

# به نام خدا

## جلسه اول

اهداف آموزشی این جلسه عبارت اند از:

- ۱- معرفی سیستم عامل اندروید
- ۲- معرفی ویژگی های اندروید
- ۳- معرفی تاریخچه اندروید
- ۴- معرفی کنسرسیوم توسعه اندروید
- ۵- معرفی نسخه های مختلف اندروید
- ۶- آشنایی با معماری اندروید
- ۷- آشنایی با مدل امنیتی اندروید
- ۸- آشنایی با زبان برنامه نویسی اندروید
- ۹- آشنایی با محیط های برنامه نویسی اندروید
- ۱۰- آشنایی با محیط برنامه نویسی Eclipse
- ۱۱- آشنایی با نسخه ADT Bundle محیط برنامه نویسی Eclipse
- ۱۲- آشنایی با محیط برنامه نویسی اندروید استودیو شرکت گوگل
- ۱۳- آشنایی با محیط برنامه نویسی Beacon Mountain شرکت اینتل
- ۱۴- آشنایی با تفاوت های اندروید استودیو و اکلیپس
- ۱۵- آشنایی با حداقل امکانات نصب
- ۱۶- آشنایی با مراحل نصب

## اندروید چیست؟

اندروید در اصل اولین سیستم عامل جامع و کامل موبایل است که به صورت Open Source منتشر شده است ( به طور خلاصه Open Source به شرایطی گفته می شود که چیزی به صورت رایگان در دسترس همگان قرار گیرد که نه تنها بتوانند از آن استفاده کنند بلکه بتوانند بسته به نیازهای خود تغییراتی هم در آن ایجاد نمایند. به عبارت دیگر دسترسی به کدهای آن برنامه آزاد می باشد).

البته راهنمای توسعه دهندگان اندروید، این سیستم عامل را یک پشته نرم افزاری برای دستگاه های همراه معرفی کرده است که مجموعه ای از نرم افزارها و کامپوننت ها را شامل می شود که یک راه حل کاملاً عملی را ارائه می دهند. این پشته نرم افزاری شامل یک سیستم عامل ( نسخه ای تغییر یافته از لینوکس )، middleware (نرم افزاری که سیستم عامل سطح پایین را به اپلیکیشن های سطح بالا متصل می سازد) که بیشتر مبتنی بر جاوا است، اپلیکیشن های کلید (key apps) که با جاوا نوشته می شوند. برای مثال یک مرورگر و یک Contact manager یا مدیر مخاطب که به اصطلاح Contacts یا همان مخاطبان شناخته می شود) تشکیل شده است.

## ویژگی های اندروید

- ◆ یک فریم ورک (فریم ورک در واقع مجموعه منسجم از کلاس ها و توابع از پیش تعریف شده است که قابلیت های بلقوه گوناگونی از یک زبان برنامه نویسی را در خود دارد و بدین ترتیب کاربر نهایی را قادر می سازد که از امکانات یک زبان استفاده کند بدون اینکه درگیر مسائل پیچیده و وقت گیر آن شود و همچنین فریم ورک مواردی مانند توسعه پذیری و ساده سازی را نیز مد نظر دارد) با امکانات استفاده مجدد و بکارگیری کامپوننت های اپلیکیشن.
- ◆ پشتیبانی از فناوری های 3G، 4G، EDGE و بلوتوث (متکی به سخت افزار مورد نیاز).
- ◆ پشتیبانی از دوربین، قطب نما، GPS و شتاب سنج (متکی به سخت افزار مورد نیاز).
- ◆ ماشین مجازی Dalvik که برای دستگاه های همراه بهینه شده است.
- ◆ پشتیبانی از GSM (متکی به سخت افزار مورد نیاز).
- ◆ مرورگر یکپارچه، مبتنی بر موتور Web Kit.
- ◆ پشتیبانی از فایل های چند رسانه ای معمول، صدا، ویدئو و عکس (MP3، H.264، AAC، AMR، MPEG4، GIF، PNG، JPG).
- ◆ گرافیک بهینه شده به کمک یک کتابخانه گرافیکی دو بعدی.
- ◆ گرافیک سه بعدی بر اساس معیارهای OpenGL ES 1.0، 1.1، 2.0 (سخت افزار شتاب سنج اختیاری است).
- ◆ پایگاه داده SQLite برای ذخیره ساخت یافته داده ها.

همچنین می توان محیط توسعه اندروید (متشکل از یک شبیه ساز دستگاه همراه و یک افزونه برای محیط Eclipse) را نیز به عنوان یکی از ویژگی های اندروید در نظر گرفت، هر چند این محیط جزو پشته نرم افزاری اندروید نیست.

## تاریخچه اندروید

بر خلاف آنچه که به طور احتمال تصور می کنید، شکل گیری اندروید از گوگل نبوده است؛ بلکه آغاز به کار آن توسط آقای اندی روبین مدیر عامل شرکت پالو آلتو و همکارانش در کالیفرنیا صورت گرفت که پس از چندی در تابستان سال ۲۰۰۵ کمپانی گوگل این شرکت را خریداری نمود.

در نوامبر سال ۲۰۰۷ گروهی از تولید کنندگان مطرح تجهیزات الکترونیکی همچون موبایل و تجهیزات بی سیم به علاوه یکسری از شرکت های نرم افزاری که از آن جمله می توان به گوگل اشاره کرد که در ابتدا ۳۴ شرکت بودند و سال به سال به تعداد اعضای آن افزوده شد، در کنار یکدیگر جمع شده و کنسرسیومی را تشکیل دادند که هدف آن ایجاد یک سیستم عامل موبایل Open Source بود تا از این طریق بتوانند هزینه های تولید اپلیکیشن های موبایل را کاهش دهند. در حقیقت از آنجا که شرکت گوگل در این کنسرسیوم رویکردی نرم افزاری داشت و از سوی دیگر صاحب اصلی این سستم عامل هم محسوب می شد، به عنوان رهبر اصلی این سیستم عامل جدید انتخاب گردید.

برای این منظور این کنسرسیوم که Open Handset Alliance نام داشت چیزی تحت عنوان SDK که در برگیرنده ابزارهای نرم افزاری مورد نیاز برای توسعه اپلیکیشن های اندروید بود را منتشر کرد (در واقع SDK مخفف واژگان Software Development Kit به معنی پکیج توسعه نرم افزار می باشد). اولین نسخه بتای Android SDK را در نوامبر ۲۰۰۷ (پاییز ۲۰۰۷) روانه بازار شد و در اکتبر سال ۲۰۰۸ اولین گوشی موبایل با سیستم عامل اندروید که T-Mobile G1 از شرکت HTC نام داشت روانه بازار شد. از آن سال به بعد شرکت های سازنده تلفن همراه یکی پس از دیگری به این سیستم عامل روی آوردند و نه تنها تلفن های همراهی مجهز به این سیستم عامل طراحی کردند، بلکه ابزارهای الکترونیکی دیگری همچون تبلت، ساعت، کتاب خوان الکترونیکی و حتی تلوزیون هم با این سیستم عامل طراحی کردند.

## معرفی نسخه های مختلف اندروید

نکته ای که در مورد نسخه های اندروید تا حدودی عجیب ولی در عین حال جالب توجه است این می باشد که نسخه های مختلف بر اساس نام دسرهای میان وعده نام گذاری شده اند. نسخه اول سیستم عامل اندروید که ۱ بود نام خاصی نداشت و بیشتر کارایی اینترنتی داشت و در عمل توان رقابت با هیچ سیستم عاملی را نداشت. سپس نسخه ۱,۱ طراحی شد که مجدداً بدون نام انتشار یافت. نسخه ۱,۵ که Cupcake به معنی کیک فنجانی بود به منزله اولین نسخه دارای نام سیستم عامل اندروید است (برای درک بهتر معنی این واژه می توان کیک یزدی را متصور شد). نسخه ۱,۶ که Donut به معنی پیراشکی نام داشت پس از نسخه ۱,۵ انتشار

یافت. نسخه بعدی که ۲,۰-۲,۱ بود Eclair یا نان خامه ای نام داشت. نسخه بعد ۲,۲ بود و تحت عنوان Froyo به معنی ماست یخی روانه بازار شد. سپس نسخه ۲,۳ تحت عنوان Gingerbread به معنی نام زنجبیلی معرفی شد.

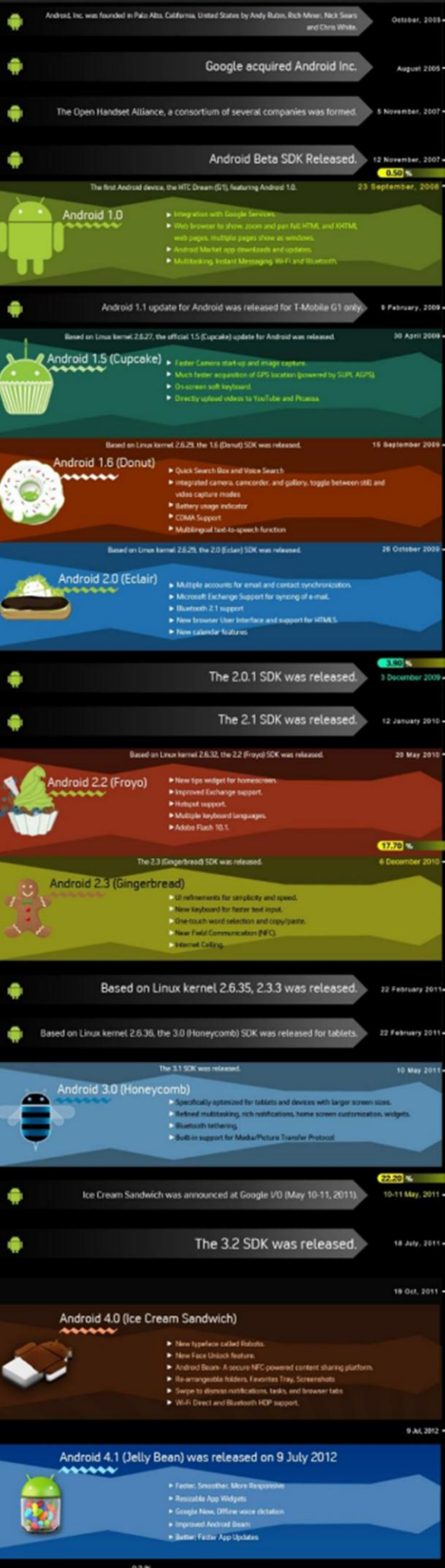
قدم بعدی اندروید بازار تبلت ها بود که برای همین منظور نسخه ۳,۰ تحت عنوان Honeycomb به معنی شانه عسل را معرفی کرد. سپس نسخه ۴,۰ که Ice cream Sandwich یا بستنی حصیری نام داشت تلفیقی از ویژگی های اندروید ۳,۰ که برای تبلت ها طراحی شده بود به علاوه یکسری ویژگی های جدید بود که این نسخه برای کلیه تجهیزات اندرویدی از موبایل گرفته تا تبلت مورد استفاده قرار می گیرد. نسخه بعدی اندروید ۴,۲ بود که Jelly Bean به معنی آب نبات پاستیلی معروف شد سپس بخاطر اینکه رقبای اندروید مانند ios بیکار ننشسته بودند، شرکت گوگل برای اینکه سهم خود را در دنیای فناوری حفظ کند در فاصله بسیار کوتاهی دو بروزرسانی انجام داد و اندروید ۴,۳ در آگوست ۲۰۱۳ و اندروید ۴,۴ را در نوامبر همان سال با نام تجاری کیت کت (Kit Kat) روانه بازار کرد. همچنین نسخه های ۴,۴,۱ و ۴,۴,۲ نیز انتشار یافتند و هم اکنون نسخه ۵ اندروید روانه بازار شده است.

نکته دیگری که در مورد نحوه نام گذاری نسخه های مختلف اندروید می بایست مد نظر قرار داده شود این است که نام دسرهای انتخابی بر اساس حروف الفبا پیش می روند. به طور مثال همانطور که در الفبای انگلیسی حرف D قبل از حرف E قرار می گیرد، نام انتخابی برای نسخه ۱,۵ معادل با Donut بود و پس از آن در نسخه ۱,۶ نام Eclair انتخاب شد (بر اساس حروف الفبا کلمه Donut پیش از کلمه Eclair قرار می گیرد).

جهت آشنایی با تاریخچه و همچنین قابلیت های هر یک از نسخ اندروید به نمودار زیر توجه نمایید:

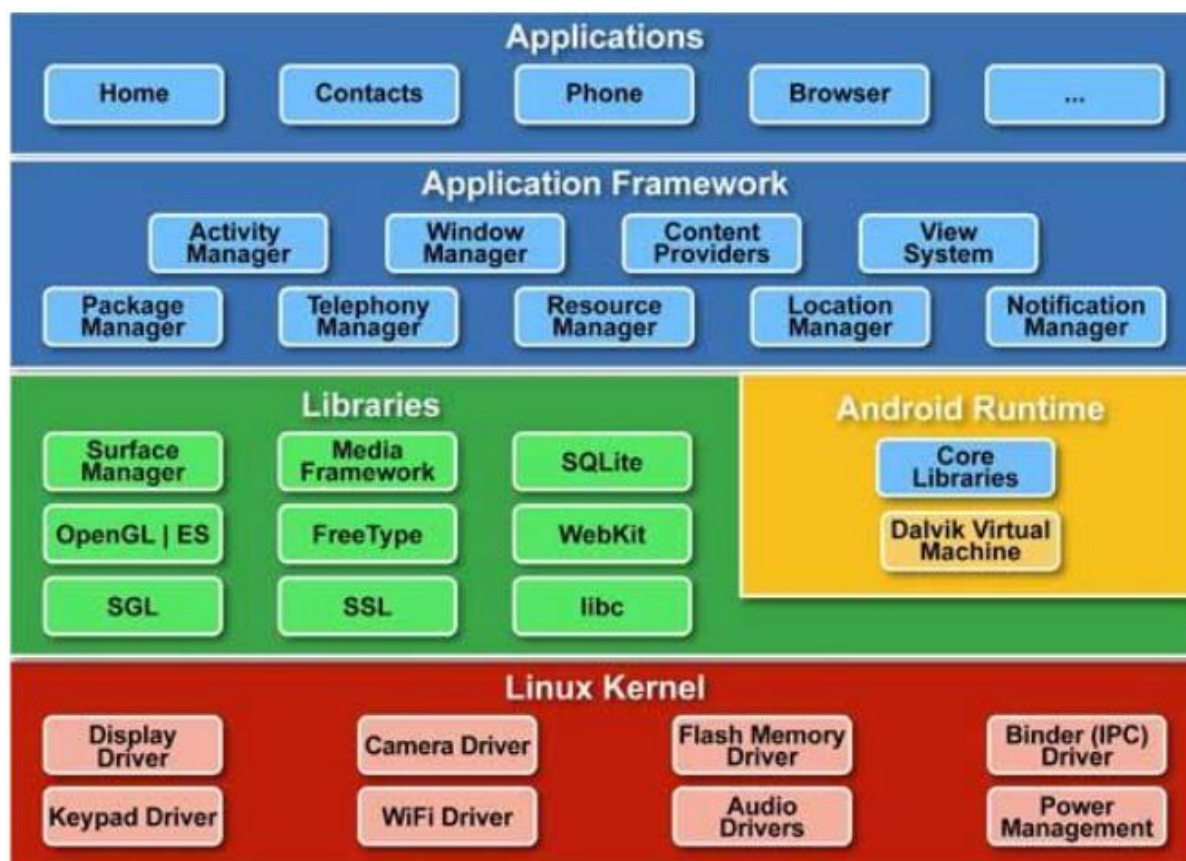
# The Android Story

Worldwide Android Market Share



## معماری اندروید

در بالاترین قسمت پشته نرم افزاری اندروید، اپلیکیشن ها قرار دارند، در قسمت میانی آن middleware (که شامل فریم ورک اپلیکیشن، کتابخانه ها و runtime اندروید است) و در نهایت هسته لینوکس با تعدادی درایور در انتها. شکل ۱-۱ این معماری لایه ای را نمایش می دهد.



شکل ۱-۱ معماری لایه ای اندروید که از چندین بخش تشکیل می شود.

بخش در خور اهمیت برای کاربران، اپلیکیشن هاست. از همین رو اندروید تعدادی اپلیکیشن پرکاربرد، مانند Browser، Contacts و Phone را برای استفاده آماده کرده است که همه آنها با زبان برنامه نویسی جاوا ایجاد شده اند. در معماری اندروید، اپلیکیشن ها لایه بالایی را تشکیل می دهند.

### نکته:

اپلیکیشن ها با استفاده از یک پیاده سازی غیر استاندارد جاوا نوشته می شوند که ترکیبی از تعاریف API های اندروید، API های جاوا ۵ و قسمتی کوچک از جاوا ۶ (برای مثال `java.io.File.setExecutable(Boolean executable, Boolean ownerOnly)`) هستند. از آنجا که اندروید بخش اعظمی از جاوا ۶ و کل جاوا ۷ را پشتیبانی نمی کند، نمی توان از API های

جدید بهره برد (برای نمونه java.lang.AutoCloseable از جاوا ۷. البته با معرفی نسخه های به روزتر اندروید، API های جدیدی نیز قابل استفاده می گردند).

هر نسخه از اندروید، عددی به همراه دارد که فرم ورک API های قابل استفاده از آن نسخه را نشان می دهد. به عنوان نمونه، این عدد که "سطح API" (API level) نامیده می شود، برای اندروید نسخه ۴,۱ سطح ۱۶ را نشان می دهد، در حالی که این عدد برای نسخه ۲,۳,۴ مقدار ۱۰ است. بدیهی است که API ها با سطح بالاتر، روی دستگاه هایی که از سطوح پایین تری بهره می برند، قابل اجرا نیست. برای مثال، نمی توان از یک API سطح ۱۶ در دستگاهی بهره برد که از سطح API ۱۰ پشتیبانی می کند. سطوح مختلف API را می توان در کلاس android.os.Build.VERSION\_CODES مشاهده کرد. همچنین با مراجعه به آدرس <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels> می توان اطلاعات بیشتری در خصوص سطوح API و سطح مربوط به هر نسخه را بدست آورد.

پس از لایه اپلیکیشن، فریم ورک اپلیکیشن قرار دارد که مجموعه ای از ابزارهای سطح بالا برای ساخت و ایجاد اپلیکیشن است. فریم ورک اپلیکیشن از پیش روی دستگاه های اندروید نصب شده است و شامل کامپوننت های زیر می باشد:

- Activity Manager: این کامپوننت چرخه حیات اپلیکیشن را فراهم می سازد و افزون بر آن، با استفاده از یک پشته، رفت و بازگشت به اپلیکیشن را مدیریت می کند.
- Content Provider: کار این کامپوننت، کپسوله کردن داده هایی است که قابل اشتراک گذاری میان اپلیکیشن ها هستند.
- Location Manager: با استفاده از این کامپوننت، دستگاه اندروید می تواند موقعیت و مکان فیزیکی خود را تشخیص دهد.
- Notification Manager: این کامپوننت، این امکان را فراهم می سازد تا با صدور اطلاعیه ای کاربر را از وقوع رویدادی خاص (برای مثال دریافت پیام جدید) آگاه کند، البته بی آنکه کار اصلی وی مختل شود.
- Package Manager: با استفاده از این کامپوننت، اپلیکیشن می تواند در باره سایر بسته های اپلیکیشن که هم اکنون روی دستگاه نصب شده اند، آگاه شود.
- Resource Manager: هر اپلیکیشن با به کارگیری این کامپوننت، می تواند به منابع خود دسترسی داشته باشد.
- Telephony Manager: این کامپوننت اپلیکیشن را قادر می سازد تا از خدمات تلفنی دستگاه مطلع شود، همچنین تماس های دریافتی یا خروجی نیز توسط همین کامپوننت کنترل می شود.
- View System: این کامپوننت مسئول مدیریت عناصر موجود در واسط گرافیکی و رویدادهایی است که در آن رخ می دهد.
- Window Manager: این کامپوننت وضعیت واقعی صفحه را نسبت به پنجره سازمان دهی می کند و از این طریق می توان کارهای مربوط به پنجره را انجام داد.

کامپوننت های فریم ورک اپلیکیشن، برای اجرا نیاز به مجموعه ای از کتابخانه های C/C++ دارند. از همین رو به کتابخانه های زیر نیاز است:

- FreeType: این کتابخانه از فونت های vector و bitmap پشتیبانی می کند.
- Libe: این کتابخانه یک پیاده سازی از کتابخانه استاندارد سیستمی C است که برای کارکرد روی دستگاه مبتنی بر لینوکس تغییر یافته است.
- LibWebCore: این کتابخانه یک موتور مرورگر مدرن و سریع ارائه می دهد که توانایی مرورگر در اندروید را بهبود می بخشد. اساس آن WebKit است و از مرورگرهای google chrome و Apple Safari استفاده می کند.
- MediaFramwork: این کتابخانه های مبتنی بر OpenCORE مربوط به PacketVideo هستند که امکان ضبط، نمایش و کار با بسیاری از فرمت های شناخته شده فیلم، عکس و صدها از قبیل AAC، MP3، H.264، MPEG4، PNG، JPEG، AMR را فراهم می سازند.
- OpenGL|ES: این کتابخانه های سه بعدی یک پیاده سازی از OpenGL بر اساس OpenGL ES 1.0/1.1/2.0 ارائه می دهند. این کتابخانه از سخت افزار سه بعدی شتاب سنج (در صورت وجود) یا نرم افزار سه بعدی rasterizer (که بسیار بهینه شده است) استفاده می کند.
- SGL: موتور زیربنایی گرافیک دوبعدی در این کتابخانه قرار دارد.
- SQLite: این کتابخانه یک موتور بانک اطلاعاتی رابطه ای بسیار سبک و در عین حال بسیار قدرتمند ارائه می دهد. این موتور برای همه ی اپلیکیشن ها قابل دسترسی است و Mozilla Firefox و گوشی iphone نیز از آن برای ذخیره دایم اطلاعات استفاده می کنند.
- SSL: (Secure Sockets Layer) این کتابخانه یک اتصال شبکه ای ایمن را ارائه می دهد.
- SurfaceManager: از طریق این کتابخانه می توان به زیر سیستم نمایش دسترسی داشت، همچنین این کتابخانه به سادگی لایه های دو بعدی و سه بعدی اپلیکیشن های مختلف را با یکدیگر ترکیب می کند.

اندروید محیطی برای زمان اجرا فراهم کرده است که از کتابخانه های هسته ای (زیرمجموعه ای از پیاده سازی Apache Harmony Java 5) و ماشین مجازی Dalvik (یک ماشین مجازی فارغ از جاوا که مبتنی بر ثبات های پردازنده است و از پشته استفاده نمی کند) تشکیل می شود.

#### نکته:

ماشین مجازی Dalvik توسط فردی به نام Dan Bornstein ساخته شده است و وی آن را Dalvik (نام روستایی که اجداد او در آن جا زندگی می کردند)، نام گذاری کرده است.



هر اپلیکیشن اندروید، به طور پیش فرض روی پردازنده لینوکس خود که نمونه ای از Dalvik در آن قرار دارد اجرا می شود. این ماشین مجازی به گونه ای طراحی شده است که چندین دستگاه بتوانند به طور همزمان و البته به بهینه ترین حالت، از چند ماشین استفاده کنند. این بهینه بودن مدیون فایل اجرایی Dalvik یا همان DEX است. DEX فرمتی است که کمترین جای پا را در حافظه باقی می گذارد.

#### نکته:

در اندروید، پردازش یک اپلیکیشن زمانی آغاز می شود که یکی از بخش های آن اپلیکیشن نیاز به اجرا داشته باشد و زمانی به پایان می رسد که دیگر به آن نیازی نباشد و نیز منابع سیستم توسط اپلیکیشنی دیگر مورد نیاز باشند.

شاید کمی عجیب به نظر آید که چگونه یک ماشین فارغ از جاوا می تواند کدهای جاوا را اجرا کند. پاسخ اینجاست که Dalvik کدهای جاوا را اجرا نمی کند. در عوض اندروید با استفاده از ابزار dx خود، کدهای کامپایل شده جاوا را به فرمت DEX تبدیل کرده و این کد توسط Dalvik اجرا می شود.

در نهایت کلاس ها و زمان اجرای اندروید وابسته به هسته لینوکس هستند (نسخه های 2.6.x و 3.0.x)؛ کارهایی از قبیل خدمات پایه ای، ریسمان دهی، مدیریت سطوح پایین حافظه، پشته های شبکه، مدیریت پردازش و مدل درایور. افزون بر آن هسته مانند یک لایه انتزاعی میان سخت افزار و مابقی پشته نرم افزاری عمل می کند.

#### مدل امنیتی اندروید

معماری اندروید شامل یک مدل امنیتی است که در آن از اجرای عملیاتی که ممکن است برای سایر کاربران، لینوکس یا سایر اپلیکیشن ها مضر باشد، جلوگیری می کند. این مدل که بیشتر از ویژگی های لینوکس استاندارد استفاده می کند، پروسه ها را در یک sandbox ایمن نگه می دارد.

به طور پیش فرض sandbox ها مانع از نوشتن و خواندن اپلیکیشن از سایر اپلیکیشن ها می شوند، همچنین دسترسی به شبکه، دوربین و غیره را مانع می شوند. بنابراین اپلیکیشن هایی که نیاز به کار با این منابع را دارند، باید نخست مجوز لازم را بدست آورند.

حصول مجوز در اندروید از راه های گوناگونی امکان پذیر است که معمول ترین آنها به صورت خودکار و با استفاده از تاییدیه ها صورت می پذیرد. مجوزهای مورد نیاز هر اپلیکیشن، در فایل manifest آن معرفی می شود.

## زبان برنامه نویسی سیستم عامل اندروید

زبان برنامه نویسی اصلی این سیستم عامل زبان جاوا است، اما این بدان معنا نیست که برنامه نویسانی با تسلط به دیگر زبان های برنامه نویسی نمی توانند برای این سیستم عامل اپلیکیشن طراحی کنند. در حقیقت کسانی که به HTML5 و JavaScript تسلط دارند خواهند توانست برای این سیستم عامل اپلیکیشن های تحت وب طراحی کنند. از سوی دیگر برنامه نویسان سی و همچنین سی پلاس پلاس می توانند با استفاده از NDK که مخفف Native Development Kit است اقدام به توسعه اپلیکیشن های اندرویدی با استفاده از زبان های فوق الذکر نمایند.

تسلط به زبان برنامه نویسی جاوا برای کسب نتیجه بهتر در این سری از آموزش های اندروید یک امر ضروری محسوب می شود و آشنایی با XML یک مزیت است که برای توسعه اپلیکیشن اندروید در این سری از آموزش ها مورد بحث قرار خواهد گرفت.

## چرا جاوا؟

هر توسعه دهنده اندروید باید از گوگل این سوال را بپرسد که چرا با اینکه اندروید از زبان های محلی یا Native پشتیبانی می کند، زبان اصلی خود را جاوا انتخاب کرده است؟

برای پاسخ به این سوال چند مورد ذکر شده است که در زیر بیان می کنیم:

- اندروید بر روی سخت افزارهای مختلفی اجرا می شود و شما نیازی ندارید که کد محلی استفاده شده در اپلیکیشن را برای بهره گیری مناسب دوباره کامپایل کنید که جاوا نیز این خصوصیت را دارا می باشد.
- تعداد زیادی از توسعه دهندگان حرفه ای نرم افزار از جاوا استفاده می کنند.
- جاوا Open Source است و کتابخانه ها و ابزار زیادی را برای راحت تر کردن چرخه توسعه نرم افزار در اختیار می گذارد.
- جاوا از خیلی از مشکلات ذاتی کد های محلی (مانند اشاره گر های که خیلی نامناسب هستند یا مدیریت نامناسب حافظه) اجتناب کرده است.
- از همه مهمتر این است که جاوا اپلیکیشن ها را در یک جعبه محافظت شده (SandBox) قرار می دهد و یک مدل امنیتی بهتری را نسبت به زبان های دیگر ایجاد می کند که باعث می شود اپلیکیشن های مخرب نتوانند سیستم عامل را از بین ببرند.

البته دلایل دیگری نیز وجود دارد که در اینجا بر اهمیت ترین آنها بیان شد. (لازم به ذکر است در توسعه این سیستم عامل از دیگر زبان های برنامه نویسی مثل زبان سی و سی پلاس پلاس نیز تا حدودی استفاده شده است).

## محیط های برنامه نویسی اندروید

ابتدا باید بدانیم محیط توسعه مجتمع چیست؟ محیط توسعه مجتمع (Integrated Development Environment) IDE یک محیط می باشد که در آن تمامی ابزار ها و امکانات برای توسعه نرم افزار به صورت مجتمع و یکپارچه قرار دارد که به برنامه نویس این امکان را می دهد که چرخه تولید نرم افزار را به راحتی پیاده سازی کند. به عبارت دیگر در این محیط برنامه نویس می تواند طراحی، کد نویسی، کامپایل، آزمایش، خطایابی و پکیج کردن نرم افزار را بدون وقفه و به ترتیب انجام دهد.

شرکت گوگل برای توسعه اپلیکیشن های اندرویدی از دو IDE به زبان جاوا پشتیبانی می کند.

1. Android Developer Tools (ADT) – <http://developer.android.com/sdk/index.html>
2. Android Studio – <http://developer.android.com/sdk/installing/studio.html>

برای توسعه اپلیکیشن های اندرویدی IDE های مختلفی حتی به زبان های برنامه نویسی غیر از جاوا وجود دارند که در لیست زیر به آنها اشاره شده است. اما در این آموزش توضیح مختصری در رابطه با سه IDE که از دیگر IDE ها پر کاربرد ترند پرداخته می شود.

نام IDE	زبان برنامه نویسی	آدرس دانلود
AIDE (Android IDE)	HTML5/C/C++	<a href="http://www.android-ide.com">http://www.android-ide.com</a>
Application Craft	HTML5	<a href="http://www.applicationcraft.com">http://www.applicationcraft.com</a>
Basic4Android	BASIC	<a href="http://www.basic4ppc.com">http://www.basic4ppc.com</a>
Cordova	HTML5	<a href="https://cordova.apache.org">https://cordova.apache.org</a>
Corona	Lua	<a href="http://coronalabs.com">http://coronalabs.com</a>
Intel XDK	HTML5	<a href="https://software.intel.com/en-us/html5/tools">https://software.intel.com/en-us/html5/tools</a>
IntelliJ IDEA	Java	<a href="https://www.jetbrains.com/idea/features/android.html">https://www.jetbrains.com/idea/features/android.html</a>
Kivy	Python	<a href="http://kivy.org/#home">http://kivy.org/#home</a>
MIT AppInventor	Blocks	<a href="http://appinventor.mit.edu/explore">http://appinventor.mit.edu/explore</a>
Monkey X	BASIC	<a href="http://www.monkeycoder.co.nz">http://www.monkeycoder.co.nz</a>
MonoGame	C#	<a href="http://www.monogame.net">http://www.monogame.net</a>
MoSync	HTML5/C/C++	<a href="http://www.mosync.com">http://www.mosync.com</a>
NS BASIC	BASIC	<a href="https://www.nsbasic.com">https://www.nsbasic.com</a>
PhoneGap	HTML5	<a href="http://phonegap.com">http://phonegap.com</a>
RAD Studio XE	Object Pascal, C++	<a href="http://www.embarcadero.com">http://www.embarcadero.com</a>
RFO Basic	BASIC	<a href="http://laughton.com/basic">http://laughton.com/basic</a>
RhoMobile Suite	Ruby	<a href="http://www.motorolasolutions.com/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite">http://www.motorolasolutions.com/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite</a>
Telerik	HTML5	<a href="http://www.telerik.com/platform#overview">http://www.telerik.com/platform#overview</a>
Titanium	JavaScript	<a href="http://www.appcelerator.com/titanium/titanium-sdk">http://www.appcelerator.com/titanium/titanium-sdk</a>
Xamarin	C#	<a href="http://xamarin.com">http://xamarin.com</a>

## محیط برنامه نویسی اکلیپس

به منظور شروع برنامه نویسی برای اندروید چند رویکرد متفاوت می توان اتخاذ کرد که در این قسمت به طور خلاصه این محیط های برنامه نویسی را معرفی می کنیم.

معروف ترین محیط برنامه نویسی برای توسعه اندروید محیط برنامه نویسی اکلیپس به همراه ابزارهای SDK و ADT است. در واقع از آنجا که محیط برنامه نویسی اکلیپس Open Source است شرکت گوگل تمایل به مراتب بیشتری برای فراهم آوردن ابزارهای توسعه اندروید برای این محیط برنامه نویسی قدرتمند از خود نشان داده است ( نحوه دانلود، نصب و بکارگیری این محیط برنامه نویسی در این دوره آموزش به طور مفصل گفته خواهد شد).

## نسخه ADT Bundle محیط برنامه نویسی اکلیپس

این محیط برنامه نویسی دقیقاً همان محیط برنامه نویسی اکلیپس است با این تفاوت که خود شرکت گوگل تغییراتی در آن صورت داده و آن را منتشر کرده است. این محیط برنامه نویسی برخلاف اکلیپس که برای توسعه اندروید می بایست تنظیمات خاصی را در آن انجام داد، در برگیرنده کلیه ابزارهای لازم برای توسعه اندروید از جمله SDK و ADT و Emulator بوده و کاربران دیگر نیازی به دانلود کردن این ابزارها به صورت مجزا نخواهند داشت. لازم به ذکر است توسعه دهندگانی که با نحوه کار با محیط برنامه نویسی اکلیپس آشنایی داشته باشند به راحتی خواهند توانست از این محیط هم در تولید اپلیکیشن اندروید استفاده کنند (نحوه دانلود و نصب این محیط برنامه نویسی در این دوره آموزش توضیح داده خواهد شد).

## محیط برنامه نویسی اندروید استودیو

اندروید استودیو (Android Studio) محیطی برای توسعه برنامه های اندرویدی است که برای اولین بار خود شرکت گوگل بر پایه محیط برنامه نویسی قدرتمند IntelliJ IDEA طراحی کرده است که همانند اکلیپس ابزارهای لازم را برای طراحی، توسعه و مشکل یابی برنامه ها را داراست. اولین نسخه از اندروید استودیو در کنفرانس سالیانه گوگل که در اوردیبهشت ماه سال ۱۳۹۲ اتفاق افتاد تحت عنوان نسخه ۱ معرفی شد (نحوه دانلود، نصب و کار با این محیط برنامه نویسی در این دوره آموزش به طور مفصل توضیح داده خواهد شد).

## محیط برنامه نویسی Beacon Mountain

شرکت معروف اینتل یک محیط برنامه نویسی جدید تحت عنوان Beacon Mountain ساخته است که ادعا می کند این ابزار جدید این امکان را به توسعه دهندگان اندروید می دهد تا بتوانند با صرف زمان کمتر و نوشتن کد کمتر دست به توسعه اپلیکیشن های اندرویدی بزنند ( از آنجا که این محیط برنامه نویسی تازه به بازار معرفی شده است و این احتمال می رود که دارای یکسری Bug باشد، از توضیح پیرامون نحوه کار با این نرم افزار خودداری می کنیم اما علاقه مندان به تست کردن این برنامه می توانند با مراجعه به آدرس <http://software.intel.com/en-us/vcsource/tools/beaconmountain> این نرم افزار را به صورت رایگان دانلود نمایند).

## تفاوت محیط های برنامه نویسی اکلیپس و اندروید استودیو

### ۱- Build Tools (ابزارهای کامپایل و تولید فایل APK):

در اکلیپس این ابزار بر پایه سازنده های قدرتمند XML طراحی شده اند که بسیاری از توسعه دهندگان جاوا از آن استفاده می کنند. اما در اندروید استودیو این ابزار بر پایه سازنده فایل Gradle طراحی شده اند که یک زبان مخصوصی را ارائه می دهد که به صورت اسکریپتی می باشد و توانایی بارگذاری خودکار فایل بتای APK را برای آزمایش فراهم می کند.

### ۲- کامپایل و دوباره سازی حرفه ای کد ها Advanced Code Completion / Refactoring:

خصوصیه هر دو IDE کامپایل اتوماتیک کد جاوا می باشد اما اندروید استودیو برای بهبود و افزایش کارایی آن اقداماتی انجام داده تا کد مخصوص اندروید را پشتیبانی کند و همچنین می تواند کد تولیدی را دوباره سازی کند در صورتی که برای اکلیپس چنین امکانی وجود ندارد.

### ۳- طراحی رابط کاربری

طراحی رابط کاربری در اندروید استودیو سریعتر صورت می گیرد و همچنین پاسخ به تغییرات و نمایش آنها سریعتر است و قابلیت شخصی سازی زیادتری نسبت به اکلیپس دارد و اما شما می توانید مانند اکلیپس به صورت دستی تنظیمات خود را در فایل XML اعمال کنید.

### ۴- سازماندهی پروژه

هر دو محیط به صورت کاملاً متفاوت پروژه ها را مدیریت می کنند. اگر شما از اکلیپس استفاده کنید تمامی پروژه ها در یک محیط کاری قرار دارند و برای تغییر محیط کاری باید اکلیپس دوباره راه اندازی شود. اما اندروید استودیو این عمل را به صورت طبیعی تر و با مدل دیگری انجام می دهد.

## ۵- نمایش و پایداری محیط توسعه

اکلیپس با اینکه چندین سال از طراحی آن می گذر اما هنوز از منابع سیستم مانند RAM و CPU بسیار بیشتر استفاده می کند و همچنین خطاهایی دارد (مانند خطایی که بعضی اوقات در تولید فایل نهایی APK رخ می دهد) که باعث بسته شدن این محیط می شود، اما اندروید استودیو تازه وارد بازار شده است و انتظار چنین باگهایی از آن می رود و همچنین از منابع سیستم کمتر استفاده می کند.

### حد اقل امکانات نصب (نیازمندی های نصب):

برای شروع به برنامه نویسی اندروید باید سیستم عامل رایانه شما یکی از موارد زیر باشد:

۱- Microsoft Windows XP یا نسخه بالاتر

۲- Mac OS X 10.5.8 یا نسخه بالاتر با چیپ اینتل

۳- Linux با کتابخانه GNU C 2.7 یا نسخه بالاتر

همچنین باید ابزارهای مورد نیاز برنامه نویسی اندروید که به صورت رایگان وجود دارند را از اینترنت دانلود کرده و نصب کنید.

در زیر لیست نرم افزار ها و ابزار های مورد نیاز برای شروع برنامه نویسی اندروید قرار دارد:

۱- Java JDK

۲- Android SDK

۳- Eclipse IDE for Java Developers

۴- Android Development Tools (ADT) Eclipse Plugin

### مراحل نصب

مراحل نصب باید به ترتیب زیر صورت گیرد

### مرحله اول نصب جاوا (Java JDK):

به منظور توسعه اندروید حتما به JRE و JDK نیاز است که می توان آخرین نسخه آن ها را از لینک زیر دریافت کرد.

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

لازم به ذکر است چنانچه شما نسخه JDK را نصب نمایید دیگر نیازی به نصب JRE نخواهید داشت چرا که JDK در بر گیرنده JRE نیز می باشد.

از آنجا که دسترسی به سایت اوراکل برای کاربران ایرانی امکان پذیر نمی باشد، پیشنهاد می شود علاقمندان JDK جاوا را از دیگر منابع دانلود نمایند. برای مثال از سایت <http://www.uncocoder.com> می توان پس از ورود به بخش مطالعه آموزش های رایگان، سپس کلیک کردن بر روی گزینه آماد کردن محیط برنامه نویسی به لینک دانلود دست پیدا کرد.

نصب JDK بسیار آسان می باشد و نیازی به توضیح ندارد. اما بعد از نصب نیاز به تنظیماتی داریم تا بتوانیم از جاوا استفاده کنیم.

در ویندوز برای تنظیمات جاوا باید در یک فایل با پسوند bat دستورات زیر را نوشته و ذخیره کنیم و در آخر آن را اجرا کنیم.

```
set PATH=C:\jdk1.6.0_15\bin;%PATH%
```

```
set JAVA_HOME=C:\jdk1.6.0_15
```

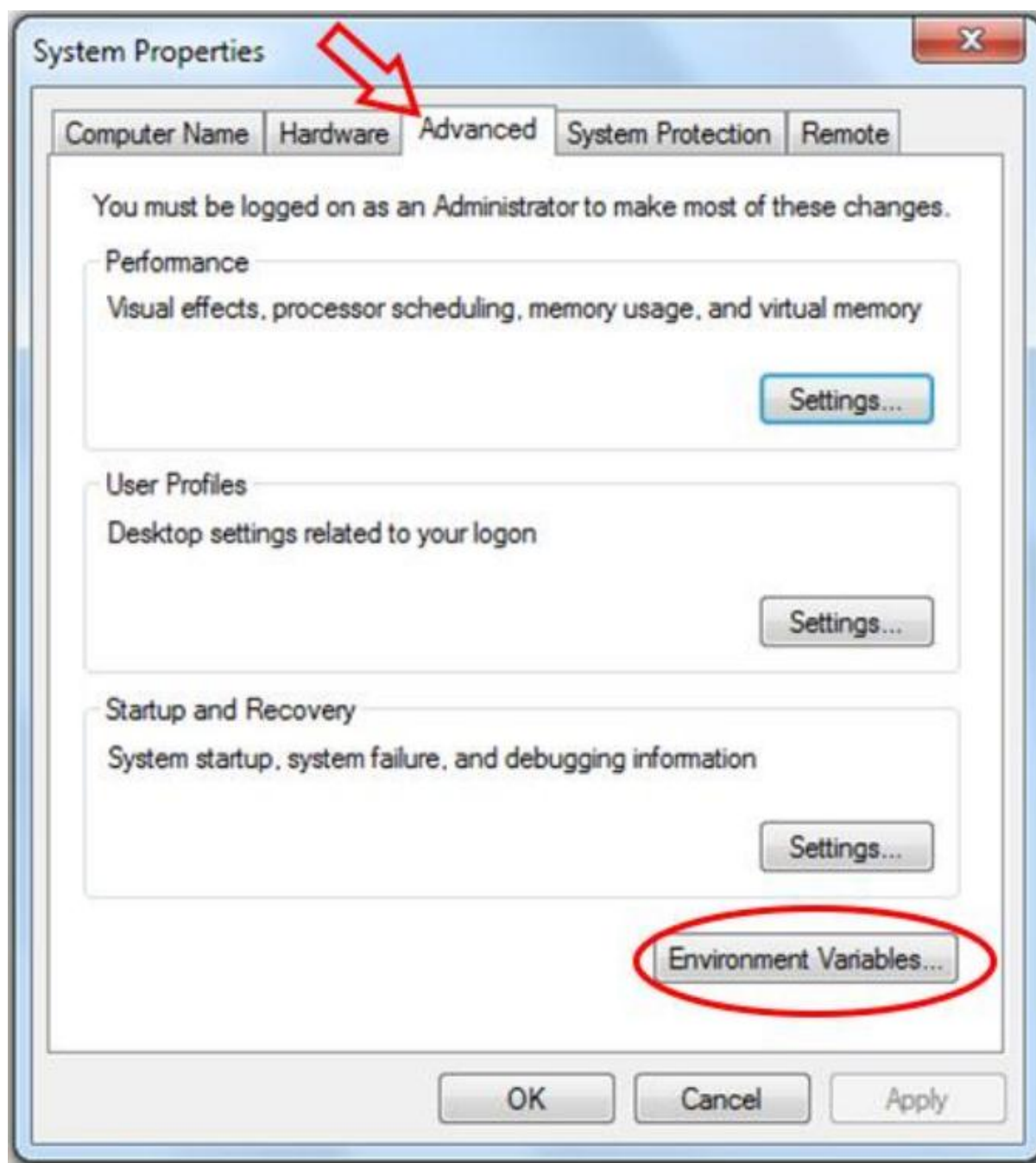
به جای فولدر jdk1.6.0\_15 نام فولدری که جاوا در آن نصب است را قرار دهید.

البته از راه دیگری نیز این کار امکان پذیر است که به صورت زیر می باشد.

روی My computer راست کلیک کرده و گزینه peroperties را کلیک کنید سپس در پنجره جدید روی گزینه Advanced System Settings کلیک کنید.

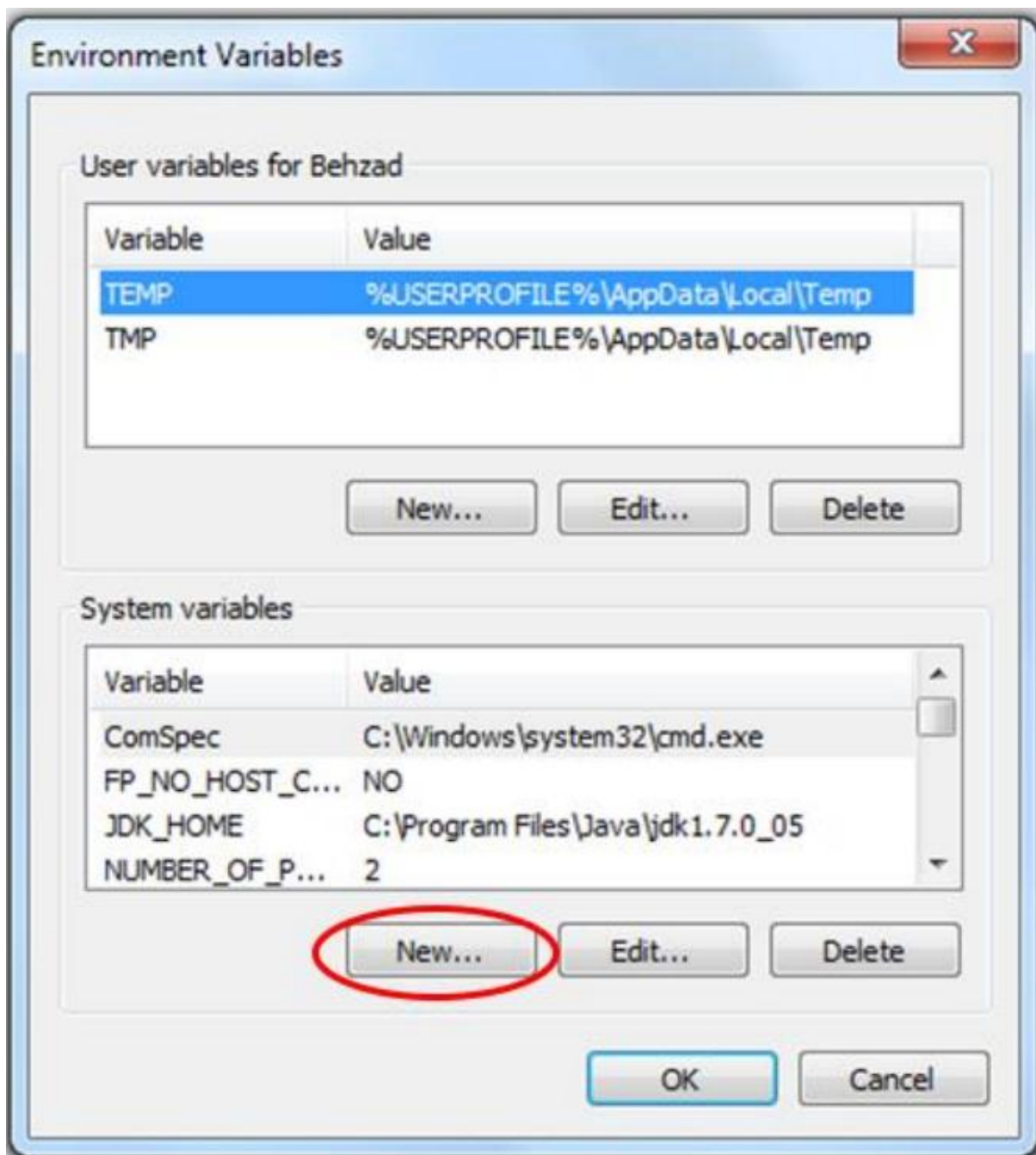


در مرحله بعد روی دکمه Environment Variable از برگه Advanced کلیک کنید.



سپس روی دکمه New در پایین پنجره کلیک کنید.





سپس در پنجره باز شده مقدار Variable name را برابر با JDK\_HOME قرار داده و مقدار Variable value را نیز آدرس مکانی که جاوا نصب می باشد را قرار دهید.



در Linux برای تنظیمات جاوا باید در یک فایل با پسوند cshrc دستورات زیر را نوشته و ذخیره کنیم و در آخر آن را اجرا کنیم.

```
setenv PATH /jdk1.6.0_15/bin:%PATH
```

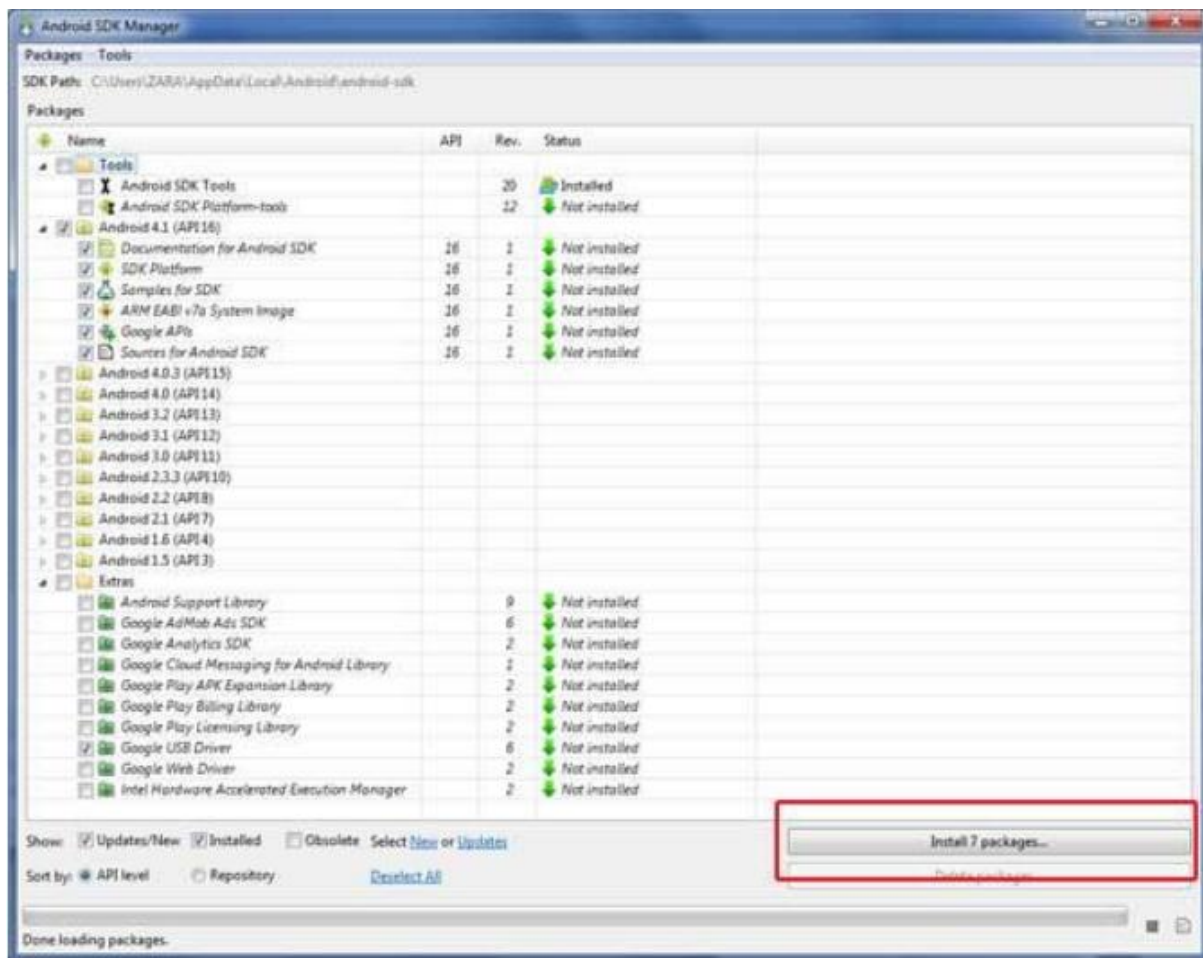
```
setenv JAVA_HOME /jdk1.6.0_15
```

به جای فولدر jdk1.6.0\_15 نام فولدري که جاوا در آن نصب است را قرار دهید.

### مرحله دوم نصب Android SDK

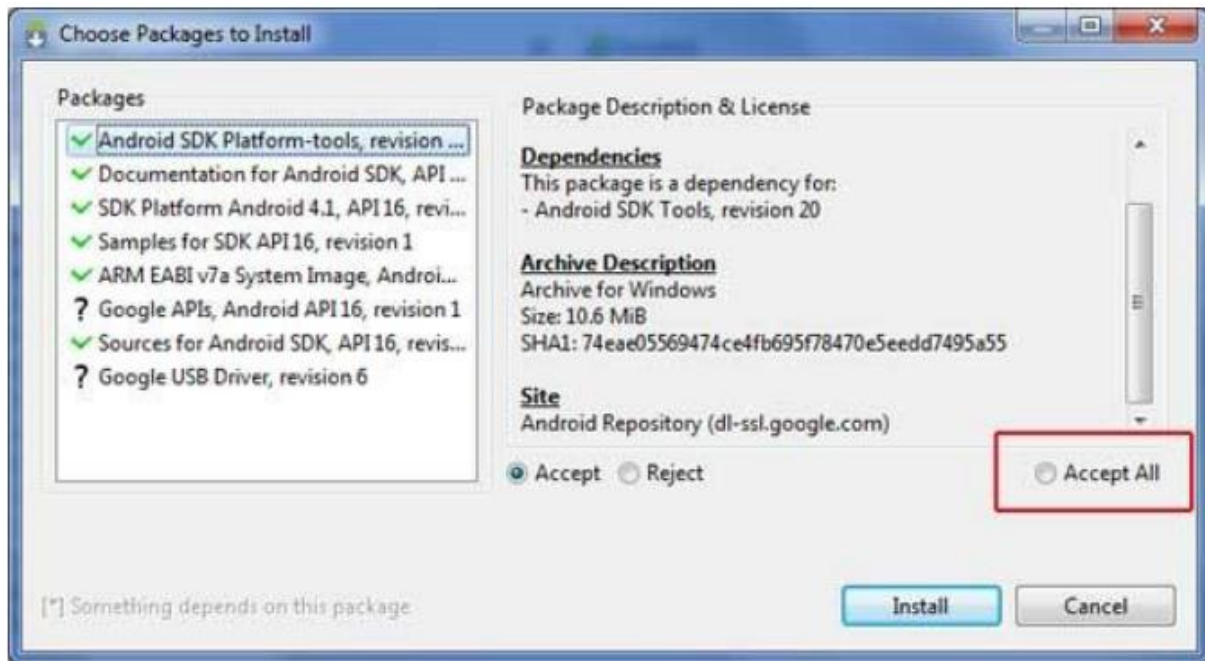
شما می توانید آخرین نسخه از Android SDK را از آدرس <http://developer.android.com/sdk/index.html> دانلود کنید. فایل installer\_rXX-windows.exe برای ویندوز و android-sdk\_rXX-macosx.zip برای Mac OS و android-sdk\_rXX-linux.tgz برای Linux می باشد.

بعد از نصب این فایل ویندوزی Android SDK را از آدرس All Programs > Android SDK Tools > SDK Manager اجرا کنید تا پنجره زیر نمایان شود



در این زمان شما پکیج های مورد نیاز را باید نصب کنید. بصورت پیش فرض شما باید ۷ پکیج را همانند تصویر بالا نصب کنید.

برای اینکار باید روی دکمه Install 7 Packages کلیک کنید تا پنجره زیر نمایش داده شود.



برای نصب بعد از وصل شدن به اینترنت دکمه رادیویی Accept All را کلیک کرده و سپس روی دکمه Install کلیک کنید.

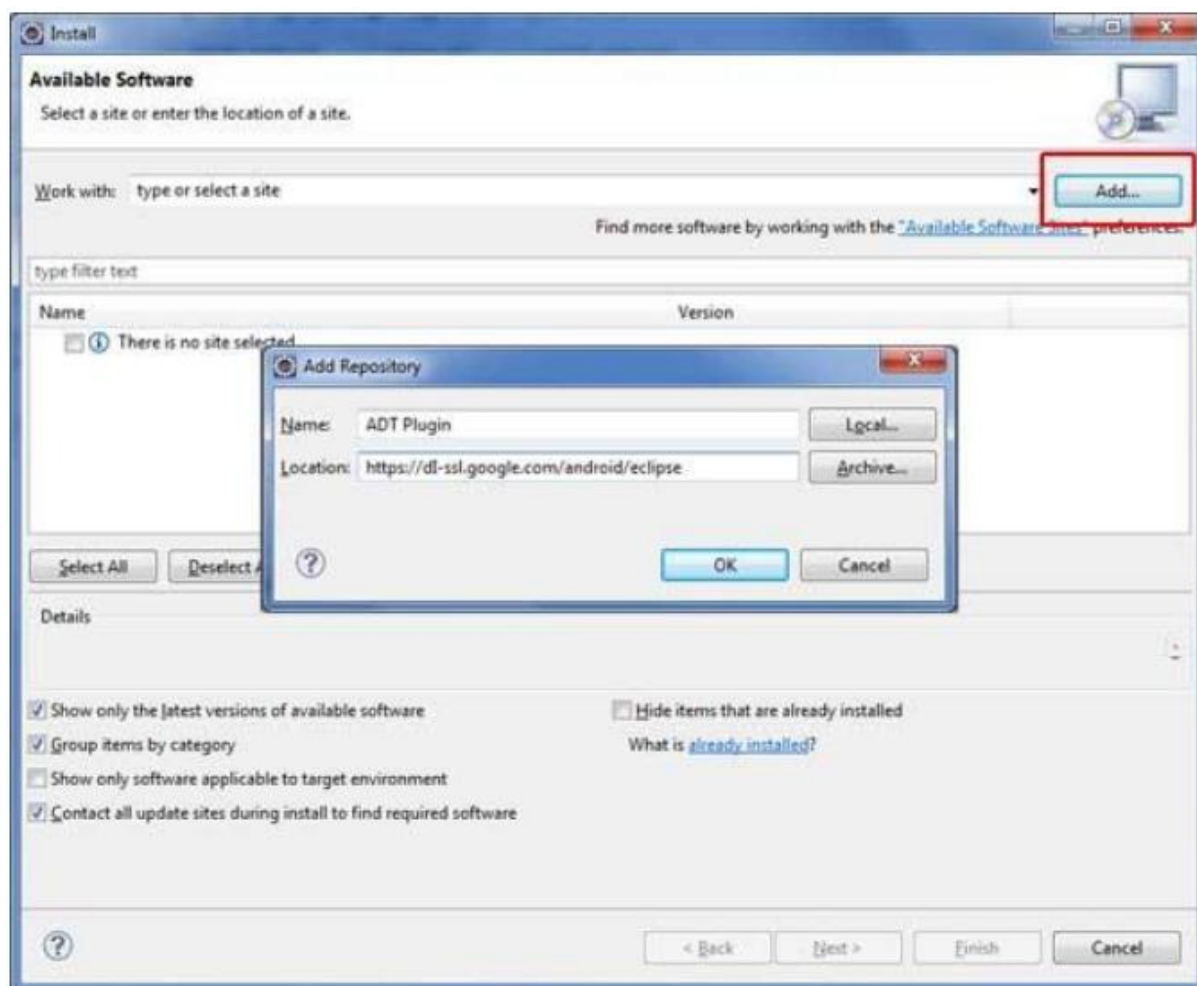
### مرحله سوم نصب Eclipse IDE

شما می توانید آخرین نسخه از Eclipse IDE را از آدرس <http://www.eclipse.org/downloads> دانلود کنید. اما نکته ای که در رابطه با آماده سازی محیط برنامه نویسی اکلیپس از اهمیت بسزایی برخوردار است آگاهی از این موضوع است، کدام نسخه اکلیپس با کدام نسخه SDK بهترین هم خوانی را دارد. از سوی دیگر از آنجا که سایت رسمی اندروید اجازه استفاده کاربران ایرانی را از امکانات رایگان این سایت نمی دهد، بالتبع کاربران ایرانی نمی توانند SDK و ADT که برای توسعه اندروید لازم و ضروری است را دانلود کنند. با جستجو در اینترنت در سایت های ایرانی می توان نسخه هایی از SDK و ADT برای دانلود یافت که همچنین نحوه نصب آنها روی اکلیپس نیز آموزش داده شده است. از جمله این وب سایت ها می توان به آدرس <http://androidcode.ir/page/android-sdk-jdk-update> اشاره کرد.

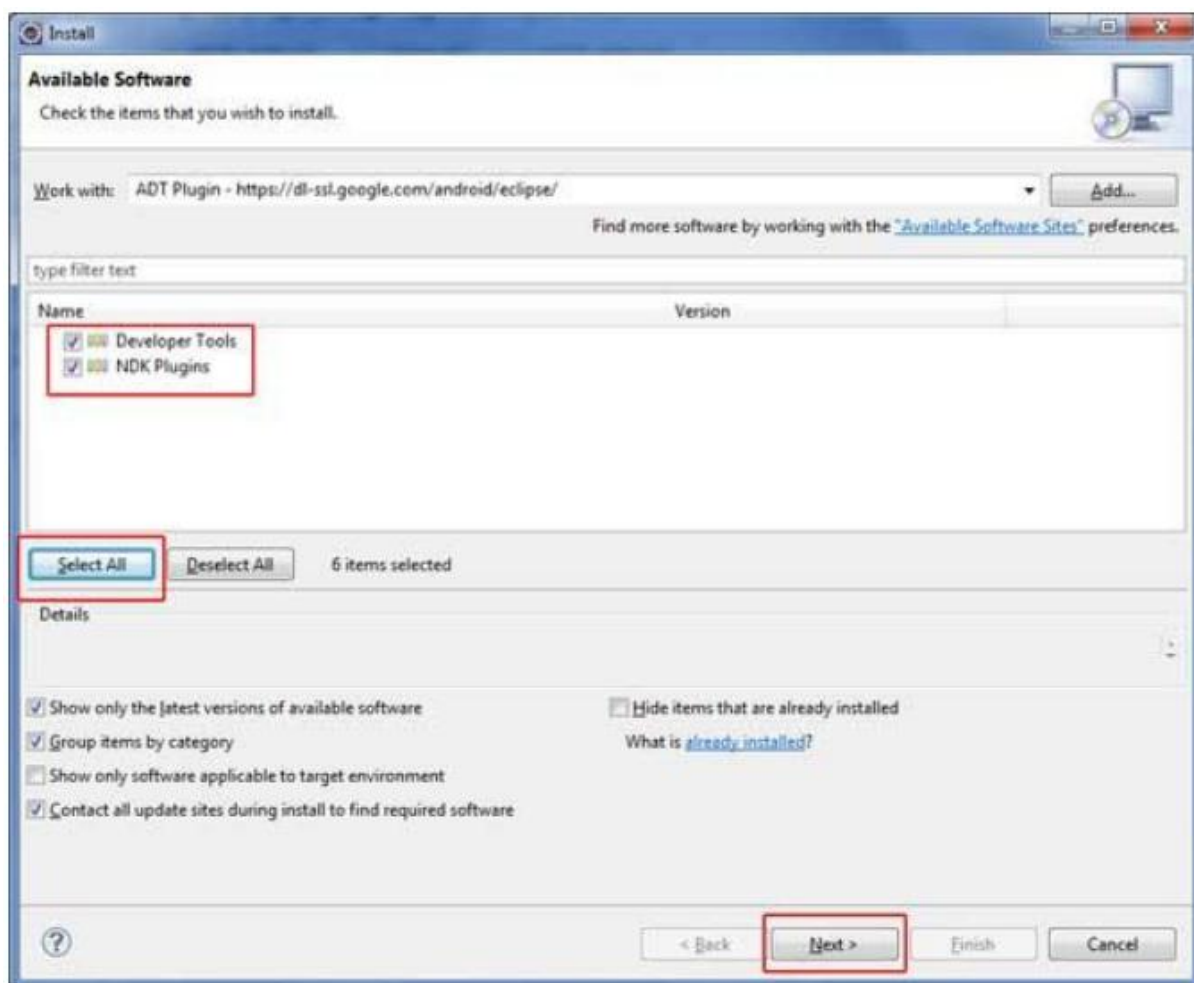
به دلیل اینکه اکلیپس Open Source است نیازی به نصب ندارد و برای استفاده از آن باید فایل فشرده آن را در پوشه ای دلخواه استخراج کنیم. بعد از استخراج روی فایل eclipse.exe درون پوشه انتخابی دوبار کلیک کنید تا اکلیپس اجرا شود.

## مرحله چهارم نصب ADT for Eclipse

شما می توانید آخرین نسخه ADT را از آدرس <http://developer.android.com/sdk/eclipse-adt.html> دانلود کنید. برای نصب ADT برای اکلیپس ابتدا منوی Help را انتخاب کنید. سپس گزینه Install new software را انتخاب کنید.



مانند تصویر بالا رو به روی قسمت Work with دکمه Add را کلیک کنید. در پنجره مشاهده شده روی دکمه Archive کلیک کنید. سپس فایل فشرده ADT را انتخاب کنید و روی دکمه Open کلیک کنید. سپس روی دکمه Ok کلیک کنید.



مانند تصویر بالا در لیستی که مشاهده می شود تمامی قسمت ها را تک زده و روی دکمه Next کلیک کنید. سپس روی دکمه Next بعدی کلیک کنید. سپس روی گزینه I accept te terms of the license agreements را انتخاب کرده و روی دکمه Finish کلیک کنید. در اینجا ADT برای اکیپس نصب شده است و ایکلیپس را دوباره راه اندازی کنید تا تغییرات اعمال شود.

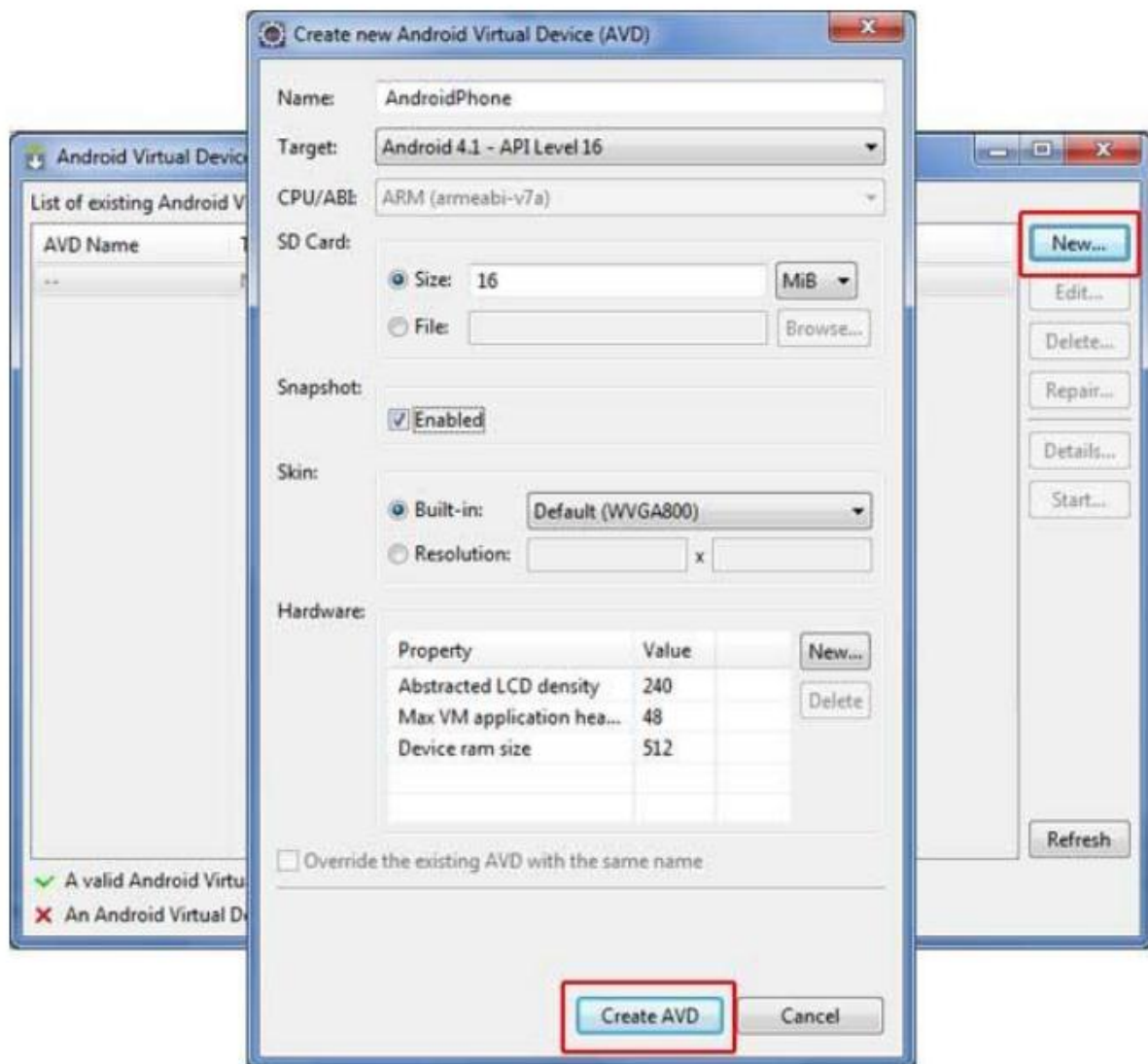
#### نکته:

در صورتی که فایل ADT را جداگانه دانلود نکرده باشید، بعد از اتصال به اینترنت در آدرس بار رو به روی قسمت Work with یکی از آدرس های موجود را انتخاب کرده و منتظر می مانیم تا لیست نمایش داده شود سپس مراحل بعدی را انجام می دهیم.

## مرحله پنجم ساخت دستگاه مجازی اندروید

برای اینکه ما برنامه های خود را آزمایش کنیم نیاز به یک دستگاه اندرویدی داریم که شرکت گوگل همراه Android SDK یک مدیر دستگاه مجازی برای ما تدارک دیده است تا بتوانیم دستگاه های مورد نیاز خود را ایجاد کنیم. (اگر دستگاه اندرویدی ما واقعی باشد بهتر است).

برای ساخت دستگاه مجازی از آدرس All Programs > Android SDK Tools > AVD Manager پنجره زیر نمایان شود، سپس روی دکمه New کلیک کنید تا دستگاهی جدید ایجاد کنیم.



در پنجره جدید تنظیمات دستگاه مجازی را انجام دهید، نام دستگاه، نسخه اندروید، اندازه کارت حافظه، اندازه صفحه نمایش، سخت افزار های مورد نیاز را وارد کنید و همچنین برای اینکه بارگذاری اولیه دوباره صورت نگیرد گزینه Snapshot را نیز انتخاب کرده و روی دکمه Create AVD کلیک کنید.



اگر دستگاه مجازی شما با موفقیت ساخته شود در لیست پنجره Android Virtual Devices نمایش داده خواهد شد و برای اجرای آن کافیست روی دکمه Start کلیک کنید

## دانلود ADT Bundle برای توسعه اندروید

علاوه بر محیط برنامه نویسی اکلیپس، شرکت گوگل نسخه ای از نرم افزار اکلیپس را تحت عنوان ADT Bundle تولید کرده است که کلیه ابزار های لازم برای توسعه اندروید از جمله خود محیط برنامه نویسی اکلیپس به علاوه SDK و ADT را داراست. این محیط برنامه نویسی به هیچ وجه نیاز به تنظیمات خاصی نداشته و صرفا با اجرای برنامه خواهیم توانست توسعه اندروید را شروع نماییم.

برای دانلود این نرم افزار می بایست به سایت <http://developer.android.com/sdk/installing/bundle.html> شرکت گوگل مراجعه نموده و این نرم افزار را به صورت کاملا رایگان دریافت نمود ولی از آنجا که امکان دسترسی کاربران ایرانی به این وب سایت امکان پذیر نمی باشد، اکثر کاربران نخواهند توانست به سادگی ADT Bundle را دانلود کنند.

کاربرانی که نمی توانند سرورهای گوگل را دور زده و این نرم افزار را دانلود نمایند، می توانند به آدرس <http://www.kamalan.com/?p=1694> مراجعه نموده و نسخه کامل ADT Bundle را دانلود نمایند. به نظر می رسد چنانچه کاربران طبق دستورات نصب در وب سایت فوق الذکر عمل کنند با هیچ مشکل خاصی مواجه نشوند.

## آماده سازی محیط برنامه نویسی اندروید استودیو

از آنجا که شرکت گوگل دسترسی به سایت اختصاصی اندروید را برای کاربران ایرانی محدود کرده است و از سوی دیگر تنها مرجع رسمی برای دانلود رایگان اندروید استودیو نیز همین وب سایت است، به نظر می رسد که دانلود اندروید استودیو برای گروه قابل توجهی از کاربران امکان پذیر نباشد. به هر حال علاقه مندان می توانند برای دانلود مستقیم اندروید استودیو به آدرس <https://developer.android.com/sdk/installing/studio.html> مراجعه نموده و نسخه سازگار با سیستم عامل خود را دانلود نمایند.

برای آن دسته از کاربرانی که امکان دسترسی به وب سایت اصلی اندروید را ندارند، سایت هایی مانند <http://www.p30download.com> و <http://androidkade.com> اقدام به آپلود این نرم افزار کرده اند، که آنها می توانند به راحتی نرم افزار را دانلود کنند.

همچنین باید فایل Gradle را نیز دانلود کنید که از آدرس <http://downloads.gradle.org/distributions/gradle-1.12-all.zip> امکان پذیر است.



ابتدا اندروید استودیو را نصب کنید سپس فایل Geadle را مسیر `c:\users\USER PC\gradle` استخراج کرده و خود فایل فشرده را نیز در آنجا قرار دهید. اندروید استودیو آماده برنامه نویسی می باشد.