МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Озерский технологический институт — филиал НИЯУ МИФИ

Кафедра прикладной математики

Вл. Пономарев

ПРАКТИКУМ

по программированию мобильных приложений

Учебно-методическое пособие

Озерск, 2019

УДК 681.3.06
П 56

Вл. Пономарев. Практикум по программированию мобильных приложений. Учебно-методическое пособие. Озерск: ОТИ НИЯУ МИФИ, 2019. — 16 с.

В пособии описываются практические работы по курсу «Современные технологии программирования». Работы второго семестра изучения дисциплины включают в себя программирование мобильных приложений, работающих под управлением операционной системы Android.

В качестве основного материала при выполнении практических работ пособие предназначено для студентов, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и специальности 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения».

Рецензенты:
1.
2.

УТВЕРЖДЕНО
Редакционно-издательским
Советом ОТИ НИЯУ МИФИ

Содержание

Введение

Эта предварительная версия документа находится в стадии написания.
Документ предназначен для студентов ОТИ НИЯУ МИФИ.

Целью работ является разработка уменьшенного варианта игры под названием «Королевский квадрат». Прочитать об этой игре можно в номере №7 журнала «Наука и жизнь» за 1980 год. Отсканированные страницы с описанием игры можно найти на сайте http://revol.ponocom.ru/ksg. В отличие от оригинальной версии, мы разрабатываем минимально возможную игру в квадрате размером 3 на 3 клетки, в которой можно ввести всего шесть коротких слов.

1. Работа МА-001. Среда разработки

Цели:
- знакомство со средой разработки
- изучение компонентов, необходимых для разработки приложений.

Задачи:
- формирование диска с Android SDK;
- создание проекта, изучение его структуры;
- создание разметки.

1.1. Рабочий диск

Для выполнения работ требуется отдельный съемный диск (флешка), на котором будут размещены SDK для Android размером более 4Г, а также разрабатываемые проекты.

Рекомендуется всегда работать за одним и тем же компьютером. Тогда установите съемный диск, откройте управление компьютером в панели инструментов и установите для диска букву "H". В дальнейшем все ссылки на SDK и на проекты будут указываться относительно этого диска.

Создайте на диске два каталога:
- anpro — для размещения проектов,
- ansdk — для размещения Android SDK.

Первая задача, которую нужно выполнить — скачать Android SDK в каталог H:\ansdk.

1.2. Пакеты разработчика мобильных приложений

Для программирования мобильных приложений под Android требуется два пакета разработчика:
- Java 2 SDK
- Android SDK

Первый пакет устанавливается на рабочий компьютер.

Пакет может быть уже установлен на компьютере. Чтобы проверить, откройте каталог Program Files\Java. Если в нем есть каталог с именем, содержащим jdk, например, jdk1.8_0_25, пакет, видимо установлен. Если нет, скачайте пакет Java2 для требуемой операционной системы с сайта производителя Java.

После приобретения нужного файла запустите его и следуйте указаниям программы установки. Если не удалось установить Java SDK, дальнейшие действия не имеют смысла, без этого пакета выполнение работ невозможно.

Пакет Android SDK скачивается из Интернет при помощи программы SDK Manager.exe, которая запускается обычно из среды разработки.

Поскольку установка данного пакета из Интернет занимает длительное время, пакет может быть предоставлен преподавателем.

Для программирования под Android можно использовать несколько разных сред разработки. Рекомендуемая среда — IntellJIdea. Если она не установлена, то можно либо скачать бесплатную версию с сайта производителя, либо использовать, например, Microsoft Visual Studio.

Если у вас есть мобильное устройство с Android, то вам нужно найти и установить на рабочий компьютер драйвер для работы с устройством через ADB (Android Debug Bridge, отладочный мост Android), если вы хотите отлаживать приложения непосредственно на мобильном устройстве.

1.3. Создание проекта

Разработка мобильных приложений будет описываться в среде разработки IntellJIdea. После запуска среды в первый раз появится диалог для открытия или создания проекта.

Рисунок 1 — Начальный диалог

В других случаях вызвать этот диалог можно, выбрав в меню File — Close Project. Если в среде уже были созданы проекты, то слева в начальном диалоге появится их список, в котором можно выбрать проект быстро. Чтобы открыть проект, можно также выбрать «Open». В других случаях этот диалог можно вызвать, выбрав в меню File — Open Project.

Мы создаем новый проект, поэтому выбираем «Create New Project». Появится первичный диалог для создания нового проекта (рисунок 2).

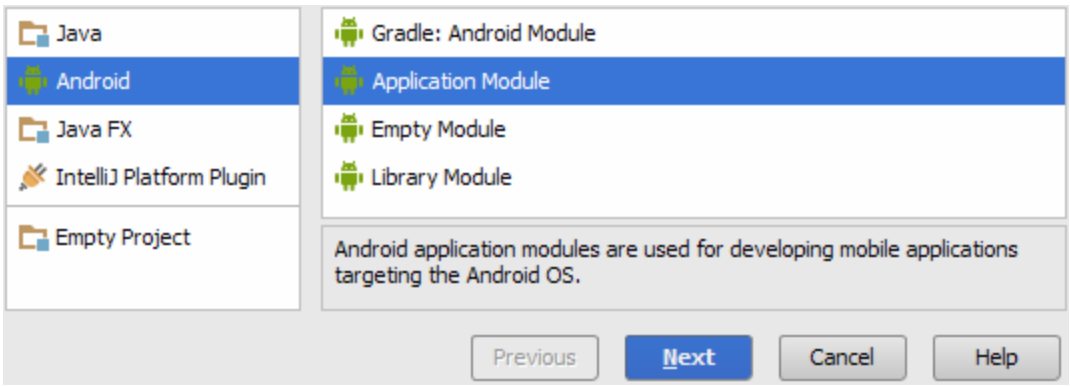

Рисунок 2 — Выбор типа проекта

Выбираем слева Android, справа — Application module, нажимаем кнопку Next. Появится диалог, в котором нужно ввести название приложения и первичной активности. Введем название проекта MiniKS (*Minimal Kings Square*), название первичной активности KSActivity (рисунок 3).

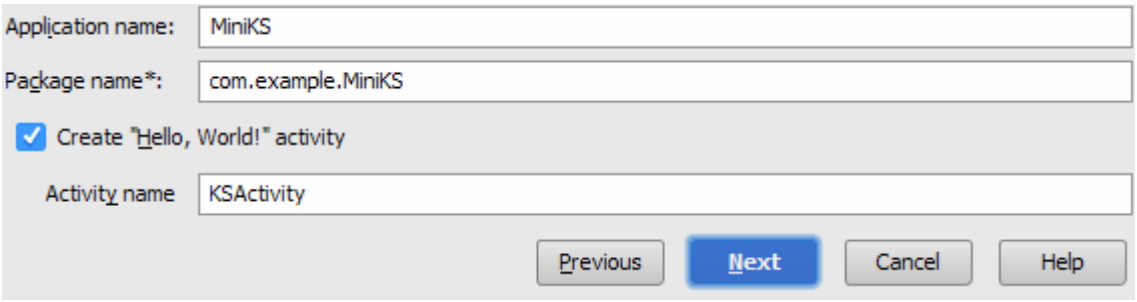

Рисунок 3 — Название приложения и первичной активности

Следующий диалог задает расположение проекта и SDK (рисунок 4).

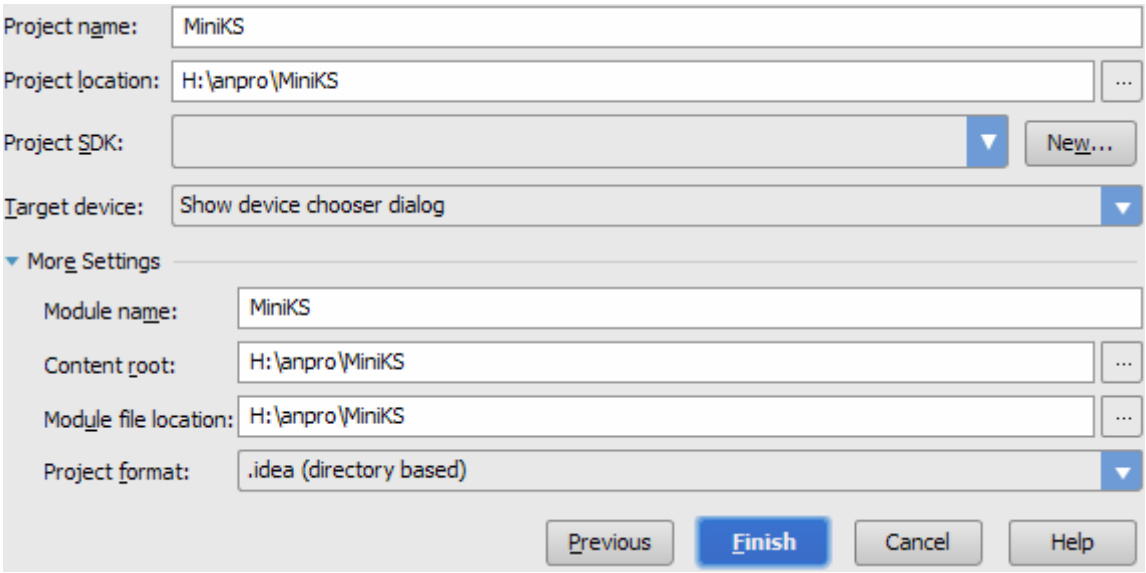

Рисунок 4 — Расположение проекта и SDK

Здесь нужно указать, что проект размещается в папке H:\anpro и выбрать версию Android SDK, обозначенную «Project SDK».

При первом старте выбирать нечего, SDK еще не заданы, нужно указать их расположение. Для этого нажмем кнопку «New…». Появится диалог предупреждения (рисунок 5).

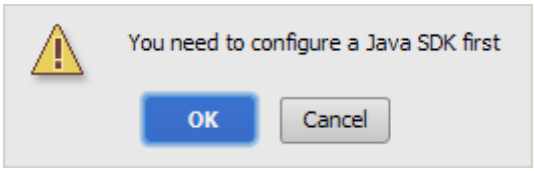

Рисунок 5 — Предупреждение о необходимости настроить Java SDK

Нажимаем OK. Появится диалог для выбора размещения Java SDK, в котором нужно указать папку, в которой находится Java SDK.

Возможно, что путь к Java SDK у вас немного другой. Неправильную папку выбрать вы не сможете. Если среда разработки не обнаружит Java SDK, то процесс создания проекта придется отложить до того момента, когда будет установлен пакет Java SDK или пока вы не найдете его.

Сразу после выбора размещения Java SDK появится еще один такой же диалог для выбора размещения Android SDK, в котором нужно указать путь H:\ansdk. Чтобы понять, что в данный момент вы выбираете, читайте заголовок окна диалога.

После того, как вы зададите расположение SDK, появится диалог, в котором можно выбрать версию Java SDK и Android SDK (рисунок 6).

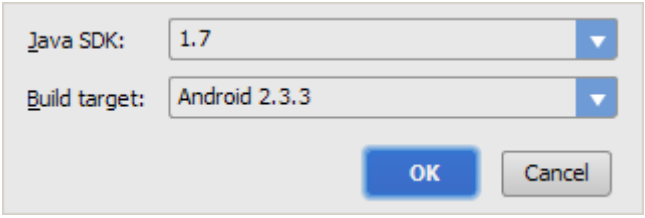

Рисунок 6 — Выбор SDK

В этом диалоге выбираем версию Android 2.3.3. Заметим, что у вас версия Java SDK может быть другой. Нажимаем кнопку Finish.

Появится диалог уведомления, в котором говорится, что для проекта автоматически будет создана папка (рисунок 7).

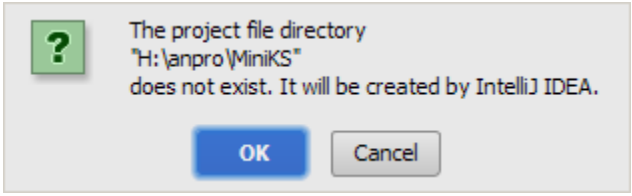

Рисунок 7 — Уведомление о создании папки проекта

Нажимаем OK и ждем, когда среда подготовит проект. Это может занять несколько минут. При первом старте после подготовки проекта появится диалог, в котором приводится совет разработчику. Выключим в этом диалоге флажок «Show Tips on Startup» и закроем диалог кнопкой «Close». При желании, конечно, можно почитать советы, выбирая их при помощи кнопки «Next Tip».

1.4. Дерево проекта

Изучим дерево проекта, которое отображает его состав. Сначала нужно открыть какой-нибудь модуль. Развернем дерево проекта (в среде разработки на левой панели вверху, там, где написано Project), разверните папки layout, values, src, в папке src дважды щелкните на файл класса KSActivity, чтобы в правой части среды появилось содержимое файла. На рисунке 8 показан примерный вид среды после выполнения указанных действий (показана только часть).

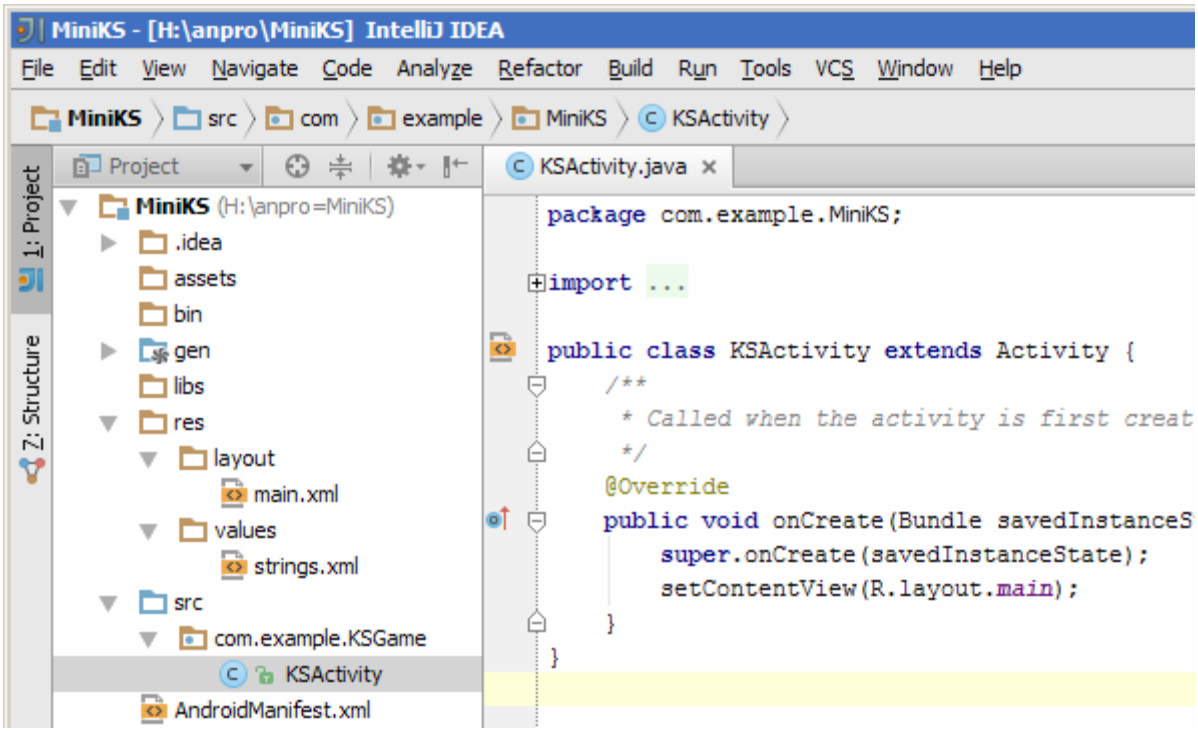

Рисунок 8 — Дерево проекта

Рассмотрим основные части дерева проекта (на левой панели).

Первые 4 папки (.idea, assets, bin, gen) нас в данный момент не интересуют, да и в последующем будут интересовать редко. Папка lib может содержать какие-то сторонние библиотеки, пока нам они не нужны.

Важной является папка res (resources). В этой папке сосредоточены ресурсы приложения. Она может содержать множество различных папок, на рисунке приведены только некоторые.

Так, папка layout содержит файлы разметки, с которыми мы познакомимся чуть позднее. Папка values содержит файлы, описывающие различные значения, например, строковые. Сейчас в ней есть файл strings.xml, содержащий строковые ресурсы приложения, мы его также рассмотрим позже. В показанном дереве нет папки drawable, в которой размещаются графические ресурсы (картинки), но у вас эта папка может быть.

Папка src (sources) содержит файлы классов Java и другие программные тексты. В ней сейчас только один файл, сгенерированный средой, это единственный файл программного кода, который сейчас есть, он описывает первичную активность нашего приложения. Текст файла вы видите в правой части среды. Обратим внимание, что фактическое расположение файла на дереве не отображается, но в третьей строке окна программы (под строкой меню) вы можете проследить фактический путь к данному файлу. Он включает в себя элементы названия пакета.

1.5 Манифест приложения

В нижней части дерева видим файл AndroidManifest.xml, манифест приложения Android. Откроем его.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.example.MiniKS"
          android:versionCode="1"
          android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10"/>
    <application android:label="@string/app_name">
        <activity android:name="KSActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUN
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Первая строка файла одинакова для всех xml-файлов, она говорит о том, что это файл xml версии 1 в кодировке «utf-8». В дальнейшем упоминать эту сроку больше не будем и приводить в листинге тоже.

К файлу манифеста мы будем обращаться в двух важных случаях:

- чтобы установить разрешения для приложения;
- чтобы скорректировать параметры активности.

Таким образом, две важные вещи есть в этом файле, — это разрешения (permissions) и активности (activity). Манифест приложения содержит и другие параметры, такие как требуемая версия Android SDK или минимальная версия SDK, но обращаться к ним приходится очень редко.

Разрешения — это те или иные средства мобильного устройства и системы Android, которые мобильному приложению разрешается использовать. Если вы устанавливали приложения на свое мобильное устройство, то, возможно, обратили внимание, что для каждого приложения есть список тех привилегий, которые приложению требуются для работы, и который вы должны принять, то есть разрешить приложению, например, использовать телефонную книгу. Нам по ходу разработки программы понадобятся некоторые разрешения, и тогда мы их добавим в файл манифеста.

Манифест должен описывать все активности приложения. Одна из активностей является стартовой. У нас одна активность, она, соответственно, является стартовой. Для каждой активности устанавливается метка, которая появляется в верхней строке экрана. В файле манифеста она указывается параметром label. Заметим, что в манифесте описывается также метка приложения. Сейчас вы можете видеть один из двух вариантов отображения метки: либо вы видите текст, написанный в файле манифеста, либо вы видите текст метки. В приведенном выше примере текст метки приведен так, как он записан в файле манифеста, то есть

```
android:label="@string/app_name">
```

Эта запись означает, что в данном месте должен быть вставлен текст из строковых ресурсов приложения, помеченный именем app_name.

По мере того, как к тексту, содержащему такую запись, обращаются все реже и реже, среда начинает вместо него отображать текст, записанный в ресурсах, при этом цвет этого текста становится другим (серым):

```
android:label="MiniKS">
```

Если при этом щелкнуть мышью на серый текст, то он развернется в то, что фактически записано в файле.

Возможно, что в файле вашего манифеста указан значок программы. В этом случае у вас должна быть папка res/drawable, в которой находится картинка со значком программы по умолчанию.

Дополнительно для активности может быть задана тема, то есть некоторое заранее подготовленное оформление. Тогда в файле манифеста появятся дополнительные строки, описывающие тему.

1.6 Разметка

Изображение на экране мобильного приложения формируется с помощью разметки. Разметка — это xml файл, описывающий, какие элементы есть на экране и как они размещены друг относительно друга. Разметки являются существенной частью разработки любого приложения и, честно говоря, головной болью начинающего Android-программиста, поскольку если опыта маловато, сформировать правильную разметку удается с большим трудом.

Откроем модуль res/layout/main.xml.

При первом открытии модуля разметки мы увидим изображение некоторого мобильного устройства. Чтобы увидеть текст, в нижней части среды выберите вкладку «Text» вместо вкладки «Design». Текст разметки имеет примерно следующий вид (без первой строки):

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andr
              android:orientation="vertical"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
        >
    <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello World, KSActivity"
            />
</LinearLayout>
```

Вторая строка открывает тег LinearLayout, описывающий представление-контейнер, которое может содержать другие представления в линейном порядке, либо горизонтальном, либо вертикальном. Порядок задан параметром orientation, и мы видим, что в данном случае это контейнер вертикального размещения внутренних представлений.

Замечание относительно терминологии. Представлением здесь называется то, что в оригинале называется View (вид). К этому классу относятся представления двух видов — контейнеры и то, что в Windows мы называем элементом управления, а в Android называется «виджет».

Разница между различными представлениями в том, что контейнеры предназначены для размещения других представлений, в оригинале они называются Layout, то есть размещение, а виджеты не могут размещать в себе других представлений, они как бы конечные представления. Виджеты могут отображать, например, текст, редактор, картинку, список.

Общим для всех представлений является то, что любое представление есть часть экрана мобильного устройства и любое представление можно отобразить тем или иным способом, хотя обычно контейнеры не предназначены для отображения.

В результате изображение на экране формируется разметкой как дерево представлений. Важно, что любая разметка начинается с контейнера, такого, как LinearLayout. Первый контейнер называется корневым (root).

В контейнеры можно вкладывать виджеты и другие контейнеры. Цель построения дерева представлений — получить правильное размещение виджетов, поскольку видим мы обычно виджеты, хотя возможен вариант использования только контейнеров.

Внутри контейнера LinearLayout, как видно из текста, размещен виджет, обозначенный тегом TextView, то есть представление текста. Заметим, что тег TextView не имеет соответствующего закрывающего тега, потому что в виджет ничего вложить нельзя.

Рассмотрим параметры (атрибуты) представлений.

В тексте есть два важнейших параметра, layout_width и layout_height. Они должны быть заданы для всех представлений и обозначают ширину и высоту представления соответственно. Эти параметры задаются обычно тремя значениями, называемыми fill_parent, wrap_content и match_parent.

Первое значение является устаревшим и его следует всегда заменять на третье значение, мы сейчас это сделаем. Таким образом, в дальнейшем мы будем использовать только два из приведенных значений, match_parent и wrap_content.

Значение match_parent указывает, что ширина или высота представления максимальна, и представление занимает всю ширину или высоту, предоставляемую контейнером. Заметим, что контейнер есть всегда. Для корневого элемента контейнером является площадь экрана, предоставляемая активности (или фрагменту), которая использует данную разметку.

Значение wrap_content указывает, что высота или ширина представления минимальна и определяется содержанием данного представления. Если представлением является виджет TextView, как в нашей разметке, то высота его будет определяться текстом и шрифтом.

Для начала научимся вводить новые значения. Удалим первое значение fill_parent и начинаем вводить значение match_parent. Достаточно ввести первые две буквы, чтобы увидеть подсказку (возможно, нужно немного подождать). Чтобы ввести значение подсказки, нажмем Enter. Заменим все значения fill_parent.

Заметим, что все параметры начинаются с android. Начинающие программисты обычно недоумевают, зачем все время нужно писать это слово. Это так называемое пространство имен, в котором находится название параметра, и это пространство не всегда android.

Добавим новый параметр в представление LinearLayout:

```
android:background="#00F"
```

Введем новую строку, начнем писать слово android, только две буквы, появится подсказка, — нажмем Enter, затем опять две буквы, появится подсказка, — опять нажмем Enter, и введем значение «#00F», означающее красный цвет. Мы увидим, как экран на эмуляторе окрасится синим цветом, и покажет, какую площадь занимает корневой элемент.

Добавим аналогичную строку в виджет, и еще одну строку, задающую цвет текста:

```
android:background="#F00"
android:textColor="#FFF"
```

Теперь мы видим границы виджета TextView.

Добавим параметр виджета, задающий выравнивание текста:

```
android:gravity="center"
```

В результате текст виджета разместится по центру.

Заменим значение параметра виджета layout_width на wrap_content:

```
android:layout_width="wrap_content"
```

Видим, что ширина виджета приняла минимальное значение. Добавим параметр виджета, задающий его выравнивание внутри контейнера:

```
android:layout_gravity="center"
```

Видим, что виджет разместился по центру контейнера по ширине.

Выровнять виджет по высоте тоже можно, но это тема другого занятия, пока оставим разметку как есть.

1.7 Строковые ресурсы

Откроем файл strings.xml. Он выглядит примерно следующим образом (без первой строки):

```
<resources>
    <string name="app_name">MiniKS</string>
</resources>
```

Все строковые значения, используемые в программе, должны описываться в строковых ресурсах. Сейчас мы добавим один строковый ресурс, название программы. Назовем ресурс ks_name, чтобы не конфликтовать с имеющимся строковым ресурсом:

```
<resources>
    <string name="app_name">MiniKS</string>
    <string name="ks_name">Kings Square</string>
</resources>
```

Вернемся к тексту разметки и заменим текст виджета TextView на текст из нового строкового ресурса:

```
android:text="@string/ks_name"
```

Смотрим на эмулятор экрана и видим, как меняется текст.

Щелкнем правой кнопкой на папку res в дереве проекта и в контекстном меню выберем «New — Directory». Появится диалог для ввода имени папки, введем название «values-ru», нажмем Enter, появится новая папка.

Щелкнем правой кнопкой на новую папку values-ru, выберем в контекстном меню «New — Values resource file». Появится диалог для ввода имени файла, введем «strings», нажмем Enter, появится новый файл.

Скопируем в новый файл описание строки ks_name.

Новый файл задает строковые значения, локализованные для России. Если в устройстве используется русский язык, то виджет должен отобразить строку на русском языке. Сейчас эмулятор экрана настроен на оригинальный язык, поэтому эмулятор отображает строку из оригинального файла strings.xml.

1.8 Эмулятор

Для тестирования проекта требуется эмулятор мобильного устройства или само устройство. Эмулятор нужен в любом случае, поэтому подготовим его. Откроем FAR. В каталоге H:\ansdk запустим AVD Manager. Нажмем кнопку Create. Вводим следующие значения:

AVD Name: AVD_10
Device: Nexus S (4.0", 480 × 800: hdpi)
Target: Android 2.3.3 - API Level 10
CPU/ABI: ARM (armeabi)
Keyboard: Hardware keyboard detect present (флажок включить)
Skin: Skin with dynamic hardware controls
Front Camera: None
Back Camera: None
Memory Options: RAM: 343 VM Heap: 32
Internal Storage: 200 MiB
SD Card: Size: 100 MiB
Emulation Options: Use Host GPU (флажок включить).
Нажимаем кнопку OK и далее, вероятно, еще раз OK.
AVD Manager должен показывать устройство (рисунок 9).

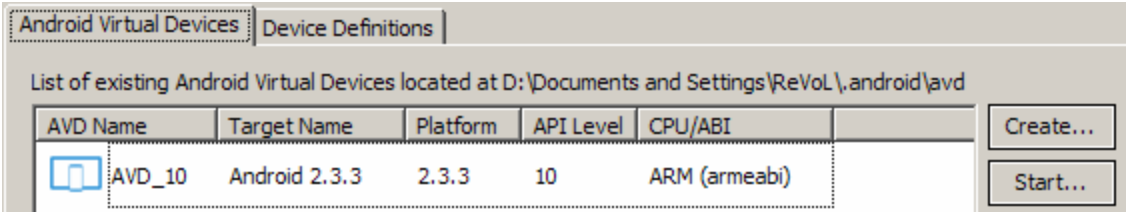

Рисунок 9 — Виртуальное устройство

1.9 Запуск программы

Если эмулятор создан, можно попробовать запустить на нем программу. Для этого нажмите Shift+F10, или используйте меню Run. В первый раз появится диалог, примерно такой, как на рисунке 10.

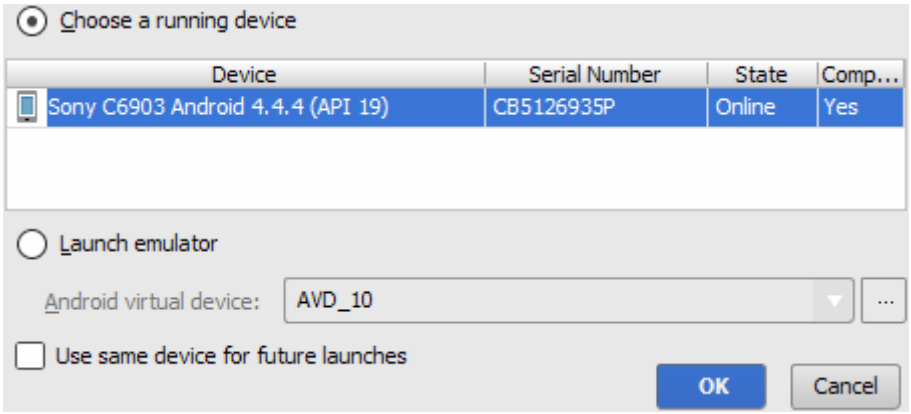

Рисунок 10 — Выбор устройства для тестирования

На рисунке видно, что есть какое-то подключенное устройство и незагруженный эмулятор. Выберите «Launch emulator» и нажмите OK.

Загрузка эмулятора занимает длительное время. Поэтому никогда не закрывайте его, а запускайте программу снова. Флажок «Use same device for future launches» означает всегда использовать выбранное устройство.

Если у вас есть мобильное устройство и установлен драйвер ADB, вы можете проверить работу программы на своем устройстве. Для этого устройство нужно сначала подготовить, переведя его в режим разработчика. После этого выберите в настройках пункт для разработчика программ. В нем нужно разрешить отладку программ через USB. Далее подключаете кабель USB и ждете появления на устройстве диалога для разрешения отладки с данного компьютера. Выберите в нем флажок «Всегда разрешать отладку с данного компьютера» и подтвердите диалог.

1.10 Код программы

Код программы состоит из класса активности. В нем переопределен метод OnCreate, который вызывается при создании активности:

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

В методе OnCreate вызывается конструктор базового класса и к активности подключается разметка с помощью метода setContentView.