Introduzione ad Android

Lezione 5

Android programming – Introduction

Ruggero Donida Labati

Laboratorio di Sistemi Operativi
Università degli Studi di Milano
Dipartimento di Informatica
A.A. 2022/2023

Sommario

1. What is an Android?
2. Writing code for Android devices
3. Android SDK
4. HelloWorld
5. APIs

What is an Android (1/6)

- The word “android” literally means a robot
- Android is an OS for the mobile devices
- Android is more than an OS, it’s a complete software stack
- Mainly developed by Google

What is an Android (2/6)

- Product of Open Handset Alliance
- Based on Linux kernel
- Ranked 1st in sales of smart devices
- 100,000 apps freely available for Android
- Code written in Java language

What is an Android (3/6)

- We can create powerful Java applications in Android
- Android has a potential market beyond mobile devices
- Some of Android code being written for non mobile applications

What is an Android (4/6)

- Android was made freely available under Apache open source license in Oct 2008
- Estimated 3 billion users

What is an Android (5/6)

- Connectivity
  - WIFI, Bluetooth and GPRS, EDGE, and 3G
- Hardware
  - Support for GPS, accelerometers and Cameras
- Graphics
  - Built in 2D/3D support including OpenGL

What is an Android (6/6)

- Storage
  - SQLLite
- Browser
  - The web browser is based on Webkit
- Interaction
  - Supports modern features like multi touch and multi tasking

Writing code for Android devices (1/7)

- To develop android applications we need
  - IDE: Officially Eclipse is used
  - Android SDK
  - Android Developer Tools plug-in for Eclipse
- Java Coding in eclipse is very intuitive
  - Rich Java environment like context
  - Sensitive help and code suggestions

Writing code for Android devices (2/7)

- Android SDK
  - Provides tools and APIs to begin developing applications on the Android using Java
  - Used to build, compile, test and debug user applications
  - Can be downloaded for Linux, Windows and Mac
  - We can add, delete and update components in Android SDK
  - To begin development we need Eclipse IDE with ADT plug-in

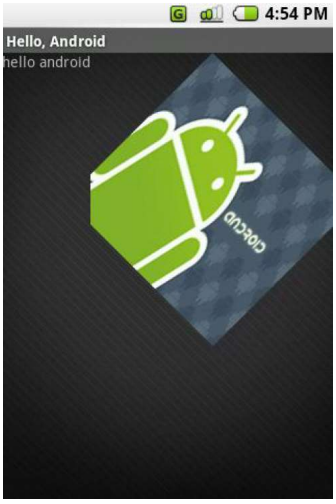
Writing code for Android devices (3/7)

- Creating and Deploying android application
  - Activity
  - Service
  - Content Provider
  - Processes and Tasks

Writing code for Android devices (4/7)

Broadcast receivers can trigger intents that start an application

Data storage provide data for your apps, and can be shared between apps – database, file, and shared preferences (hash map) used by group of applications



Activity is the presentation layer of your app: there will be one per screen, and the Views provide the UI to the activity

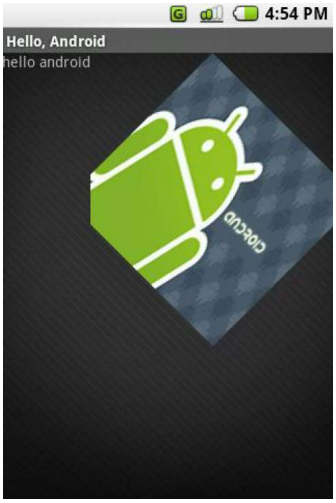
Intents specify what specific action should be performed

Services run in the background and have no UI for the user – they will update data, and trigger events

Writing code for Android devices (5/7)

Views such as lists, structure grids, text boxes, buttons, and even an embeddable web browser

Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data



An **Activity Manager** that manages the life cycle of applications and provides a common navigation backstack

A **Notification Manager** that enables all apps to display custom alerts in the status bar

A **Resource Manager,** providing access to non-code resources such as localized strings, graphics, and layout files

Writing code for Android devices (6/7)

- How it works
  1. Write app in Java
  2. Compiled in Java
  3. Transformed to Dalvik bytecode
  4. Loaded into Dalvik VM

Writing code for Android devices (7/7)

- Dalvik Virtual Machine
  - Responsible for running android Java programs
  - Optimized for low memory.
  - Designed to allow multiple VM instances to run
  - Relies on OS for process isolation, memory management and threading.
  - Executes Dalvik(DEX) files
  - DEX files are zipped into android package (APK)

Android SDK (1/9)

android studio

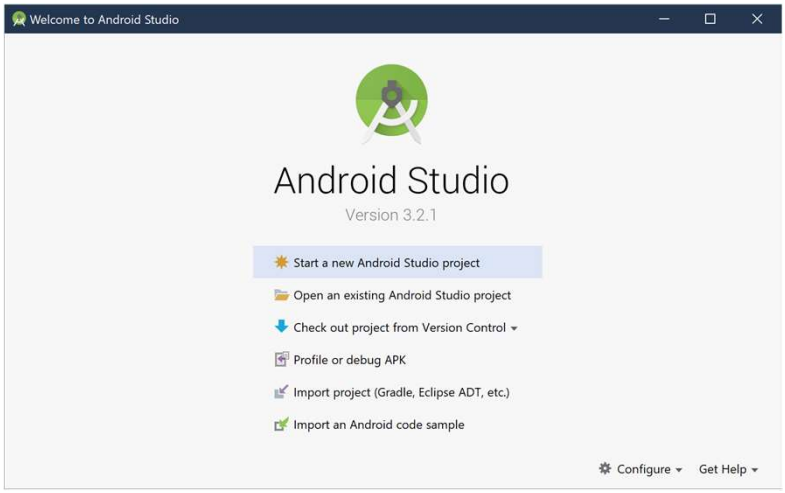
Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

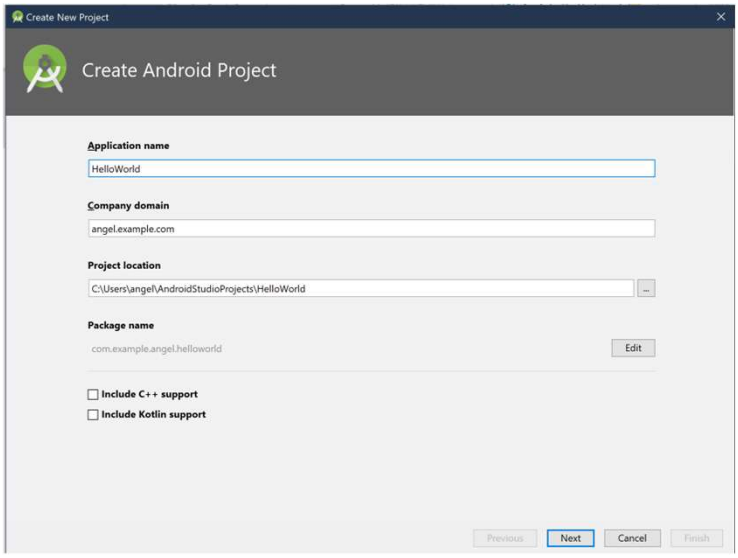
3.6.2 for Windows 64-bit (748 MB)

https://developer.android.com/studio

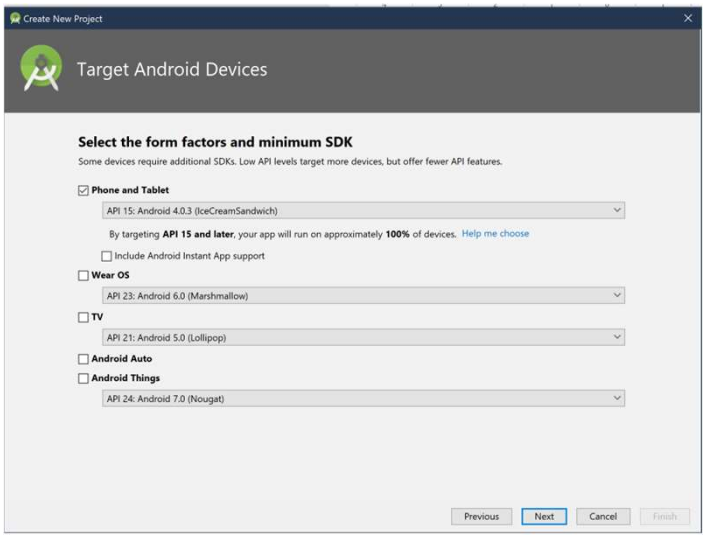
Android SDK (2/9)



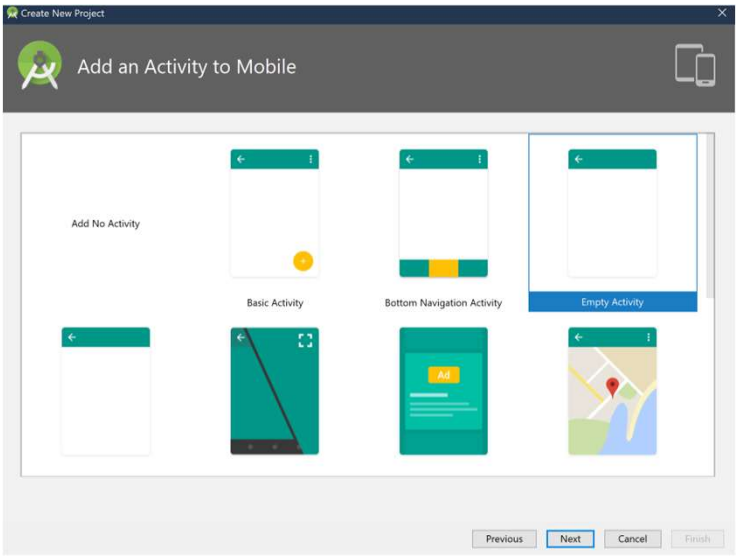
Android SDK (3/9)



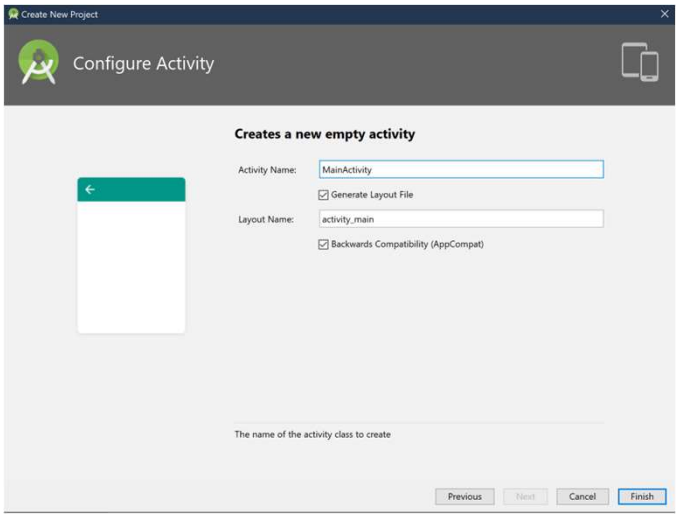
Android SDK (4/9)



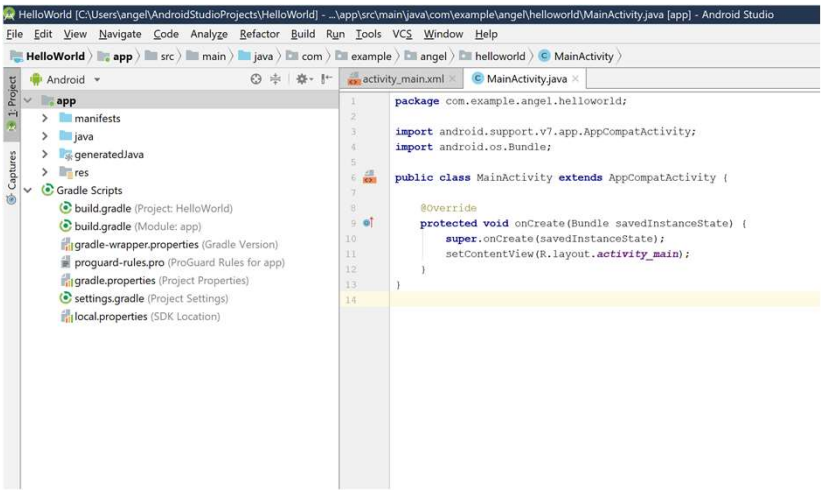
Android SDK (5/9)



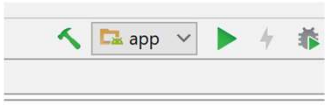
Android SDK (6/9)

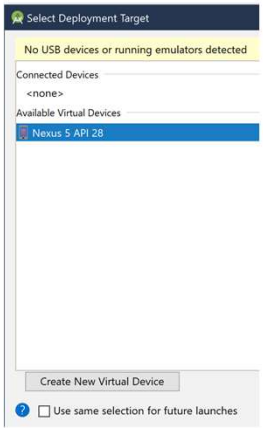


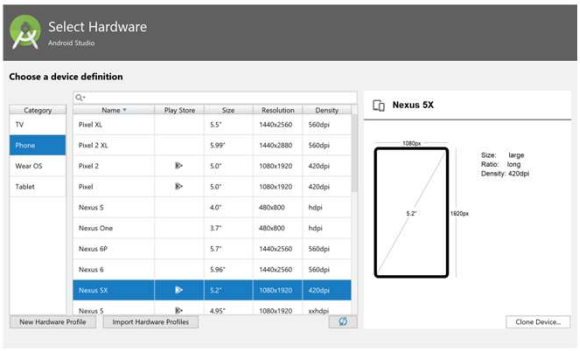
Android SDK (7/9)



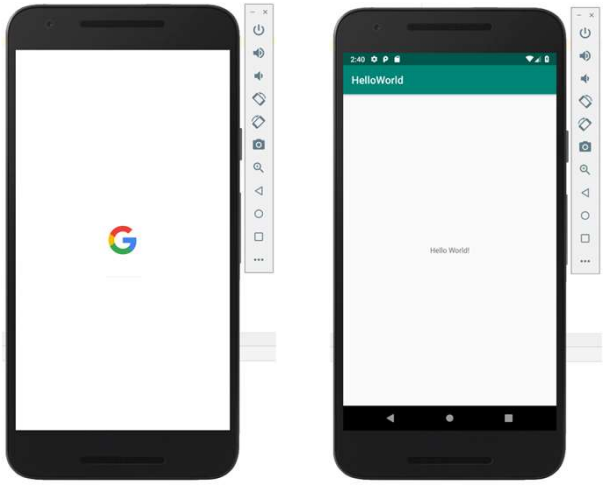
Android SDK (8/9)







Android SDK (9/9)



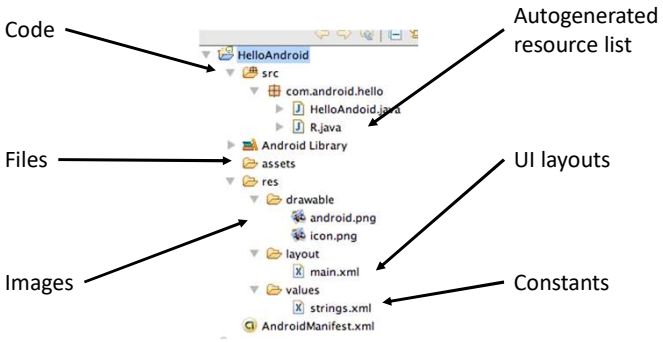
HelloWorld (1/5)

```
package com.example.angel.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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


HelloWorld (2/5)

- Applications can be run on the device or emulator

HelloWorld (3/5)



HelloWorld (4/5)

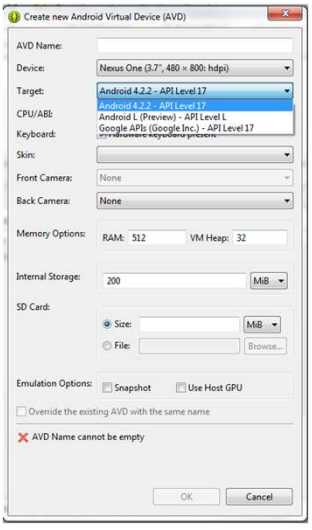
- Every application must provide a file named "AndroidManifest.xml"
- Contains the configuration information for correctly installing it
- Contains three things
  1. Class names
  2. Events
  3. Permissions

HelloWorld (5/5)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.my_domain.app.helloactivity">
  <application android:label="@string/app_name">
    <activity android:name=".HelloActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category
android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
```


APIs (1/14)

- When you create an android virtual machine there are two options for the target device (same API Level or Android version)
  - Android
  - Google APIs



APIs (2/14)

- Android
  - Basic functions included in all android devices (compatible with the API level)
- Google APIs
  - Adds to the basic functions, additional features implemented by Google
    - Google Maps
    - Google Drive
    - USB Open Accessory libraries
    - ...
- If you don't need the features provided by Google it is recommended to use basic Android

APIs (3/14)

- Android APIs
  - android
    - Contains resource classes used by applications included in the platform and defines application permissions for system features
  - android.accessibilityservice
    - Used for development of accessibility services that provide alternative or augmented feedback to the user
  - android.animation
    - Provides functionality for the property animation system, which allows you to animate object properties of any type
  - android.app
    - Contains high-level classes encapsulating the overall Android application model

APIs (4/14)

- Android APIs
  - android.app.admin
    - Provides device administration features for security-aware applications. Useful in enterprise settings, in which IT professionals require rich control over employee devices
  - android.app.backup
    - Contains the backup and restore functionality available to applications
  - android.appwidget
    - Contains the components necessary to create "app widgets", which users can embed in other applications (such as the home screen) to quickly access application data and services without launching a new activity

APIs (5/14)

- Android APIs
  - android.Bluetooth
    - Provides classes that manage Bluetooth functionality
  - android.content
    - Contains classes for accessing and publishing data on a device
  - android.content.pm
    - Contains classes for accessing information about an application package
  - android.content.res
    - Contains classes for accessing application resources as colors, media or other files in the package, plus important device configuration details (orientation, input types, etc.) that affect how the application behaves

APIs (6/14)

- Android APIs
  - android.database
    - Contains classes to explore data returned through a content provider
  - android.database.sqlite
    - Contains the SQLite database management classes that an application would use to manage its own private database
  - android.drm
    - Provides classes for managing DRM (digital rights management) content and determining the capabilities of DRM plugins
  - android.gesture
    - Provides classes to create, recognize, load and save gestures

APIs (7/14)

- Android APIs
  - android.graphics
    - Provides low level graphics tools such as canvases, color filters, points, and rectangles that let you handle drawing to the screen directly
  - android.graphics.drawable
    - Provides classes to manage a variety of visual elements that are intended for display only, such as bitmaps and gradients
  - android.graphics.drawable.shapes
    - Contains classes for drawing geometric shapes
  - android.graphics.pdf
    - Contains classes for manipulation of PDF content

APIs (8/14)

- Android APIs
  - android.hardware
    - Provides support for hardware features, such as the camera and other sensors
  - android.inputmethodservice
    - Base classes for writing input methods (such as software keyboards)
  - android.location
    - Contains the framework API classes that define Android location-based and related services
  - android.media
    - Provides classes that manage various media interfaces in audio and video

APIs (9/14)

- Android APIs
  - android.mtp
    - Provides APIs that let you interact directly with connected cameras and other devices
  - android.net
    - Classes that help with network access, beyond the normal java.net.* APIs (Wifi, RTP, SIP…)
  - android.nfc
    - Provides access to Near Field Communication (NFC) functionality, allowing applications to read NDEF message in NFC tags
  - android.opengl
    - Provides an OpenGL ES static interface and utilities

APIs (10/14)

- Android APIs
  - android.os
    - Provides basic operating system services, message passing, and inter-process communication on the device
  - android.print
    - Provides classes for implementing print support in applications
  - android.printservice
    - Provides classes for implementing print services
  - android.provider
    - Provides convenience classes to access the content providers supplied by Android

APIs (11/14)

- Android APIs
  - android.renderscript
    - Provides support for high-performance computation across heterogeneous processors
  - android.sax
    - A framework that makes it easy to write efficient and robust SAX handlers
  - android.security
    - Provides access to facilities of the Android security subsystems
  - android.support.*
    - Support classes to access some of the android.app package features introduced in old APIs in a backwards compatible fashion

APIs (12/14)

- Android APIs
  - android.telephony
    - Provides APIs for monitoring phone information, network type and connection state, plus utilities for manipulating phone number strings
  - android.telephony.cdma
    - Provides APIs for using CDMA-specific telephony features
  - android.telephony.gsm
    - Provides APIs for utilizing GSM-specific telephony features, such as text/data/PDU SMS messages
  - android.test
    - A framework for writing Android test cases and suites

APIs (13/14)

- Android APIs
  - android.text
    - Provides classes used to render or track text and text spans on the screen
  - android.transition
    - Enables "scenes & transitions" functionality for view hierarchies
  - android.util
    - Provides common utility methods such as date/time manipulation, base64 encoders and decoders, string and number conversion methods, and XML utilities
  - android.view
    - Provides classes that expose basic user interface classes that handle screen layout and interaction with the user

APIs (14/14)

- Android APIs
  - android.webkit
    - Provides tools for browsing the web
  - android.widget
    - Contains (mostly visual) UI elements to use on your Application screen