

توسعه برنامه های

موبایل

جلسه هشتم مجازی

بخش دوم

سحر صادقی

آموزش ساخت برنامه اندروید به زبانی ساده؛

نرم افزار App Inventor به ما کمک کرد تا راحت تر از همیشه به ساخت برنامه اندروید بپردازیم. بعد از ایجاد نرم افزار اندرویدی خود، در آخرین قسمت کمی آن را توسعه می دهیم.

نرم افزار تحت وب App Inventor بهانه ای شد تا یک دوره برنامه نویسی مقدماتی و بسیار پایه ای اندروید را آغاز کنیم. این برنامه آنلاین که توسط محققان دانشگاه مطرح MIT توسعه پیدا کرده است، ابزارهای آماده زیادی در دل خود دارد که سبب می شود برای ساخت اپلیکیشن اندرویدی، تا حد ممکن نیاز به کدنویسی از بین برود. تنها کافی است قطعه کدهایی که به صورت بخش بندی شده تحت بلاک های گوناگون قرار گرفته اند را با استفاده از تکنیک کشیدن و رها کردن به درون قسمت های مختلف برنامه خود بیندازید.

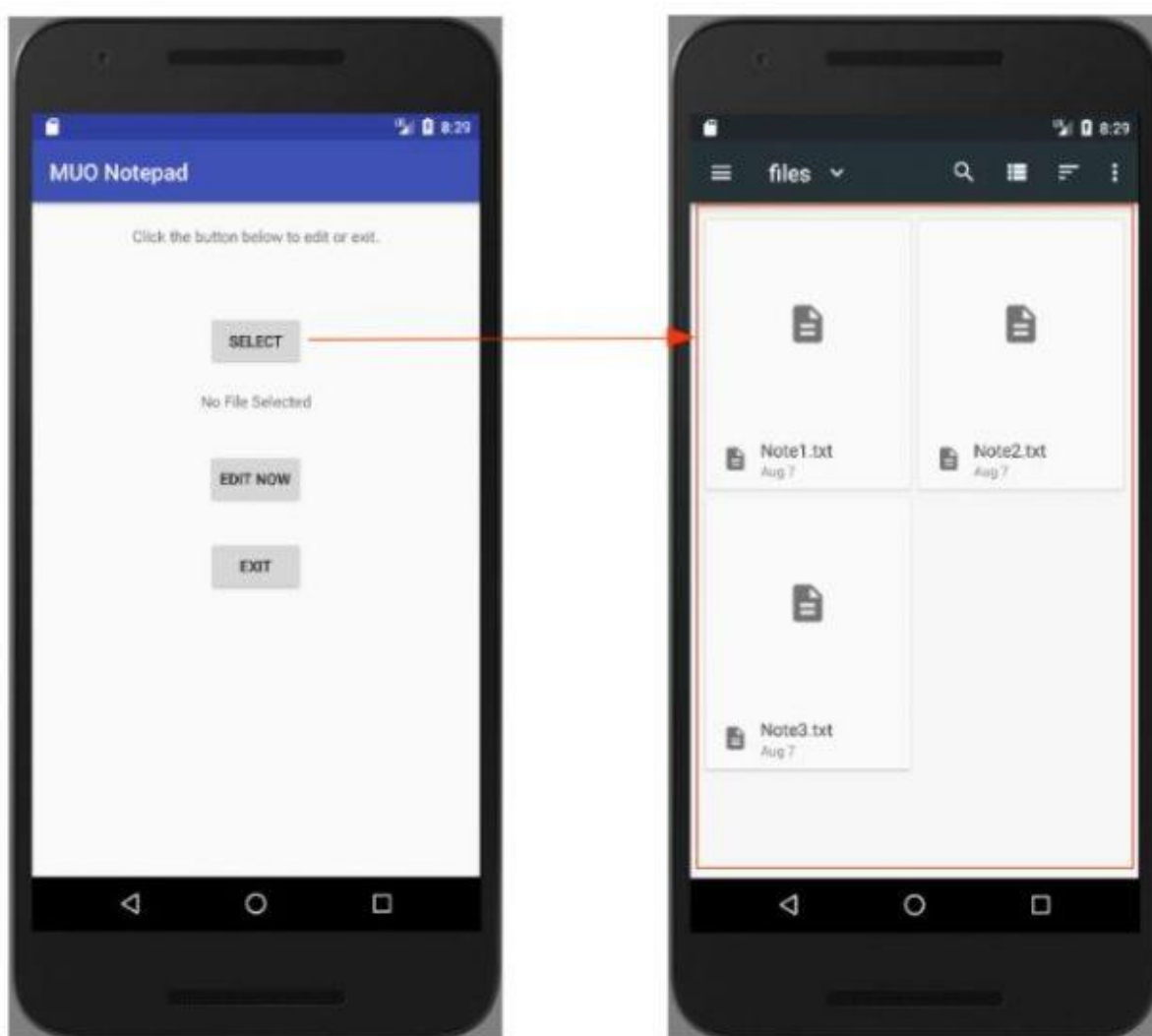
این قسمت ساخت برنامه اندروید آخرین مطلب بوده و در آن به توسعه اپلیکیشن نوت برداری ساده ای که ایجاد کرده ایم می پردازیم. پس در صورت عدم مطالعه قسمت های پیشین، ابتدا آن ها را به ترتیب از قسمت اول مرور کنید تا در جریان نحوه پیشرفت پروژه و ساخت نرم افزار اندروید مورد بحث قرار بگیرید.

اکنون که ورژن قابل استفاده اپلیکیشن خود را در قسمت قبل با استفاده از اندروید استودیو و زبان برنامه نویسی جاوا ایجاد کردیم، آن را با استفاده از همین محیط برنامه نویسی توسعه می دهیم. قابلیتی که قرار است بعد از ساخت برنامه اندروید به آن بیفزاییم، امکان انتخاب از میان چندین فایل داخل دایرکتوری استاندارد است. با افزودن این ویژگی، نرم افزار ما بیشتر شبیه به یک ابزار نوت برداری شده و دیگر تنها یک چرک نویس معمولی نخواهد بود.



همان طور که در اولین قسمت استفاده از اندروید استودیو به منظور آموزش برنامه نویسی اندروید به آن اشاره کردیم، App Inventor تا یک حد مشخص جوابگوی نیازهای ما است و برای جلو رفتن در پروژه، خواه ناخواه باید درگیر کدنویسی شویم. یکی دیگر از پیش نیازهای موفقیت در دنیای برنامه نویسی اندروید، تسلط به اندروید استودیو است. پس باید نحوه ایجاد یک کپی از پروژه کنونی را بلد باشید. در غیر این صورت از صفحه آموزش روش انجام این کار در وب سایت Stackoverflow کمک بگیرید.

ما از یک اندروید Intent برای فراخوانی صفحه (Activity) ویرایشگر از صفحه اصلی استفاده کردیم، اما روش ساده تری نیز به منظور صدا کردن دیگر اپلیکیشن ها وجود دارد. با اضافه کردن چندین خط کد، Intent ما درخواستی را برای اپلیکیشن های مدیریت فایل ارسال کرده و منتظر پاسخ می نشیند.



وادر کردن Intent به پس دادن یک رشته (شی متن جاوا) که بتوان از آن استفاده کرد کار ساده ای نبود، اما یک جستجوی ساده در گوگل سبب شد پاسخ پرسش خود را دریافت کنیم. این پاسخ ها برخی شامل کدهای آماده ای بودند که توانستیم خیلی راحت از آن ها حین ساخت برنامه اندروید خود استفاده کنیم.

On creation (initialization) of this Activity create some files...

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

    File workingDir = new File(String.valueOf(this.getExternalFilesDir(null)));
    File workingFile = null;
    String[] files = {"Note1.txt", "Note2.txt", "Note3.txt"};
    for (String s : files) {
        workingFile = new File(workingDir, s);
        if (!workingFile.exists())
            try {
                workingFile.createNewFile();
            } catch (Exception e) {
                Context context = getApplicationContext();
                CharSequence toastText = e.toString();
                Toast errorToast = Toast.makeText(context, toastText, Toast.LENGTH_LONG);
                errorToast.show();
            }
    }
}
```

... when the "Select" button is clicked, call file managers for a result, then run "onActivityResult..."

```
final Button selectButton = (Button) findViewById(R.id.selectButton);

selectButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent fileSelectIntent = new Intent(Intent.ACTION_OPEN_DOCUMENT);
        fileSelectIntent.setType("text/plain");
        fileSelectIntent.addCategory(Intent.CATEGORY_OPENABLE);

        try {
            startActivityForResult(fileSelectIntent, PICK_FILE_CODE);
        } catch (ActivityNotFoundException e) {
            Context context = getApplicationContext();
            CharSequence toastText = e.toString();
            Toast errorToast = Toast.makeText(context, toastText, Toast.LENGTH_LONG);
            errorToast.show();
        }
    }
});
```

... when a result is provided, assign it to a new label...

```
selectedFile = (TextView) findViewById(R.id.selectedFile);
```

با اعمال این تغییرات و کمک گرفتن از چند خط کد آماده، اکنون می‌توان از یک اپلیکیشن مدیریت فایل استفاده کرد تا به منظور ذخیره‌سازی محتوای خود، یک فایل انتخاب کنیم. در این بین می‌توان چند قابلیت ساده اما کاربردی دیگر را نیز به اپلیکیشن اضافه کرد:

- می‌توان از بین چندین فایل موجود یکی را انتخاب کرد، اما در حال حاضر، امکان ایجاد آن‌ها وجود ندارد. به این منظور، به یک ویژگی نیاز خواهیم داشت تا یک اسم فایل فراهم کرده و بعد از ایجاد، آن فایل را انتخاب کند.
- می‌توان کاری کرد تا اپلیکیشن نوت‌برداری ما به درخواست‌های اشتراک‌گذاری پاسخ بدهد. به این ترتیب می‌توان (به عنوان نمونه) حین مرور صفحات وب، یک آدرس اینترنتی را با برنامه به اشتراک گذاشت و آن را در یکی از فایل‌های نوت خود ذخیره کرد.
- ما بعد از ساخت برنامه اندروید خود با یک سری متن خالی سر و کله می‌زنیم، اما در این دسته از اپلیکیشن‌ها، معمولاً باید بتوان محتوایی غنی‌تر ایجاد کرد. این دسته از نوت‌ها حداقل باید دارای قابلیت اضافه شدن عکس باشند.

جاوا به اندازه‌ای گسترده است تا این قابلیت‌ها و هر چیز دیگری که به ذهنتان می‌رسد با استفاده از آن قابل پیاده‌سازی باشد.

توزیع اپلیکیشن اندرویدی ایجاد شده

اکنون که پروژه به اتمام رسیده، باید بدانید هدف نهایی تان چیست و می‌خواهید چه کاری با آن انجام بدهید. شاید تنها هدف شما از ساخت برنامه اندروید در اختیار داشتن یک اپلیکیشن تماماً شخصی و سفارشی‌سازی شده است. این برنامه شاید به اندازه‌ای شخصی توسعه پیدا کرده که هیچ کاربر دیگری نتواند از آن استفاده کند. توصیه این است که هرگز چنین اپلیکیشن اندرویدی ایجاد نکرده و حتی در صورت اطمینان از تصمیم خود برای عدم انتشار عمومی نرم‌افزار، آن را به صورت استاندارد ایجاد کنید.



بعد از ساخت برنامه اندروید و پیش از نصب آن بر روی تبلت یا گوشی موبایل خود باید چند قدم بردارید. باید یاد بگیرید چگونه اپلیکیشن خود را در قالب یک پکیج قابل نصب در بیاورید.

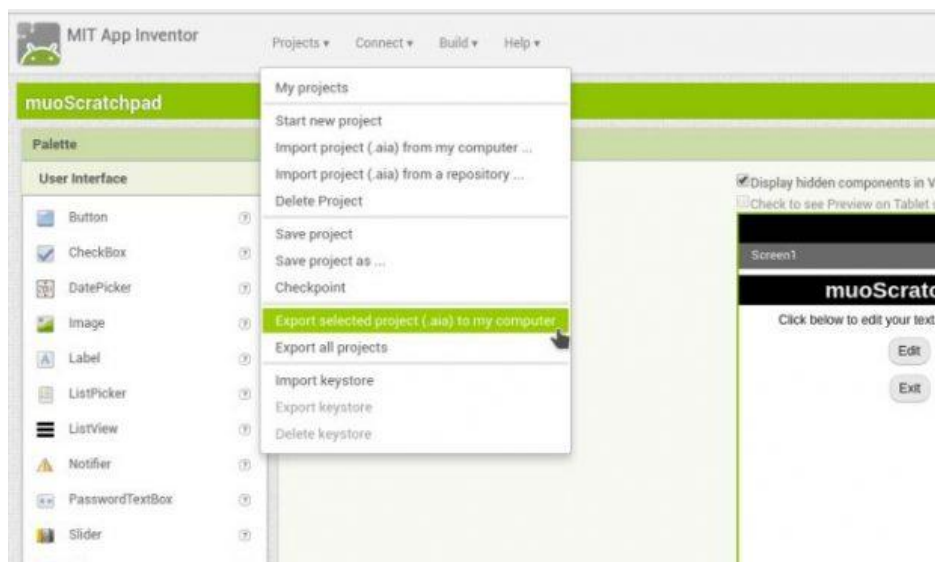
توزیع کد منبع اپلیکیشن

همه کاربران چنین تفکری نداشته و اغلب دوست دارند برنامه‌شان را با دیگران به اشتراک بگذارند. در هر صورت دنیای پلتفرم‌های متن‌باز، روی دست توسعه‌دهندگان می‌چرخد و هر حرکت می‌تواند کمکی به رشد آن‌ها باشد. یکی دیگر از مزایای قرار دادن کد منبع نرم‌افزار در اختیار دیگران، بهره بردن از تجربیات آن‌ها است. سایر افراد بر روی اپلیکیشن شما کار کرده و خطاهای آن را رفع می‌کنند و یا قابلیت‌های تازه به آن می‌افزایند. به این ترتیب خودتان نیز از مزایای اشتراک‌گذاری کد منبع برنامه سود خواهید برد.

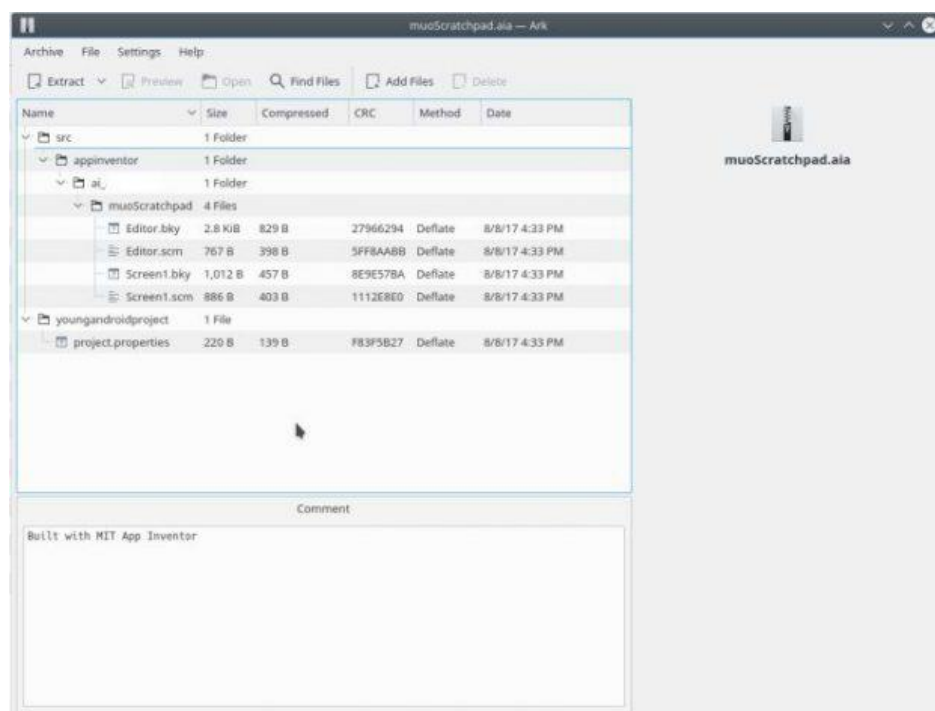
یکی از مهمترین ویژگی‌های App Inventor که آن را تبدیل به ابزاری خوب برای ساخت برنامه اندروید می‌کند، مخفی کردن و عدم نمایش کدهای اغلب در هم پیچیده برنامه به کاربر است. در واقع پشت تمامی بلاگ‌هایی که در جای جای اپلیکیشن قرار می‌دهید، بسته به کاربرد هر یک، چندین خط کد قرار دارد و کاربر

آن را به صورت حاضر و آماده استفاده می کند. اما برای گرفتن خروجی از کد اپلیکیشن نوشته شده در App Inventor چه باید کرد؟

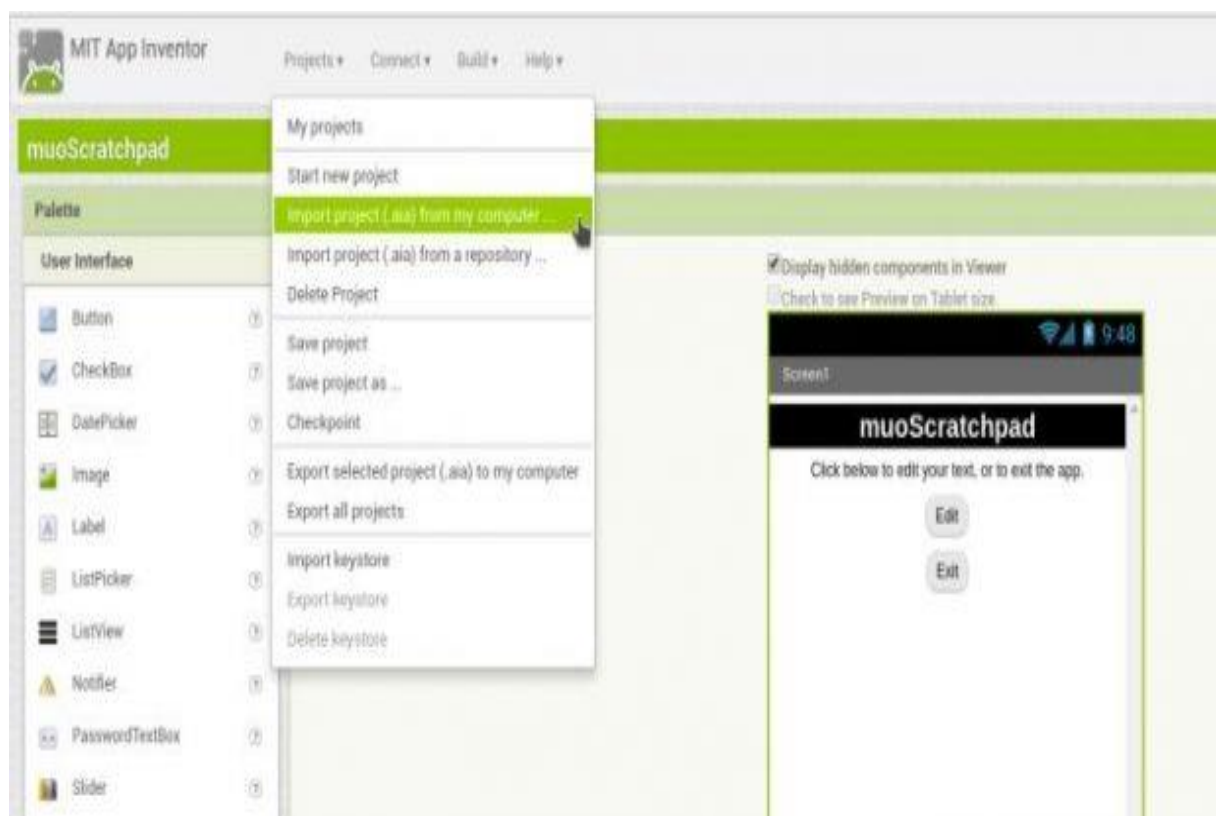
ابتدا پروژه خود را باز کرده و سپس از منوی Projects، گزینه Export Selected Project (.aia) To My Computer را انتخاب کنید.



با این کار، فایل دارای پسوند aia دانلود خواهد شد. به نظر می رسد این فرمت آرشیو App Inventor است. اما در نهایت با یک فایل Zip سر و کار خواهید داشت. آن را با استفاده از یک نرم افزار همچون Winrar کنید.

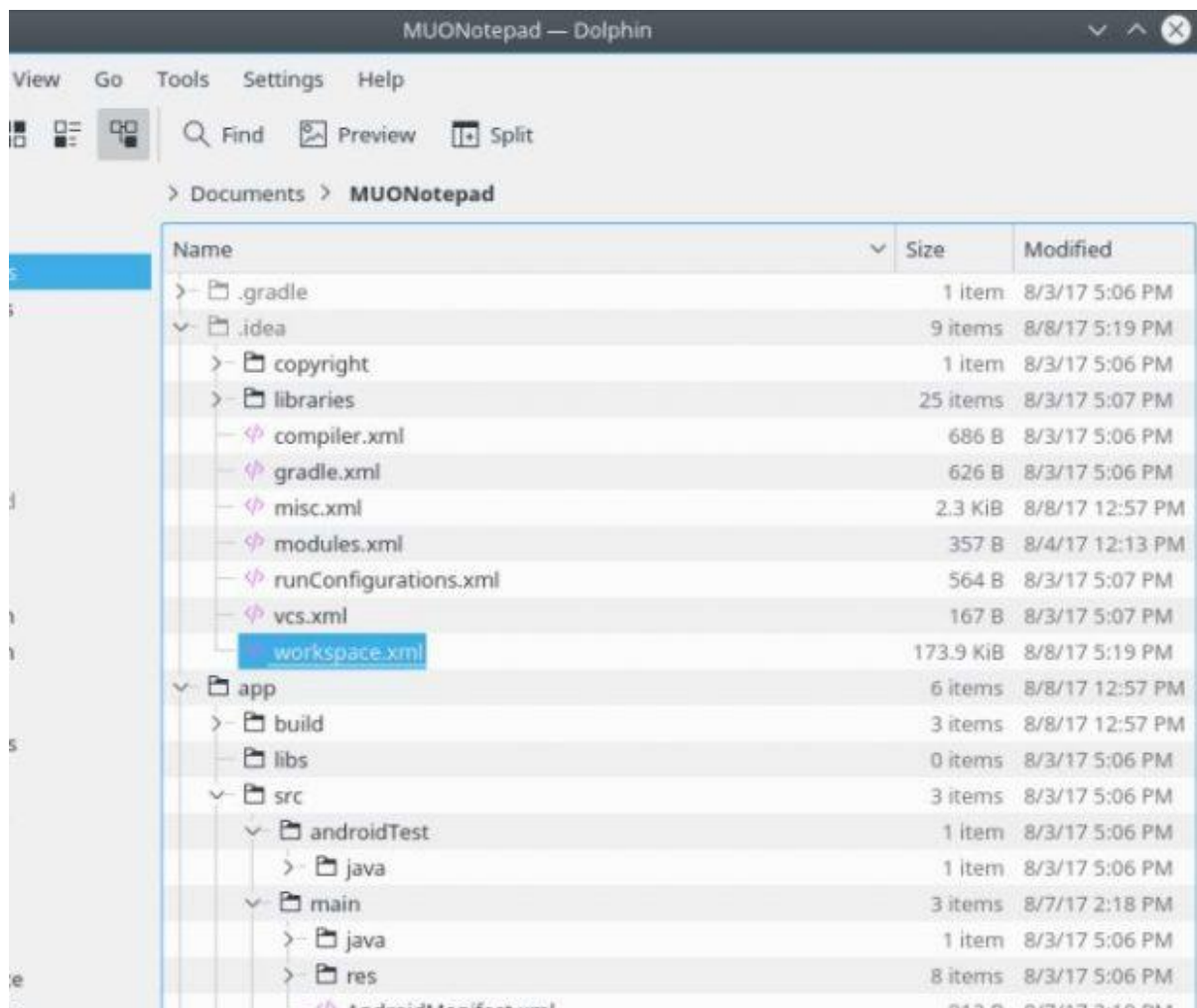


توجه کنید محتویات فولدر `appinventor/ai_[your user id]/[project name]` و `BKY` و `SCM` است. این‌ها کدهای منبعی که در اندروید استودیو مشاهده می‌کنیم نیستند، به همین دلیل قادر به باز کردن آن‌ها در محیط‌های برنامه نویسی قدیمی و کامپایل کردنشان نخواهید بود. با این حال می‌توان در هر زمان دیگر، آن‌ها را در `App Inventor` بارگذاری (Import) کرده و کار توسعه و یا تغییر کدهای نوشته شده را از سر گرفت.



دریافت کد اپلیکیشن حین توسعه آن در اندروید استودیو کار ساده‌تری است. کافی است به قسمت پروژه‌ها وارد شده و فولدر مربوط به این پروژه را تبدیل به یک فایل Zip کنید. سپس آن را به محلی دیگر انتقال داده و به صورت عادی باز کنید.

اندروید استودیو تنظیمات پروژه (workspace.xml) را خوانده و همه چیز را به حالت عادی و زمان اولیه ساخت برنامه اندروید برمی‌گرداند.

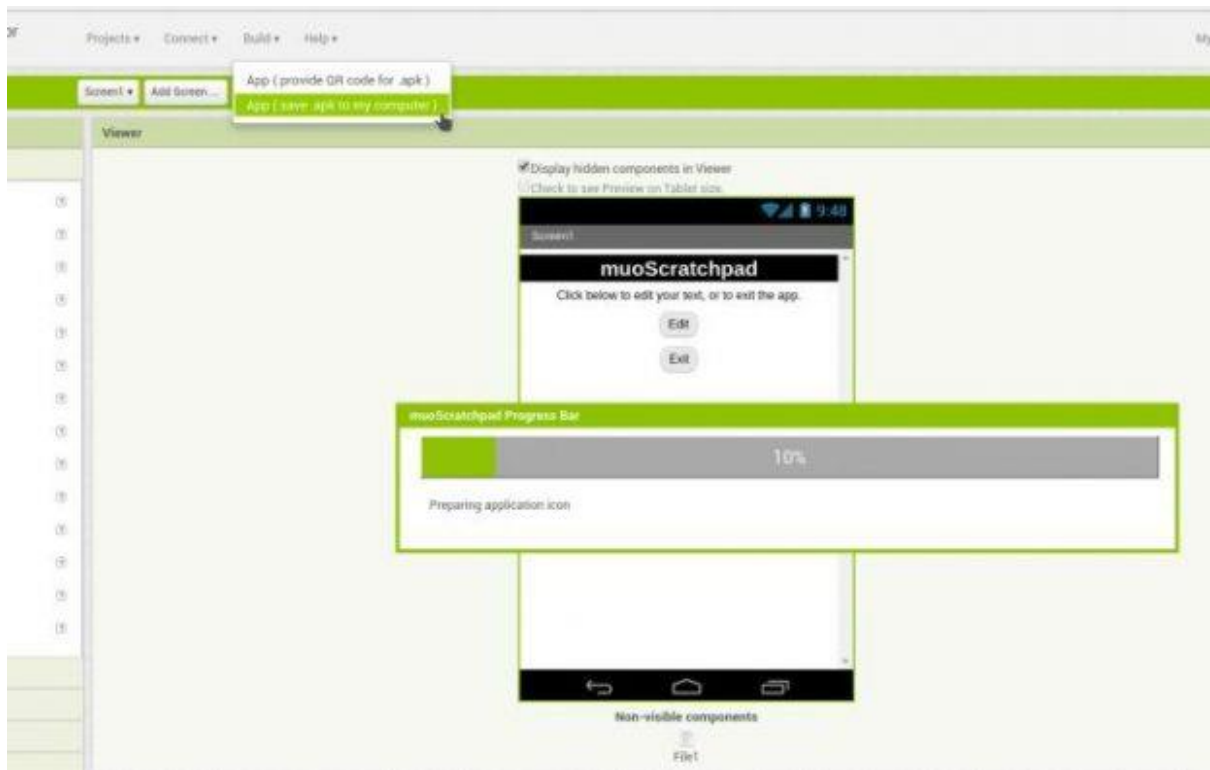


البته با فشردن کل فولدر، چندین فایل اضافی نیز ذخیره می‌شوند، در حالی که امکان ایجاد مجدد آن‌ها
 حین سری‌های بعدی بیلد کردن پروژه وجود دارد. در هر صورت بودنشان ضرری ندارد، اما با آن‌ها، آرشیو کد
 منبع اپلیکیشن تمیزتر خواهد بود.

اگر می‌خواهید بعد ساخت برنامه اندروید خود آن را برای تست یا هر دلیل دیگر در اختیار فرد دیگری قرار
 دهید، بهترین و سراسرترین کار، ارسال فایل نصبی دارای پسوند apk برای وی است. کمک گرفتن از وب
 سایت‌هایی همچون F-Droid می‌تواند پیشنهاد خوبی باشد.

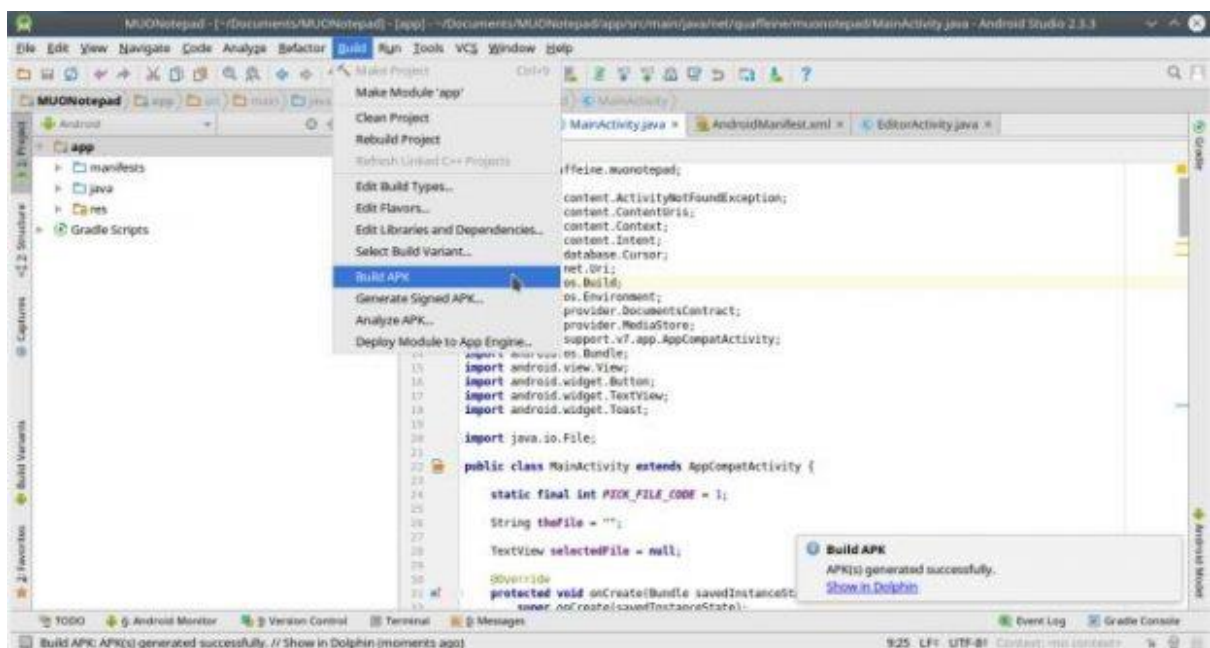
ایجاد فایل دارای پسوند apk در App Inventor

به این منظور به منوی Build رفته و گزینه (App (Save .apk To My Computer را انتخاب کنید. با این
 کار ساخت فایل نصبی اپلیکیشن آغاز می‌شود. بعد از اتمام قادر به دریافت فایل نهایی خواهید بود.



ایجاد فایل دارای پسوند apk در اندروید استودیو

کافی است بعد از ساخت برنامه اندروید در منوی Build، گزینه Build APK را انتخاب کنید تا خیلی راحت بسته نصبی اپلیکیشن ایجاد شود. به محض اتمام فرآیند، یک پیغام، مسیر ذخیره شدن فایل را نمایش می دهد.



اما قرار دادن اپلیکیشن در گوگلی پلی استور نیاز به صرف زمان بیشتری دارد. متأسفانه این کار برای توسعه‌دهندگان ایرانی به سختی امکان‌پذیر است. پیش از همه باید ۲۵ دلار به منظور ثبت نام پرداخت کنید. چندین عمل جزئی دیگر نیز وجود دارد که یکی از مهمترین آن‌ها، ایجاد یک کلید رمزنگاری به منظور امضا کردن اپلیکیشن‌های خود با آن است. اگر به هر دلیلی این کلید را گم کنید، دیگر قادر به به‌روزرسانی اپلیکیشن نخواهید بود.

اما برای آن دسته از توسعه‌دهندگان که بعد از ایجاد چندین اپلیکیشن اندرویدی، اکنون به طور جدید تصمیم دارند فعالیت در گوگلی پلی استور را آغاز کنند، سه فرآیند مهم وجود دارد:

- ثبت نام به عنوان یک توسعه‌دهنده: با ورود به این صفحه در وب سایت Google می‌توانید پروفایل خود را به عنوان یک توسعه‌دهنده تکمیل کنید. در ادامه وارد یک صفحه ثبت نام می‌شوید که شامل چندین گزینه است. یکی از بخش‌های ثبت نام، پرداخت ۲۵ دلاری است که در بالا به آن اشاره کردیم.
- آماده کردن اپلیکیشن برای فروشگاه: نسخه‌های شبیه‌ساز برنامه که تاکنون آن‌ها را تست می‌کردید، به عنوان نسخه‌های اشکال‌زدایی شناخته می‌شوند. پس آن‌ها قطعات کد به مراتب بیشتری دارند که مربوط به عیب‌یابی و ورود به سیستم هستند که البته ضرورتی نیز ندارند. حتی ممکن است وجودشان نگرانی‌هایی از لحاظ حریم خصوصی ایجاد کند. پیش از انتشار در گوگل پلی استور، می‌بایست یک نسخه نشر با استفاده از روش‌های ذکر شده در وب سایت Developer.Android ایجاد کنید. یکی از مراحل این فرآیند، قرار دادن یک کلید رمزنگاری در اپلیکیشن است.



- سازمان خود را بنا کنید: یکی دیگر از مراحل اشتراک‌گذاری در گوگل پلی بعد از ساخت برنامه اندروید، ایجاد یک صفحه فروشگاه برای خود است. سازمان و زیربنای شما ممکن است شامل چندین سرور باشد که قرار است اپلیکیشن، داده‌های خود را با آن‌ها همگام‌سازی کند. در صورت نیاز باز هم می‌توانید از وب سایت Developer.Android کمک بگیرید.

- در نهایت اگر می‌خواهید پرداختی بابت این اپلیکیشن به شما صورت بگیرد، می‌بایست مشخصات پرداختتان را تکمیل کنید. این مورد جزییات زیادی داشته و کمی زمان‌بر است.

خلاصه کلاس ساخت برنامه اندروید و نتیجه‌گیری

به بهانه App Inventor یک دوره پنج جلسه‌ای برای آموزش ساخت برنامه اندروید ترتیب دادیم که سبب شد افراد بدون تجربه برنامه‌نویسی اندروید کمی با مقدمات آشنا شده و افراد دارای تجربه بیشتر، به معلومات خود اضافه کنند. در آخر متوجه شدیم اگرچه ابزارهای آنلاین و حاضر و آماده مختلفی وجود دارند، اما از یک جایی به بعد، دست و پنجه نرم کردن با کدنویسی غیر قابل چشم‌پوشی است و نمی‌توان یک اپلیکیشن بسیار خوب و کاربردی را تنها با چند کلیک ساده ایجاد کرد. درس‌هایی که از این دوره آموزشی ساخت برنامه اندروید گرفتیم را به صورت زیر خلاصه می‌کنیم:

- دو روش را برای ایجاد یک اپلیکیشن اندرویدی بررسی کردیم؛ نخست از تکنیک‌های ساده کشیدن و رها کردن و کار با موس به واسطه ابزار آنلاین App Inventor استفاده کرده و سپس اپلیکیشن را از ابتدا، با کدنویسی در اندروید استودیو ایجاد کردیم. مورد نخست نیاز به آموزش کمتری داشت و با وجود محدودیت‌های آن، کاربر می‌توانست از قطعه‌های مختلف بدون درگیر شدن با کدهای پشت آن‌ها استفاده کند. مورد دوم، نیاز به زمان بیشتری داشته و نمی‌توان از آن طی مدت زمانی کوتاه استفاده کرد. اما از طرفی دیگر، بدون محدودیت است. با زبان جاوا هر کاری می‌توان انجام داد.



- هر یک از این روش‌ها مزایا و معایب خود را دارند، پس می‌توان از هر دوی آن‌ها به صورت ترکیبی استفاده کرد. به عنوان نمونه، می‌توان همانند روشی که ما در این دوره انجام دادیم، چهارچوب کلی اپلیکیشن و نمونه اولیه آن را خیلی ساده و سریع در App Inventor ایجاد و سپس قابلیت‌های بیشتر را به وسیله اندروید استودیو

به آن اضافه کرد. برای این کار باید روش دریافت کد منبع از App Inventor که در همین قسمت آموزش به آن اشاره کردیم را بلد باشید.

- ابتدا بر روی کاربرد اپلیکیشن و روش کار آن تمرکز کرده و سپس برای زیباسازی آن زمانی را صرف کنید. اگرچه کاربردی بودن نرم افزار اصلی ترین کلید است، اما نباید از اهمیت زیبایی رابط کاربری غافل شد.
- مهمترین چیز در برنامه نویسی، توانایی یافتن نیازهای خود از وب است. اگر در جایی به مشکل برخورد کردید، ناامید نشوید، چرا که حتی بهترین و خبره ترین توسعه دهندگان نیز درگیر پیچیدگی چند صد و یا هزار خط کد می شوند. مشکل را به زبانی ساده و دارای مفهوم در گوگل جستجو کرده و در بین نتایج، به دنبال راه حل مشکل خود بگردید.
- حین ساخت برنامه اندروید هر از گاهی آن را تست کنید. به این ترتیب ایرادیابی بسیار راحت تر خواهید بود. اگر بعد از اضافه کردن چندین قابلیت و نوشتن چند صد خط کد، حین تست در شبیه ساز به مشکل برخورد کنید، پیدا کردن ایراد شدیداً طاقت فرسا خواهد بود.