

موسسه آموزش عالی علامه دهخدا

مستندسازی پروژه نهایی برای دریافت درجه کارشناسی

عنوان:

بازی آندروید:تست حافظه

استاد راهنما:

جناب آقای مهندس خزایی

دانشجو:

محمدامین امینی

9111520

**شهریور1393**

تقدیم به

تمام کسانی که در این راه مرا حمایت کردند ؛ به خصوص استاد گرامی ، جناب آقای مهندس خزایی.

**چکیده**:

این مستندسازی برای پروژه ی یک بازی آندرویدی است.این پروژه با زبان جاوا در محیط اکلیپس برای سیستم عامل آندروید ساخته شده است.

به همین منظور در این مستند سازی ابتدا درمورد سیستم عامل آندروید و سپس درباره ی زبان برنامه نویسی جاوا و محیط اکلیپس توضیحاتی مختصر داده شده است و سپس کدهای ساخت بازی قرار داده شده اند.

حداقل نسخه ی سیستم عامل برای اجرای این بازی آندروید "کندوی عسل" و نسخه هدف بازی "آبنبات ژله ای" میباشند.

در این بازی شما با نه خانه روبرو هستید که با تکیه بر شانس و حافظه باید عکس های مشابه را انتخاب کنید و حواستان به بمب باشد.اگر هر چهار عکس را بتوانید با مشابه های خود انتخاب کنید بازی را برده اید.

کلمات کلیدی : بازی ، سیستم عامل ، آندروید ، زبان برنامه نویسی ، جاوا.

***فهرست فصل ها***

1– مقدمه . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

2– سیستم عامل آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

2-1-آندروید چیست؟. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2-1-1-معنای آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2-1-2-تعریف آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2-1-3-تاریخچه . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2-1-4-کپی رایت. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2-1-5-لوگو. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2-1-6-ویژگی ها. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2-1-7-فرمت فایل های پشتیبانی شده. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2-2- آرت. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2-3- محیط برنامه نویسی. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2-4- نسخه های آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2-4-1-نگاهی به نسخه های مختلف آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11 2-4-4- پراکندگی نسخه های مختلف آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2-5- سندباکس. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2-6- آندروید مارکت . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

2-7- گوگل تی وی . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

2-8- معماری آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2-8-1- تشریح لایه های مختلف معماری . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

2-9- رابطه آندروید و جاوا . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2-10- ماشین مجازی دالویک . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2-11- بازی های آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2-12- رده های سنی بازی آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2-13- ورژن های مختلف بازی های آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2-14- قابلیت های بازی آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2-15- معایب بازی های آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3 – جاوا . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3-1-تاریخچه . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3-2-اهداف اولیه . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3-3-نسخه ها . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3-4- برنامه های جاوا و اپلت ها . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3-5- .Net رقیب جاوا؟. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3-6- بین .Net و جاوا کدام رو انتخاب کنیم؟ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

3-7-خط مشی جاوا . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

3-8-پیاده سازی . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

3-9- اداره خودکار حافظه. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

3-10- گرامر. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

3-11- توزیع های جاوا. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

3-12- کتابخانه های جاوا. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

3-13- ویرایش ها. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

3-14- ایرادات مطرح شده. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

3-15- پاسخ به ایرادات. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

3-16- یک اشتباه متداول! . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

3-17- برنامه نویسی برای آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

3-17-1-نحوه نصب SDK. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

3-17-2- AVD. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .41

3-17-3- نصب برنامه شبیه ساز در کامپیوتر. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

3-18- Eclipse و برنامه نویسی با آن. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3-18-1- برنامه نویسی آندروید چگونه است؟. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3-18-2- Eclipse چیست؟. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3-18-2-1- معماری. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3-18-2-2- سکوی غنی مشتری. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

3-18-2-3- تاریخچه. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .46

3-18-2-4- نسخه ها. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .47

3-18-2-5- افزونه ها. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .47

3-18-2-6- شروع کار. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

3-18-2-7- نمونه هایی از برنامه های جاوا. . . . . . . . . . . . . . . . . . . . . . . . . 51

3-18-2-8- کلاس های خاص. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

4 – معرفی پروژه . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

5 – توضیحات فنی و کدها . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

***فهرست جدول ها***

جدول 1 – نسخه های آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

جدول2 – نسخه های جاوا . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

جدول3- نسخه های Eclipse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

***فهرست شکل ها***

شکل1-لوگوی آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

شکل2-معماری سیستم عامل آندروید . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

شکل3-لوگوی جاوا . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

شکل4-SDK Manager . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

شکل5-مراحل نصب شبیه ساز. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .41

شکل6-شبیه ساز آندروید. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

شکل7-کدهای نصب شبیه ساز . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

شکل8-نرم افزار اکلیپس. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

شکل9-محیط برنامه نویسی اکلیپس. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .49

شکل10-کادر ورود اطلاعات برای ساختن اپ. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

شکل11-صفحه ی خانه ی بازی. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .57

شکل12-صفحه ی اصلی بازی. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

شکل13-هدف بازی. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

شکل14-پیغام پایان بازی. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

شکل15-محتویات پروژه. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

***فصل اول***

**مقدمه**

١

توسعه ی روزافزون بازار تلفن های همراه و پیشرفت چشمگیر ابزارها و برنامه های کاربردی مختلف در طی دهه های گذشته ، باعث شده است که تلفن همراه به عنوان جزیی جدایی ناپذیر از زندگی افراد تبدیل شود. این مساله تا حدی است که امکانات بعضی از نمونه های تلفن همراه با کامپیوترهای امروزی برابری میکند. از یک طرف،به دلیل قالب جمع و جور تلفن های همراه که امکان حمل و نقل آسان آن را فراهم میکند و از طرف دیگر،پیشرفت چشمگیری که در فشرده سازی هرچه بیشار مدارهای الکترونیکی حاصل شده است،باعث شده تا بازار تلفن های همراه ،هرروزه شاهد ظهور مدل های جدید با قابلیت های بهتر نسبت به مدل های قبلی باشد. این پیشرفت ها فقط محدود به توسعه های سخت افزاری نمی شود و توسعه های نرم افزاری را نیز شامل میشود.به همین دلیل است که شرکت های بزرگی همچون گوگل نیز در برابر وسوسه ورود به این بازار پررونق نمیتوانند مقاومت کنند..

سیستم عامل اندروید که از طرف گوگل عرضه شده و به صورت منبع باز منتشر شده است،راه را برای توسعه هرچه بیشتر این بازار فراهم آورده است. اندروید به سرعت درحال رشد و پیشرفت است و به طور سالانه درحدود ده هزار برنامه کاربردی برای آن معرفی میشود که این مساله به وجود مرجعی برای توسعه و ایجاد برنامه های کاربردی مناسب نیاز دارد. به طور کلی،یادگیری سیستم عامل اندروید بسیار ساده است و گوگل کتابخانه های مختلفی را برای پیاده سازی هرچه بهتر برنامه های کاربردی مختلف عرضه کرده است.تمرکز اصلی این مستند سازی بر قابلیت توسعه ی برنامه های کاربردی اندروید گذاشته شده است.

برای فهم این مستندسازی آشنایی مختصر با محیط برنامه نویسی و زبان های برنامه نویسی لازم است .هرچند که در ادامه به معرفی سیستم عامل آندروید و زبان برنامه نویسی جاوا و محیط توسعه ی Eclipse پرداخته ام ؛ اما بطور کلی ، آشنایی نسبی با ساختارهای برنامه نویسی شی گرا و رویداد محور میتواند در هرچه بهتر فهمیدن این مقاله موثر باشد.

بطور خلاصه این مقاله بصورت زیر ادامه می یابد:

در فصل دوم، به معرفی سیستم عامل آندروید و مروری به جنبه های مختلف آن پرداخته میشود.این فصل حاوی برنامه نویسی نیست. در ادامه این فصل نسخه های مختلف این سیستم عامل و سپس تاریخچه ای کوتاه از بازی های آندروید و حواشی و مزایا و معایب آنها را خواهیم خواند.

٢

در فصل سوم ، معرفی ای از زبان برنامه نویسی جاوا ، چگونگی و چرایی پدید آمدن آن را خواهیم داشت. در ادامه ی این فصل ، محیط توسعه ی Eclipse و کار با آن را توضیح خواهم داد.

حالا اندکی برای برنامه نویسی و تولید یک برنامه کاربردی آماده ایم.

در فصل چهارم ، توضیحی کوتاه درمورد برنامه ی کاربردی تولیدی خودم ، که یک بازی میباشد خواهم داد ؛ و در فصل پنجم ، شاهد توضیحات فنی و کدهای برنامه نویسی این برنامه هستیم.در این فصل ، که فصل آخر مستندسازی است، سعی برآن شده تا توضیحات و کدها بصورت آموزشی برای خواننده نوشته شود و امید به آن است که قدمی هرچند کوتاه برای شما خواننده ی عزیز در مسیر بلند برنامه نویسی آندرویدی باشد.

موفق باشید.

٣

***فصل دوم***

**سیستم عامل آندروید**

٤

**-21-آندروید چیست؟**

**-21-1-معنای آندروید:**

پیش از ورود به اطلاعات مربوط به آندروید، نخست به نام آن می‌پردازیم. بنابر ترجمه دیکشنری کمبریج، آندروید این گونه تعریف شده است: «یک ربات (ماشینی که به وسیله کامپیوتر کنترل می‌شود) که به گونه‌ای ساخته شده تا شکل ظاهری شبیه به انسان داشته باشد.» شاید بتوان نزدیک‌ترین معنی در زبان فارسی به آندروید را آدم آهنی یا آدم ماشینی دانست.

**-21-2-تعریف آندروید:**

اندروید1 نام یک سیستم عامل موبایل است که توسط شرکت گوگل توسعه داده میشود. این سیستم عامل اوپن سورس است و برپایه هسته لینوکس بنا شده است. اندروید بر خلاف سیستم عامل iOS آیفون که فقط پردازنده های ARM را پشتیبانی میکند، بر روی انواع مختلفی از پردازنده ها (ARM, MIPS, Power Architecture, x86) قابل نصب است. از سال 2008 تاکنون تلفن های همراه متعددی با استفاده از این سیستم عامل به بازار ارائه شده اند. همچنین چندین Tablet PC نیز با استفاده از این سیستم عامل به بازار ارائه شده اند.

**-21-3-تاریخچه:**

تاسیس آندروید و کار گروهی برای ساخت آن به ماه اکتبر 2003 میلادی بازمیگردد؛ زمانی که اندی رابین ، ریچ ماینر ، نیک سیرز و کریس وایت تصمیم به ساخت سیستم عاملی برای گوشی های هوشمند گرفتند که امکانات بیشتری به کاربران خود بدهد.

اولین قدم در تولید سیستم عامل آندروید زمانی برداشته شد که گوگل در سال 2005 شرکـــت ناشر اولیه ی آندروید را در پالو آلتوی کالیفرنیا خریداری کرد ، شرکت کوچک آندروید در زمینه تولید نرم افزار و برنامه های کاربردی برای تلفن های همراه فعالیت می کرد . اندی رابین مدیر ارشد اجرایی این شرکت پس از پیوستن آندروید به گوگل به سمت قائم مقام مدیریت مهندسی این شرکت و مسئول پروژه آندروید در گوگل منصوب شد.در واقع می توان رابین را پایه گذار آندروید دانست ، چرا که وی علاوه بر اینکه ایده تولید آندروید

1.Android

٥

را در شرکت کوچک خود پرورش داد ، در سمت مدیر این پروژه در شرکت گوگل توانست ایده خود را پیاده سازی کند و سیستم عامل اندروید را با نام شرکت کوچک پیشین خود روانه بازار نماید.

تیم آندروید به رهبری رابین فعالیت خود را برای تولید platform مبتنی بر هسته لینوکس آغاز کردند. درز اخباری از فعالیت های این تیم به خارج از گوگل ، سبب بروز شایعاتی مبتنی بر تمایل گوگل به تولید تلفن های همراه در اواخر سال 2006 گردید . با اعلام زمان کنفرانس خبری شرکت گوگل در نوامبر سال 2007 ، تمامی رسانه ها و افکار عمومی جهان چشم انتظار مشاهده نخستین تلفن همراه ساخت گوگل بودند . ولی غافلگیری بزرگ رخ داد . هیچ خبری از "یک" گوشی تلفن همراه نبود بلکه خبر داغ آن روز درمورد ورود صدها تلفن همراه در سال های پیش رو بود که توسط شرکت های مختلف تولید می شد.

گوگل به شرکت های تولیدکننده تلفن همراه ، وعده ی سیستم عامل انعطاف پذیر و قابل به روزرسانی را داده بود به همین منظور گوگل شرکت های بسیاری را با خود برای رسیدن به این هدف همراه کرده بود.

"اتحادیه گوشی باز" یا Open Handset Alliance در روز پنج نوامبر 2007 اعلام موجودیت کرد.

سی و چهار شرکت فعال در زمینه تولید نرم افزار ، تولید گوشی های تلفن همراه ، اپراتور تلفن همراه و تولید کننده نیمه رساناها و پردازنده های تلفن همراه اعضای موسس این اتحادیه بودند . در میان نام های مشهور در بین اعضای موسس ، شرکت هایی چون Samsung ، LG ، Motorola ، HTC ، T-Mobile ، eBay ، Nvidia ، intel و . . . و البته Google به چشم می خوردند . اریک اشمیت مدیر ارشد اجرایی گوگل در این مراسم گفت : (( اعلام امروز بسیار جاه طلبانه تر از معرفی تنها "یک" تلفن همراه گوگلی است که در چند هفته اخیر توسط رسانه ها پیش بینی شده بود . از دیدگاه ما پلتفرمی که ما ارائه کرده ایم ، هزاران تلفن همراه گوناگون را به بازار روانه خواهد کرد.))

نخستین گوشی مبتنی بر آندروید توسط شرکت HTC با همکاری T-Mobile تولید شد . این گوشی که به فاصله کمتر از یک سال از تشکیل اتحادیه Open Handset Alliance یعنی در بیست و دوم اکتبر 2008 تولید شد ، در بازارهای مختلف به نام های HTC Dream ، T-Mobile G1 ، Era G1 به بازار عرضه گردید . سپس گوگل کدهای مربوط به آندروید را به عنوان یک پروژه ی منبع باز منتشر کرد.

پس از وقایع مذکور گوگل به طور رسمی از Android Open Source Project رونمایی کرد . پروژه ای که به گفته ناشرین آن هدفش ارتقای کیفیت تلفن های همراه و تجربه ای که کاربران از تلفن های همراه خود دارند است . AOSP همچنین با رونمایی خود ، از مفهوم «دستگاه های مبتنی بر آندروید» نیز پرده برداری کرد. دستگاه هایی که قابلیت پشتیبانی و اجرای نرم افزارهای نوشته شده از سوی شرکت های Third Party را دارند

۹ دسامبر ۲۰۰۸ چهارده عضو جدید از شرکت‌های صنعت تلفن همراه جهان به اتحادیه گوشی باز پیوستند. در بین این نام‌ها باید به سونی اریکسون، اریکسون، توشیبا، ایسوز، گارمین، هواوی و آرم هولدینگز اشاره کرد. روند پیوستن شرکت‌های بزرگ به اتحادیه تا به امروز نیز ادامه داشته‌است و شرکت‌هایی چون ایسر، آلکاتل، لنوو، شارپ، فاکس‌کان، ان‌ای‌سی، کیوسرا، ان‌ایکس‌پی، اس‌تی-اریکسون، مارول، زدتی‌ئی و دل نیز از جمله شرکت‌هایی بوده‌اند که به جمع پشتیبانی‌کنندگان اندروید پیوسته‌اند.

6

در ۳ سپتامبر ۲۰۱۳ توسعه‎دهندگان اندروید به‌طور رسمی اعلام کردند که با شرکت نستله، که از شرکت‎های مطرح صنعت شکلات‌سازی جهان می‎باشد، همکاری خواهند کرد. در همین راستا نگارش ۴.۴ سیستم‎عامل اندروید، کیت‌کت نام گرفت. کیت کت از مارک‎های معروف شکلات است که توسط شرکت نستله تولید می‎شود.

**-21-4-کپی‌رایت و حق امتیاز:**

حق امتیاز آندروید به صورت اپن سورس بر اساس حق امتیاز آپاچی1 ارائه می‌گردد. بر این اساس شرکت‌های عضو اتحادیه می‌توانند با دسترسی به کدهای اصلی آندروید آن را مطابق دلخواه خود تغییر دهند و کد تغییر یافته را بدون عودت دادن برای خود حفظ کنند. با اینکه سعی می‌شود تا اکثریت قسمت‌های این سیستم عامل بر اساس همین مجوز ارائه گردد، استثناهایی نیز وجود دارد. برای مثال هسته لینوکس موجود در این سیستم عامل با پروانه عمومی همگانی گنو نسخه ی دو۲ منتشر شده است.

1.Apache License

2. GPLv2

٧

**-21-5-لوگوی اندروید:**

طراح لوگوی مشهور اندروید “آیرینا بلاک” است. سه سال پیش از آیرینا بلاک و تیم طراحی گوگل تقاضا شد تا لوگویی برای اندروید طراحی کنند که به سرعت با کاربر ارتباط برقرار کند و به آسانی قابل شناسایی باشد؛ همچنین به بلاک گفته شد که این لوگو باید حتماً تصویری از ربات باشد چرا که اندروید به معنای روبات است. آیرینا پس از مطالعه زیاد در مورد اسباب‌بازی‌ها و شخصیت‌های فانتزی و تخیلی در آخر تصمیم گرفت لوگوی اندروید را از یک منبع عجیب الهام بگیرد: دستشویی! هر کسی علامت روی در دستشویی‌ها را دیده و الهام از این علامت‌ها می‌تواند باعث شود در یک نگاه لوگوی اندروید شناخته شود. یک نکته جالب دیگر در مورد لوگوی اندروید این است که گوگل برخلاف دیگر شرکت‌ها که سعی در حفاظت از لوگویشان داشتند تصمیم گرفت تا لوگو را به صورت منبع باز قرار دهد تا هر کس بتواند آن را به دلخواه خودش تغییر دهد. گوگل در این‌باره می‌گوید: “ما تصمیم گرفتیم این لوگو می‌تواند یک لوگوی اشتراکی و تعاملی باشد که هر کس در دنیا بتواند آن را تغییر دهد. این تصمیم بسیار شجاعانه بود. ” شاید با اپلیکیشن Androidify و یا اسباب بازی‌ها و Action Figureهای کوچک اندرویدی بامزه آشنا باشید؛ همه این‌ها به لطف آزاد بودن لوگوی اندروید امکان‌پذیر شده است.



شکل1-لوگوی آندروید

**-21-6-ویژگی‌های اندروید:**

اندروید تمامی تکنولوژی‌های اتصال1 شامل GSM/EDGE, CDMA, EV-DO, UMTS, ، بلوتوث و وای-فای را پشتیبانی می‌کند.

اندروید از فرمت‌های مختلف فایل‌های مالتی مدیا مثل MPEG, MP3, AAC, AMR, JPEG, PNG, GIF پشتیبانی می‌کند.

اندروید برای ارسال پیغام‌های متنی یا همان اس ام اس از فرم‌های SMS, MMS و XMPP پشتیبانی می‌کند.

مرورگر موجود در اندروید بر اساس فریم ورک اوپن سورس WebKit توسعه یافته‌است.

اندروید برای ذخیره داده‌ها و مدیریت بانک‌های اطلاعاتی سبک از نرم‌افزار SQLite استفاده می‌کند.

تمام برنامه‌های اندروید به زبان جاوا نوشته می‌شوند. برای اجرای برنامه‌های جاوایی روی این سیستم عامل، کدهای جاوا به کدهای Dalvik تبدیل می‌شوند و سپس روی ماشین مجازی جاوایی 2 اجرا می‌شوند.(درمورد ماشین مجازی دالویک درادامه توضیح خواهم داد)

1. Connectivity

2. Dalvik virtual machine

٨

ابزارهای مختلف اندروید برای توسعه دهندگان به راحتی در دسترس است و توسط شرکت گوگل پشتیبانی می‌شوند. این ابزارها شامل کتابخانه‌ها، خطایاب، شبیه‌ساز گوشی و یک پلاگین برای اکلیپس است.

اندروید از سخت‌افزارهای مختلف همچون جی پی اس و دوربین‌های متنوع پشتیبانی می‌کند.

تصاویر و فایل‌های گرافیکی بوسیله OpenGL پردازش می‌شوند که کیفیت بالاتری خواهند داشت.

با استفاده از تکنولوژی نسبتاً جدید شرکت Adobe با نام AIR نیز می‌توان به توسعه برنامه‌های کاربردی تحت این سیستم عامل پرداخت.

**-21-7-فرمت فایل‌های پشتیبانی شده:**

اندروید در حالت پیش‌فرض فایل‌های mp3, aac , ogg , amr , midi , mpeg4 , wav , bmp , gif , png , jpg را پشتیبانی می‌کند. اندروید Adobe Flash را نیز پخش می‌کند و می‌تواند فایل‌های GIF متحرک را با حرکت پخش کند. برای پخش فایل‌های جریان دار مانند صوت و ویدئو نیز می‌توانید از تگ ویدئو html5 و همچنین تکنولوژی Adobe Flash Streaming استفاده کنید. در نسخه‌های جدید اندروید، موتور جاوااسکریپت مرورگر کروم که سرعت بسیار بالایی در اجرای کدهای جاوا اسکریپت دارد به مرورگر اندروید متصل شده‌است. (در ضمن مرورگر اندروید کدهای HTML5 را پشتیبانی می‌کند)

**-22-آرت :**

آرت ران تایم جدید اندروید است که جایگزین دالویک شده است. شرکت گوگل برای اولین بار در اندروید ۴.۴ آرت را در کنار دالویک قرار داد و کاربران می‌توانستند با مراجعه به تنظیمات آن را فعال کنند.

**-23-محیط برنامه‌نویسی اندروید:**

مجموعه برنامه نویسی اندروید یا Android SDK شامل یک دیباگر، کتابخانه‌های اندروید، شبیه‌ساز سیستم عامل، مستندات اندروید و فایل‌های نمونه و آموزشی است که به کاربر در ایجاد برنامه‌ها کمک می‌کند. هم اکنون این SDK بر روی یک سیستم ۳۲ بیتی که لینوکس، ویندوز و یا mac OSX داشته باشد اجرا می‌شود. پیش نیازهای نصب این SDK عبارتند از JDK و Apache Ant و python 2.2.

٩

**-24- نسخه های آندروید:**

گوگل ویرایش های گوناگون آندروید را علاوه بر شماره ویرایش ، با نام یک شیرینی یا دسر معرفی میکند. این نام البته از ترتیب حروف الفبا برای حرف اول آن نام نیز پیروی میکند به گونه ای که ویرایش های منتشر شده آندروید تا به امروز به این نام ها بوده اند:

جدول1- نسخه های آندروید

|  |  |  |  |
| --- | --- | --- | --- |
| نسخه | نام لاتین | نام فارسی | تاریخ انتشار |
| 1.0 | Alpha | - | ۲۳ دسامبر ۲۰۰۸ |
| 1.1 | Beta | - | ۹ فوریه ۲۰۰۹ |
| 1.5 | Cupcake | کیک فنجانی | ۳۰ آوریل ۲۰۰۹ |
| 1.6 | Donut | پیراشکی | ۱۵ سپتامبر ۲۰۰۹ |
| 2.0 و 2.1 | Éclair | نان خامه ای | ۲۶ اکتبر ۲۰۰۹ |
| 2.2 | Froyo | ماست یخ زده | ۲۰ می۲۰۱۰ |
| 2.3 | Gingerbread | نان زنجبیلی | ۶ دسامبر ۲۰۱۰ |
| 3.0 ، 3.1 و 3.2 | Honeycomb | کندوی عسل | ۲۲ فوریه ۲۰۱۱ |
| 4.0 | Ice Cream Sandwich | بستنی حصیری  (ساندویچ بستنی) | نوامبر ۲۰۱۱ |
| 4.1 | Jelly Bean | آبنبات ژله ای | ژوئیه ۲۰۱۲ |
| 4.2 | Jelly Bean | آبنبات ژله ای | اکتبر ۲۰۱۲ |
| 4.3 | Jelly Bean | آبنبات ژله ای | ژوئیه ۲۰۱۳ |
| 4.4 | KitKat | کیت کت | اکتبر ۲۰۱۳ |
| 5.0 | Lollipop | آبنبات چوبی | ۵ جولای ۲۰۱۴ |

١٠

**-24-1- نگاهی به نسخه های مختلف آندروید:**

**نسخه ی 1.5 یا Cupcake:**

نسخه ی 1.5 آندروید در 30 آوریل 2009 منتشر شد.این نسخه مبتنی بر هسته لینوکس 2.6.27 بود.از جمله قابلیت هایی که در این ویرایش گنجانده شده بود شامل :

* امکان ضبط فیلم از طریق دوربین فیلمبرداری
* فرستادن فیلم به سایت یوتیوب و عکس به سایت پیکاسا به صورت مستقیم از روی گوشی
* صفحه کلید مجازی با قابلیت پیش بینی کلمات وارد شده
* پشتیبانی از پخش استریوی موسیقی از طریق بلوتوث و کنترل پخش موسیقی یا ویدیو از طریق بلوتوث
* قابلیت اتصال اتوماتیک به دستگاه های بلوتوث
* امکان شخصی سازی صفحه اصلی با استفاده از ویجت ها و یا پرونده های شخصی

جابجایی انیمیشنی تصاویر به هنگام عوض شدن صفحات

**اندروید نسخه 1.6 یا Donut:**

در سپتامبر 2009 اندروید نسخه 1.6 را منتشر کرد.این نسخه مبتنی بر هسته لینوکس 2.6.29 بود و قابلیت های زیر را به اندروید افزود:

* بهبود در سرویس Android Market
* رابط کاربری یکپارچه برای دوربین عکسبرداری ، دوربین فیلمبرداری و گالری تصاویر
* امکان انتخاب چند عکس برای پاک کردن در منوی گالری
* به روزرسانی ویژگی جست و جوی صوتی
* به روزرسانی ویژگی جست و جو با قابلیت جست و جو در موارد نشانه گذاری شده،تاریخچه،اسامی و وب از صفحه اصلی
* پشتیبانی از تکنولوژی به روز شده CDMA/EVDO ، X802.1 ، VPN و موتور Text To Speech
* پشتیبانی از رزولشن WVGA برای صفحه نمایش

١١

**اندروید نسخه 2 و 2.1 یا Eclair:**

هر دو نسخه 2 و 2.1 اندروید مانند نسخه 1.6 مبتنی بر هسته لینوکس 2.6.29 طراحی شده بودند.اهم امکانات اضافه شده در این نسخه به شرح زیر است :

* سرعت سخت افزاری بهبود یافته
* ویژگی چندلمسی
* پشتیبانی از رزولشن های بیشتر برای صفحه نمایش
* رابط کاربری به روز رسانی شده
* مرورگر اینترنتی با قابلیت پشتیبانی از HTML5
* دفترچه تلفن به روز رسانی شده
* Google Map نسخه 3.1.2
* پشتیبانی از Microsoft Exchange
* افزوده شدن امکان فلاش داخلی برای دوربین
* افزوده شدن زوم دیجیتال دوربین
* به روزرسانی صحه کلید مجازی
* پشتیبانی از بلوتوث نسخه 1.2
* اضافه شدن قابلیت Live Wallpaper
* اضافه شدن امکان ارسال فایل با استفاده از بلوتوث

**نسخه 2.2 یا Froyo:**

اندروید نسخه 2.2 را در می 2010 معرفی شد.این ویرایش اندروید مبتنی بر هسته لینوکس نسخه 2.6.32 است و قابلیت های زیر به آن اضافه شده است :

* افزایش سرعت سیستم عامل ، حافظه و عملکرد سیستم بین 2 تا 5 برابر نسخه 2
* افزایش سرعت اجرای برنامه های کاربردی با استفاده از تکنیک های JIT
* اضافه شدن موتور نرم افزار مرورگر اینترنتی مبتنی بر موتور Webkit که با موتور Chrome JavaScript V8 همراه شدن است که سرعت بسیار بالایی در اجرای کدهای JavaScript دارد.
* افزایش پشتیبانی از Microsoft Exchange با قابلیت هایی چون سیاست حریم شخصی به روز شده ، همسان سازی تقویم و . . .

١٢

* Android Market به روز شده با قابلیت به روزرسانی خودکار برنامه های کاربردی
* شماره گیری صوتی و انتقال دفترچه تلفن از طریق بلوتوث
* امکان نصب برنامه های کاربردی روی حافظه های جانبی
* پشتیبانی از فلش نسخه 10.1
* بهبود عملکرد دوربین در حالت های عکس و فیلمبرداری

**اندروید نسخه 2.3 یا Gingerbread:**

اندروید نسخه 2.3 یا نان زنجبیلی در دسامبر 2010 معرفی شد.امکانات اضافه شده دراین نسخه به شرح زیر هستند:

* بهینه سازی و تغییر در طراحی رابط کاربری به همراه ساده تر و سریعتر کردن آن
* پشتیبانی از صفحه های بزرگتر و رزولشن های بیشتر WXGA و بیشتر
* پشتیبانی از تماس اینترنتی یا تماس های VoIP
* افزودن قابلیت NFC یا همان ارتباط حوزه نزدیک
* بهبود در نمای گرافیکی آندروید و صفحه ی کیبورد لمسی و قابلیت های دیگر

**اندروید نسخه 3 و 3.1 و 3.2 یا Honeycomb که مخصوص Tablet ها می باشد:**

* بهبود در بسیاری از ویژگی های گرافیکی آندروید
* قابلیت پشتیبانی از پردازنده های چند هسته ای
* پشتیبانی از USB Host Mode برای انتقال مستقیم اطلاعات از دوربین و دیگر دستگاه ها
* اضافه شدن برنامه های Google Movies و Google Books
* بهینه سازی آن برای صفحات نمایش عریض

**اندروید نسخه 4.0 یا Ice Cream Sandwich :**

اندروید نسخه 4.0 یا بستنی حصیری در اکتبر 2011 معرفی شد . در این نسخه تغییرات عمده ای در سیستم عامل نسبت به نسخه های پیشین به وجود آمد.رابط کاربری به طور کلی بازنویسی و دگرگون شد و همه چیز از نو بهینه سازی شده است.

١٣

نسخه 4.0 اندروید را میتوان متحد کننده ی دو نسخه ی 2.3.x و نسخه ی 3.x این سیستم عامل دانست که بدین ترتیب از این پس نسخه های سیستم عامل اندروید ویژه گوشی های هوشمند و تبلت ها یکسان خواهد بود.

ضمن اینکه ، امکانات جدید زیادی به اندروید چهارم اضافه شد.قابلیت های تازه و تغییرات شامل :

* امکان استفاده از دکمه های مجازی در رابط های کاربری به جای استفاده از دکمه های فیزیکی در پایین گوشی.
* قرار دادن ویجت ها در Tab هایی مشابه با لیست برنامه ها
* ساخته شدن پوشه ها به آسانی و با کشیدن و انداختن
* نرم افزار جدید برای تلفن
* امکان زوم در تقویم
* اضافه کردن امکان جستجوی آفلاین در میان رایانامه ها در Gmail و خطی کردن پیش نمایش رایانامه ها
* امکان گرفتن Screenshot با نگه داشتن دکمه ی پاور و دکمه ی صدا
* بهبود سیستم غلط یابی صفحه کلید
* دسترسی مستقیم از صفحه اصلی گوشی به برنامه ها
* بهبود Copy و Paste کردن
* بهبود سیستم تشخیص صدا
* خارج کردن سیستم از حالت قفل با سیستم تشخیص چهره
* پشتیبانی مرورگر جدید از Tab که میتواند تا 16 تب همزمان را پشتیبانی کند.
* سیستم Font جدید که گوگل آن را "روبوتو" نامیده است که برای صفحه نمایش با رزولشن بالا بهینه سازی شده است.
* امکانات نظارت و مدیریت بر میزان مصرف دیتا و مشخص کردن سقف مصرف دیتا
* امکان توقف برنامه هایی که در پس زمینه از اینترنت استفاده کردند.
* بهبود برنامه دوربین ، رساندن تاخیر Shooter به صفر و امکان Zoom هنگام فیلمبرداری
* برنامه ویرایش عکس
* گالری مدیریت تصاویری جدید بر اساس موقعیت و افراد
* برنامه شبکه های اجتماعی به نام People ادغام شده با Google+
* اندروید Beam امکانی است که با کمک NFC اجازه میدهد اطلاعاتی مانند وب سایت ها ، دفترچه تماس ، آدرس ،فیلم و ... را سریعا به فرد دیگری انتقال داد.

1. Drag & Drop

١٤

* موتور قدرتمند ورود صوتی متن ( با استفاده از این موتور کاربران حتی میتوانند هنگام صحبت مکث نموده و دوباره صحبت خود را از سر گیرند)
* ارائه صفحه کلید فارسی

**-24--2 پراکندگی نسخه‌های اندروید:**

طبق آماری که شرکت گوگل در آپریل ۲۰۱۴ منتشر کرده، میزان پراکندگی نسخ اندروید بدین صورت می‌باشد:

فرویو (نسخه ۲٫۲): ۰٫۷ درصد

جینجری برد (نسخه ۲٫۳): ۱۳٫۶ درصد

ایس کریم سندویچ (نسخه ۴٫۰): ۱۰٫۶ درصد

جلی بین (نسخه ۴٫۱ و ۴٫۲ و ۴٫۳): ۵۴٫۲ درصد

کیت کت (نسخه ۴٫۴): ۲۰٫۹ درصد

**Sandbox 2-5 :**

کلیه ی نرم افزار‌های آندروید در فضای ایزوله ای از سیستم عامل به نام Sandbox  اجرا میشوند، فضایی که به دیگر قسمتها و منابع سیستم عامل دسترسی‌ ندارد .هنگامی که کاربر قصد دریافت یک نرم افزار از Android Market‌ را دارد کلیه بخش هایی که آن نرم افزار قصد دسترسی‌ به آن را دارد برای کاربر به نمایش در می‌‌آید. به عنوان مثال یک بازی‌ ممکن است بخواهد قابلیت لرزش گوشی را فعال کند اما نباید به پیغام‌های ذخیره شده بر روی گوشی دسترسی‌ پیدا کند.

علی‌ رغم توضیحات داده شده ، باز هم نرم افزار‌های جاسوسی و دزدی اطلاعات بر روی Android Market‌ دیده شده است. به گزارش Kaspersky Lab در اگوست سال ۲۰۱۰، بر روی Android Market‌ ، یک تروجان برای دزدی SMS‌ها منتشر شد که تعدادی از دستگاه‌ها را نیز آلوده ی خود کرد. گوگل خیلی‌ سریع به این موضوع واکنش نشان داد و مسئولیت غیر فعال کردن این گونه App ها را از Android Market‌ها را بر عهد گرفت. با این وجود نرم افزار‌های آنتی ویروس بسیاری نظیر AVG Kaspersky، F-Secureو McAfee برای آندروید منتشر شده اند تا امنیت اطلاعات کاربران را تامین کنند.

١٥

**-2-6آندروید مارکت:**

آندروید مارکت سرویس فروش نرم‌افزار‌های کاربردی برای گوشی‌های آندروید است. یک برنامه کاربردی ویژه آندروید مارکت به صورت از پیش بارگذاری شده بر روی گوشی‌های آندروید نصب گردیده و به کاربران امکان می‌دهد نرم‌افزار‌های مورد نیاز خود را خریداری و دانلود کنند. البته تمامی نرم‌افزار‌های موجود در آندروید مارکت فروشی نیستند بلکه بیش از نیمی از نرم‌افزارهای موجود در آندروید مارکت به صورت رایگان عرضه می‌شوند و از این نظر آندروید بیشترین درصد نرم‌افزارهای رایگان را در بین تمامی سیستم‌های عامل تلفن‌های همراه هوشمند در اختیار کاربران قرار می‌دهد.

هر برنامه‌نویس با ثبت نام، امکان فروش برنامه‌های خود در آندروید مارکت را دارد. ۷۰ درصد از مبلغ فروش برنامه‌های کاربردی به برنامه‌نویس تعلق می‌گیرد و ۳۰ درصد مابقی بین اپراتورها توزیع می‌شود. بر اساس سیاست‌های گوگل در حال حاضر تمامی برنامه نویسان عضو پروژه آندروید از سراسر جهان می‌توانند برنامه‌های کاربردی رایگان خود را از طریق آندروید مارکت در ۴۶ کشور عرضه کنند. برای اینکار کافی است برنامه نویسان فرمی مختصر را تکمیل کرده و البته ۲۵ دلار حق عضویت هم به گوگل بپردازند.

ولی تنها برنامه‌نویسان ساکن در نه کشور اتریش، فرانسه، آلمان، ایتالیا، ژاپن، هلند، اسپانیا، انگلستان و ایالات متحده آمریکا می‌توانند برنامه‌های خود را برای فروش در ۱۳ کشور استرالیا، اتریش، کانادا، فرانسه، آلمان، ایتالیا، ژاپن، هلند، نیوزلند، اسپانیا، سوییس، انگلستان و ایالات متحده آمریکا در معرض بازدید خریداران قرار دهند و سایر کشورها امکان مشاهده و خرید برنامه‌های غیر رایگان را ندارند.

نکته جالب توجه در این زمینه عدم حضور حتی یک کشور از منطقه خاورمیانه و شمال آفریقا (به غیر از اسرائیل) در لیست چهل و شش کشوری است که امکان دسترسی به آندروید مارکت را دارند! این در حالی است که گوشی‌های مجهز به آندروید از سوی اغلب شرکت‌های بزرگ از جمله موتورولا، HTC، سونی اریکسون و ال جی

١٦

در این منطقه مدت‌ها است که به بازار عرضه شده‌اند.

تعداد برنامه‌های کاربردی موجود در آندروید مارکت تاکنون نزدیک به ۵۰۰۰۰ بوده است این در حالی است که برنامه‌های کاربردی موجود در iTune برای گوشی آیفون به ۱۷۵۰۰۰ برنامه بالغ می‌شود و از این منظر به نظر می‌رسد که آندروید راهی دراز برای سبقت گرفتن از آیفون در پیش دارد. ولی در هر حال باید این نکته را هم در نظر داشت که هرچند تعداد برنامه‌های کاربردی نشانگر اقبال برنامه‌نویسان به پلت فرم موردنظر است، ولی تعداد بسیار زیاد برنامه‌ها برای کاربران همیشه هم خوب نیست.

چرا که آنان را مجبور می‌سازد تا جستجویی سخت و طاقت فرسا را برای دستیابی به برنامه مورد علاقه خود تجربه کنند. گوگل کوشیده است دسترسی به برنامه‌های کاربردی آندروید را به شدت محدود ساخته و تمامی دسترسی کاربران را از طریق نرم‌افزار آندروید مارکت نصب شده بر روی گوشی کاربران کانالیزه نماید. گوگل تا بدانجا پیش رفته است که حتی دسترسی به اطلاعات و امکان جست‌وجوی برنامه‌های کاربردی آندروید از وب سایت رسمی آندروید مارکت از طریق کامپیوترهای شخصی، امکانپذیر نیست.

از طرف دیگر عدم امکان نصب نرم‌افزارهای دانلود شده بر روی کارت حافظه محدودیت دیگری است که کاربران برای دانلود برنامه‌ها از هر جای دیگری به غیر از آندروید مارکت با آن مواجه هستند. اما هنوز هم راه‌هایی برای دور زدن گوگل برای دستیابی به برنامه‌های کاربردی وجود دارد. بسیاری از برنامه‌نویسان از جمله شرکت‌های تولید برنامه‌های کاربردی نسخه قابل نصب برنامه‌های خود را علاوه بر آندروید مارکت از طریق وب سایت‌های خود نیز در اختیار کاربران قرار می‌دهند.

علاوه بر این برخی وب‌سایت‌ها اقدام به جمع‌آوری و در اختیار قراردادن برنامه‌های کاربردی پرطرفدار آندروید نموده‌اند. نمونه‌ای از این سایت‌ها Androlib و Androidzoom هستند. همچنین با نصب یک نرم‌افزار بر روی گوشی خود امکان خواهید یافت برنامه‌های دانلود شده بر روی کارت حافظه خود را بر روی گوشی نصب نمایید.

١٧

اگرچه با استفاده از راه‌هایی که گفته شد امکان نصب برخی نرم‌افزارها بر روی گوشی خود را خواهید داشت، ولی باید اذعان کرد که اغلب برنامه‌های آندروید به ویژه برنامه‌های اصلی آن که توسط خود گوگل طراحی شده‌اند، مانند نقشه‌های گوگل یا برنامه گاگلز تنها از طریق آندروید مارکت قابل دسترسی هستند. پس اگر می خواهید به برنامه های کاربردی اصلی آندروید دسترسی داشته باشید، باید از آندروید مارکت نصب شده بر روی گوشی خود استفاده کنید.

**-2-7آندروید و گوگل تی وی:**

خب اگر فکر کرده‌اید که کار شما با آندروید تمام شده است سخت در اشتباه هستید. اگر کار شما هم با آندروید تمام شده باشد، کار آندروید با شما تمام نشده است. بله این آدم آهنی سبز رنگ پس از رسوخ در تلفن‌های همراه شما قصد دارد وارد تلویزیون‌های شما هم بشود. به چشمان خود شک نکنید. درست خوانده‌اید آندروید به زودی در تلویزیون‌های شما نیز خواهد بود. در همایش Google I/O در ماه مه ۲۰۱۰ شرکت‌های گوگل، سونی، اینتل، لاجیتک، بست بای، ادوبی و دیش نتورک از عرضه تلویزیون‌های مبتنی بر آندروید خبر دادند.

تلویزیون‌هایی که به طور بی‌سیم به اینترنت متصل می‌شوند و علاوه بر اینکه امکان اتصال به شبکه‌های آنلاین پخش فیلم را دارند، از برنامه‌های کاربردی که برای نصب بر روی این تلویزیون‌ها تهیه می‌شوند نیز بهره خواهند برد.

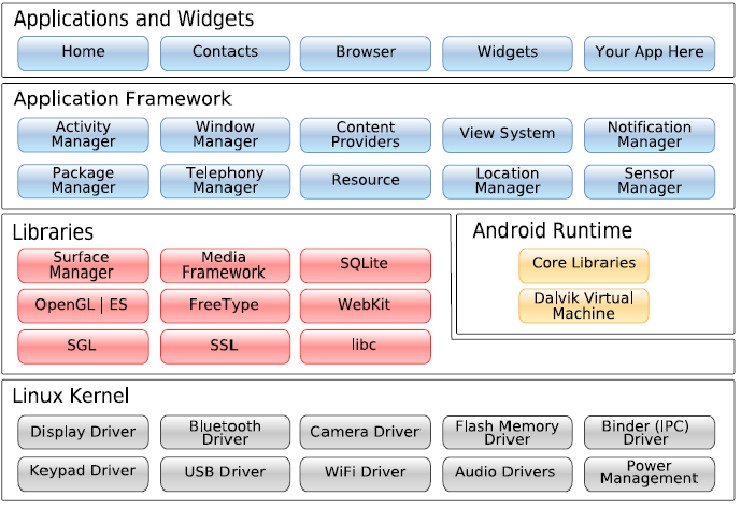
١٨

**-2-8-معماری آندروید :**

این سیستم عامل براساس هسته سیستم عامل لینوکس توصعه یافته است. درواقع مدیریت و بهینه سازی حافظه،ارتباط با سخت افزار و سرویس های سیستم و مدیریت منابع دستگاه موبایل را هسته لینوکس انجام میدهد و آندروید فقط لایه ایست که ارتباط میان کاربر و سیستم عامل را برقرار میکند.این لایه نرم افزاری به وسیله جاوا پیاده سازی شده است. سیستم عامل آندروید از نظر معماری یک پشته نرم افزاری به حساب می آید ، به این معنی که مجموعه ای از برنامه های کوچک متصل به هم است که همگی به صورت یک سیستم عامل واحد کار میکنند.

**نمودار معماری اندروید:**

تصویر زیر لایه ها و بخش های اصلی سیستم عامل آندروید را نشان میدهد:



شکل٢-معماری سیستم عامل اندروید

١٩

**-2-8-1تشریح لایه های مختلف معماری:**

**Linux Layer:**

در پایین ترین بخش معماری اندروید ، هسته1 این سیستم عامل قرار دارد.Kernel هسته مرکزی سیستم عامل و ابتدایی ترین بخش آن را تشکیل میدهد.گوگل از نسخه 2.6 لینوکس برای طراحی هسته اندروید استفاده کرد که برای انجام سرویس های اصلی شامل برنامه های مدیریت حافظه ، شبکه ، ایمنی ، پشته و Driver های سیستم میشود.این هسته همچنین مانند یک لایه انتزاعی2 مابین سخت افزار و سایر نرم افزارها عمل میکند.

**Libraries:**

بخش دیگر پشته اندروید ، کتابخانه های سیستم عامل است.این بخش شامل دستورالعمل های مختلفی است که به دستگاه دستور میدهد با داده های مختلف چطور رفتار کند.برای مثال کتابخانه Media Framework شامل اطلاعات اجرای انواع فرمت های فایل های عکس ، موزیک و فیلم است.

**Android Runtime:**

در سطح Libraries از پشته اندروید ، Android Runtime قرار دارد که شامل کتابخانه های جاوا است که در ساخت برنامه های اندروید به کار میرود و برای اجرای آن ها کاملا حیاتی میباشد.

اندروید شامل مجموعه ای از کتابخانه های اصلی است که اکثر عملکردهای قابل دسترس را با استفاده از زبان جاوا ممکن میسازد.هر برنامه کاربردی اندروید در فرایند مخصوص به خودش اجرا میشود و دسترسی مخصوص به خود در ارتباط با ماشین مجازی دالویک دارد.

بخش دیگر پشته Runtime ، Davlik Virtual Machine است.یکی از مزایای استفاده از Virtual Machine در سیستم عامل اندروید این است که هیچ برنامه ای به منابع دیگر وابسته نیست و اگر یکی از برنامه ها از کار بیفتد ، کارکرد برنامه های دیگر تحت تاثیر قرار نمیگیرد.این امر مدیریت حافظه نیز بسیار ساده میسازد.

1.Kernel

2. Abstarction Layer

٢٠

**Application Framework:**

Application Framework امکان استفاده مجدد و جایگزینی اجزاها را فراهم میکند. با فراهم آوردن Open Development Platform ، اندروید برنامه سازان را قادر کرده است تا برنامه های کاربردی خلاقانه و غنی بسازند.معماری برنامه های کاربردی به منظور ساده سازی استفاده مجدد از اجزاها طراحی شده است. توسعه دهندگان آزادی کامل دارند تا از ویژگی هایی مانند دسترسی به سخت افزار ، دسترسی به اطلاعات محلی(موقعیت جغرافیایی) ، اجرای سرویس های پس زمینه ، تنظیم زنگ ساعت ، اضافه کردن اطلاعیه ها به نوار وضعیت وبسیاری دیگر در برنامه هایی که میسازند ، استفاده کنند.هر برنامه ای میتواند قابلیت های خود را در اختیار دیگر برنامه ها قرار دهد و همچنین از قابلیت های دیگر برنامه ها استفاده کند(البته به محدودیت های امنیتی هم بستگی دارد)

برنامه های زیربنایی ، مجموعه ای از سرویس ها و سیستم های زیرند:

* مجموعه View System ها که برای ساخت برنامه های کاربردی استفاده میشوند ، مانند متن ، کادر متنی ، کلیدها و . . .
* Content Providers که برنامه ها را قادر میسازند تا به اطلاعات برنامه های دیگر مانند لیست تماس ، دسترسی پیدا کنند یا حتی اجازه دسترسی به اطلاعات خود را به برنامه های دیگر دهند.
* Notification Manager که از این طریق برنامه ها را قادر میکند تا هشدارهای خود را نوار وضعیت نشان دهند.
* Resource Manager اجازه دسترسی به منابعی که کد-برنامه نیستند را فراهم میکند مانند دسترسی به رشته ها ی محلی (Localized Strings) ، تصاویر و فایل های مربوط به طرح برنامه
* Activity Manager که مدیریت چرخه زندگی (Lifecycle) برنامه ها را در دست دارد و به نحوه اجرای برنامه ها نظارت میکند.

**Applications and Widgets:**

در بالاترین سطح پشته اندروید ، چارچوب نرم افزارها قرار دارد.این لایه شامل برنامه های کاربردی سیستم عامل که به همراه سیستم عامل ارائه شده نظیر برقراری تماس ، استفاده از دوربین و ذخیره شماره تلفن ، نقشه ، ایمیل و. . . میباشد. تمام این برنامه ها با استفاده از زبان برنامه نویسی Java نوشته شده اند.

٢١

**2-9-رابطه اندروید و Java :**

نرم افزارهای جانبی اندرویدی با استفاده از زبان جاوا نوشته میشود و برای ارتباط با لایه های زیرین سیستم عامل میتوانند از کتابخانه های جاوایی اندروید استفاده کنند. بخش رابط کاربری سیستم عامل اندروید با زبان جاوا

نوشته شده است. اما این سیستم عامل ، ماشین مجازی جاوا1 ندارد. برای اجرای برنامه های جاوایی روی این سیستم عامل ، کدهای جاوا به کدهای دالویک تبدیل میشوند و سپس روی Dalvik Virtual Machine اجرا میشوند.

**2-10- ماشین مجازی Dalvik** 2**:**

Dalvik یک ماشین مجازی جاوایی است که برای سیستم عامل اندروید بهینه شده است تا هم RAM و هم CPU و هم باطری کمتری مصرف کند. ماشین مجازی دالویک برای اجرای قابلیت های اساسی مانند مدیریت حافظه کم ، متکی بر هسته لینوکس است و وظیفه بهینه سازی کدها را برای اجرا بر روی موبایل را دارد.

این ماشین به گونه ای ساخته شده است که هر دستگاهی میتواند چندین ماشین مجازی را به طور همزمان اجرا کند. هر ماشین مجازی دالویک فایل ها را به فرمت (.dex) اجرا میکند که این کار باعث بهینه سازی در دستگاه هایی که حافظه پایینی دارند ، میشود . این ماشین مجازی مبتنی بر رجیسترها و کلاس هایی که توسط کامپایلر Java ساخته شده است را اجرا میکند.البته همانطور که قبلا نیز گفته شد هم اکنون ران تایم جدید اندروید به نام "آرت" برای اندرویدهای 4.4 یا همان نسخه ی "کیت کت" به بعد قابل دسترسی است.

1. Java Virtual Machine

2.Dalvik Virtual Machine

٢٢

**2-11-بازی های آندروید :**

آندروید یک سیستم عامل متن باز است که در حال حاضر بیش از ۳۰ شرکت مختلف به آن پیوسته اند. همانطور که می دانید بازی هایی که برای سیستم عامل آندروید عرضه می شوند یکی از زیباترین و گرافیکی ترین بازی های جهان می باشند. این بازی ها همچنین دارای گیم پلی روان و بسیار جذابی نیز هستند که می توانند کاربر را به سمت خود سوق دهند. بازی های اندروید با فرمت apk منتشر می شوند. برخی از بازی های آندروید دارای فایل دیتا می باشند. فایل های دیتا در اصل اطلاعات کامل کننده فایل apk می باشند که بعد از نصب کردن apk شما باید فایل دیتا را نیز در مسیری که معمولا سایت منتشر کننده آن بازی می گوید کپی کنید تا بتوانید آن بازی آندروید را شروع کنید. بازی های آندروید دارای حجم های مختلفی می باشند. برخی از این بازی ها از یک مگابایت و حتی کمتر و برخی بازی های اندروید دارای دیتای چند گیگابایتی می باشند. طبعا هرچه دیتای یک بازی بیشتر باشد، گرافیک آن نیز بیشتر خواهد بود و کاربر گیم پلی جذاب تری تجربه خواهد کرد. امروزه یکی از بزرگترین کمپانی هایی که در زمینه تولید بازی های آندروید فعالیت دارد، کمپانی گیم لافت می باشد.

**2-12-رده های سنی بازی آندروید:**

بازی های آندروید مانند بازی سایر سیستم عامل ها و کنسول ها امروزه برای رده بندی های سنی مختلفی ارائه میشوند. اما نکته ای که در مورد بازی آندروید جالب است بدانید این است بازی های آندروید معمولا به گونه ای ساخته می شوند که اکثر رده بندی های سنی بتوانند به سادگی از آن استفاده کنند و از آن لذت ببرند. اما در کل بازی های آندروید را می توان برای سه سطح کودک، نوجوان و بزرگ سال در نظر گرفت. بازی های کودکان و نوجوانان به طبع دارای حجم کمتری نسبت به بازی هایی که برای بزرگسالان عرضه میشوند، می باشند. معمولا کسانی که منتشر کننده بازی های آندروید هستند در کنار عرضه آن باید رده بندی آن را نیز نمایش دهند. در سایت های مختلف ارائه دهنده ی اپلکیشن های آندرویدی دسته بندی های اختصاصی برای بازی های کودکان را میشود پیدا کرد، که کودکان می توانند بازی های مناسب خود را از طریق این دسته بندی انتخاب کنند و از آنها لذت ببرند.

**2-13-ورژن های مختلف بازی های آندرویدی :**

امروز شاهد انتشار نسخه هایی مختلف سیستم عامل آندروید هستیم. سیستم عامل آندرویدی که امروزه قدیمی

٢٣

ترین آنها شناخته می شود، ورژن ۱.۵ و بالاترین سیستم عامل آندروید فعلی ۵ می باشد. بازی هایی که برای سیستم عامل بالاتر عرضه می شوند دارای گرافیک و گیم پلی جذاب تری نسبت به بازی هایی که هستند که برای سیستم های پایین تر عرضه میشوند. اما نمی توان از این نکته غافل بود که گاها ممکن است بازی برای سیستم عامل آندروید ورژن پایین عرضه شود که دارای گرافیک و گیم پلی فوق العاده ای باشد. شما در سایت های دانلود برنامه های آندرویدی دسته بندی های گوناگونی را میبینید که از بین آن ها میتوانید به سادگی تنها بازی ها و برنامه هایی را دانلود کنید که مناسب ورژن آندروید آن گوشی شما باشد. این کار سبب می شود زمان کمتری برای جستوجوی نرم افزار و بازی آندروید مورد علاقه خود صرف کنید.

**2-14-قابلیت های بازی های آندرویدی:**

یکی از قابلیت های بارز بازی های آندروید که آن را نسبت به بازی سایر سیستم عامل ها متمایز می کند داشتن گرافیکی بسیار قوی و خیره کننده می باشد. البته نمی توان از گیم پلی عالی بازی های آندروید نیز چشم پوشی کرد. اگر بخواهیم در بین سیستم عامل های مختلف یک بررسی کامل داشته باشیم می توان گفت که بازی های عرضه شده برای سیستم عامل آندروید دارای بالاترین قدرت از لحاظ گرافیک و گیم پلی می باشد. گرافیک خیره کننده بازی های آندروید یکی از مواردی است که توانسته امروزه میلیون ها کاربر مختلف را به سمت خود بکشاند و آنها را به خود معتاد کند.

**2-15-معایب بازی های آندرویدی:**

آن قدر دنیای آندروید و بازی های آندروید بزرگ و قدرتمند است که نمی توان عیبی برای آن برشمرد. اما همانطور که می دانید هیچ چیز بدون مشکلی وجود ندارد. یکی از معایبی که در بازی های آندروید به چشم می خورد این است که برخی از بازی های آندروید دارای حجم بالایی هستند و این می تواند برای کاربرانی که حافظه گوشی آنها کم می باشد، دردسر ساز باشد. همچنین بالا بودن حجم بازی های آندروید برای آن دسته از کاربرانی که بازی های خود را از طریق اینترنت دانلود می کنند نیز می تواند مشکل باشد، چرا که دانلود کردن بازی که دارای چندین گیگ حجم می باشد که می تواند ساعت ها زمان ببرد.

24

***فصل سوم***

**جاوا**

٢٥

جاوا1 یک زبان برنامه‌نویسی شئ‌گراست که برای اولین بار توسط جیمز گاسلینگ در شرکت سان مایکروسیستمز ایجاد شد و در سال ۱۹۹۵ به عنوان بخشی از سکوی جاوا منتشر شد. زبان جاوا شبیه به ++C است اما مدل شیءگرایی آسان‌تری دارد و از قابلیت‌های سطح پایین کمتری پشتیبانی می‌کند. یکی از قابلیت‌های اصلی جاوا این است که مدیریت حافظه را بطور خودکار انجام می‌دهد. ضریب اطمینان عملکرد برنامه‌های نوشته‌شده به این زبان بالا است و وابسته به سیستم‌عامل خاصی نیست، به عبارت دیگر می‌توان آن را روی هر رایانه با هر نوع سیستم‌عاملی اجرا کرد. برنامه‌های جاوا به صورت کدهای بیتی همگردانی (کامپایل) می‌شوند. که مانند کد ماشین هستند و به ویژه وابسته به سیستم‌عامل خاصی نیستند.

نام جاوا از Just Another Vague Acronym گرفته شده است و دلیل آن این بود که تیم برنامه نویسان جاوا به قهوه خیلی علاقه داشتند. این موضوع در لوگوی جاوا هم مشهود است!!



شکل٣-لوگوی جاوا

**3-1-تاریخچه :**

در مقایسه با زبان‌های دیگر، همچون C++ یا بیسیک یا فورترن، جاوا زبان نسبتاً جدیدتری است. شرکت سان مایکروسیستم٢ در سال ۱۹۹۱ یک پروژه تحقیقاتی به نام گرین٣ را آغاز کرد. هدف این پروژه ایجاد زبانی جدید شبیه به C++ بود که نویسنده اصلی آن، جیمز گاسلینگ، آن را بلوط نامید. اما بعدها به دلیل برخی مشکلات حقوقی از میان لیستی از کلمات تصادفی نام آن به جاوا تغییر کرد.

1.Java

2. Sun Microsystems

3. Green

٢٦

پروژه گرین به دلیل مشکلات بازاریابی در شرف لغو شدن بود تا اینکه گسترش وب در سال ۱۹۹۳ باعث نمایش توانایی‌های وافر جاوا در این عرصه گشت. اینگونه بود که شرکت سان مایکروسیستمز در مه ۱۹۹۵ جاوا را رسماً به بازار عرضه کرد.

جاوا یک زبان برنامه‌نویسی است که در آغاز توسط شرکت سان مایکروسیستمز ایجاد شده‌است و در سال ۱۹۹۵ به عنوان بخش اصلی سکوی جاوا منتشر شد. این زبان قسمت‌های بسیاری از گرامر خود را از C و C++ گرفته اما دارای مدل شی‌گرایی ساده‌ای است و امکانات سطح پایین کمی دارد. کاربرد جاوا در کامپایل به صورت بایت کد است که صرف نظر از معماری و خصوصیات آن کامپیوتر، قابلیت اجرا روی تمامی ماشین‌های شبیه‌سازی جاوا را داشته باشد. اجزای اصلی کامپایلرهای جاوا، ماشین‌های پیاده‌سازی و کتابخانه‌های آن توسط این شرکت از سال ۱۹۹۵ منتشر شد. در مه ۱۹۹۷ این شرکت، نرم‌افزار رایگان این زبان را فراهم کرد. دیگران هم کاربردهای دیگری از این زبان را منتشر کردند مثل کامپایلر GNU برای جاوا.

مرورگرهای اصلی وب، به هم پیوستند تا به طور مطمئن جاوا اپلت را بدون صفحات وب اجرا کنند و به این صورت جاوا خیلی زود معروف و محبوب شد. با پیدایش java2، نسخه ی جدید توانست ترکیب‌های جدیدی را برای نوع‌های مختلف پلت فرم‌ها ایجاد کند. به عنوان مثال J2EE، باهدف کاربرد برای تشکیلات اقتصادی، و نسخه ی سکوی جاوا، نسخه میکرو برای تلفن همراه منتشر شد. در سال ۱۹۹۶ با هدف بازاریابی، این شرکت نسخه ی جدید J2 را با نام‌های سکوی جاوا، نسخه سازمانی، سکوی جاوا، نسخه میکرو و سکوی جاوا، نسخه استاندارد منتشر کرد. در سال ۱۹۹۷ شرکت سان میکروسیستمز، ISO/IEC JTC1 standards body و Ecma International را به فرمول جاوا تغییر داد. شرکت Sun بسیاری از کاربردهای جاوایش را بدون هیچ هزینه‌ای فراهم آورد. شرکت Sun با فروش مجوز برای بعضی از کاربردهای خاصش مثل Java Enterprise System درآمدی را بدست آورد. در ۱۳ نوامبر ۱۹۹۶ شرکت Sun نرم‌افزار جاوا را به صورت رایگان و با مجوز عمومی برای همه منتشر کرد.

۷2

**3-2- اهداف اولیه:**

* این زبان باید ساده، شی‌گرا و مشهور باشد.
* مطمئن و بدون خطا باشد.
* وابسته به معماری کامپیوتر نبوده و قابل انتقال باشد.
* باید با کارایی بالا اجرا شود.
* باید به صورت پویا و نخ‌کشی‌شده باشد

**3-3- نسخه ها :**

نسخه های جاوا و تاریخ انتشارشان به شرح زیر است :

جدول2-نسخه های جاوا

|  |  |
| --- | --- |
| نسخه | تاریخ انتشار |
| JDK 1.0 | ۲۱ ژانویهٔ ۱۹۹۶ |
| JDK 1.1 | ۱۹ فوریهٔ ۱۹۹۷ |
| J2SE 1.2 | ۸ دسامبر ۱۹۹۸ |
| J2SE 1.3 | ۸ می ۲۰۰۰ |
| J2SE 1.4 | ۶ فوریهٔ ۲۰۰۲ |
| J2SE 5.0 | ۳۰ سپتامبر ۲۰۰۴ |
| Java SE 6 | ۱۱ دسامبر ۲۰۰۶ |
| Java SE 7 | ۲۸ ژوئیه ۲۰۱۱ |
| Java SE 8 | ۱۸ مارچ ۲۰۱۴ |

۲۸

**3-4- برنامه های جاوا و اپلت ها :**

جاوا برای نوشتن انواع برنامه‌های کاربردی مناسب است. با جاوا می‌توان انواع برنامه‌های زیر را نوشت:

* برنامه های تحت وب
* برنامه نویسی سیستم های کوچک مانند تلفن های همراه و تبلت ها
* برنامه های کاربردی بزرگ**1**
* برنامه های رومیزی**2**

قابلیت خاصی در جاوا وجود دارد بنام اپلت. اپلت‌ها امکانات فراوانی برای نوشتن برنامه‌های تحت وب در اختیار برنامه‌نویسان قرار می‌دهند که دیگر زبان‌های برنامه‌نویسی فاقد آن هستند. البته وجود ماشین مجازی جاوا برای اجرای اپلت لازم است. اپلت‌ها نظیر فناوری اکتیوایکس شرکت مایکروسافت هستند که برنامه‌نویسان را قادر می‌سازد تا امکاناتی را به مرورگر کاربر بیفزایند. البته تفاوت این دو در امنیت می‌باشد به گونه‌ای که اپلت‌ها بدلیل اینکه در محیطی به نام sandbox اجرا می‌شوند امن هستند ولی اکتیوایکس‌ها فاقد چنین امنیتی هستند.

**.Net -5-3 رقیب Java ؟**

.NET فرمی است که به وسیله مایکروسافت 3 برای ساخت سایت ها با قابلیت های زیاد و متفاوت ایجاد شده است به طوری که قسمتی از یک پروژه را می توان با C++ و بخش دیگر را با برنامه ای دیگر نوشت و در نهایت کل آن توسط یک واسطه به نام MSIL ترجمه شده و در محیط .NET قابل اجرا است.

J2EE به وسیله شرکت SUN و با همکاری شرکت های IBM و HP تهیه شده است. در این قالب تنها زبانی که مورد استفاده قرار می گیرد جاوا است، و بر خلاف .NET که فقط روی محیط های سازگار با ویندوز عمل می کند، مستقل از سخت افزار است.

1. Enterprise

2. Desktop

3. MicroSoft

٢٩

**3-6- بین .NET و JAVA کدام را انتخاب کنیم؟**

- قابلیت انتقال برنامه ها روی سیستم های مختلف و شعار اصلی جاوا یعنی: "یک بار بنویس و هر جا استفاده کن" (Writeonce-Run anywhere) موضوعی بسیار حایض اهمیت است که بسته به نیاز شما و نوع برنامه شما می تواند خیلی مهم باشد.

- در محیط هایی که چند نوع سیستم سخت افزاری وجود دارد قطعا برنامه های تحت جاوا مناسب تر است ولی اگر در محیط مورد نظر شما از سیستم عامل ویندوز استفاده می شود استفاده از .NET هزینه کمتری در برخواهد داشت.

.NET ویژگی هایی دارد که با آن می توان سایت های جذاب و زیباتری ساخت اما در سایت های پیچیده و بزرگ و جایی که کارایی مهمتر از ظاهر است بهتر است از جاوا استفاده کنیم.

- تصور کنید نیاز به برنامه ای دارید که در آن چیزهایی به صورت دینامیک ایجاد می شوند و پس از طی یک سری عملیات بر روی آنها از بین می روند، در این نوع برنامه ها تعقیب اشیایی که در برنامه ساخته می شوند، تخصیص و مدیریت حافظه تماما بر عهده برنامه نویس است. اما جاوا دارای یک سیستم خودکار مدیریت و پاکسازی حافظه است که بسیاری از مشکلات را آسان کرده است.

هر تکنولوژی که ایجاد می شود؛ هر برنامه ای که مورد توجه عموم قرار می گیرد قطعا مورد توجه نفوذگران هم واقع می شود، هر سیستمی که بخواهد بماند و پیشرفت کند نیاز به ایمنی دارد و اینجاست که سدهای حفاظتی جاوا و مدل های امنیتی آن که حاصل تلاش زبردست ترین برنامه نویسان دنیاست مورد توجه قرار می گیرد. مدل چهار لایه امنیتی جاوا جلوی خیلی از اقدامات خرابکارانه و نفوذی را می گیرد، ولی هیچ سدی صددرصد غیر قابل نفوذ نیست!!

در هر حال انتخاب بین این دو نیاز به بررسی دقیق، آینده نگری و مشخص شدن اهداف دراز مدت و میزان سرمایه گذاری شما دارد زیرا هزینه هایی که یک برنامه جاوا دربر دارد در اکثر موارد چندین برابر یک برنامه .NET خواهد بود.

٣٠

بطور خلاصه بدلایل زیر جاوا مورد استقبال تر است:

**سیستم‌عامل:** هر چقدر زبانهای net. قوی باشند تنها بر روی پلت فرم ویندوز اجرا می‌شوند و برخی ویندوز را سیستم‌عامل غیر قابل اعتمادی در برنامه نویسی Enterprise می‌دانند.ولی جاوا از این نظر انتخاب خوبی است

**قابلیت حمل:** جاوا بر روی سکوهای رایانش گوناگونی قابل اجرا است، از ATM و ماشین رختشویی گرفته تا سرورهای سولاریس با قابلیت پشتیبانی از 1024 cpu برای پردازش.

**جاوا بیشتر از یک زبان است**: جاوا فقط یک زبان نیست و انجمن‌هایی متشکل از بزرگان صنایع و برنامه‌نویسان زیادی مشغول به توسعه و ایجاد استانداردهای جدید و به روز هستند.

**3-7- خط مشی جاوا:**

یکی از ویژگی‌های جاوا قابل حمل بودن آن است. یعنی برنامه ی نوشته شده به زبان جاوا باید به طور مشابهی در کامپیوترهای مختلف با سخت‌افزارهای متفاوت اجرا شود. و باید این توانایی را داشته باشد که برنامه یک بار نوشته شود، یک بار کامپایل شود و در همه کامپیوترها اجرا گردد. به این صورت که کد کامپایل شده ی جاوا را ذخیره می‌کند، اما نه به‌صورت کد ماشین بلکه به‌صورت بایت‌کد جاوا. دستورالعمل‌ها شبیه کد ماشین هستند، اما با ماشین‌های مجازی که به طور خاص برای سخت‌افزارهای مختلف نوشته شده‌اند، اجرا می‌شوند. در نهایت کاربر از سکوی جاوا نصب شده روی ماشین خود یا مرورگر وب استفاده می‌کند. کتابخانه‌های استاندارد یک راه عمومی برای دسترسی به ویژگی‌های خاص فراهم می‌کنند. مانند گرافیک، نخ‌کشی و شبکه. در بعضی از نسخه‌های ماشین مجازی جاوا، بایت‌کدها می‌توانند قبل و در زمان اجرای برنامه به کدهای محلی کامپایل شوند. فایده ی اصلی استفاده از بایت‌کد، قسمت کردن است. اما ترجمه ی کلی یعنی برنامه‌های ترجمه شده تقریباً همیشه کندتر از برنامه‌های کامپایل شده ی محلی اجرا می‌شوند. این شکاف می‌تواند با چند تکنیک خوش‌بینانه که در کاربردهای JVM قبلی معرفی شد، کم شود. یکی از این تکنیک‌ها JIT است که بایت‌کد جاوا را به کد محلی ترجمه کرده و سپس آن را پنهان می‌کند. در نتیجه برنامه خیلی سریع‌تر نسبت به کدهای ترجمه شده ی خالص شروع و اجرا می‌شود. بیشتر VMهای پیشرفته، به‌صورت کامپایل مجدد پویا، در آنالیز VM، رفتار برنامه ی اجرا شده و کامپایل مجدد انتخاب شده و بهینه‌سازی قسمت‌های برنامه، استفاده می‌شوند. کامپایل مجدد پویا می‌تواند کامپایل ایستا را بهینه‌سازی کند. زیرا می‌تواند قسمت hot spot برنامه و گاهی حلقه‌های

٣١

داخلی که ممکن است زمان اجرای برنامه را افزایش دهند را تشخیص دهد. کامپایل JIT و کامپایل مجدد پویا به برنامه‌های جاوا اجازه می‌دهد که سرعت اجرای کدهای محلی بدون از دست دادن قابلیت انتقال افزایش پیدا کند.

تکنیک بعدی به عنوان کامپایل ایستا شناخته شده‌است. که کامپایل مستقیم به کدهای محلی است مانند بسیاری از کامپایلرهای قدیمی. کامپایلر ایستای جاوا، بایت‌کدها را به کدهای شی محلی ترجمه می‌کند.

کارایی جاوا نسبت به نسخه‌های اولیه بیشتر شد. در تعدادی از تست‌ها نشان داده شد که کارایی کامپایلرJIT کاملا مشابه کامپایلر محلی شد. عملکرد کامپایلرها لزوماً کارایی کدهای کامپایل شده را نشان نمی‌دهند. یکی از پیشرفت‌های بی نظیر در در زمان اجرای ماشین این بود که خطاها ماشین را دچار اشکال نمی‌کردند. علاوه بر این در زمان اجرای ماشینی مانند جاوا وسایلی وجود دارد که به زمان اجرای ماشین متصل شده و هر زمانی که یک استثنا رخ می‌دهد، اطلاعات اشکال زدایی که در حافظه وجود دارد، ثبت می‌کنند.

**3-8- پیاده سازی:**

شرکت سان میکروسیستم مجوز رسمی برای پلت فرم استاندارد جاوا را به مایکروسافت ویندوز, لینوکس، و سولاریس (سیستم‌عامل). داده‌است. همچنین محیط‌های دیگری برای دیگر پلت فرم‌ها فراهم آورده‌است. علامت تجاری مجوز شرکت سان میکروسیستم طوری بود که با همه ی پیاده‌سازی‌ها سازگار باشد. به علت اختلاف قانونی که با ماکروسافت پیدا کرد، زمانی که شرکت سان ادعا کرد که پیاده‌سازی ماکروسافت از RMI یا JNI پشتیبانی نکرده و ویژگی‌های خاصی را برای خودش اضافه کرده‌است. شرکت سان در سال ۱۹۹۷ پیگیری قانونی کرد و در سال ۲۰۰۱ در توافقی ۲۰ میلیون دلاری برنده شد. در نتیجه کمی بعدماکروسافت جاوا را به ویندوز فرستاد. در نسخه ی اخیر ویندوز، مرورگر اینترنت نمی‌تواند از جاوا پلت فرم پشتیبانی کند. شرکت سان و دیگران یک سیستم اجرای جاوای رایگان برای آنها و نسخه‌های دیگر ویندوز فراهم آوردند.

٣٢

**3-9- اداره ی خودکار حافظه :**

جاوا از حافظه ی بازیافتی خودکار برای اداره ی حافظه در چرخه ی زندگی یک شی استفاده می‌کند. برنامه‌نویس زمانی که اشیا به وجود می‌آیند، این حافظه را تعیین می‌کند. و در زمان اجرا نیز، زمانی که این اشیا در استفاده ی زیاد طولانی نباشند، برنامه نویس مسئول بازگرداندن این حافظه‌است. زمانی که مرجعی برای شی‌های باقی‌مانده نیست، شی‌های غیر قابل دسترس برای آزاد شدن به صورت خودکار توسط بازیافت حافظه، انتخاب می‌شوند. اگر برنامه‌نویس مقداری از حافظه را برای شی‌هایی که زیاد طولانی نیستند، نگه دارد، چیزهایی شبیه سوراخ حافظه اتفاق می‌افتند.

یکی از عقایدی که پشت سر مدل اداره ی حافظه ی خودکار جاوا وجود دارد، این است که برنامه‌نویس هزینه ی اجرای اداره ی دستی حافظه را نادیده می‌گیرد. در بعضی از زبان‌ها حافظه لازم برای ایجاد یک شی، به صورت ضمنی و بدون شرط، به پشته تخصیص داده می‌شود. و یا به‌طور صریح اختصاص داده شده و از heap بازگردانده می‌شود. در هر کدام از این راه‌ها، مسئولیت اداره ی اقامت حافظه با برنامه‌نویس است. اگر برنامه شی را برنگرداند، سوراخ حافظه اتفاق می‌افتد. اگر برنامه تلاش کند به حافظه‌ای را که هم‌اکنون بازگردانده شده، دستیابی پیدا کند یا برگرداند، نتیجه تعریف شده نیست و ممکن است برنامه بی‌ثبات شده و یا تخریب شود. این ممکن است با استفاده از اشاره‌گر مدتی باقی بماند، اما سرباری و پیچیدگی برنامه زیاد می‌شود. بازیافت حافظه اجازه دارد در هر زمانی اتفاق بیفتد. به‌طوری که این زمانی اتفاق می‌افتد که برنامه بی‌کار باشد. اگر حافظه ی خالی کافی برای تخصیص شی جدید در هیپ وجود نداشته باشد، ممکن است برنامه برای چند دقیقه متوقف شود. در جایی که زمان پاسخ یا اجرا مهم باشد، اداره ی حافظه و منابع اشیا استفاده می‌شوند.

جاوا از نوع اشاره‌گر ریاضی C و ++C پشتیبانی نمی‌کند. در جایی که آدرس اشیا و اعداد صحیح می‌توانند به جای هم استفاده شوند. همانند ++C و بعضی زبان‌های شی‌گرای دیگر، متغیرهای نوع‌های اولیه ی جاوا شی‌گرا نبودند. مقدار نوع‌های اولیه، مستقیماً در فیلدها ذخیره می‌شوند. در فیلدها (برای اشیا) و در پشته (برای توابع)، بیشتر از هیپ استفاده می‌شود. این یک تصمیم هوشیارانه توسط طراح جاوا برای اجرا است. به همین دلیل جاوا یک زبان شی‌گرای خالص به حساب نمی‌آید.

33

**3-10- گرامر :**

گرامر جاوا وسیع‌تر از ++C است و برخلاف ++C که ترکیبی است از ساختارها و شی‌گرایی، زبان جاوا یک زبان شی‌گرای خالص می‌باشد. فقط نوع داده ی اصلی از این قاعده مستثنی است. جاوا بسیاری از ویژگی‌ها را پشتیبانی می‌کند و از کلاس‌ها برای ساده‌تر کردن برنامه‌نویسی و کاهش خطا استفاده می‌کند.

**3-11- توزیع های جاوا :**

منظور از توزیع جاوا پیاده‌سازی‌های مختلفی است که برای کامپایلر جاوا و همچنین مجموعه کتابخانه‌های استاندارد زبان جاوا (JDK) وجود دارد. در حال حاضر چهار توزیع‌کننده ی عمده جاوا وجود دارند:

* **سان میکروسیستمز:** توزیع کننده اصلی جاوا و مبدع آن می‌باشد. در اکثر موارد هنگامی که گفته می‌شود جاوا منظور توزیع سان می‌باشد.
* **GNU Classpath:** این توزیع از سوی موسسه نرم‌افزارهای آزاد منتشر شده و تقریباً تمامی کتابخانه استاندارد زبان جاوا در آن بدون بهره‌گیری از توزیع شرکت سان از اول پیاده‌سازی شده‌است. یک کامپایلر به نام GNU Compiler for Java نیز برای کامپایل کردن کدهای جاوا توسط این موسسه ایجاد شده‌است. فلسفه انتخاب نام Classpath برای این پروژه رها کردن تکنولوژی جاوا از وابستگی به علامت تجاری جاوا است بطوریکه هیچ وابستگی یا محدودیتی برای استفاده آن از لحاظ قوانین حقوقی ایجاد نشود و از طرفی به خاطر وجود متغیر محیطی classpath در تمامی محیط‌های احرایی برنامه‌های جاوا، این نام به نوعی تکنولوژی جاوا را برای خواننده القا می‌کند. کامپایلر GNU توانایی ایجاد کد اجرایی (در مقابل بایت کد توزیع سان) را داراست. لازم به ذکر است که در حال حاضر شرکت سان تقریباً تمامی کدهای JDK را تحت مجوز نرم‌افزارهای آزاد به صورت متن باز منتشر کرده‌است و قول انتشار قسمت بسیار کوچکی از این مجموعه را که به‌دلیل استفاده از کدهای شرکت‌های ثانویه نتوانسته به صورت متن باز منتشر نماید در آینده نزدیک با بازنویسی این کدها داده‌است.
* مایکروسافت #J: این در حقیقت یک توزیع جاوا نیست. بلکه زبانی مشابه می‌باشد که توسط مایکروسافت و در چارچوب.net ارائه شده‌است. انتظار اینکه در سیستم‌عاملی غیر از ویندوز هم اجرا شود را نداشته باشید

34

**3-12- کتابخانه های جاوا :**

کتابخانه‌های جاوا که به صورت بایت کد از کد اصلی کامپایل شده‌اند، برای پشتیبانی از بعضی از کاربردهای جاوا، توسطJRE منتشر شده‌است. مثال‌هایی از این کتابخانه‌ها عبارتند از:

* کتابخانه‌های مرکزی
* کتابخانه‌هایی که برای ساختار داده کاربرد دارند. مثل لیست‌ها، درخت‌ها، مجموعه‌ها، مترجم‌ها.
* کتابخانه ی پرداز XML (تجزیه، تغییر شکل، اعتبار)
* کتابخانه‌های موضعی و بین‌المللی
* کتابخانه‌های انتگرال گیری که امکان تایپ کردن توسط سیستم‌های بیرونی را می‌دهند.
* JDBC برای دستیابی به داده‌ها
* JNDI برای مراجعه و کشف کردن
* CORBA & RMI برای توسعه ی کاربرد توزیع کردن
* کتابخانه‌های واسط کاربر
* AWT (توابع پنجره‌ای مجرد) که قسمت‌هایی از GUI را فراهم می‌کنند.
* کتابخانه‌های swing که در AWT ساخته شده‌اند اما کاربردهایی از AWT widgetry را فراهم می‌کنند.
* APL ها برای ضبط صدا، پردازش و بازنواختی
* کاربردهای وابسته ی پلت فرم ماشین‌های مجازی جاوا
* Plugins که توانایی اجرا شدن در مرورگرهای وب را به اپلت می‌دهد.
* java web start
* دادن مجوز و مستندسازی

**3-13- ویرایش ها :**

شرکت سان میکروسیستم، ۴ نوع ویرایش از کاربردهای مختلف جاوا را ارائه داده‌است:

35

1. Java card for smartcard
2. JavaME
3. JavaSE
4. JavaEE

**3-14- ایرادات مطرح شده :**

مهم‌ترین ایرادی که برنامه نویسان سایر زبان‌ها به زبان جاوا می‌گیرند سرعت اجرایی پایین جاوا در مقایسه با زبان‌ها سطح پایین‌تر مانند C++ و اسمبلی است. یک برنامه جاوا به صورت بایت کد می‌باشد و باید در ماشین مجازی جاوا اجرا گردد. به همین دلیل سرعت اجرای پایینی را در مقابل زبان‌هایی همچون C++دارد. به صورت دیگر یک برنامه C به طور متوسط تا ۱۰ برابر سریعتر از برنامه مشابه جاوا اجرا می‌گردد.

جاوا علی‌رغم شیءگرا بودن در بخشی از قسمت‌ها برخی از اصول شی‌گرایی را نادیده گرفته‌است. از جمله این قسمت‌ها قابلیت بازتابش1 می‌باشد. هدف اصلی بازتابش بررسی (مشاهده) و ایجاد تغییر در برنامه در حال اجرا است ولی این مهم با زیر پا گذاشتن بعضی اصول ممکن شده‌است. برای نمونه با استفاده از بازتابش (و در صورت داشتن مجوز لازم ضمن اجرای برنامه) می‌توان به متدهای خصوصی دیگر کلاس‌ها دسترسی داشت.

زبان جاوا در مقابل زبانی مثل C++ ساده‌تر و یادگیری آن آسانتر است. این آسانتر بودن با حذف بسیاری از موارد که باعث قدرتمندتر بودن زبان C++ بوده‌اند ایجاد شده‌است. مهم‌ترین این موارد اشاره گرها و وراثت چندگانه بوده‌اند که در زبان جاوا یافت نمی‌شوند.

از آنجایی که جاوا زبانی با عدم وابستگی به بستر می‌باشد پس استفاده از توابع سیستم‌عامل در برنامه را به طور مستقیم نمی‌پذیرد. به همین صورت نمی‌توان به طور مستقیم از واسط‌های برنامه نویسی غیر از جاوا در آن استفاده نمود.

1.Reflection

36

**3-15- پاسخ به ایرادات :**

سرعت پایین برنامه‌های جاوا در محیطی که اجرا می‌شوند ملاک کارایی نبوده زیرا در محیط وب مسئله‌ای که سرعت را کند می‌سازد، شبکه بوده و ابتدا باید سربار شبکه را از روی برنامه‌ها برداشت. از طرف دیگر در برنامه‌های رومیزی هم در JDK 5.0 و ۶٫۰، بهینه‌سازی بسیاری بوجود آمده که این مسئله باعث شده که در آخرین تست کارایی که انجام شده یک برنامه جاوا در محدودهٔ ۰٫۸ تا ۱٫۳ همان برنامه در ++C کارایی داشته باشد که ۱٫۳ آن مربوط به بخش واسط کاربری و سرعت ۰٫۸ آن مربوط به بسته تخلیه حافظه می‌شده که هیج الگوریتمی نتوانست از الگوریتم Garbage Collector جاوا پیشی بگیرد. همچنین سال ۱۹۹۹ در مقاله‌ای آقای Lutz Prechelt به این مسئله را ثابت کردند که تجربه برنامه‌نویسی که برنامه‌ای را می‌نویسد از انتخاب زبانی که برنامه برروی آن نوشته می‌شود در کارایی تأثیر بیشتری دارد و این بدان معناست که کارایی یک برنامه را برنامه‌نویس مشخص می‌کند و نه زبان برنامه‌نویسی (ایشان در همان مقاله از زبان جاوا استفاده نمودند تا ذهنیت بد را از بین ببرند)

همچنین در صنعت نرم‌افزار هزینه اصلی مربوط به ساخت نرم‌افزار است و نه تهیه سخت‌افزار برای دستیابی به سرعت بیشتر.

حذف اشاره‌گرها در جاوا به دلیل مشکلاتی بوده که آنها در طول تاریخشان بوجود آورده‌اند، اگرچه این موارد در برنامه‌های سیستمی لازم به‌نظر می‌رسد ولی در محیط‌های تحت‌وب که بستر اصلی جاوا هستند می‌توانند اثراتی به مراتب شدیدتر نسبت به آنچه در برنامه‌های سیستمی دارند داشته باشند و باعث می‌شود که توجه برنامه‌نویسان از مسائلی چون کارایی، قابلیت اطمینان و مقیاس‌پذیری برنامه به تنظیم اشاره‌گرها معطوف گردد.

وجود وراثت چندگانه در زبانی مانند C++ ، باعث ایجاد مشکلات اساسی‌ای می‌گردید که اکثر برنامه‌نویسان C++ از آن دوری می‌کرده و هنوز هم می‌کنند. ولی قابلیت چندریخته شدن یک کلاس از لحاظ شی گرایی بسیار مهم بوده و بنابراین توجیهی برای وجود وراثت چندگانه را فراهم می‌نمود.

37

در جاوا با وارد شدن مفهومی به نام واسط برنامه‌سازی 1، دیگر نیازی به وجود وراثت چندگانه احساس نشد و بنابراین از زبان جاوا حذف گردید. در حال حاضر اکثر طراحان برنامه‌ها حتی به این نتیجه رسیده‌اند که وراثت تکی هم باعث ایجاد مشکل بوده و تا آنجایی که می‌شود باید از Composition استفاده نمود و در تمامی کتاب‌های طراحی که از سال ۲۰۰۰ به این طرف چاپ شده به آن اشاره نموده‌اند.

از ابتدای بوجود آمدن جاوا، کتابخانه JNI - Java Native Interface در آن وجود داشته که قابلیت فراخوانی و دستکاری برنامه‌هایی در C++ و... را می‌داده که از نمونه‌های آن می‌توان به Jtwain که یک بسته‌ایست که از کتابخانه‌های ویندوز برای اسکن عکس استفاده می‌کند، یا SWT که یک بسته نرم‌افزاریست که از کتابخانه‌های ویندوز و لینوکس (برحسب سیستم‌عامل) برای ساخت واسط کاربری۲ استفاده می‌کند، نام برد.

بسیاری از موارد یاد شده به عنوان ایرادات به جاوا به عنوان ایرادات به طراحی زبان‌های سطح بالا هستند و نه جاوا.

**3-16-یک اشتباه متداول !**

برخی مردم به علت شباهت اسمی، جاوا و جاوااسکریپت را با هم اشتباه می‌گیرند. در حالیکه این دو زبان گرچه در ظاهر و کلمات شبیهند ولی بطور ساختاری با یکدیگر متفاوتند. جاوا اسکریپت محصول شرکت نت اسکیپ است. جاوا برای اجرا باید به زبان ماشین مجازی ترجمه شود اما جاوااسکریپت زبانی است که معمولاً در صفحات وب نوشته می‌شود و توسط مرورگر تفسیر می‌گردد. در جاوا متغیرها همگی بر اساس نوعشان معرفی می‌شوند اما در جاواسکریپت نوع متغیرها به صورت ضمنی مشخص می‌شود.

1. Interface

2. UI : User Interface

38

**3-17- برنامه نویسی برای آندروید :**

برای شروع کار با آندروید به منظور برنامه نویسی برای آن ابتدا چند برنامه مهم از جمله یک IDE ( که بهترین پیشنهاد Eclipse است ، اما از موارد دیگر مانند Netbeans هم میتوانید استفاده کنید) را نصب کنید.دیگر مواردی که باید داشته باشید :

* Java : رابط برنامه نویسی است.
* ADT : یک پلاگین برای Eclipse است که برای شناساندن برنامه به اندروید به کار میرود.
* USB Drive : زمانی به کار میرود که بخواهید برنامه نوشته شده را به گوشی انتقال دهید.
* API : مخفف Application Programming Interface میباشد که به معنای رابط برنامه سازی کاربردی است و توابعی هستند که در برنامه نویسی به کار میروند.
* AVD : مخفف Android Virtual Device است که شبیه ساز سیستم عامل اندروید است.

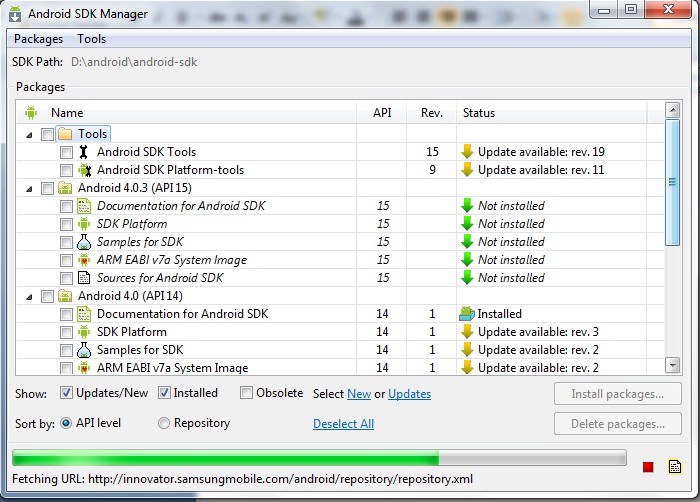
**3-17--1نحوه ی نصب** **SDK** :

SDK مخفف Software Development Kit است که درواقع مجموعه ای از زبان برنامه نویسی ، کتابخانه های اندروید ، شبیه ساز سیستم عامل ، مستندات آندروید و IDE (که پیشنهاد گوگل همان Eclipse است) و فایل های نمونه و آموزشی که تمام کلاس ها و رابط های مورد استفاده در برنامه نویسی آندروید را کاملا توضیح میدهد و API های لازم برای شروع برنامه نویسی بر روی پلتفرم اندروید با زبان برنامه نویسی جاوا را مهیا کرده است.

برای شروع کار با SDK ابتدا باید برنامه Java نصب شود ، چون Java رابط با برنامه نویسی است و ابزارهای توسعه اندروید به آن نیاز دارند. بعد از آن Installer Google را نصب کنید تا بتوانید SDK را دانلود کنید.

در صفحه ی بعد میتوانید تصویر SDK Manager را مشاهده نمایید.

39



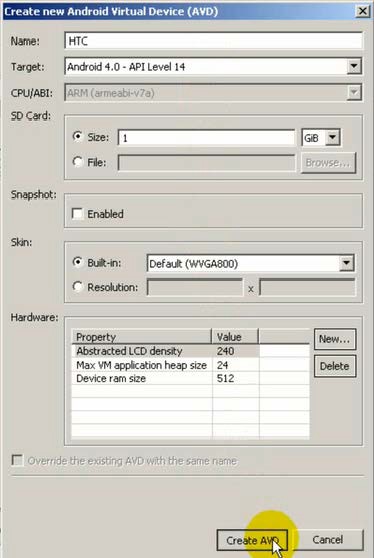
شکل4-SDK Manager

40

**3-17--2 AVD:**

AVD شبیه ساز سیستم عامل اندروید است که یکی از کاربردهای آن تست نرم افزار است.

قبل از اینکه بتوانید برنامه ای را که توسط Eclipse نوشته اید بر روی شبیه ساز مشاهده کنید باید شبیه ساز را تنظیم کنید مثلا بگویید ایت شبیه ساز قرار است کدام نسخه از اندروید را برای شما شبیه سازی کند یا کدام سطح API را میخواهید پشتیبانی کنید.یکی از مزایای ماشین مجازی اندروید نسبت به سایر شبیه سازها مانند نوکیا یا اپل این است که به تعداد نامحدودی میتوانید شبیه ساز داشته باشید. ابتدا به محلی که SDK را نصب کردید بروید و AVD را اجرا کنید.سپس پنجره ی اصلی Android Virual Device Manager را مشاهده میکنید.در قسمت راست بر روی New کلیک کنید تا پنجره ی Create new Android Virtual Device – AVD باز شود.اطلاعات موردنظر را وارد فیلدهای مربوطه کنید.در آخر بر روی دکمه Create AVD کلیک کنید.



شکل5-مراحل نصب شبیه ساز

41

Name : به دلخواه خود یک نام انتخاب کنید.از آنجا ممکن است به چند ماشین مجازی نیاز داشته باشید،نامی را انتخاب کنید که بتوانید تشخیص دهید چه ماشینی است و کدام سطح از API را قرار است پشتیبانی کند.

Target : در این قسمت مشخص میکنید که ماشین کدام ورژن از اندروید ، با چه سطح API ای را شبیه سازی کند. دقت کنید اگر برنامه ای برای نسخه 2.2 نوشته شود قابل اجرا روی نسخه های پایینتر یا قبلی نیست.

SD Card : این فضا برای اجرا و نصب یک برنامه از فضای کامپیوتر شما کسر میشود.

Skin : اندازه صفحه نمایش را مشخص میکند.

Hardware : اگر نیاز به قابلیت خاصی دارید ( مثلا استفاده از سنسور شتاب دهنده) میتوانید با کلیک بر روی New آن را به لیست اضافه کنید.

با کلیک بر روی Create AVD به پنجره قبلی باز میگردید و میبینید که ماشین مجازی که ساختید به لیست اضافه شده است. حال در برنامه Eclipse وقتی برنامه تان را اجرا کنید ، Eclipse برنامه شما را به این ماشین مجازی میفرستد و شما میتوانید برنامه ای را که نوشتید آزمایش کنید.



شکل6-شبیه ساز اندروید

42

**3-17-3- نصب برنامه در شبیه ساز توسط کامپیوتر:**

برنامه ها و بازی های نوشته شده برای اندروید با پسوند .apk ذخیره میشوند. Apk ها فایل های فشرده شده ای هستند که آندروید قابلیت شناسایی آنها را دارد. برای نصب برنامه های موردنظر مطابق زیر عمل کنید: (مراحل یک تا چهار برای نصب در گوشی و از آن به بعد برای گوشی و شبیه ساز یکسان است)

یک. در گوشی به مسیر Setting 🡪 Aplication بروید و گزینه ی Uknown Sources را فعال کنید.

دو. حال به Setting 🡪 Application 🡪 Development بروید و گزینه USB Debugging را فعال کنید.

سه. گوشی را با کابل به کامپیوتر متصل کنید و منتظر باشید تا آن را بشناسد ( اگر گوشی شناسایی نشد از این دو درایور استفاده کنید : Google USB Driver Rev و Android SDK

چهار. حال SDK را در آدرس C:\AndroidSDK ذخیره کنید.

پنج. سپس برنامه ی مورد نظر با پسوند .apk را به مسیر AndroidSDK\platform-tools انتقال دهید.

شش. یک CMD در ویندوز باز کنید و به آدرس برنامه بروید.

هفت. سپس در CMD دستورات زیر را تایپ کنید:

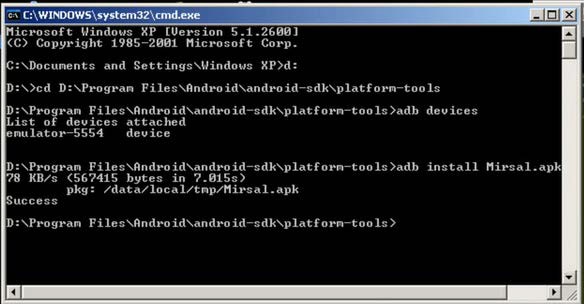
Adb devices

هشت. حال باید شماره سریال دستگاه را ببینید که نشان میدهد مراحل را درست انجام داده اید.حال این دستور را برای نصب برنامه بنویسید: ( مثلا اگر نام برنامه Mirsal.apk است به این شکل:)

Adb install Mirsal.apk

نه. سپس ارتباط با گوشی برقرار شده و در انتها “Seccess” نوشته میشود.

43



شکل7-کدهای مربوط به نصب برنامه در شبیه ساز توسط CMD

برای Uninstall کردن برنامه ، به Setting 🡪 Application 🡪 Manage Applications گوشی یا شبیه ساز بروید.در آنجا اطلاعات کاملی از نرم افزارهای نصب شده روی گوشی یا شبیه ساز میبینید . سپس روی نرم افزار مورد نظر کلیک کنید و Uninstall را انتخاب کنید.بدین صورت برنامه مورد نظر پاک میشود.

44

**3-18- Eclipse و برنامه نویسی با آن:**

**3-18-1- برنامه نویسی آندروید چگونه است؟**

آندروید دارای کتابخانه های برنامه نویسی قدرتمندی است که کار برنامه نویسان را بسیار ساده میکند.از جمله پایگاه داده رابطه ای کوچک شده ی SQLite ، توابع گرافیکی سه بعدی openGL ، موتور مرورگر webkit (همانند مرورگر Chrome ) و موتور گرافیکی SGL. برنامه نویسی برای آندروید به زبان جاوا انجام میگیرد و در نهایت کامپایلر Dalvik برنامه نوشته شده به زبان جاوا را به کدهای مقصد که قابل فهم برای سیستم عامل اندروید باشند تبدیل میکند و در آخر یک برنامه Package Manager که فایلی قابل نصب است با پسوند .apk تولید میکند.

**3-18-2-Eclispe چیست؟**

اکلیپس یک محیط توسعه نرم‌افزاری چندزبانه برای محیط توسعه مجتمع با قابلیت اضافه کردن افزونه می‌باشد. این محیط توسعه در ابتدا با زبان جاوا و برای توسعه برنامه‌های این زبان استفاده می‌شد. در ادامه با افزوده شدن افزونه‌هایی به آن امکان توسعه زبان‌هایی چون سی، سی++، روبی، کوبول، پایتون، پرل، پی اچ پی، لاتک و آندروید را فراهم می‌کند.

**3-18-2-1- معماری:**

سامانه ی زمان اجرای اکلیپس مبتنی بر اکویناکس که پیاده سازی استانداردی، سازگار با او اس جی ای است، می‌باشد. اکلیپس از افزونه هارا برای فراهم آوردن امکان استفاده از تمام قابلیت‌هایش بر روی سامانه اجرا، استفاده می‌نماید که برخلاف با بعضی از برنامه‌هایی که به صورت معمول با قابیلت کد سخت می‌باشند، است.

ساز و کار افزونه‌های در اکلیپس صورتی سبک تر از چهارچوب مهندسی نرم‌افزار بر اساس مولفه است. به علاوه برای اینکه اکلیپس قابل گسترش برای توسعه سایر زبان‌های برنامه نویسی مانند سی و پایتون باشد، چهارچوب افزونه‌ها به اکلیپس امکان کار با زبان‌های حروفچینی مانند لاتک و برنامه‌های شبکه‌ای مانند تلنت و سامانه مدیرت پایگاه داده را فراهم می‌آورد. جاوا و پشتیبانی از سی وی اس در کیت توسعه نرم‌افزاری اکلیپس فراهم شده است.

45

به جز بخش کوچکی از هسته زمان اجرا، تمام بخش‌های در اکلیپس به صورت افزونه‌ای است. این بدان معناست تمام افزونه‌های در اکلیپس به شکل دقیقاً یکسانی توسعه می‌یابند و تمام خصوصیات آنها با هم برابر است. اکلیپس برای بخش عظیمی از خصوصیات افزونه‌هایی را فراهم می‌نماید، که بعضی از آنها از طرف شخص ثالت توسعه به اشکال رایگان یا تجاری توسعه می یاند. نمونه‌ای از این افزونه‌ها، افزونه یو ام‌ال برای نمودار ترتیبی و سایر دیاگرام‌های یو ام‌ال است، یا افزونه‌ای برای کاوش در پایگاه داده.

**3-18-2-2- سکوی غنی مشتری:**

اکلیپس، سکوی غنی مشتری اکلیپس را برای توسعه برنامه‌های همه منظوره را فراهم می‌آورد. اجزا زیر این بستر را فراهم می‌آورد:

* اکویناکس – یک چهارچوب بسته بندی استاندارد
* هسته سکو – بوت اکلیپس ، اجرای افزونه ها
* جی فیس – کلاسهای نمایشگر برای انتقال حالت از مدل کنترل نمایشگر به اس دبلیو تی،بافر فایل ها ، ویرایشگر متن ها
* میزکار اکلیپس ۱– نمایش ها ، ویرایشگرها ، چشم اندازها.

**3-18-2-3- تاریخچه:**

اکلیپس به عنوان پروژه‌ای در شرکت ای بی ام شعبه کانادا ایجاد گردید که برای جایگزینی مخیطهای توسعه خانواده ی ویژوال ایج بود ویژوال ایج مبتنی بر اسمال تاک بود در حالی که اکلیپس توسط او تی ای بر مبنای جاوا توسعه یافت. در نوامبر سال ۲۰۰۱، کنسرسیومی برای توسعه اکلیپس به عنوان یک نرم‌افزار متن باز ایجاد شده که در سال ۲۰۰۴، بنیاد اکلیپس تاسیس شد.

نگارش سوم اکلیپس(منتشر شده در ۲۱ ژوین سال ۲۰۰۴) سکوی خدماتی او اس جی ای را به عنوان معماری زمان اجرای خود برگزید. اکلیپس در ابندا تحت مجوز متداول عمومی و بعداً تحت مجوز عمومی اکلیپس منتشر شد. نرم‌افزار آزاد و متن‌باز بعدها اعلام نمود که هردو مجوز از نوع نرم‌افزار آزاد هستند، ولی با اجازه‌نامه عمومی همگانی گنو کاملاً مطابق نیستند.

1. Eclipse Workbench

46

براساس گفته لی نکمن، مدیر ارشد فناوری شرکت ای بی ام بخش رشنال در آن زمان و که بعدها رییس بخش توسعه وپشتیبانی این بخش شد، انتخاب نام اکلیپس، برای مقابله با ویژوال استودیو متعلق به شرکت مایکروسافت بود و نه شرکت سان میکروسیستم جالب آنکه وی اکنون کارمند شرکت مایکروسافت است.

**3-18-2-4نسخه ها:**

|  |  |  |
| --- | --- | --- |
| نسخه | تاریخ انتشار | نگارش سکو |
| مارس | ۲۵ ژوین ۲۰۱۵ (تیر ۱۳۹۴) (برنامه‌ریزی شده برای انتشار) | 4.5 |
| لونا | ۲۵ ژوین ۲۰۱۴ (تیر ۱۳۹۳) | 4.4 |
| کپلر | ژوین ۲۰۱۳ (تیر ۱۳۹۲) | 4.3 |
| جونو | ۲۷ ژوین ۲۰۱۲ (تیر ۱۳۹۱) | 4.2 و 3.8 |
| ایندیگو | ژوین ۲۰۱۱ (تیر ۱۳۹۰) | 3.7 |
| هلیوس | ۲۳ژوین ۲۰۱۰ (تیر ۱۳۸۹) | 3.6 |
| گالیله | ۲۴ژوین ۲۰۰۹ (تیر ۱۳۸۸) | 3.5 |
| گانیمده | ۲۵ژوین ۲۰۰۸ (تیر ۱۳۸۷) | 3.4 |
| اروپا | ۲۹ژوین ۲۰۰۷ (تیر ۱۳۸۶) | 3.3 |
| کالیستو | ۳۰ژوین ۲۰۰۶ (تیر ۱۳۸۵) | 3.2 |
| اکلیپس 3.1 | ۲۸ ژوین ۲۰۰۵ (تیر ۱۳۸۴) | 3.1 |
| اکلیپس 3.0 | ۲۸ ژوین ۲۰۰۴ (تیر ۱۳۸۳) | 3.0 |

جدول3-نسخه های Eclipse

**3-18-2-5افزونه ها :**

برای برنامه اکلیپس افزونه‌های بسیاری ساخته شده است که هریک برای زبان و اعمالِ خاصی طراحی شده‌است.

47

**3-18-2-6شروع کار :**

ابتدا باید Android-SDK را که از قبل نصب کردید در درایو ویندوز کپی کنید.سپس برنامه Eclipse را اجرا کنید: (با این توضیح که Eclipse نیازی به نصب شدن ندارد و شما باید آنرا از اینترنت دانلود کنید و سپس از حالت zip خارج کنید)



شکل8-نرم افزار Eclipse

برای شروع کار مطابق زیر عمل کنید:

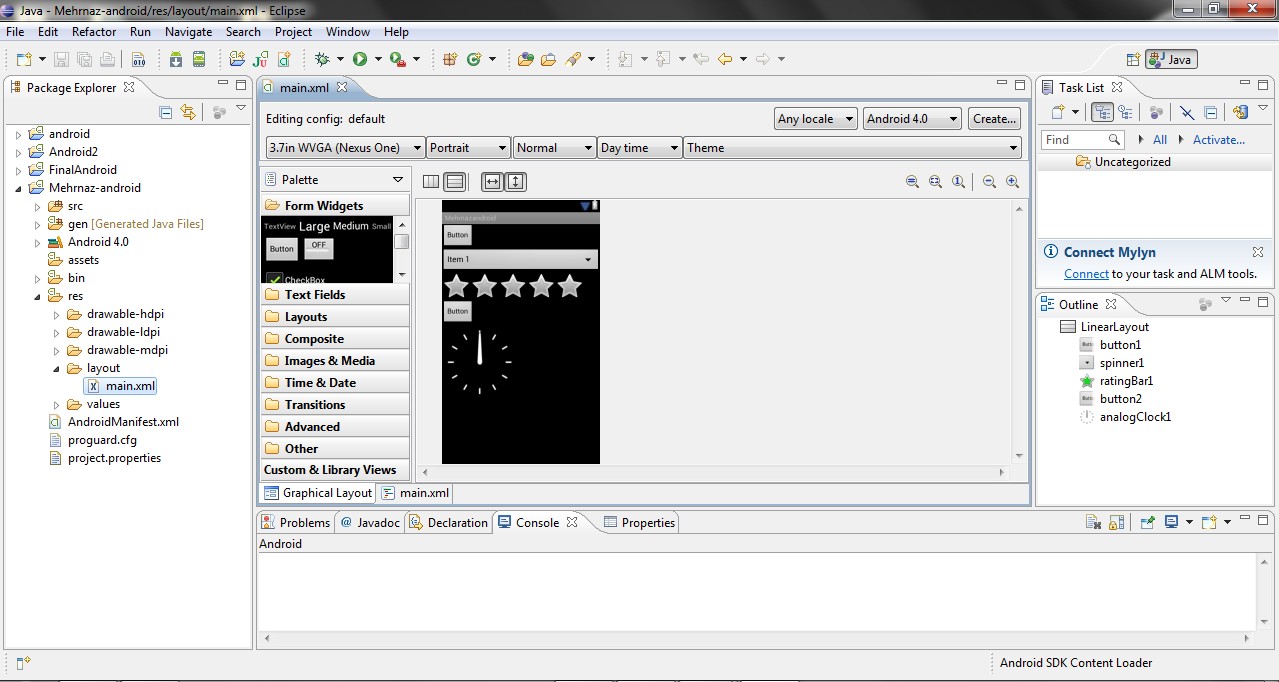
یک. به Help 🡪 Install new software بروید و پس از کلیک روی Add ، در Name نام دلخواه را وارد کنید و در قسمت Location مسیر ADT مورد نظر را که از قبل دانلود کردید انتخاب کنید.

دو. در پنجره ی Available Software تیک Developer Tools را انتخاب کنید و پس از آن Next را بزنید.

سه. سپس به Window 🡪 Preferences و شاخه ی Android بروید و مسر SDK را در SDK Location وارد کنید.

48

چهار. سپس File 🡪 New 🡪 Project 🡪 Android Project را انتخاب کنید.پس از وارد کردن نام و انتخاب نسخه اندروید موردنظر برای برنامه ، Eclipse برای شروع کدنویسی آماده خواهد شد.

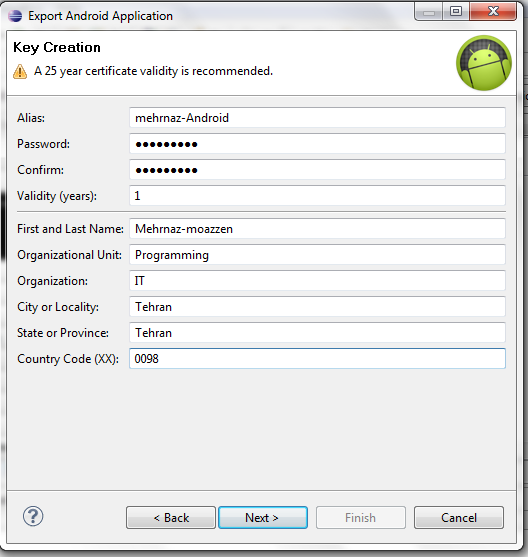


شکل9- محیط برنامه نویسی Eclipse

توسط main.xml در شاخه ی layout برنامه (مطابق شکل بالا) ، میتوانید با Drag & Drop به راحتی اجزای موردنظر را روی صفحه بکشید و کارهای گرافیکی برنامه خود را انجام دهید . به این صورت میتوانید نوشتن اولین برنامه خود را آغاز کنید.

پس از کدنویسی به File 🡪 Export بروید. سپس از شاخه Android ، Android Export Application را انتخاب کنید. پس از وارد کردن محل ذخیره سازی،نام و رمز موردنظر برای برنامه پنجره زیر نمایش داده میشود:

49



شکل10-کادر ورود اطلاعات برای ساختن اپ

پس از وارد کردن اطلاعات مورد نظر و کلیک کردن روی Next ، برنامه با پسوند .apk ساخته خواهد شد.

50

**3-18-2-7نمونه های از برنامه های جاوا :**

در زیر نمونه‌ای از برنامه‌ای که در جاوا نوشته شده‌است آورده شده‌است. البته برای کامپایل کردن این برنامه به کیت توسعه جاوا که قبلا نصب کرده اید نیاز دارید.

**public** **class** Test{

**public** **static** **void** main(String[] args) {

System.out.println("HelloWorld!");

}

}

برای اجرای برنامه بالا، ابتدا باید یک فایل به نام Test.java ساخته شود و سپس کامپایل شود:

$ javac Test.java

سپس یک فایل خروجی به نام Test.class دریافت می‌شود. بعد با استفاده از دستور زیر برنامه قابل اجرا است:

$ java Test

**مثال:**

برنامه Hello world به این صورت در زبان جاوا می‌تواند نوشته شود:

*// HelloWorld.java*

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

System.out.println("Hello, World!");

}

}

51

بر طبق قرارداد فایل ها بعد از کلاس‌های عمومی نام گذاری می‌شوند. سپس باید پسوند java را به این صورت اضافه کرد: Hello world.java. این فایل اول باید با استفاده از کامپایلر جاوا به بایت کد کامپایل شود. در نتیجه فایل Hello world.class ایجاد می‌شود. این فایل قابل اجرا است. فایل جاوا ممکن است فقط یک کلاس عمومی داشته باشد. اما می‌تواند شامل چندین کلاس با دستیابی عمومی کمتر باشد.

کلاسی که به صورت خصوصی تعریف می‌شود ممکن است در فایل.java ذخیره شود. کامپایلر برای هر کلاسی که در فایل اصلی تعریف می‌شود یک کلاس فایل تولید می‌کند. که نام این کلاس فایل همنام کلاس است با پسوند.class

کلمه کلیدی publicبرای قسمت‌هایی که می‌توانند از کدهای کلاس‌های دیگر صدا زده بشوند، به کار برده می‌شود. کلمه ی کلیدی static در جلوی یک تابع، یک تابع ایستا را که فقط وابسته به کلاس است و نه قابل استفاده برای نمونه‌هایی از کلاس، نشان می‌دهد. فقط تابع‌های ایستا می‌توانند توسط اشیا بدون مرجع صدا زده شوند. داده‌های ایستا به متغیرهایی که ایستا نیستند، نمی‌توانند دسترسی داشته باشند.

کلمه ی کلیدی void نشان می‌دهد که تابع main هیچ مقداری را بر نمی‌گرداند. اگر برنامه ی جاوا بخواهد با خطا از برنامه خارج شود، باید system.exit() صدا زده شود. کلمه ی main یک کلمه کلیدی در زبان جاوا نیست. این نام واقعی تابعی است که جاوا برای فرستادن کنترل به برنامه، صدا می‌زند. برنامه جاوا ممکن است شامل چندین کلاس باشد که هر کدام دارای تابع main هستند.

تابع main باید آرایه‌ای از اشیا رشته‌ای را بپذیرد. تابع main می‌تواند از آرگومان‌های متغیر به شکل public static void main(string…args) استفاده کند که به تابع main اجازه می‌دهد اعدادی دلخواه از اشیا رشته‌ای را فراخوانی کند. پارامترstring[]args آرایه‌ای از اشیا رشته ایست که شامل تمام آرگومان‌هایی که به کلاس فرستاده می‌شود، است.

52

چاپ کردن، قسمتی از کتابخانه ی استاندارد جاوا است. کلاس سیستم یک فیلد استاتیک عمومی به نام out تعریف کرده‌است. شی out یک نمونه از کلاس printstream است و شامل تعداد زیادی تابع برای چاپ کردن اطلاعات در خروجی استاندارد است. همچنین شامل println(string) برای اضافه کردن یک خط جدید برای رشته ی فرستاده شده اضافه می‌کند.

**3-18-2-8کلاس های خاص:**

**برنامک(برنامه های کاربردی کوچک) :**

اپلت جاواها برنامه‌هایی هستند که برای کاربردهایی نظیر نمایش در صفحات وب، ایجاد شده‌اند. واژه ی import باعث می‌شود کامپایلر جاوا کلاس‌های javaapplet.Applet وjava.awt.Graphics را به کامپایل برنامه اضافه کند. کلاس Hello کلاس Applet را توسعه می‌دهد. کلاس اپلت چارچوبی برای کاربردهای گروهی برای نمایش و کنترل چرخه ی زندگی اپلت، درست می‌کند. کلاس اپلت یک تابع پنجره‌ای مجرد است که برنامه‌های کوچکی با قابلیت نشان دادن واسط گرافیکی برای کاربر را فراهم می‌کند. کلاس Hello تابع موروثی print(Graphics) را از سوپر کلاس container باطل می‌کند، برای اینکه کدی که اپلت را نمایش می‌دهد، فراهم کند. تابع paint شی‌های گرافیکی را که شامل زمینه‌های گرافیکی هستند را می‌فرستد تا برای نمایش اپلت‌ها استفاده شوند. تابع paint برای نمایش "Hello world!" تابع drawstring(string,int,int) را صدا می‌زند.

**JSP :**

صفحه ی سرور جاوا قسمتی از سرور javaEE است که پاسخ تولید می‌کند. نوعاً صفحات HTML به درخواست‌های HTTP از مشتری.JSPها کد جاوا در صفحه ی HTML را با استفاده از حائل <%and%> اضافه می‌کنند.JSP به javaservlet کامپایل می‌شود.

**جاوا سرولت :**

تکنولوژی servlet جاوا گسترس وب را به آسانی فراهم می‌کند. و شامل مکانیزم‌هایی برای توسعه ی تابعی سرور وب و برای دسترسی به سیستم‌های تجاری موجود است.servlet قسمتی از javaEE است که به درخواست‌های مشتری پاسخ می‌دهد.

53

*// Hello.java*

**import** java.io. \*;

**import** javax.servlet. \*;

**public** **class** Hello **extends** GenericServlet {

**public** **void** service(ServletRequest request, ServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

**final** PrintWriter pw = response.getWriter();

pw.println("Hello, world!");

pw.close();

}

}

واژه ی import کامپایلر جاوا را هدایت می‌کند که تمام کلاس‌های عمومی و واسط‌ها را از بسته‌های java.io وjava.servlet را در کامپایل وارد کند.

کلاس Hello کلاس Genericservlet را توسعه می‌دهد. کلاس Genericservlet واسطی برای سرور فراهم می‌کند تا درخواست را به servlet بفرستد و چرخه ی زندگی servlet را کنترل کند.

**Generics :**

قبل از کلاس‌های عمومی برای هر متغیر باید یک نوع خاص تعریف می‌کردیم. به عنوان مثال برای کلاس‌های ظرف این امر مشکل بود زیرا را آسانی برای ایجاد یک container وجود نداشت که نوع‌های خاصی از اشیا را بپذیرد. کلاس‌های عمومی اجازه می‌دهند نوع زمان کامپایل، بدون نیاز به ایجاد تعداد زیادی از container، چک شود. همه آنها کدهای مشابهی دارند.

**سویینگ :**

Swing کتابخانه ی واسط گرافیکی کاربر است برای پلت فرم javaSE. ابزاری مشابه پنجره، GTK و motif توسط شرکت sun فراهم شده‌اند. این مثال کاربرد swing یک پنجره ی واحد همراه با Hello world را ایجاد می‌کند.

54

*// Hello.java (Java SE 5)*

**import** java.awt.BorderLayout;

**import** javax.swing. \*;

**public** **class** Hello **extends** JFrame {

**public** Hello() {

**super**("hello");

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

setLayout(**new** BorderLayout());

add(**new** JLabel("Hello, world!"));

pack();

}

**public** **static** **void** main(String[] args) {

**new** Hello().setVisible(**true**);

}

}

اولین جمله ی import کامپایلر جاوا را هدایت می‌کندتا کلاس Borderlayout را از بسته ی java.awt در جاوا به کامپایل اضافه کند. و import دوم همهٔ کلاس‌های عمومی و واسط آن‌ها را از بسته ی javax.swing اضافه می‌کند. کلاس Hello کلاس Jframe را توسعه می‌دهد. کلاس Jframe یک پنجره با میلهٔ عنوان و کنترل بستن است.

زمانی که برنامه آغاز می‌شود، تابع main با JVM صدا زده می‌شود. این یک نمونه ی جدید از کلاس Hello را ایجاد کرده و با صدا زدن تابع setvisible(Boolean) با مقدار true نمایش داده می‌شود.

55

***فصل چهارم***

**معرفی پروژه :**

**بازی آندروید**

**"تست حافظه"**

56

بسیارخب ، تا حالا با سیستم عامل آندروید و زبان برنامه نویسی جاوا آشنا شدید. حالا به عنوان نمونه شروع می کنیم پروژه ی خودم رو قدم به قدم پیش میبریم.

اولین قدم ، ایده ی بازی است. ایده ی بازی من از بازی های تست حافظه ای کوچک که شامل چندین جفت عکس بودند گرفته شده ، با این تفاوت که من یک عامل برای باخت هم درنظر گرفتم.

دومین قدم ، نوشتن یه سناریو برای بازی ست ، بازی چجوری شروع شه ، چجوری ادامه پیدا کنه ، کجا تموم بشه ؛ مثلا در اینجا بازی با 9 خانه ی یک شکل شروع میشه ، اینجوری ادامه پیدا میکنه که شما به طور شانسی باید خانه ها را یک به یک کلیک کنید و به خاطر بسپارید پشت هر خانه چه شکلی است و شکل های یکسان را با هم جفت کنید. اگر خانه ی بمب را کلیک کنید همانجا بازی تمام میشود و اگر بتوانید همه ی عکس ها را جفت کنید بدون اینکه رو بمب کلیک کنید بازی را برده اید و زمان شما به عنوان امتیاز بازی ثبت میشود تا اگر از بهترین زمان کمتر باشد ، جایگزین آن شود.

قدم بعدی ، شروع کردن به نوشتن و کد زدن برای بازی است و قدم آخر طراحی مراحل یا همان انتخاب پس زمینه و شکل ها و باقی مسائل مربوط به شکل ظاهری اپلیکیشن می باشد.

در اینجا چند عکس از مراحل مختلف بازی را برایتان می گذارم :



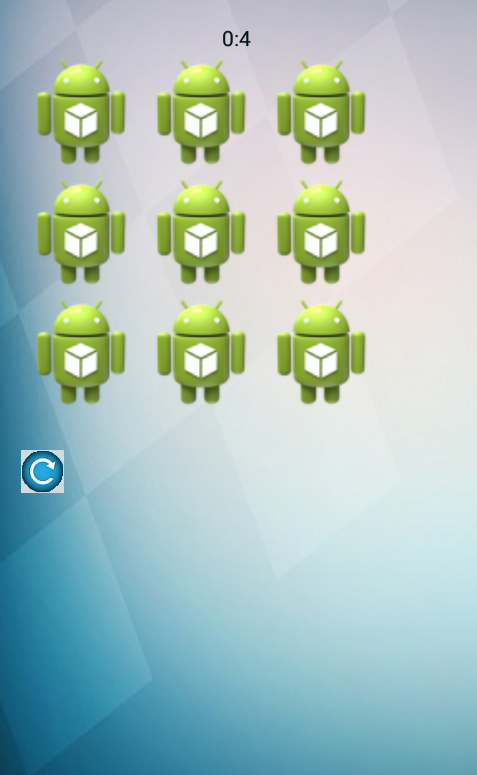
شکل11-صفحه ی خانه ی بازی

57

"صفحه خانه" بازی صفحه ی شروع بازی است که شامل دو گزینه ی Play برای رفتن به صفحه ی اصلی بازی و شروع آن و گزینه ی Exit برای خارج شدن از اپلیکیشن می باشد.

با زدن گزینه ی Play به صفحه ی زیر هدایت میشوید که در اینجا نه خانه ی بازی را مشاهده می کنید ، در پشت هر کدام از این خانه ها یک عکس قرار دارد که با کلیک کردن بر آن عکس شروع به چرخیدن میکند. در بالای تصویر Timer بازی را مشاهده میکنید که زمان پایان بازی را محاسبه میکند.

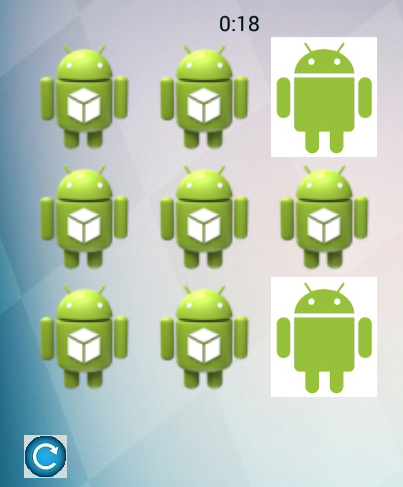
در پایین خانه ها هم دکمه Reload بازی برای بازگشتن به حالت اولیه قرار دارد.



شکل12-صفحه ی اصلی بازی

58

خب ، همانطور که گفتم با کلیک کردن روی هر خانه ، آن عکس شروع به چرخیدن میکند و عکس پشت خود را نشان میدهد ، هدف بازی این است که مانند شکل زیر ، عکس های یکسان با هم جفت شوند تا همه عکس ها به غیر از بمب پیدا شود و بازی تمام شود.

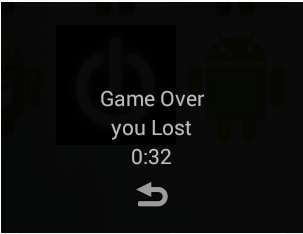


شکل13 – هدف بازی

59

و در پایان یک پیغام به شما نشان داده میشود که حکایت از آن است که شما برنده یا بازنده شده اید.

در عکس زیر پیغام تمام شدن را مشاهده میکنید ، که در آن میگوید شما بازی را باخته اید و در پایین پیغام زمان بهترین برد را میتوانید ببینید.



شکل14 – پیغام پایان بازی

با زدن دکمه ی Reload پیغام بازی به حالت اولیه برمیگردد و زمان سنج صفر میشود.

خب ، چگونگی شروع ، ادامه و پایان بازی را مشاهده کردید ، حال در فصل آینده بصورت فنی به چگونگی ساخت بازی میپردازیم.

60

***فصل پنجم***

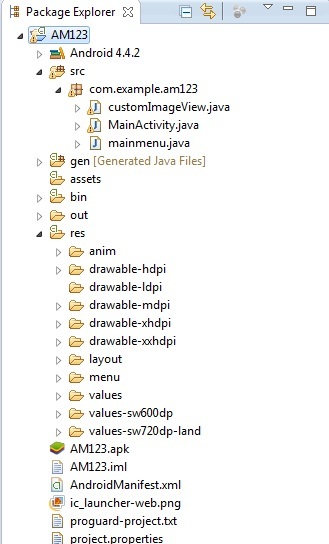
**توضیحات فنی**

**و**

**کدها**

61

برای شروع بهتر است بدانید که اکثر کار ما برای نوشتن این بازی ، با دو پوشه src و res میباشد.محتویات این دو پوشه را در شکل زیر میتوانید ببینید :



شکل15-محتویات پروژه

پوشه ی src برای ایجاد کلاس ها و نوشتن کدهای پروژه و پوشه ی res برای نوشتن Layout صفحات برنامه ، ذخیره ی عکس ها ، ذخیره ی اسم و آیکون برنامه و باقی خصوصیات ظاهری میباشد.

62

در ابتدا به کلاس ها و کدهای مربوط به آنها میپردازیم:

برای راحت کار کردن با کدها و اجرای عملیات ، کلاس CustomeImageView را میسازیم که کلاس ImageView را Extend کرده و با رابط1 AnimationListener از زیرکلاس Animation آن را ایمپلمنت2 میکنیم:

**public** **class** customImageView **extends** ImageView **implements** Animation.AnimationListener

حال مقادیر زیر را نوشته و مقداردهی میکنیم:

**Context** context;

**boolean** isMiddle=**false**;

**boolean** showFace;

**int** tile=R.drawable.*ic\_launcher*;

**int** face;

**private** **boolean** freez=**false**;

**public** **boolean** tempFreez=**false**;

**public** customImageView(Context context) {

**super**(context);

**this**.context= context;

}

**public** customImageView(Context context, AttributeSet attrs, **int** defStyle, **int** mode){

**super**(context, attrs, defStyle);

}

**public** customImageView(Context context, AttributeSet attrs){

**super**(context,attrs);

}

1. Interface

2. Implement

63

سه ساختمان1 CustomeImageView نوشته شده برای آن است که بعدا بتوانیم از آن بصورت com.example.am123.customImageView به عنوان رابط بین XML و جاوا استفاده کنیم.

حال سه ساختمان کد زیر را به صورت Override برای استفاده از Animation و چرخش عکس هنگام کلیک شدن مینویسیم:

@Override

**public** **void** onAnimationStart(Animation animation) {

}

@Override

**public** **void** onAnimationEnd(Animation animation) {

**if** (isMiddle){

**if** (showFace){

setImageResource(face);

}

**else**{

setImageResource(tile);

}

clearAnimation();

Animation anim2=AnimationUtils.*loadAnimation*(context,R.anim.*flip2*);

anim2.setAnimationListener(**this**);

setAnimation(anim2);

startAnimation(anim2);

isMiddle=**false**;

}

tempFreez=**false**;

}

1. Constracture

64

@Override

**public** **void** onAnimationRepeat(Animation animation) {

}

در مورد دستور OnAnimationEnd ابتدا بررسی میکند آیا بلوک انتخاب شده در حالت وسط قرار دارد یا نه.اگر بود عکس مورد نظر تصادفی را جایگزین میکند و در غیر اینصورت عکس کاشی (عکس پشت ای که در حالت اولیه ما هر نه خانه را با آن عکس میبینیم) را قرار میدهد.

در ادامه دستور بارگذاری انیمیشن و حرکت کردن بلوک توسط کد flip2 (کدهای flip2 و flip1 را در ادامه خواهیم دید) و بعد از آن false کردن "وسط بودن" بلوک و از freeze درآوردن باقی بلوک ها را مشاهده میکنیم.

کد بالا برای زمانی وسط اجرا میباشد و کد زیر برای زمان از لحظه ی انتخاب شدن بلوک تا لحظه ای که به حالت وسط میرسد است :

**public** **void** startFlip(**boolean** showFace,**int** face){

**this**.face=face;

**this**.showFace=showFace;

clearAnimation();

Animation anim1= AnimationUtils.*loadAnimation*(context,R.anim.*flip1*);

anim1.setAnimationListener(**this**);

setAnimation(anim1);

startAnimation(anim1);

isMiddle=**true**;

}

65

دو دستور زیر نیز در رابطه با ثابت کردن1 بلوک ها میباشد :

**public** **void** setFreez(){

freez=**true**;

}

**public** **boolean** isFreez(){

**return** freez;

}

در پایین دو دستور flip را مشاهده میکنیم که این دستورها بصورت xml در پوشه ی anim زیرشاخه ی پوشه ی res پروژه ی ما قرار دارند.طرز کار این دستور به این صورت است که ما مقدار چرخش و زمان طول کشیدن را مشخص میکنیم ؛flip1 :

<scale

xmlns:android=*”http://schemas.android.com/apk/res/android”*

android:fromXScale=*”1.0”* android:toXScale=*”0.0”*

android:pivotX=*”50%”*

android:fromYScale=*”1.0”* android:toYScale=*”1.0”*

android:pivotY=*”50%”*

android:duration=*”1000”* />

flip2:

<scale

xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:fromXScale=*"0.0"* android:toXScale=*"1.0"*

android:pivotX=*"50%"*

android:fromYScale=*"1.0"* android:toYScale=*"1.0"*

android:pivotY=*"50%"*

android:duration=*"1000"* />

1. freeze

66

حال برای ادامه و درک بهتر کلاس mainmenu.java ما ابتدا فایل activity\_mainmenu.xml را که در پوشه ی layout زیرشاخه ی res قرار دارد را بررسی میکنیم.فایل های xml این پوشه برای کارهای گرافیکی و ساختن نما و مشخص کردن اندازه ها میباشند.

کد زیر مشخص کننده ی اندازه ی کل صفحه ، نحوه دید بصورت عمودی یا افقی و عکس پس زمینه است:

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:background=*"@drawable/background"*>

دو دستور زیر نیز مشخص کننده ی اندازه ی دکمه ها ، مقدار فاصله از عنصر بالایی ، ساختن یک شناسه برای ارتباط دادن با کلاس جاوا ، مکان قرار گرفتن در تصور (وسط یا گوشه) و عکس پس زمینه ی دکمه ها هستند:

<ImageView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"150dp"*

android:id=*"@+id/playGameImageView"*

android:layout\_gravity=*"center"*

android:background=*"@drawable/play"*/>

<ImageView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"50dp"*

android:id=*"@+id/ExitImageView"*

android:layout\_gravity=*"center"*

android:cropToPadding=*"false"*

android:background=*"@drawable/exit"*/>

67

نکته : تفاوت مفهوم wrap\_contant و fill\_parrent در این است که در اولی اندازه ی عنصر به میزان لازم مقادیر داخل آن است و فضای بیشتری را اشغال نمیکند اما در fill\_parrent اندازه ی عنصر به اندازه ی عنصر والد آن می باشد.

برای درک بهتر کدهای بالا بهتر است یک بار دیگر به شکل 11-صفحه ی خانه ی بازی (صفحه 57) مراجعه کنید.

اکنون به کدهای کلاس جاوای مربوط به صفحه ی خانه میپردازیم.یک کلاس میسازیم که کلاس Activity را گسترش دهد و دو شی ImageView برای دو دکمه ی صفحه میسازیم:

**public** **class** mainmenu **extends** Activity {

ImageView playGameImageView;

ImageView exitImageView;

در تابع زیر شی های ساخته شده را با شناسه ی فایل های xml بالا مرتبط میکنیم:

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_mainmenu*);

playGameImageView=(ImageView)findViewById(R.id.*playGameImageView*);

exitImageView=(ImageView)findViewById(R.id.*ExitImageView*);

و سپس دستور هرکدام را مینویسیم که به صفحه ی اصلی بازی برویم یا از بازی خارج شویم:

playGameImageView.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View view) {

startActivity(**new** Intent(mainmenu.**this**,MainActivity.**class**));

}

});

68

exitImageView.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View arg0) {

finish();

}

});

حال میرسیم به کدهای اصلی بازی در کلاس MainActivity و فایل activity\_main.xml .

باز هم ابتدا به فایل xml میپردازیم ، کد زیر مشخص کننده ی مقادیر اصلی صفحه ، عکس پس زمینه و context مربوطه ، فاصله عناصر داخلی و گرایش1 آن میباشد:

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:background=*"@drawable/background"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:orientation=*"vertical"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*".MainActivity"*

android:layout\_gravity=*"center"*

android:gravity=*"center"* >

کد LinearLayout زیر مقادیر کلی عناصر صفحه را مشخص میکند:

1. orientation

69

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

android:orientation=*"vertical"*

android:gravity=*"center"*

android:layout\_gravity=*"center"*>

کد TextView زیر برای زمان سنج بازی و مشخصات اندازه و رنگ و ساختن شناسه آن است :

<TextView

android:id=*"@+id/timerTextView"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:textColor=*"#011210"* />

کد LinearLayout زیر برای یک ردیف از خانه های صفحه اصلی بازی است که شامل سه بلوک میباشد و در آنها اندازه ردیف و بعد آن اندازه ی هر بلوک ، عکس کاشی (پشت بلوک) ، تگ آن و ارتباط دادن آن با CustomeImageView نوشته شده است:

<LinearLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"* >

<com.example.am123.customImageView android:id=*"@+id/image1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:src=*"@drawable/ic\_launcher"*

android:tag=*"1"*/>

کد بالا را برای هر نه خانه ی بازی و هر سه ردیف مینویسیم و سپس کد زیر را برای دکمه "restart" مینویسیم:

70

<LinearLayout android:id=*"@+id/panel"*

android:orientation=*"vertical"*

android:layout\_height=*"fill\_parent"*

android:layout\_width=*"fill\_parent"*>

<Button android:id=*"@+id/restartButton"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:background=*"@drawable/reset"*

android:layout\_marginTop=*"25dp"*/>

</LinearLayout>

برای دیدن گرافیکی کدهای بالا به شکل12-صفحه ی اصلی بازی(صفحه58) مراجعه کنید.

حالا میرسیم به قسمت کدهای کلاس جاوا MainActivity . کلاس را ساخته و کلاس اصلی Activity را در آن گسترش میدهیم:

**public** **class** MainActivity **extends** Activity {

نُه شی از کلاس CustomeImageView میسازیم:

customImageView image1;

customImageView image2;

customImageView image3;

customImageView image4;

customImageView image5;

customImageView image6;

customImageView image7;

customImageView image8;

customImageView image9;

71

مقادیر زیر را نیز برای استفاده های آتی میسازیم:

**Button** resartButton;

**TextView** timerTextView;

**Handler** handler=**new** Handler();

**int**[] bmpArray;

**boolean** isFirst=**true**;

**customImageView** tempCustomImageView;

**int** score=0;

**int** bombRes= drawable.*bomb*;

**int** time=0;

**boolean** isGameOver=**false**;

حال تابع زیر را مینویسیم که در آن به فایل activity\_main.xml وصل شده ، سپس یک آرایه ی نه خانه ای ساخته و عکس های بازی را به هرخانه ی آن مقداردهی میکنیم تا سپس با تابع randomize بتوانیم آنها را درهم آمیخته کنیم.

دوتابع InitView() و fitInScreen() را هم که کدهای آنها را در ادامه خواهیم دید مینویسیم تا در تابع اولی هر شی ساخته شده ی CustomeImageView را به یک خانه ی فایل xml صفحه اصلی بازی وصل کنیم.در تابع دومی نیز معین میکنیم که اندازه ی خانه ها در هر دستگاه چگونه باشد.

بعد از نحوه ی کار کردن زمان سنج را مشخص کرده ایم که در ابتدا زمان بازی 0 باشد و با پیشرفت بازی میزان زمان به دقیقه و ثانیه تبدیل میشود :

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

bmpArray= **new** **int**[9];

72

bmpArray[0]=drawable.*a*;

bmpArray[1]=drawable.*b*;

bmpArray[2]=drawable.*c*;

bmpArray[3]=drawable.*d*;

bmpArray[4]=drawable.*a*;

bmpArray[5]=drawable.*b*;

bmpArray[6]=drawable.*c*;

bmpArray[7]=drawable.*d*;

bmpArray[8]=drawable.*bomb*;

bmpArray=randomize(bmpArray);

InitView();

fitInScreen();

timerTextView.setText("0:0");

handler.postDelayed(**new** Runnable() {

@Override

**public** **void** run() {

time++;

**int** min=time/60;

**int** sec=time%60;

timerTextView.setText(min+":"+sec);

**if**(!isGameOver)

handler.postDelayed(**this**,1000);

}

},1000);

}

73

نکته : استفاده از handler سه خاصیت دارد ؛ اول-میتواند کدی را با تاخیر اجرا کند. دوم-thread است ، یعنی زیرمجموعه ای از برنامه است . سوم-به context اصلی برنامه دسترسی دارد.

کد تابع randomize() :

**private** **int**[] randomize(**int**[] bmparr){

Random rgen = **new** Random(); // Random number generator

**for** (**int** i=0; i<bmparr.length; i++) {

**int** randomPosition = rgen.nextInt(bmparr.length);

**int** temp = bmparr[i];

bmparr[i] = bmparr[randomPosition];

bmparr[randomPosition] = temp;

}

**return** bmparr;

}

کد تابع fitInScreen() :

**private** **void** fitInScreen(){

android.view.Display display = ((android.view.WindowManager) getSystemService(Context.*WINDOW\_SERVICE*))

.getDefaultDisplay();

LinearLayout.LayoutParams llp1=**new** LinearLayout.LayoutParams((display.getWidth()\*25)/100, (display.getWidth()\*25)/100);

llp1.gravity= Gravity.*CENTER*;

image1.setLayoutParams(llp1);

image2.setLayoutParams(llp1);

74

image3.setLayoutParams(llp1);

image4.setLayoutParams(llp1);

image5.setLayoutParams(llp1);

image6.setLayoutParams(llp1);

image7.setLayoutParams(llp1);

image8.setLayoutParams(llp1);

image9.setLayoutParams(llp1);

}

کد تابع InitView() :

**private** **void** InitView(){

image1=(customImageView)findViewById(R.id.*image1*);

image2=(customImageView)findViewById(R.id.*image2*);

image3=(customImageView)findViewById(R.id.*image3*);

image4=(customImageView)findViewById(R.id.*image4*);

image5=(customImageView)findViewById(R.id.*image5*);

image6=(customImageView)findViewById(R.id.*image6*);

image7=(customImageView)findViewById(R.id.*image7*);

image8=(customImageView)findViewById(R.id.*image8*);

image9=(customImageView)findViewById(R.id.*image9*);

image1.context=getBaseContext();

image2.context=getBaseContext();

image3.context=getBaseContext();

image4.context=getBaseContext();

image5.context=getBaseContext();

image6.context=getBaseContext();

75

image7.context=getBaseContext();

image8.context=getBaseContext();

image9.context=getBaseContext();

image1.setOnClickListener(onclick());

image2.setOnClickListener(onclick());

image3.setOnClickListener(onclick());

image4.setOnClickListener(onclick());

image5.setOnClickListener(onclick());

image6.setOnClickListener(onclick());

image7.setOnClickListener(onclick());

image8.setOnClickListener(onclick());

image9.setOnClickListener(onclick());

resartButton=(Button)findViewById(R.id.*restartButton*);

resartButton.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View view) {

recreate();

}

});

timerTextView=(TextView)findViewById(R.id.*timerTextView*);

}

76

از دیگر اتفاقاتی که در تابع بالا رخ داده اینست که context هر شی تصویر از context اصلی برنامه گرفته شده و هر شی تصویر موقع کلیک شدن تابع onclick() را فراخوانی میکند.

تابع recreate() در دستور دکمه ی restart نیز همانطور که از اسمش پیداست صفحه ی بازی را بارگذاری مجدد میکند.

حالا میرسیم به اصلی ترین قسمت کد برنامه ، جایی که ما کلیک را انجام میدهیم و برنامه بررسی میکند که آیا کلیک اول است یا دوم و درنهایت نتیجه ی آن کلیک ها چیست (برد ، باخت ، یا هیچکدام)

کد زیر را برای ساختن تابعی که اشیا تصویر ما روی آن setOnClickListener شده اند مینویسیم:

**private** OnClickListener onclick(){

**return** **new** View.OnClickListener() {

تابع onClick() را بصورت Override مینویسیم:

**public** **void** onClick(**final** View arg0) {

**int** imageRes=bmpArray[Integer.*parseInt*(((customImageView)arg0).getTag().toString())-1];

اتفاقی که در بالا رخ میدهد بدین شرح است که : بلوکی که روی آن کلیک کرده ایم در arg0 قرار میگیرد و تگ آن تصویر (که در فایل xml به هر تصویر دادیم) گرفته میشود ، چون آبجکت است به string تبدیل میشود و سپس آن را parse میکنیم به int و درنهایت یکی از آن کم میکنیم (چون آرایه ی ما از صفر شروع شده و تگ های ما از یک)

حالا کد زیر را برای وقتی که ما کلیک سوم را انجام داده ایم و به هیچ نتیجه ای نرسیده ایم ، مینویسیم.منظور از کلیک سوم اینست که ما عکس اول و دوم را انتخاب کردیم اما نه بردی بدست آورده ایم و نه بمب را مشاهده کرده ایم.پس لازم است تا عکس ها از حالت freeze دربیایند:

**if** (((customImageView)arg0).isFreez() || ((customImageView)arg0).tempFreez || ((customImageView)arg0).equals(tempCustomImageView) ) {

**return**;

}

77

در غیر اینصورت پس یا کلیک اول است یا کلیک دوم ، پس لازم است بلوک انتخاب شده شروع به چرخیدن کند:

((customImageView) arg0).startFlip(**true**,imageRes);

بررسی میکنیم آیا بلوک انتخاب شده "بمب" است یا خیر ، اگر بود بازی تمام شده است :

**if** (((customImageView)arg0).face==bombRes){

gameOver(**true**);

**return**;

}

اگر بمب هم نبود ، حالا اگر کلیک اول باشد آن را در tempCustomeImageView قرار میدهیم تا با کلیک بعدی آن را مقایسه کنیم ، سپس isFirst را درحالت false قرار میدهیم:

**if** (isFirst){

tempCustomImageView=((customImageView) arg0);

isFirst=**false**;

}

و اگر کلیک دوم بود کد زیر را نوشته ایم تا دو کلیک انجام شده با هم مقایسه شوند و در صورت برابری freeze شوند و یک score اضافه شود.بعد بررسی میشود اگر score ما به 4 رسیده بود به معنای آنست که بازی با برد تمام شده و زمان نهایی برای ارزش گذاری ثبت میشود.در غیر اینصورت بازی ادامه پیدا میکند و باقی بلوک ها unfreeze میشوند و isFirst باز true میشود :

**else**{

freezAll();

**int** tempImageRes=bmpArray[Integer.*parseInt*(tempCustomImageView.getTag().toString())-1];

**if** (tempImageRes==imageRes){

tempCustomImageView.setFreez();

((customImageView) arg0).setFreez();

score++;

78

**new** Timer().schedule(**new** TimerTask() {

@Override

**public** **void** run() {

MainActivity.**this**.runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

unFreezAll();

}

});

}

},2500);

**if** (score==4) gameOver(**false**);

}**else**{

**new** Timer().schedule(**new** TimerTask() {

@Override

**public** **void** run() {

MainActivity.**this**.runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

tempCustomImageView.startFlip(**false**, 0);

((customImageView) arg0).startFlip(**false**, 0);

unFreezAll();

}

});

}

}, 2500);

} isFirst=**true**; }

79

کدهای freeze و unfreeze کردن بلوک ها:

**private** **void** freezAll(){

image1.tempFreez=**true**;

image2.tempFreez=**true**;

image3.tempFreez=**true**;

image4.tempFreez=**true**;

image5.tempFreez=**true**;

image6.tempFreez=**true**;

image7.tempFreez=**true**;

image8.tempFreez=**true**;

image9.tempFreez=**true**;}

**private** **void** unFreezAll(){

image1.tempFreez=**false**;

image2.tempFreez=**false**;

image3.tempFreez=**false**;

image4.tempFreez=**false**;

image5.tempFreez=**false**;

image6.tempFreez=**false**;

image7.tempFreez=**false**;

image8.tempFreez=**false**;

image9.tempFreez=**false**;

}

آخرین تابع بازی هم تابع gameover() است که مربوط به پایان بازی و نشان دادن dialog می باشد.حال ابتدا به فایل dialog\_layout.xml میپردازیم :

80

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"200dp"*

android:layout\_height=*"100dp"*

android:orientation=*"vertical"*>

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Game Over"*

android:id=*"@+id/gameOverTextView"*

android:layout\_gravity=*"center\_horizontal"*/>

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"time"*

android:id=*"@+id/gameOverDialogTimeTextView"*

android:layout\_gravity=*"center"*/>

<TextView android:id=*"@+id/bestScore"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"center"*/>

<ImageButton

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:id=*"@+id/restartButton"*

android:background=*"@android:drawable/ic\_menu\_revert"*

android:layout\_gravity=*"center"*/>

</LinearLayout>

81

مثل فایل های xml قبلی در اینجا لازم است از لحاظ گرافیکی دکمه ها را بسازیم و به آنها اندازه و درصورت نیاز تصویر یا شناسه ی جدید بدهیم.برای دیدن dialog box به شکل14-پیغام پایان بازی (صفحه ی60) مراجعه کنید.

حال دوباره به کلاس جاوا MainActivity برمیگردیم.تابع gameover() را مینویسیم و هر کدام از اجزای آن را به قسمت مربوطه در xml مرتبط میکنیم:

**private** **void** gameOver(**boolean** lost){

isGameOver=**true**;

**final** Dialog dialog=**new** Dialog(**this**);

dialog.setContentView(R.layout.*dialog\_layout*);

TextView timeTextView=(TextView)dialog.findViewById(R.id.*gameOverDialogTimeTextView*);

TextView bestScoreTextView =(TextView)dialog.findViewById(R.id.*bestScore*);

ImageButton resetButton=(ImageButton)dialog.findViewById(R.id.*restartButton*);

یک SharedPreferences میسازیم تا به فضای در اختیار اپلیکیشن (خارج از SD Card) دسترسی داشته باشیم :

SharedPreferences shp =getSharedPreferences("score",MODE\_WORLD\_WRITEABLE);

دستور مربوط به دکمه ی restart :

resetButton.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View view) {

recreate();

dialog.dismiss();

}

});

82

دستور مربوط به پایان بازی ، اگر بازی با باخت تمام شده بود ، پیغام "You Lost" نمایش داده میشود و اگر بازی را برده باشیم زمان برد محاسبه میشود و با "بهترین امتیاز" مقایسه میشود تا در صورت پایین تر و بهتر بودن زمان جایگزین آن شود سپس با پیغام "You Win! Your Time is" به شما نشان داده میشود:

**if** (lost){

timeTextView.setText("you Lost");

}**else**{

timeTextView.setText("you win. your time "+ time/60+":"+time%60);

bestScoreTextView.setText("");

**if** (shp.getInt("bestScore",0)>time){

SharedPreferences.Editor shpeditor=shp.edit();

shpeditor.putInt("bestScore",time).commit();

}

}

محاسبات بهترین زمان و ذکر این نکته که اگر بهترین زمانی وجود نداشت نیازی به محاسبه نیست:

**int** x=shp.getInt("bestScore", 0);

**if** (x!=0)

bestScoreTextView.setText(x/60+":"+x%60);

و در نهایت نشان دادن dialog :

dialog.show();

حالا نوشتن بازی را به پایان رسانده اید.

برای تعیین اسم و آیکون بازی : به فایل string.xml در پوشه ی Values زیرشاخه ی res بروید و کد زیر را بنویسید و اسم بازی تان را انتخاب کنید ( در اینجا Memory Test نام بازی میباشد)

<string name=*"app\_name"*>Memory Test</string>

برای آیکون بازی هم عکس دلخواه خود را در پوشه ی drawable-hdpi (که پوشه قرار گرفتن همه ی عکس های بکار رفته در بازی است) زیرشاخه ی res قرار دهید.

83

حالا به فایل AndroidManifest.xml در پکیج برنامه ی خود بروید و کدهای زیر را در تگ application بنویسید:

android:icon=*"@drawable/iconam"*

android:label=*"@string/app\_name"*

و کد زیر را هم بنویسید تا نسخه ی سیستم عامل هدف و حداقل نسخه ی سیستم عامل برای اجرای بازی را مشخص کرده باشید.(البته برای تعیین نسخه سیستم عامل باید شماره sdk آن را بنویسید)

در اینجا حداقل سیستم عامل لازم برای نصب بازی نسخه 3.0 آندروید و نسخه ی هدف بازی آندروید 4.4 میباشد :

<uses-sdk

android:minSdkVersion=*"11"*

android:targetSdkVersion=*"19"* />

اکنون شما تمام قسمت های مربوط کدزنی و ساختن بازی را گذرانده اید ، حالا به آخرین قسمت پروژه که ساختن فایل نصب برنامه با پسوند .apk هست میرسیم :

روی بسته (Package) اصلی پروژه راست-کلیک کنید و گزینه ی Export را بزنید.پوشه ی Android را باز کنید و گزینه ی Export Android Application را انتخاب کنید و به صفحه ی بعد بروید . در این صفحه برای بررسی پروژه ایست که میخواهید از آن خروجی بگیرید. نام پروژه ی شما در کادر نوشته شده است پس به صفحه ی بعد بروید.

برای ساخت فایل apk نیاز به یک keystore دارید ؛ اگر از پیش ساخته اید مکان آن را انتخاب کنید و رمزعبور خود را بنویسید و به صفحه ی بعد بروید. اگر هم keystore ندارید یکی بسازید.

\*برای ساخت keystore در صفحه ی بعد باید نام و رمزعبور ، مقدار اعتبار و حداقل یکی از خانه های مشخصات دلخواه (مثل اسم و فامیل) را پرکنید.

در صفحه ی آخر مسیر ایجاد فایل را انتخاب کنید و Finish را بزنید.

84

**منابع:**

1. آندروید برای برنامه نویسان – مولفان : جیمز استیل،نلسون تو – مترجم:احمدرضا بقالی
2. درگاه ویکی پدیا
3. Android.developer.com
4. Github.com
5. Wrox java learning for android developers

**Abstract**:

This documentary is for an android game’s project. This project has been written in java language in an eclipse environment for android operating systems.

For this reason in this documentary there are first some short explanations about java programming language and then about eclipse environment and finally game building codes.

The minimum operating system’s versions for operating this android game are "honeycomb" and game’s objective version "jelly bean".

In this game you will face with nine blocks that depending on chance and memory must choose similar photos and take care of bomb. If you choose four similar photos you will have won the game.

**Keywords**: Game, Operating System, Android, Programming Language, Java.



**Allameh Dekhoda Higher Education Institute**

**DEPARTMENT OF COMPUTER ENGINEERING**

**“B.Sc” THESIS**

**SUBJECT:**

**Andriod Game: Memory Test**

**THESIS ADVISOR: Mr.Khazaei**

**BY:**

**Mohammad Amin Amini**

September 2014