МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИСТиД(филиал) в г.Пятигорске

Методические указания по выполнению самостоятельных работ По дисциплине «Программирование мобильных устройств»

Направление подготовки	09.03.02 Информационные системы и технологии
Профиль	**Информационные системы и технологии**
Квалификация выпускника	Бакалавр
Форма обучения	очный
Учебный план	2020
Изучается в 5 семестре	

Пятигорск, 2020 г.

Рассмотрено и утверждено на заседании кафедры систем управления и информационных технологий протокол № ___ от _____2020

Зав.кафедрой СУИТ _____И.М. Першин

Тема №1. Введение в мобильное программирование.

Операционная система Windows Phone 7

Самостоятельная работа №1

Цель работы: знакомство с системой Windows Phone, изучение оболочки Microsoft Visual Studio 2010 Express for Windows Phone, получение навыков составления программ для данной операционной системы.

Методические указания

Windows Phone - операционная система для мобильных устройств, разработанная фирмой Microsoft в 2010 году. Windows Phone поддерживает две популярные в настоящее время платформы разработки - Silverlight и XNA.

Silverlight предоставляет большие возможности для разработки сложных пользовательских интерфейсов, предоставляя традиционные элементы управления, высококачественный текст, векторную графику, мультимедиа, анимацию и привязку данных, которые могут выполняться во множестве браузеров и на разных платформах.

XNA - это игровая платформа Microsoft, поддерживающая основанную на спрайтах 2D-графику и BD-графику с традиционной архитектурой игрового цикла.

И Silverlight, и XNA могут использоваться в качестве платформы приложений для Windows Phone. В состав инструментария входит специальная версия среды разработки Microsoft Visual Studio 2010 Express for Windows Phone, Expression Blend 4 for Windows Phone (ориентированный на дизайн интерфейсов для мобильных приложений), среда конструирования игр XNA Game Studio for Windows Phone, эмулятор мобильных устройств Windows Phone Series Emulator для тестирования, а также технология Silverlight for Windows Phone.

Далее будет рассмотрено, как создать простое приложение в системе Windows Phone. После запуска Visual Studio 2010 Express For Windows Phone в меню Файл выбираем пункт Создать проект. Далее слева выбираем пункт Silverlight for Windows Phone и шаблон Приложение Windows Phone. Здесь можно сразу задать требуемое имя для создаваемого проекта.

После нажатия кнопки OK, на экране появятся следующие окна: слева отображается полнофункциональный интерактивный дизайнер пользовательского интерфейса с изображением телефона; в центре - код XAML, основанный на XML языке разметки; справа

- Обозреватель решений. Можно заметить, что проект состоит из следующих файлов:

App.xaml/App.xaml.cs - содержит точку входа программы, инициализирует ресурсы

программы и выводит программу на экран;

MainPage.xaml/MainPage.xaml.cs - содержит страницу с пользовательским интерфейсом;

ApplicationIcon.png - файл значка в формате PNG, который выводится в списке приложений телефона;

Background.png - файл изображения в формате PNG, который выводится на стартовой странице;

SplashScreenImage.jpg - файл изображения, которое выводится во время загрузки приложения;

Properties/AppManifest.xml - манифест, необходимый для создания сборки;

Properties/AssemblyInfo.cs - содержит метаданные о имени и версии приложения, которые встраиваются в сборку;

Properties/WMAppManifest.xml - манифест, который содержит специальные данные о приложении для Windows Phone Silverlight;

папка Ссылки - содержит различные библиотеки, которые обеспечивают работоспособность приложения.

Далее рассмотрим процесс сборки и тестирования программы. Для начала необходимо включить окно вывода с помощью команд: Отладка | Окна | Вывод. Далее в меню Отладка выбираем команду Построить решение для компиляции. Окно Вывод содержит сообщения, генерируемые во время компиляции приложения, включая финальное сообщение, в котором подводится окончательный итог и количество предупреждений и ошибок. Также можно использовать окно Список ошибок (Вид | Другие окна| Список ошибок), которое показывает ошибки, предупреждения и сообщения, выдаваемые компилятором. Можно сделать двойной щелчок на описании ошибки, чтобы автоматически оказаться в нужном месте исходного кода.

Далее рассмотрим запуск программы в эмуляторе. Для начала необходимо убедиться, что в системе установлен Windows Phone Emulator в выпадающем списке устройств на панели инструментов. После этого нажмите клавишу F5 или щелкните по зеленому треугольнику для запуска программы в Windows Phone Emulator. На экране появится эмулятор устройства и начнется процесс установки приложения на эмулятор. Через некоторое время приложение появится в эмуляторе.

Порядок выполнения работы

1. Изучить методические указания к самостоятельной работе №1.

2. Запустить на выполнение Microsoft Visual Studio 2010 Express for Windows Phone.

3. Создать в среде Visual Studio 2010 Express For Windows Phone проект, рассмотренный в самостоятельной работе №1.

4. Произвести компиляцию созданного проекта.

5. Запустить разработанное приложение в среде Windows Phone Emulator.

6. Изменить параметры поля TextBlock (имя, текст, расположение) для созданного приложения.

7. По заданию преподавателя разработать, отладить и запустить в эмуляторе программы для системы Windows Phone.

Отчет должен содержать:

1. Название и цель работы.
2. Листинг созданного приложения в среде Microsoft Visual Studio 2010 Express for Windows Phone.
3. Результаты работы приложения в эмуляторе Windows Phone Emulator.
4. Выводы по проделанной работе.

Самостоятельная работа №2. Введение в разработку Android- приложений

Цель работы: знакомство с инструментами разработки Android- приложений.

Методические указания

Android - операционная система для мобильных устройств: смартфонов, планшетных компьютеров, КПК. В настоящее время именно Android является самой широко используемой операционной системой для мобильных устройств. Это бесплатная операционная система, основанная на Linux с интерфейсом программирования Java.

Платформа Android объединяет операционную систему, построенную на основе ядра ОС Linux, промежуточное программное обеспечение и встроенные мобильные приложения. Разработка и развитие мобильной платформы Android выполняется в рамках проекта AOSP (Android Open Source Project) под управлением OHA (Open Handset Alliance), руководит всем процессом Google.

Android поддерживает фоновое выполнение задач; предоставляет развитую

библиотеку элементов пользовательского интерфейса; поддерживает 2D- и BD-графику, используя OpenGL стандарт; поддерживает доступ к файловой системе и встроенной базе данных SQLite.

Приложение для Android пишется на языке Java, а среду разработки можно выбрать, например, из популярных средств разработки, таких как Android Studio или Eclipse.

Android Studio - среда разработки под Android, основанная на IntelliJ IDEA. Она предоставляет интегрированные инструменты для разработки и отладки. Для отладки приложений используется эмулятор телефона - виртуальная машина, на которой будет запускаться созданное приложение.

Чтобы создать эмулятор телефона, выбираем в меню Android Studio пункты SDK Manager | Tools | AVD Manager | Device Definitions.

Далее нужно выбрать нужную версию виртуального устройства и при необходимости отредактировать его параметры.

Далее рассмотрим этапы создания нового проекта. После запуска Android Studio выбираем New Project, появится диалоговое окно мастера.

Поле Application name - имя, которое будет отображаться в заголовке приложения.

Поле Company Domain служит для указания адреса вашего сайта.

Поле Package name формирует специальный Java-пакет на основе вашего имени из предыдущего поля.

Поле Project location позволяет выбрать место на диске для создаваемого проекта.

После нажатия на кнопку Next переходим к следующему окну. Здесь выбираем типы устройств, под которые будем разрабатывать приложение. В большинстве случае мы будет писать для смартфонов и планшетов, поэтому оставляем флажок у первого пункта. Также можно создавать приложения для Android TV, Android Wear и Glass.

Далее необходимо выбрать внешний вид экрана приложения. Выберем, например, вариант Blank Activity. После нажатия кнопки Finish студия формирует проект и создаёт необходимую структуру из различных файлов и папок.

В левой части среды разработки на вкладке Android появится иерархический список из папок, которые относятся к проекту. Вкладка Android содержит две основные папки: app и Gradle Scripts. Первая папка app является отдельным модулем для приложения и содержит все необходимые файлы приложения - код, ресурсы картинок и т.п. Вторая папка служит для различных настроек для управления проектом. Раскрываем папку app. В ней находятся три папки: manifest, java, res.

Папка manifest содержит единственный файл манифеста AndroidManifest.xml. В

этом файле должны быть объявлены все активности, службы, приёмники и контент-провайдеры приложения. Также он должен содержать требуемые приложению разрешения. «AndroidManifest.xml» можно рассматривать, как описание для развертывания Android-приложения.

Папка java содержит три подпапки - рабочую и для тестов. Рабочая папка имеет название пакета и содержит файлы классов. Сейчас там один класс MainActivity.

Папка res содержит файлы ресурсов, разбитых на отдельные подпапки.

Запуск программы осуществляется выбором команды Run в пункте меню Run. После этого в эмуляторе загрузится разработанная программа, на экране появится окно приложения с надписью «Hello World!» и заголовком программы.

Порядок выполнения работы

1. Изучить методические указания к самостоятельной работе №2.
2. Запустить на выполнение Android Studio.
3. Если будет использоваться внешний эмулятор для запуска Android-приложений (например, BlueStacks или Genymotion), загрузить его.
4. Создать в среде Android Studio проект, рассмотренный в самостоятельной работе №2.
5. Изучить содержимое файла AndroidManifest.xml из папки manifest и папку res, содержащую файлы ресурсов проекта.
6. Запустить созданное приложение в эмуляторе Android и наблюдать за появлением этого приложения и результатов его работы в окне приложений эмулятора.
7. Изучить содержимое файлов проекта activity_main.xml и MainActivity.java. Отредактировать их содержимое для получения нового текста на экране и запустить приложение в эмуляторе Android.

Отчет должен содержать:

1. Название и цель работы.
2. Листинг созданного приложения в Android Studio на языке Java.
3. Результаты работы приложения в эмуляторе телефона.

4. Выводы по проделанной работе.

Самостоятельная работа №3. Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android

Цель работы: научиться создавать приложения с элементами управления.

Методические указания

Часто при работе приложения на экране дисплея должны находиться элементы управления. Элементы управления - это доступные для манипулирования экранные объекты. Их можно разделить на четыре основные категории:

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

Командные элементы управления выполняют некоторые действия. Главным командным элементом является кнопка, которая обладает множеством вариантов отображения. Действие выполняется сразу после нажатия на кнопку. Часто особым образом выделяется кнопка по умолчанию, соответствующая наиболее часто используемому действию. Кнопки, помещенные на панель инструментов, обычно становятся квадратными, теряют текстовую надпись и обзаводятся пиктограммой - пояснением в виде графического значка.

Элементы выбора позволяют пользователю выбрать из группы допустимых объектов тот, с которым будет совершено действие.

Элементы выбора применяются также для действий по настройке. Распространенными элементами выбора являются флажки и списки. Щелкнув по флажку, пользователь немедленно увидит появившуюся галочку. Элементы управления типа «список» позволяют осуществлять выбор из конечного множества текстовых строк, каждая из которых представляет команду, объект или признак. Пользователь может выбрать единственную строку текста, нажав на нее.

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

Элементы управления отображением используются для управления визуальным представлением информации на экране. Типичными примерами элементов отображения

являются разделители и полосы прокрутки. Сюда же входят разделители страниц, линейки, направляющие, сетки и рамки.

Каждому объекту нужно задать размеры, координаты, цвет, текст и т.д. Android поддерживает способ, основанный на XML- разметке, который напоминает разметку веб-страницы. Можно использовать и визуальный способ перетаскивания объектов с помощью мыши.

Файлы XML-разметки находятся в папке res/layout проекта. Папка содержит ресурсы, не связанные с кодом. Кроме разметки, там же содержатся изображения, звуки, строки для локализации и т.д.

Откроем созданный на самостоятельной работе №2 проект. Когда разметка открыта в графическом представлении, то слева от основной части редактора кода можно увидеть панель инструментов, в которой сгруппированы различные элементы по категориям: Widgets, Texts, Layouts и т.д.

В группе Images найдите элемент ImageButton, перетащите его на форму и отпустите. Далее необходимо выбрать изображение для кнопки.

Во вкладке Component Tree выберем элемент Constraint Layout (экран). В панели свойств Properties отобразятся самые употребительные свойства выбранного компонента. К ним относятся идентификатор, ширина и высота. Выбираем View all properties (две стрелочки), чтобы открыть все свойства компонента. Найдите свойство background. Щелкните рядом с этим словом во второй колонке. Появится текстовое поле, в которое можно ввести значение вручную и кнопка с тремя точками, которая запустит диалоговое окно для создания ресурса. Переходим на вкладку Color и выбираем нужный цвет.

Далее можно поменять картинку для графической кнопки. Находим подходящее изображение и копируем его в папку res/drawable проекта. Затем выделяем элемент ImageButton на форме и в панели свойств выбираем свойство srcCompat. Снова щелкаем на кнопке с тремя точками и выбираем ресурс в категории Drawable - там должен появиться ресурс с именем добавленного ранее файла. Там же в окне свойств находим свойство onClick и вручную прописываем onClick - это будет именем метода для обработки нажатия на кнопку.

Далее установите курсор мыши внутри текста "onClick" у кнопки (переключившись в режим Text) и нажмите комбинацию Alt+Enter. Во всплывающем окне выбрать вариант Create 'onClick(View)' in 'MainActivity'. В коде класса MainActivity появится заготовка для обработки щелчка кнопки. Необходимо набрать следующий текст:

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity; import android.view.View; import
```

```
android.widget.TextView;
public class MainActivity extends AppCompatActivity { private TextView mHelloTextView;
@Override
protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); mHelloTextView=(TextView)findViewById(R.id.textView);
}
public void onClick(View view) { mHelloTextView.setText("Hello!");
}
}
```

Далее необходимо запустить программу в эмуляторе Android, нажать на созданную кнопку и наблюдать за изменением текста на экране.

Порядок выполнения работы

1. Изучить методические указания к самостоятельной работе №3.
2. Запустить на выполнение Android Studio.
3. Если будет использоваться внешний эмулятор для запуска Android-приложений (например, BlueStacks или Genymotion), загрузить его.
4. Открыть в среде Android Studio проект, разработанный в ходе выполнения лабораторной работы №2 или создать новый проект.
5. Разместить на форме элементы управления ImageButton и TextView, настроить их параметры.
6. В окне java-кода проекта добавить строки обработки нажатия на кнопку.
7. Запустить созданное приложение в эмуляторе Android и наблюдать за появлением этого приложения и результатов его работы в окне приложений эмулятора.
8. Добавить в проект другие элементы управления, настроить их свойства и проверить работу приложения в эмуляторе Android.

Отчет должен содержать:

1. Название и цель работы.
2. Листинг созданного приложения в Android Studio с использованием командных элементов управления, элементов выбора, элементов ввода и элементов отображения.
3. Результаты работы приложения в эмуляторе телефона.
4. Выводы по проделанной работе.

Самостоятельная работа №4. 2Б-анимация, создание и использование служб в приложениях под Android

Цель работы: знакомство с возможностями создания Android- приложений с использованием анимации.

Методические указания

Android предоставляет мощные API для анимации элементов пользовательского интерфейса и построения 2D и 3D изображений. Платформа Android предоставляет две системы анимации: анимация свойств, появившаяся в Android 3.0, и анимация компонентов пользовательского интерфейса (наследников класса View).

Анимация свойств (Property Animation) позволяет определить анимацию для изменения любого свойства объекта, независимо от того изображается оно на экране или нет.

Анимация компонентов пользовательского интерфейса используется для реализации анимации преобразований над наследниками класса View. Для расчета анимации преобразований используется следующая информация: начальная точка, конечная точка, размер, поворот и другие общие аспекты анимации. Анимация преобразований может выполнять серии простых изменений содержимого экземпляра класса View. Например, для текстового поля можно перемещать, вращать, растягивать, сжимать текст, если определено фоновое изображение, оно должно изменяться вместе с текстом. Пакет android.view.animation предоставляет все классы, необходимые для реализации анимации преобразований.

Для задания последовательности инструкций анимации преобразований используется XML код. Такие файлы с XML кодом должны располагаться в папке res/anim/ проекта.

Рассмотрим последовательность разработки приложения, использующую

анимацию.

Создадим новый проект в Android Studio. Далее подготовим несколько дополнительных файлов, которые будут использоваться при анимации. В папке drawable проекта создадим файл sun.xml с таким содержимым:

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android" android:dither="true" android:shape="oval" >
<gradient android:endColor="#ffff6600" android:gradientRadius="150" android:startColor="#ffffcc00" android:type="radial"
android:useLevel="false" />
<size
android:height="150dp" android:width="150dp" />
</shape>
```

Здесь для изображения солнца используется овал, а также для цвета применяется градиент - плавное изменение цвета от темножелтого (startColor) к светло-желтому (endColor).

В той же папке drawable создадим новый файл sky.xml следующего содержания:

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android" android:dither="true" android:shape="rectangle" >
<gradient android:angle="90"
android:endColor="#ff000033 "
android:startColor="#ff0000ff"
/>
</shape>
```

Здесь задана фигура в виде прямоугольника (rectangle) с голубым градиентом от нижнего края к верхнему.

В той же папке drawable создадим новый файл grass.xml следующего содержания:

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android" android:dither="true" android:shape="rectangle" >
<gradient android:angle="90"
```

android:endColor="#ff003300" android:startColor="#ff009900

" />

</shape>

Здесь задан зеленый прямоугольник с градиентом.

Далее в файл strings.xml в папке res/values добавим следующие строки:

<string name="sun">Солнце</string>

<string name="grass">Трава</string>

<string name="sky">Небо</string>

Затем откроем разметку главной активности activity_main.xml и добавим в неё несколько элементов ImageView. Должно получиться следующее содержимое файла:

<RelativeLayout xmlns:android= "http://schemas.android.com/apk

/res/android"

xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" tools:context=".MainActivity" > <ImageView

android:id="@+id/sky" android:layout_width="flll_parent" android:layout_height="fill_parent" android:contentDescription="@string/sk y" android:src="@drawable/sky" />

<ImageView android:id="@+id/sun"

android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_centerHorizontal="true" android:contentDescription="@string/sun" android:scaleType="fitCenter" android:src="@drawable/sun" />

<ImageView android:id="@+id/grass" android:layout_width="fill_parent"

android:layout_height="150dp" android:layout_alignParentBottom="true" android:contentDescription="@string/grass" android:src="@drawable/grass" />

</RelativeLayout>

У всех элементов ImageView в атрибуте android:src прописаны созданные фигуры, которые теперь можно видеть на экране.

Далее создадим новую папку res/anim, в которой будут находиться файлы анимации. В этой папке создадим новый файл sun_rise.xml со следующим содержимым:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas. android.com/apk/res/android" android:duration=" 5000"

android:fillAfter="true" android:interpolator="@android:ani m

/accelerate_decelerate_interpolator" android:shareInterpolator="false" >

<scale

android:fromXScale="1.0" android:fromYScale="1.0" android:pivotX="50%" android:pivotY="50%" android:toXScale="1.5" android:toYScale="1.5" />

<translate android:fromYDelta="80

%p" android:toYDelta="10%p"

/>

<alpha

android:fromAlpha="0.3" android:toAlpha="1.0" />

</set>

В блоке set установлены параметры анимации. Например, параметр android:duration показывает, что анимация должна совершиться в течение 5 секунд. Параметр fillAfter управляет состоянием анимации - она не должна прыгать в начало. Параметр android:interpolator использует системную константу для небольшого ускорения от начала к середине анимации и торможения от середины к концу анимации.

Внутри блока set устанавливаются специальные блоки, отвечающие за характер анимации: изменение размеров, позиции и прозрачности. Например, фигура солнца будет увеличиваться от своего начального размера в полтора раза равномерно от своей середины (scale). Элемент translate двигает солнце по экрану вертикально вверх. Мы отталкиваемся относительно родительского элемента, используя суффикс "p". Солнце начинает движение в позиции 80% от родительского элемента по оси Y и заканчивает движение в позиции 10%. При движении также меняется прозрачность солнца от полной прозрачности до полной непрозрачности (alpha).

Далее в файле MainActivity.java записываем программный код приложения:

import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.ImageView;

public class MainActivity extends AppCompatActivity { @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);

// Получаем ссылку на изображение солнца ImageView sunImageView =

(ImageView) findViewById(R.id.sun);

// Анимация для восхода солнца Animation sunRiseAnimation = AnimationUtils.loadAnimation(this,R.anim.sun_rise);

// Подключаем анимацию к нужному изображению View sunImageView.startAnimation(sunRiseAnimation);

}

}

Порядок выполнения работы

1. Изучить методические указания к самостоятельной работе №4.
2. Запустить на выполнение Android Studio.
3. Если будет использоваться внешний эмулятор для запуска Android-приложений (например, BlueStacks или Genymotion), загрузить его.
4. Создать в среде Android Studio проект, разработанный в самостоятельной работе №4.
5. Создать в соответствующих директориях проекта файлы grass.xml, sky.xml, sun.xml.
6. Добавить в проект графические изображения неподвижных и движущихся изображений.
7. Создать новую папку res/anim и добавить в нее файл sun_rise.xml.
8. В окне java-кода проекта добавить строки для получения анимированного изображения.
9. Запустить созданное приложение в эмуляторе Android и наблюдать за появлением этого приложения и результатов его работы в окне приложений эмулятора.
10. Создать новое приложение с применением анимированного изображения.

Отчет должен содержать:

1. Название и цель работы.
2. Листинг созданного приложения в Android Studio с использованием элементов графики и анимации.
3. Результаты работы приложения в эмуляторе телефона.
4. Выводы по проделанной работе.

Самостоятельная работа №5. Работа с Android Market

Цель занятия: изучить этапы работы пользователя с сервисом Android Market (Google Play).

Методические указания

Google Play (прошлое название - Android Market) - магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения. На сегодняшний день Google Play является ведущим магазином для распространения Android приложений.

Прежде чем публиковать приложение, надо провести несколько подготовительных действий. Одно из них - локализация. Рассмотрим процесс локализации приложения. Создадим приложение, которое будет поддерживать английскую и русскую локализации.

Для русской локализации необходимо создать новый подкаталог values-ru в каталоге res. Для этого щелкаем правой кнопкой мыши на папке res и выбираем New | Android resource directory. В диалоговом окне в левой части Available qualifiers: выбираем пункт Locale и переносим его в правую часть Chosen qualifiers. В появившейся третьей колонке выбираем нужные языки, например, русский. Вы увидите, что в поле Directory name автоматически появится нужное название папки.

Затем надо скопировать файл res/values/strings.xml в новую папку и изменить его содержимое на следующее:

<resources>

<string name= "app_name"> Локализованное приложение

</string> <string name-'hello_world">Здравствуй, M^!</string>

</resources>

Если на эмуляторе или реальном устройстве выбраны русские настройки, то вверху экрана увидим строку «Локализованное приложение».

Далее надо изменить ярлык для созданного приложения. Есть несколько способов создания иконок для приложений. Например, можно воспользоваться бесплатным онлайн сервисом Android Asset Studio по созданию иконок для Android приложений. Можно также создать собственное изображение, сохранить его в формате png, скопировать в папку проекта res/drawable и заменить строку для задания иконки приложения в файле AndroidManifest.xml на следующую:

android:icon="@drawable/имя файла иконки".

Любое приложение, отправляемое на Google Play, должно иметь подписанный

сертификат. Сертификат позволяет идентифицировать разработчика как автора программы. Создадим подписанный APK- файл (архивный исполняемый файл приложения для Android). Необходимо выбрать пункты меню Build | Generate Signed APK. Появится диалоговое окно мастера, которое необходимо заполнить данными. В первом поле следует указать путь к хранилищу ключей. Если хранилище создается первый раз, то выбираем кнопку Create new. Появится новое диалоговое окно.

В первом поле Key store path нужно выбрать папку и ввести имя для файла с хранилищем, которому будет присвоено расширение jks. Затем нужно заполнить поля Password и Confirm. Далее создаем ключ для приложения. В поле Alias (псевдоним) надо задать название ключа. Для ключа также нужно задать пароль и подтвердить его. В поле Validity (years) задается период действия ключа. В нижних полях заполняется информация о разработчике.

После нажатия кнопки Finish будет создан APK-файл.

Порядок выполнения работы

1. Изучить методические указания к самостоятельной работе №5.
2. Запустить на выполнение Android Studio.

3.Создать новый проект или загрузить проект, созданный при выполнении предыдущих лабораторных работ.

4. Выполнить локализацию приложения.
5. Создать ярлык для разработанного приложения.
6. Создать подписанный сертификат разработанного приложения и сгенерировать APK-файл для публикации приложения на Android Market (Google Play).

Отчет должен содержать:

1. Название и цель работы.
2. Листинг локализованного приложения в Android Studio, использующего созданный для данного приложения ярлык.
3. Результаты работы приложения в эмуляторе телефона.
4. Выводы по проделанной работе.

Тема самостоятельного изучения № 6. Введение в программирование для мобильных устройств.

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Введение: обзор современных мобильных устройств (Android, IPhone, Windows Phone), технологии разработки мобильных приложений на этих платформах. Языки программирования: Java (Android), Swift (IPhone), Javascript (Windows Phone и другие).

Тема самостоятельного изучения № 7 Обзор платформы Android.

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Преимущества и недостатки платформы. Архитектура Android. Основные компоненты. Обзор среды разработки Android Studio: установка, настройка, использование. Эмулятор мобильного устройства.. Пример: разработка первого мобильного приложения.

Тема самостоятельного изучения № 8. Пользовательский интерфейс

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Класс Application. Меню. Разметка. Представления. События. Анимация.

Тема самостоятельного изучения № 9. Намерения, данные.

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Адаптеры. Намерения в Android: явные и неявные. Запуск

Активностей с помощью Намерений. Работа с настройками и состоянием приложения. Работа с файлами.

Тема самостоятельного изучения № 10. Работа с СУБД

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Базы данных в Android. СУБД SQLite. Работа с БД в Android: выполнение запросов, получение и изменение данных. Применение адаптеров.

Тема самостоятельного изучения № 11. Использование сетевых сервисов

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Контент-провайдеры: создание, использование. Интернет-сервисы: использование. Широковещательные Приемники: регистрация, применение, жизненный цикл. Broadcast.

Тема самостоятельного изучения № 12. Развертывание мобильного приложения в маркете

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Подготовка к публикации разработанного мобильного приложения. Развертывание приложения в Google-маркете.

Тема самостоятельного изучения № 13. Установка Android Studio

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Версии ОС Android, средства необходимые для начала

разработки под ОС Android, основные преимущества и недостатки ОС Android, ОС, под которыми возможно разрабатывать программное обеспечение под ОС Android.

Тема самостоятельного изучения № 14. Создание нового проекта

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Типы устройств, которые поддерживает IDE Android Studio, основные файлы проекта, созданного по умолчанию, способы добавления сторонней библиотеки в разрабатываемое приложение, apk-файл – характеристики и способы получения.

Тема самостоятельного изучения № 15. Жизненный цикл Activity

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Элемент Activity, элементы жизненного цикла Activity, методы Activity, вызываемые при смене ориентации устройства, связывание интерфейса с Activity.

Тема самостоятельного изучения № 16. Использование ресурсов приложения

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Файлы ресурсов приложения, способы получения доступа к элементу файла ресурса приложения, изменения в файлах R.java.

Тема самостоятельного изучения № 17. Layout-файл в activity. Смена ориентации экрана

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Каталоги приложения, в которых хранятся файлы ресурсов, отличия элемента <LinearLayout> от элемента <RelativeLayout>, действия в программе, которые необходимо предусмотреть при смене ориентации экрана.

Тема самостоятельного изучения № 18. Всплывающие уведомления / toast notification

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Варианты разработки приложения под ОС Android, в которых использование уведомлений Toast оправдано, отображение Toast уведомления на базе собственной разметки, параметры, которыми регулируется время отображения уведомления на экране.

Тема самостоятельного изучения № 19. Уведомления / push notification

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Принцип работы push-нотификаций, программное отслеживание доставки push-нотификация пользователю, изменение разметки отображения push-нотификации, навыки, необходимые разработчику, для использования push-нотификаций.

Тема самостоятельного изучения № 20. Локализация приложения

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Цели, в которых используется локализация в приложении, структура хранения констант в файлах, добавление локализации к уже созданному приложению.

Тема самостоятельного изучения № 21. Переключение между экранами

Вид деятельности студентов: самостоятельное изучение учебно-методической

литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Способы перехода между «окнами» в приложении на ОС Android, класс Intent в Android приложении, типы данных, которые можно передавать через область extraData в классе Intent.

Тема самостоятельного изучения № 22. Организация сервиса в приложении

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Сервис в Android приложении, отличия сервиса от Activity, способы организации автоматически перезапускаемого сервиса.

Тема самостоятельного изучения № 23. Сохранение данных в приложении

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Способы сохранения данных во внутренней и внешней памяти, методы создания объекта Shared preferences, типы директорий во внешней памяти.

Тема самостоятельного изучения № 24.

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План конспекта: Случаи использования Shared Preferences и БД, Преимущества и недостатки использования БД на мобильном устройстве, класс, используемый для открытия соединения с БД.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чезарини, Ф. Программирование в Erlang / Ф. Чезарини, С. Томпсон. - М. : ДМК Пресс, 2012. - 487 с. - (Функциональное программирование). - ISBN 978-5-94074-617-1 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=232035

2. Операционная система Android / . - М. : МИФИ, 2012. - 64 с. - ISBN 978-5-7262-1780-2 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=231690

3. Михеева, Е. В. Информационные технологии в профессиональной деятельности : учеб. пособие / Е.В. Михеева. - 14-е изд., стер. - М. : Академия, 2016. - 384 с.

4. Гохберг, Г. С. Информационные технологии : учебник / Г.С. Гохберг, А.В. Зафиевский, А.А. Короткин. - 9-е изд., перераб. и доп. - М. : Академия, 2014. - 240 с.

5. Хлебников, А. А. Информационные технологии : учебник / А. А. Хлебников. – М. :КноРус, 2014. – 472 с.

6. Васильев, А. Н. Java. Объектно-ориентированное программирование : [учеб. пособие] / А.Н. Васильев. - СПб. : Питер, 2012. - 400 с. : ил. - (Учебное пособие). - Прил.: с. 379-395. - Библиогр.: с. 377.

7. Винокуров Н.А. Практика и теория программирования. В 2 Кн. Кн. 1 Ч. I и II [Текст]: учеб. издание/ Н.А. Винокуров, А.В. Ворожцов. – М.: Физматкнига, 2008. – 192 с.

8. Винокуров Н.А. Практика и теория программирования. В 2 Кн. Кн. 2 Ч. III и IV [Текст]: учеб. издание/ Н.А. Винокуров, А.В. Ворожцов. – М.: Физматкнига, 2008. – 288 с.