

#### **Интернет-ресурсы**

1. Электронная библиотечная система Znanium: сайт.- URL: <https://znanium.com/> – Текст: электронный.
2. Электронная библиотечная система Юрайт: сайт. - URL: <https://urait.ru/> -Текс: электронный.