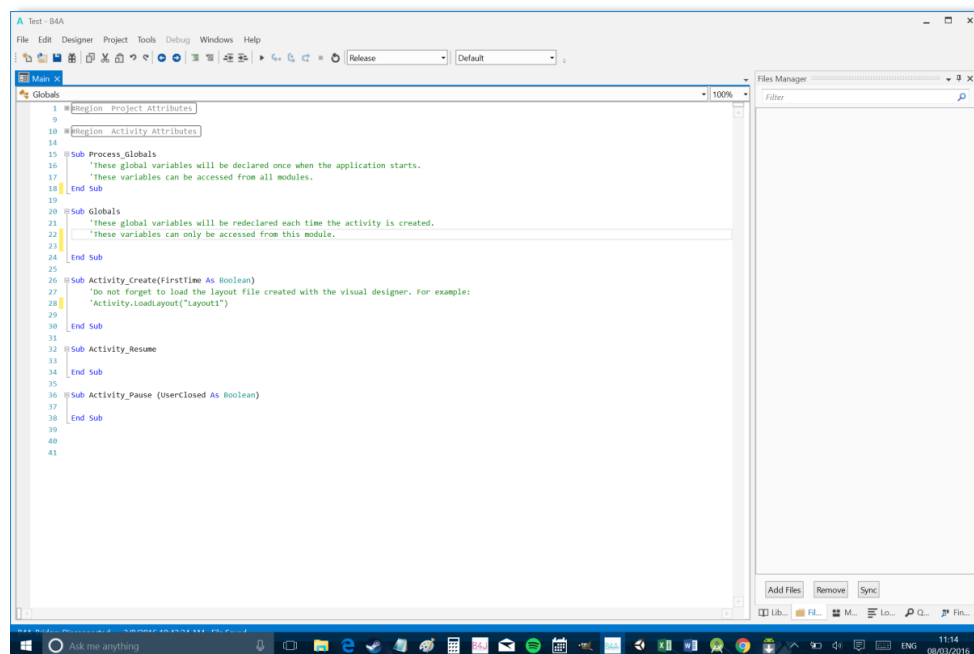


ساخت اولین برنامه با B4A

1- مقدمه‌ای بر B4A

کلید انجام کارهای زیاد این است که مطمئن شوید ابزار مناسبی برای کار دارید و وقتی صحبت از توسعه موبایل می‌شود، این به معنای انتخاب IDE مناسب برای برنامه خاصی است که می‌خواهید ایجاد کنید. اگر می‌خواهید چیزی بسازید که با زیبایی‌های اندروید سازگار باشد یا اگر می‌خواهید برای اطمینان از حداکثر پشتیبانی از رایج‌ترین مسیر استفاده کنید، Android Studio عالی است. Unity برای ساخت بازی‌های سه بعدی بدون نیاز به استخدام یک تیم کامل از توسعه‌دهندگان بسیار مناسب است. و همچنین Basic4Android یکی از بهترین گزینه‌ها برای مواردی است که می‌خواهید به سرعت یک برنامه اندروید بسازید. همچنین فکر می‌کنیم این یک ابزار نسبتاً خوب برای مبتدیان و ایجاد برنامه‌های بین پلتفرمی (cross platform) است.

قبل از ادامه کار، توجه داشته باشید که Basic4Android یک نرم افزار رایگان نیست. اگر به IDE‌های پرداخت‌کردنی علاقه ندارید، می‌توانید روش دیگری را امتحان کنید. همچنین یک دوره آزمایشی رایگان در وبسایت (www.b4x.com) وجود دارد و ثبت سفارش نیز باعث دسترسی شما به انجمن می‌شود (که با افراد بسیار مفیدی پر شده است). Basic4Android یک ابزار IDE (محیط توسعه یکپارچه) و "RAD" (توسعه سریع برنامه) است. همانطور که از این عنوان پیداست، هدف آن تسهیل ایجاد سریع و آسان برنامه‌ها است. در عین حال، سعی می‌کند این کار را بدون محدود کردن شما به هیچ وجه در حد توانایی برنامه‌های شما انجام دهد.



Basic4Android یک برنامه ساز (app builder) که کاربران بدون کدنویسی برنامه بسازند نیست. شما باید برای ساخت برنامه کدنویسی کنید اما برای مبتدیان دسترسی به آن بیشتر است و بسیاری از ویژگی‌های جالبی برای ساده‌سازی روند وجود دارد و به هیچ وجه محدود به آنچه می‌توانید ایجاد کنید نیستید. سناریوهایی وجود دارد که Basic4Android بهترین انتخاب برای پروژه شما نخواهد بود. همانطور که در ابتدا گفتیم، انتخاب ابزار مناسب برای کار بخشی از این چالش است. اما به عنوان کسی که می‌تواند از IDEهای متعددی استفاده کند، بازهم به لطف ساده و سرراست بودن این زبان، برای انجام چندین کار مختلف به استفاده از B4A روی می‌آوریم. در ادامه شرحی بر زبان BASIC که در Basic4Android مورد استفاده قرار می‌گیرد داریم.

اگر در برنامه‌نویسی تازه‌کار هستید، ممکن است دریابید که BASIC کمی راحت‌تر از جاوا است. نحو BASIC اغلب بیشتر شبیه به انگلیسی ساده و روان است و این یکی از ویژگی‌هایی است که باعث می‌شود کاربر پسند باشد. برای نشان دادن این مورد، این دو خط کد را که یک کار را انجام می‌دهند مقایسه کنید:

```
BASIC
```

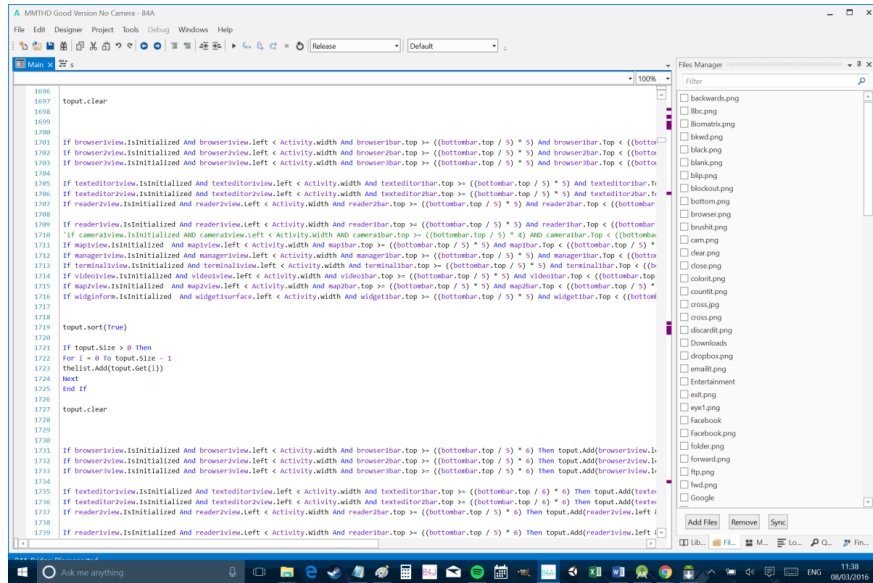
```
IF level = 3 THEN titlebar.Text = "level 3"
```

در قبال:

```
JAVA
```

```
if (level == 3){  
    titlebar.setText("level 3");  
}
```

مورد اول بسیار مشهودتر و ساده‌تر دومی است و اگر در برنامه‌نویسی تازه‌وارد باشید راحت‌تر آن را درک می‌کنید. (و هیچ وجود ندارد که آن را فراموش کنید!) جاوا همچنین شی‌گراتر از Basic4Android است. این مورد ممکن است توسط برخی به عنوان یک مزیت شناخته شود و برای پروژه‌های بزرگتر مفید باشد. هرچند برای یک مبتدی، کمی کار بیشتر لازم است تا با آن سر و کله بزند.



Basic4Android کارهای زیادی را در پشت صحنه برای شما انجام می‌دهد. به عنوان مثال، برای آنکه کد دوم در Java/Android Studio کار کند، باید کلاس مربوطه را import کنید. به همین ترتیب، ایجاد متغیرهای سراسری بسیار آسان‌تر است، نوشتن رشته‌ها نیز آسان است، همچنین تنظیم تایمرها و ... در هر دو مورد لازم است ویجتی را که به آن رجوع می‌کردید را مقداردهی اولیه کنید، اما Basic4Android انجام این کار را برای شما ساده‌تر کرده است. حتی راه‌اندازی Basic4Android مراحل بسیار کمتر و بسیار آسان‌تر از راه‌اندازی Android Studio دارد. در مورد اشکال‌زدایی و تست نیز همین مورد وجود دارد. می‌توانید از B4A Bridge (یک برنامه تلفن همراه رایگان) برای آزمایش برنامه‌های دستگاه خود از طریق بلوتوث استفاده کنید و سپس هنگام اجرا، اشکال‌زدایی آنها را انجام دهید.

1-1- آیا Basic4Android می‌تواند هر آنچه را که نیاز دارید انجام دهد؟

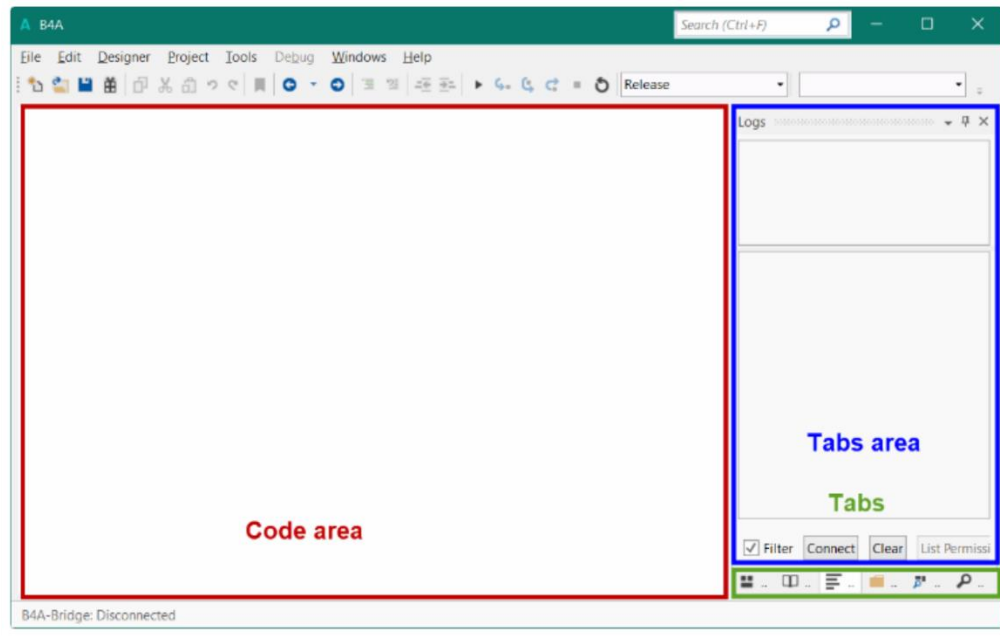
سوالی که احتمالاً بسیاری از افراد می‌پرسند این است: "آیا واقعاً می‌توانید در Basic4Android کاری انجام دهید که در Android Studio انجام می‌دهید؟". در کل پاسخ این سوال مثبت است. Basic4Android به هر کاربر اجازه می‌دهد تا کتابخانه‌های خود را که به زبان جاوا نوشته شده‌اند ایجاد و به اشتراک بگذارد. هر کدی که می‌توانید با Android Studio استفاده کنید، قابلیت قرارگیری در کتابخانه B4A را دارند. از آنجا که نحو برای Basic4Android، Basic4Java و Basic4iOS تا حد زیادی یکسان است، می‌توانید یک برنامه ساده در رایانه ایجاد کنید و سپس برخی از کتابخانه‌ها و ویژگی‌های خاص پلتفرم را برای انتقال آن به Android و iPhone تغییر دهید. همچنین می‌توانید از B4J (که رایگان است) برای ایجاد برنامه‌های کوچک دستکاپ برای اهداف خود یا ایجاد برنامه‌هایی که نیاز به برقراری ارتباط با رایانه دارند (مانند کنترلرهای رسانه) استفاده کنید.

2- نصب B4A و Android SDK

B4A یک ابزار 100٪ توسعه آزاد برای برنامه‌های آندروید است و همچنین شامل تمام ویژگی‌های مورد نیاز برای توسعه سریع هر نوع برنامه آندروید است. از دیگر ویژگی‌های مهم B4A این است که دیگر درگیر کار با xml نخواهید بود. شما می‌توانید با استفاده از موارد زیر برنامه‌ها را اشکال‌زدایی و توسعه دهید:

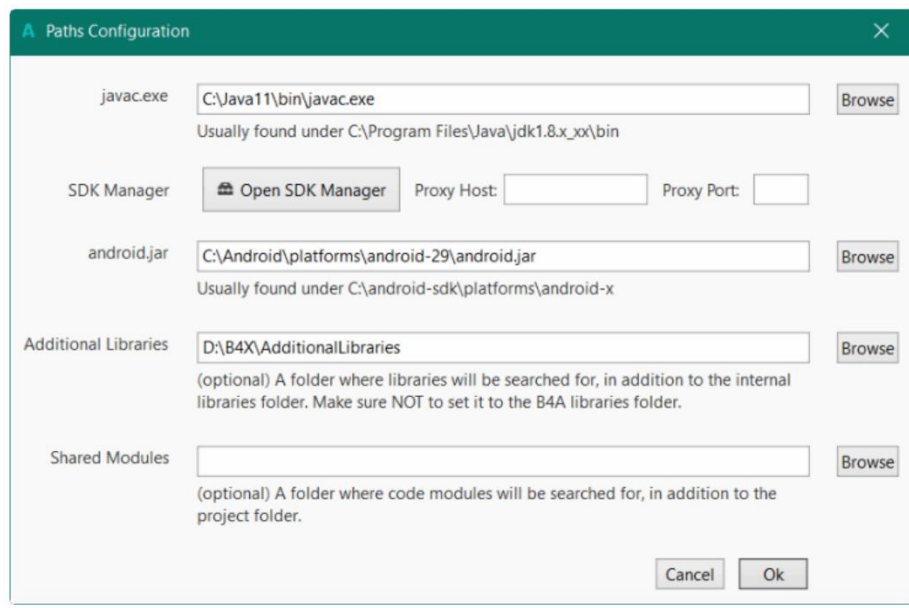
- یک دستگاه واقعی متصل از طریق B4A Bridge
- یک دستگاه واقعی متصل از طریق کابل USB
- یا یک شبیه‌ساز Android.

برای شروع کار ابتدا باید نرم‌افزار B4A را از اینترنت دانلود کنید و نصب کنید. زمانی که نرم‌افزار را اجرا می‌کنید با تصویر زیر روبرو خواهید شد.



2-1- پیکربندی مسیرهای B4A در IDE

نرم افزار را باز کنید. روی گزینه **Configure Paths** در منوی **Tools** کلیک کنید. پس از آن تصویر زیر را مشاهده خواهید کرد.



اگر از قبل جاوا روی سیستم شما نصب باشد (فرض می‌کنیم که جاوا نصب است)، مقدار فیلد `javac.exe` پر شده می‌باشد. در تمام این مراحل فرض شده است که **Android SDK** به طور کامل روی سیستم شما نصب شده است. مسیر را برای "`android.jar`" با عبارت مقابل تنظیم کنید:

`C:\Android\platforms\android-28\android.jar`

توصیه می‌شود یک پوشه خاص برای کتابخانه‌های اضافی ایجاد کنید. **B4A** از دو نوع کتابخانه استفاده می‌کند:

- کتابخانه‌های استاندارد که همراه **B4A** هستند و در پوشه **Libraries** نرم افزار **B4A** قرار دارند. هنگام نصب نسخه جدید **B4A**، این کتابخانه‌ها به طور خودکار به روز می‌شوند.
- کتابخانه‌های اضافی، که بخشی از **B4A** نیستند و بیشتر توسط برنامه‌نویسان نوشته می‌شوند. این کتابخانه‌ها باید در یک پوشه خاص متفاوت از پوشه کتابخانه‌های استاندارد ذخیره شوند.

2-2- اتصال یک دستگاه واقعی

روش‌های مختلفی برای اتصال یک دستگاه واقعی وجود دارد:

- **USB**، به پشتیبانی دستگاه از **ADB debugging** نیاز دارد و باید **USB Debugging** را روی دستگاه فعال کنید.
- **B4A Bridge**، از طریق **WiFi**.

2-2-1- اتصال از طریق USB

دراپور USB Google قبلاً به طور پیش فرض روی گوشی شما بارگیری شده است. اگر این دراپور کار نمی کند، باید به دنبال دستگاه دیگری باشید. برای اتصال دستگاه از طریق USB باید USB debugging را روی دستگاه فعال کنید. در صورت استفاده از شبیه ساز، این مورد نیز مورد نیاز است. روی تنظیمات گوشی کلیک کنید. وارد قسمت Developer options شوید و USB debugging را فعال کنید.

2-2-2- اتصال از طریق B4A-Bridge

همیشه توصیه می شود به جای شبیه ساز اندروید از یک دستگاه واقعی استفاده کنید چون شبیه ساز در مقایسه با دستگاه واقعی بسیار کند است (به خصوص با نصب برنامه ها). با این وجود همه دستگاه ها از ADB debugging پشتیبانی نمی کنند. این دلیل استفاده از ابزار B4A-Bridge است. B4A-Bridge از دو جز ساخته شده است. یک مؤلفه بر روی دستگاه اجرا می شود و به مؤلفه دوم که بخشی از IDE است امکان اتصال و ارتباط با دستگاه را می دهد. اتصال از طریق شبکه انجام می شود (B4A-Bridge در صورت عدم وجود شبکه نمی تواند کار کند). پس از اتصال، B4A-Bridge از تمام ویژگی های IDE پشتیبانی می کند که شامل: نصب برنامه ها، مشاهده LogCat و طراح بصری و ... می باشد.

3- شروع برنامه نویسی B4A با ساخت برنامه Hello World!

بیایید با یک "Hello World" ساده در Basic4Android شروع کنیم. روی گزینه new کلیک کنید و سپس پروژه خود را در جایی ذخیره کنید که بتوانید دوباره آن را پیدا کنید. برای شروع، ما قصد داریم یک برچسب ایجاد کنیم. ما می توانیم این کار را به صورت بصری انجام دهیم اما در این قسمت به صورت کدنویسی این کار را انجام می دهیم. این کار سریع و آسان است.

ابتدا یک برچسب را در Globals تعریف می کنیم. خط زیر را اضافه کنید:

```
Sub Globals
Dim label1 As Label
End Sub
```

در اینجا sub مانند یک متد رفتار می کند. اکنون می توانید از هر جای کد به label1 ارجاع دهید.

در ادامه باید یک متد به نام Activity_Create بنویسیم. Activity_Create متدی است که هنگام شروع activity برای اولین بار اجرا می‌شود. در اینجا قصد داریم مقدار اولیه label1 را مقداردهی کنیم و بگوییم که می‌خواهیم به چه صورت باشد. از کد زیر استفاده کنید:

```
Sub Activity_Create(FirstTime as Boolean)
    label1.Initialize("Label1")
    Activity.AddView(label1, 0%x, 0%y, 100%x, 100%y)
```

ما از این طریق برجسب را ایجاد کردیم و سپس آن را به activity (بخشی از برنامه در حال اجرا) اضافه کردیم در حالی که مکان و اندازه آن را نیز تعیین کردیم. مکان '0%x, 0%y' به این معنی است که در گوشه بالا سمت چپ صفحه است. عرض و ارتفاع '0%x, 0%y' به این معنی است که 100٪ عرض و 100٪ ارتفاع را اشغال می‌کند. اکنون برجسب ما دقیقاً همان اندازه صفحه است که البته نامرئی است. از آنجا که ما از درصد استفاده می‌کنیم، اندازه برجسب متناسب با اندازه نمایشگری که در حال اجرا است تغییر می‌کند. این خطوط پیام سلام ما را اضافه می‌کنند و اطمینان حاصل می‌کنند

```
label1.Gravity = Gravity.CENTER
label1.Text = "Hello World!"
```

که در مرکز برجسب شناور (float) است:

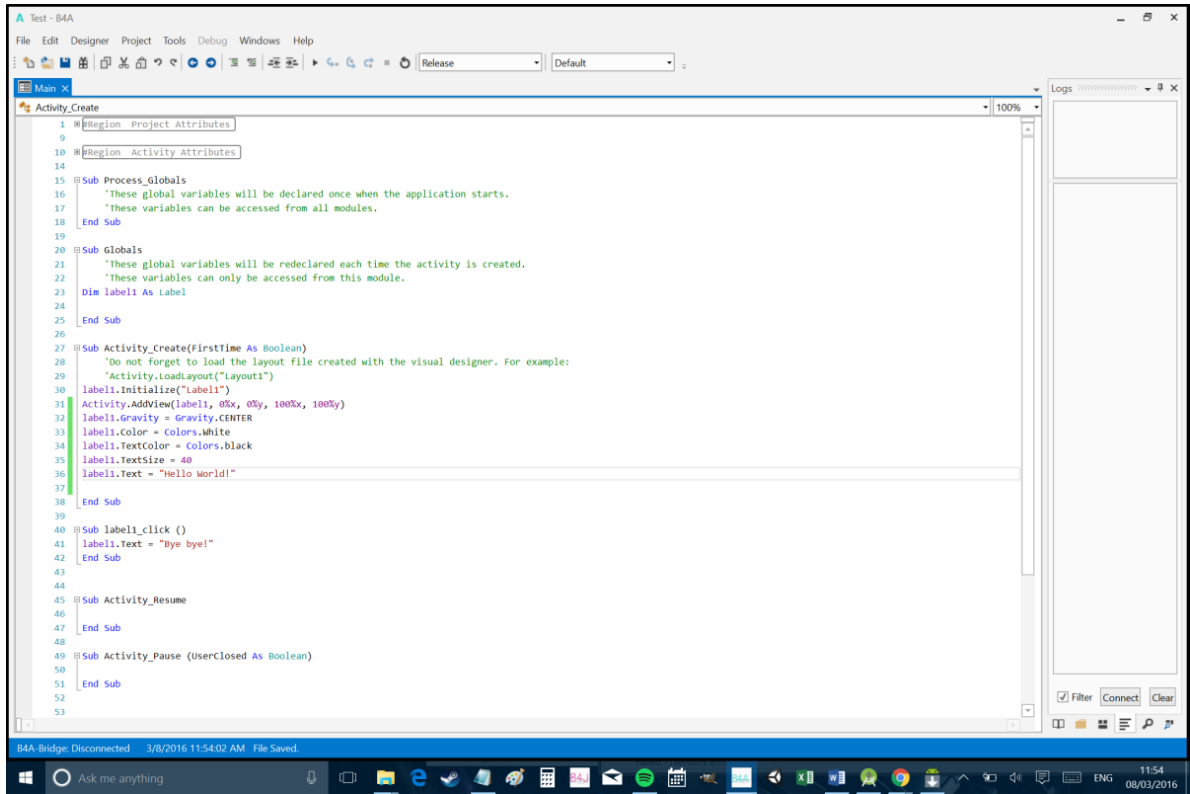
همچنین می‌توانیم خطوط زیر را اضافه کنیم:

این کار کمی سبک (style) به برجسب می‌دهد. در این قسمت "Hello World!" به پایان رسیده است اما برای ایجاد تعاملی بیشتر در کارها، می‌توانیم متد دیگری ایجاد کنیم:

با این کار به ازای کلیک روی برجسب عبارت Hello World! به Bye bye! تغییر میابد.

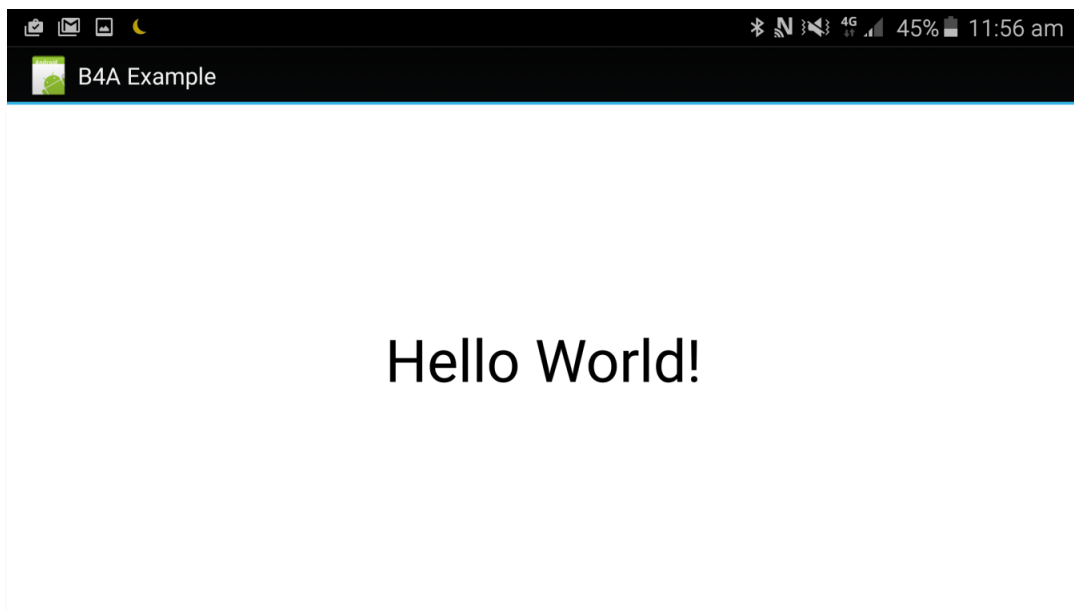
```
label1.Color = Colors.White
label1.TextColor = Colors.black
label1.TextSize = 40
Sub label1_click ()
    label1.Text = "Bye bye!"
End Sub
```

اگر کار را درست انجام داده باشید، چیزی شبیه به عکس زیر خواهد بود:



```
1 #Region Project Attributes
9
10 #Region Activity Attributes
14
15 @Sub Process_Globals
16     'These global variables will be declared once when the application starts.
17     'These variables can be accessed from all modules.
18 End Sub
19
20 @Sub Globals
21     'These global variables will be redeclared each time the activity is created.
22     'These variables can only be accessed from this module.
23 Dim label1 As Label
24
25 End Sub
26
27 @Sub Activity_Create(FirstTime As Boolean)
28     'Do not forget to load the layout file created with the visual designer. For example:
29     'Activity.LoadLayout("Layout1")
30 label1.Initialize("label1")
31 Activity.AddView(label1, 0%, 0%, 100%, 100%)
32 label1.Gravity = Gravity.CENTER
33 label1.Color = Colors.White
34 label1.TextColor = Colors.Black
35 label1.TextSize = 40
36 label1.Text = "Hello World!"
37
38 End Sub
39
40 @Sub label1_click ()
41     label1.Text = "Bye bye!"
42 End Sub
43
44
45 @Sub Activity_Resume
46
47 End Sub
48
49 @Sub Activity_Pause (Userclosed As Boolean)
50
51 End Sub
52
53
```

با اجرای آن روی دستگاه واقعی یا شبیه‌ساز با تصویر زیر روبرو می‌شوید:



4- شرح بیشتر برنامه نویسی B4A با ساخت برنامه ریاضی ساده

در زیر عکس برنامه‌ای که در ادامه اقدام به ساخت آن خواهیم کرد آمده است:

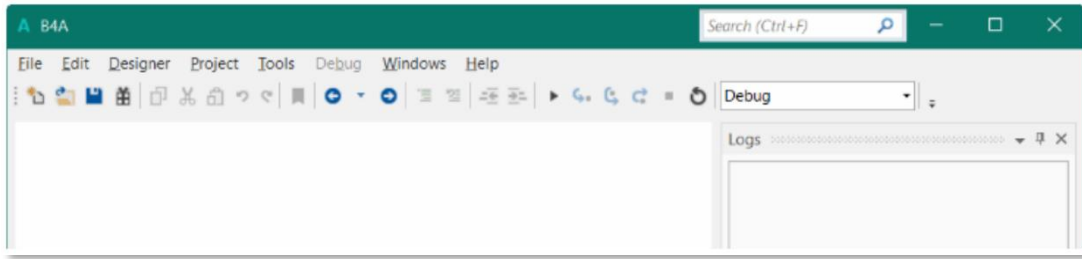


در عکس بالا، ما موارد زیر را خواهیم داشت:

- 2 برچسب که اعداد ایجاد شده به طور تصادفی را نمایش می‌دهند (بین 1 تا 9).
 - 1 برچسب با علامت ریاضی (+).
 - 1 نمای EditText که در آن کاربر باید نتیجه را وارد کند.
 - 1 دکمه، برای تأیید زمان پایان کار کاربر هنگام وارد کردن نتیجه استفاده می‌شود یا محاسبه جدیدی را ایجاد می‌کند.
 - 1 برچسب با نظر درباره نتیجه.
- ما طرح رابط کاربری را با VisualDesigner طراحی می‌کنیم و مرحله به مرحله کل مراحل را طی می‌کنیم.

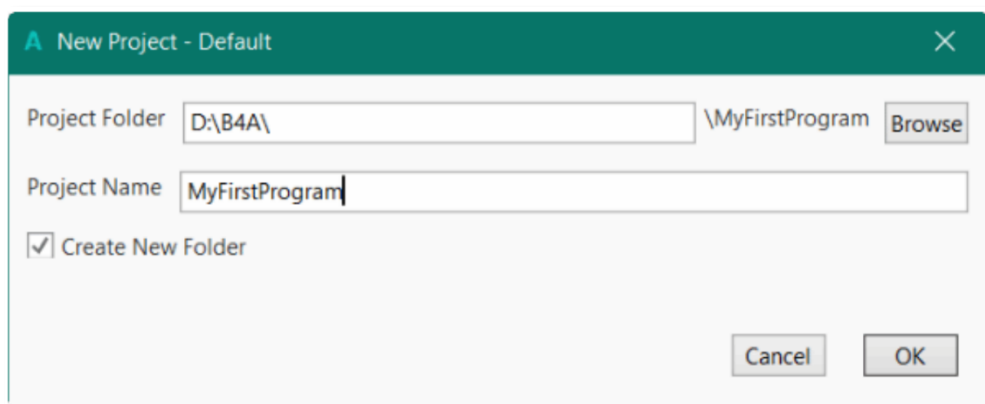
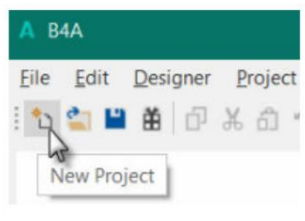
4-1- IDE را اجرا کنید

وقتی IDE را باز می‌کنید همه چیز خالی است.

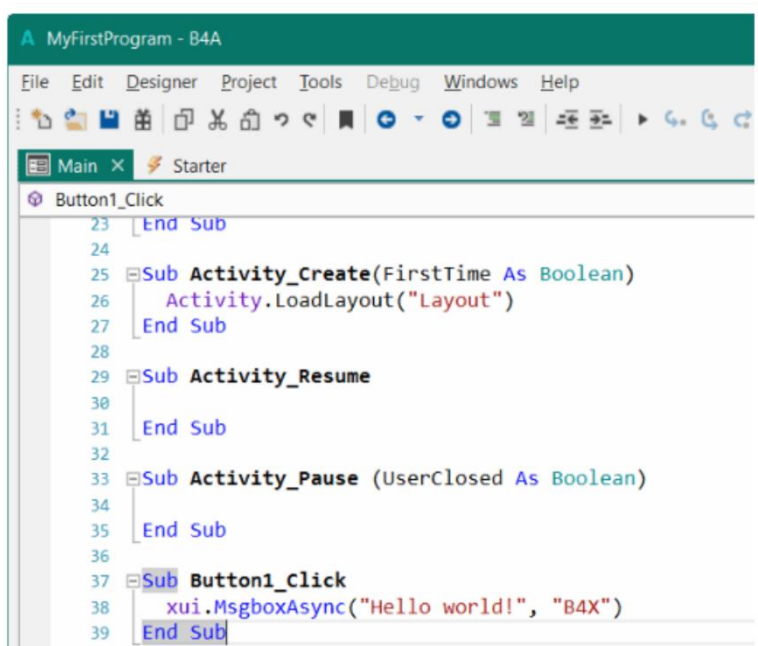


روی دکمه New Button کلیک کنید.

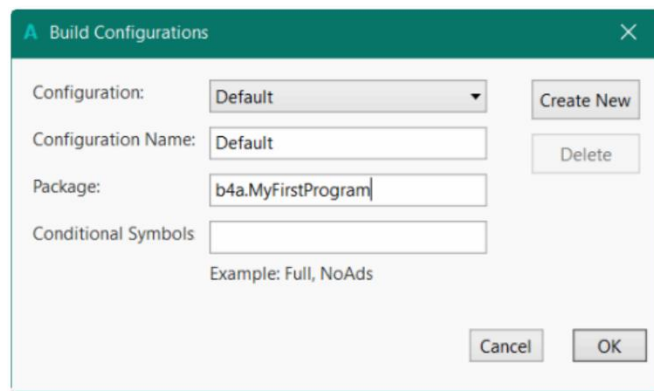
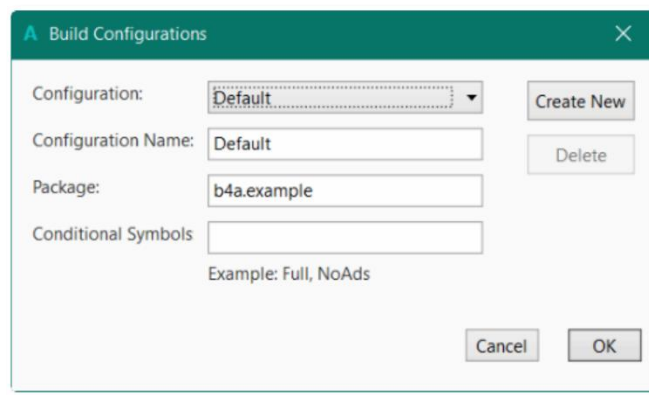
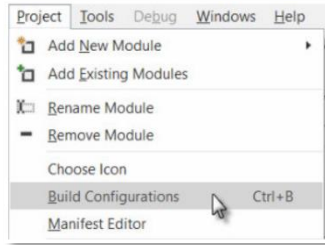
در ادامه از شما درخواست می‌شود که پروژه خود را ذخیره کنید:



در قسمت Project Name، MyFirstProgram را وارد کنید. همچنین مسیر Project Folder را تعیین کنید. برای این کار می‌توانید هر پوشه‌ای را وارد کنید. ما از D:\B4A\ به عنوان پوشه عمومی برای پروژه‌های B4A استفاده می‌کنیم. روی ok کلیک کنید و راهی مرحله بعد شوید. در ادامه با یک فایل main مواجه خواهید شد که قالب پیش فرض یک فایل B4A هست.

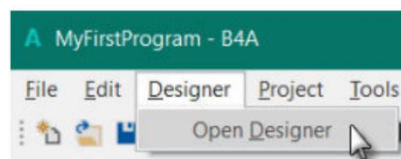


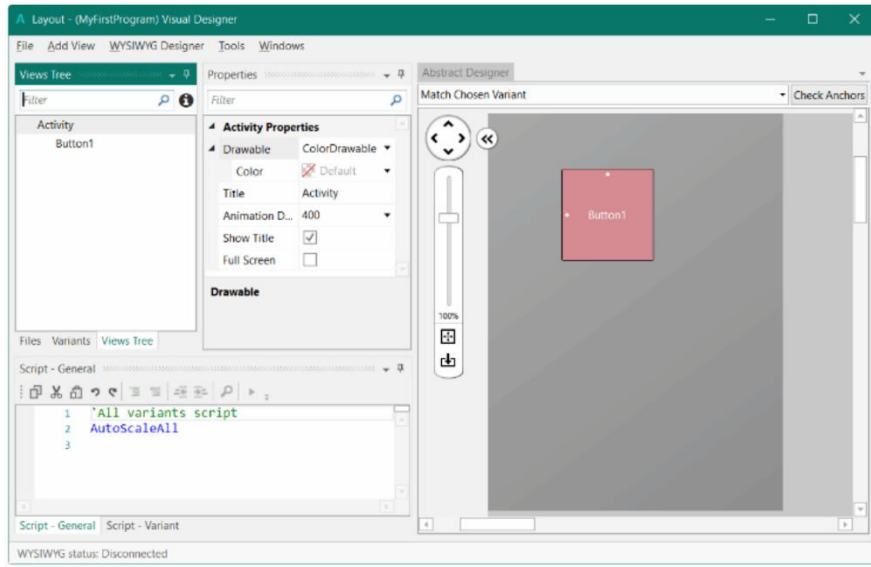
هر برنامه به یک نام بسته نیاز دارد. از منو روی گزینه Project کلیک کنید و پس از آن روی Build Configuration کلیک کنید. در پنجره باز شده، نام پیش فرض بسته b4a.example می باشد که ما آن را تغییر داده و b4a.MyFirstProgram را جایگزین آن می کنیم و روی دکمه ok می زنیم.



4-3- در IDE، designer را اجرا کنید و کد را بنویسید

از منو designer اجرا کنید. پنجره باز شده باید به شکل زیر باشد:



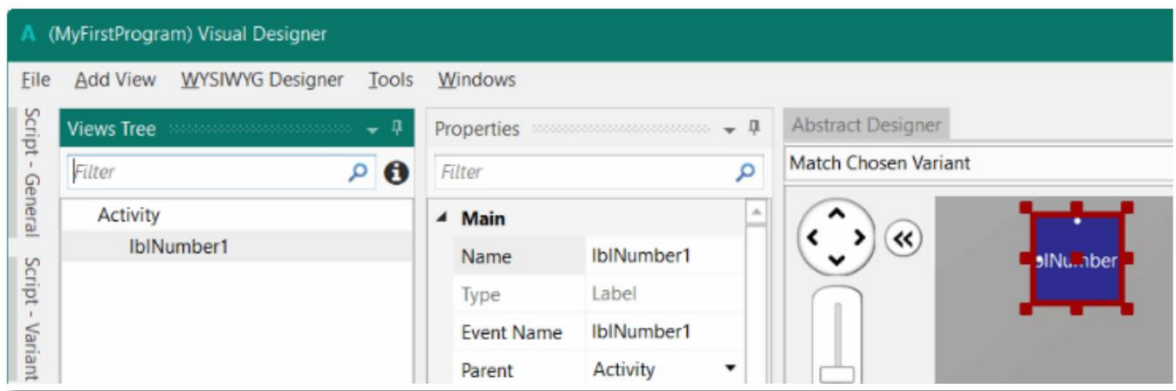


پنجره های مختلفی وجود دارد:

- Views Tree: تمام نماها را به صورت درخت نشان می‌دهد.
- Properties: تمام خصوصیات نمای انتخاب شده را نشان می‌دهد.
- Abstract Designer: نماها را روی صفحه نمایش می‌دهد.
- Script - General: اجازه می‌دهد تا تنظیمات دقیق را روی لایه‌ها انجام دهید.

در ادامه نمای Button1 که به صورت پیش‌فرض در لایه موجود بود را حذف می‌کنیم. از قسمت Add View 2 تا

برچسب یا label اضافه می‌کنیم. این دو برچسب به ازای 2 عدد توضیح داده شده در بالا می‌باشد. می‌بینیم که برچسب در صفحه ایجاد شده که قابلیت تغییر اندازه را نیز داراست. حال بیاید کمی تغییرات در آن ایجاد کنیم. این تغییرات را می‌توان از قسمت properties انجام داد. به طور پیش‌فرض، نام، Label با یک عدد است که در اینجا Label1 است. بگذارید نام آن را به lblNumber1 تغییر دهیم. سه حرف "lbl" در ابتدا به معنای "Label" است و "Number1" به

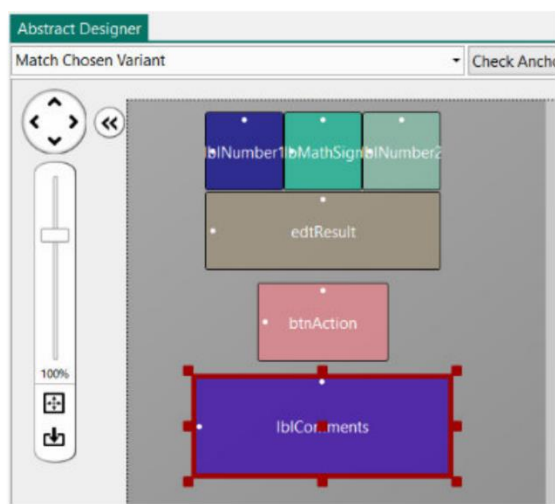


معنی شماره اول است. توصیه می‌شود از نام‌های معنی‌دار برای نماها استفاده کنید تا مستقیماً بدانیم هدف آن چیست.

روی برچسب ایجاد شده کلیک راست کنید و گزینه **duplicate** را بزنید تا یک برچسب دیگر تکرار شود و نام آن را **lblNumber2** بگذارید. در ادامه باز هم مثل قبل عمل کرده و یک برچسب دیگر ایجاد کرده ویژگی **text** آن را **+** و نام آن را **lblMathSign** قرار می‌دهیم. در مرحله بعدی از قسمت **Add View** یک **EditText** انتخاب می‌کنیم و مکان آن را زیر برچسب‌ها قرار می‌دهیم و نام آن را **edtResult** می‌گذاریم. در قسمت بعدی یک دکمه یا **Button** انتخاب می‌کنیم و نام آن را **btnAction** می‌گذاریم و همچنین ویژگی **Text** آن را روی **ok** تنظیم می‌کنیم. یک برچسب دیگر اضافه می‌کنیم و مکان آن را در زیر دیگر نماها قرار می‌دهیم و نام آن را **lblComments** می‌گذاریم.

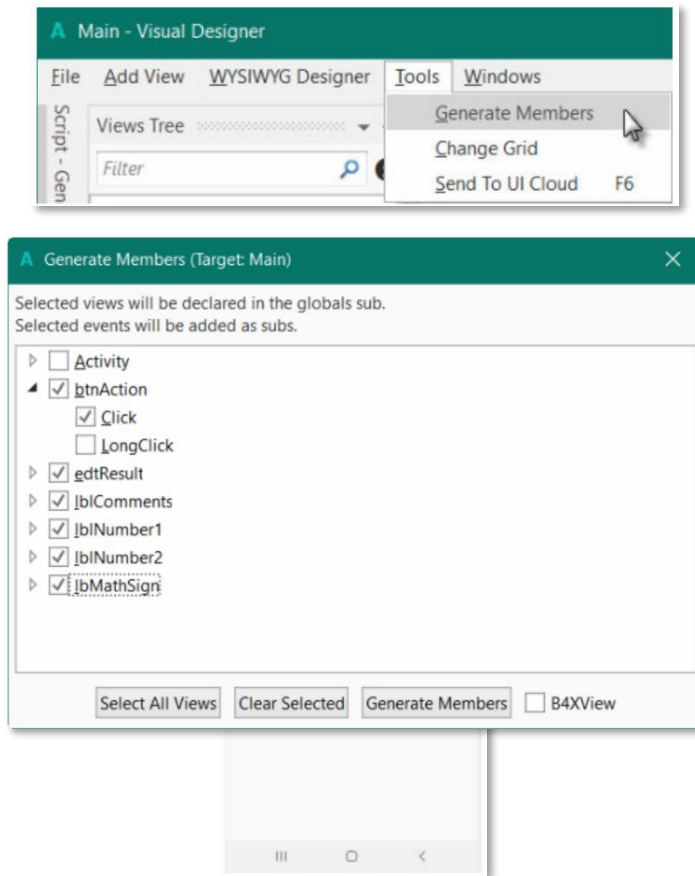
توجه: ویژگی‌ها یا **Properties** دیگری مانند رنگ متن، فونت و ... هم برای تغییر وجود دارند که به دلیل سادگی آن توضیح داده نشده است.

صفحه نهایی در **designer** به شکل زیر خواهد بود:



و اگر آن را اجرا کنیم روی شبیه‌ساز یا دستگاه واقعی به شکل زیر خواهد بود:

پروژه را ذخیره کرده و ادامه می‌دهیم. برای نوشتن روال‌های پروژه، باید Viewها را در کد ارجاع دهیم. این کار را می‌توان با Generate Memberstool در Designer انجام داد. از منوی tools روی Generate Members کلیک کنید.



در اینجا تمام نماهای اضافه شده به لایه فعلی را می‌بایم و چک باکس همه نماها را می‌زنیم. زدن چک باکس نمای edtResult باعث ایجاد مرجع آن در زیر روال (sub) Globals در کد می‌شود. این امر برای شناخته شدن نما توسط سیستم و اجازه دادن به عملکرد تکمیل خودکار، لازم است. روی دکمه Generate Members کلیک کرده تا چیزی شبیه به این در کد ظاهر شود.

```
Private btnAction As Button
Private edtResult As EditText
Private lblComments As Label
Private lblMathSign As Label
Private lblNumber1 As Label
Private lblNumber2 As Label
```

```
Sub btnAction_Click
End Sub
```

حالا برای وارد کردن کد به IDE برمی گردیم. ابتدا، برای بارگذاری لایه، به Activity خود نیاز داریم. این مورد از قبل در زیرروال "Activity_Create" آماده شده است

```
Activity.loadlayout("Layout")
```

ما می خواهیم به محض شروع برنامه یک problem جدید ایجاد کنیم. بنابراین، ما یک فراخوانی به زیرروال NewProblem اضافه می کنیم.

```
Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("Main")
    NewProblem
End Sub
```

توجه داشته باشید که NewProblem به رنگ قرمز است به معنی خطا است. این خطا 'Undeclared variable ...' می باشد. در این حالت به این معنی است که سیستم این نام را تشخیص نمی دهد. ایجاد یک مسئله جدید به معنای تولید دو مقدار تصادفی جدید بین 1 تا 9 برای Number1 و Number2 است، سپس مقادیر را با استفاده از خصیصه "Text" lblNumber1 و lblNumber2 نشان می دهیم. برای این کار کد زیر را وارد می کنیم:

در Sub Globals دو متغیر برای دو عدد اضافه می کنیم.

```
Public Number1, Number2 As Int
End Sub
```

و زیرروال "NewProblem":

```
Sub NewProblem
    Number1 = Rnd(1, 10)    ' Generates a random number between 1 and 9
    Number2 = Rnd(1, 10)    ' Generates a random number between 1 and 9
    lblNumber1.Text = Number1    ' Displays Number1 in label lblNumber1
    lblNumber2.Text = Number2    ' Displays Number2 in label lblNumber2
    lblComments.Text = "Enter the result" & CRLF & "and click on OK"
    edtResult.Text = ""        ' Sets edtResult.Text to empty
End Sub
```

این تابع Rnd(1, 10) یک عدد تصادفی از 1 تا 10 ایجاد می کند، بنابراین بین 1 و 9 است. خط زیر نظر را در lblCommentsview نمایش می دهد:

```
lblComments.Text = "Enter the result" & CRLF & "and click on OK"
```

CRLF کاراکتر LineFeed یا خط بعد است.

اکنون کد رویداد کلیک دکمه را اضافه می کنیم. ما دو حالت داریم:

- وقتی متن دکمه برابر با "O K" باشد (با فاصله ای بین O و K) ، به این معنی است که یک مشکل جدید نمایش داده می شود و برنامه منتظر است کاربر نتیجه ای وارد کند و دکمه را فشار دهد

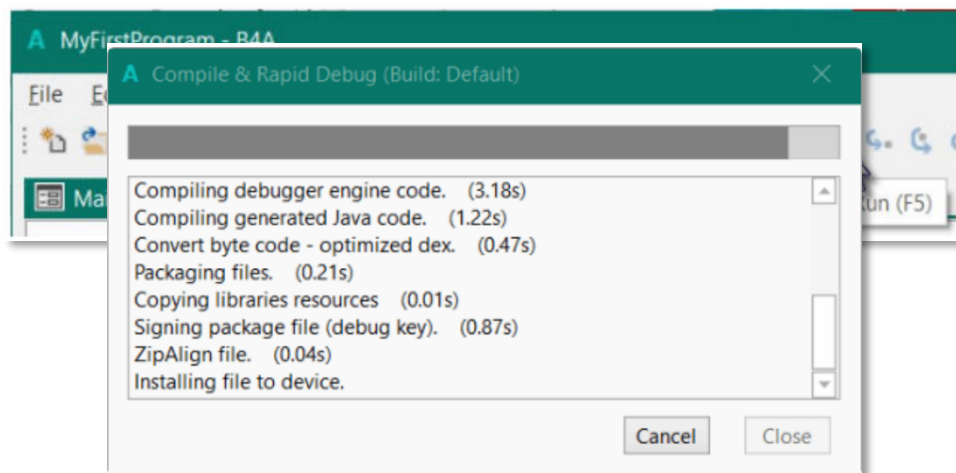
- وقتی متن دکمه برابر با "NEW" باشد، به این معنی است که کاربر پاسخ صحیحی را وارد کرده است و هنگامی که کاربر روی دکمه کلیک می‌کند مشکل جدیدی ایجاد می‌شود. در حال حاضر روال رویداد کلیک دکمه در کد وجود دارد.
کد زیر را اضافه کنید:

```
Sub btnAction_Click
    If btnAction.Text = "O K" Then
        If edtResult.Text = "" Then
            MsgBox("No result entered", "E R R O R")
        Else
            CheckResult
        End If
    Else
        NewProblem
        btnAction.Text = "O K"
    End If
End Sub
```

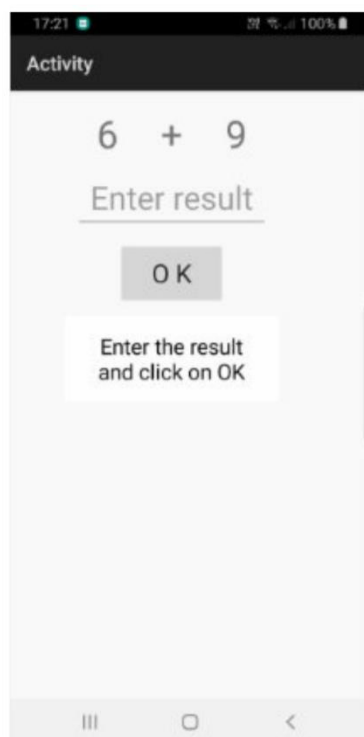
آخرین روال نتیجه را بررسی می‌کند.

```
Sub CheckResult
    If edtResult.Text = Number1 + Number2 Then
        lblComments.Text = "G O O D result" & CRLF & "Click on NEW"
        btnAction.Text = "N E W"
    Else
        lblComments.Text = "W R O N G result" & CRLF & "Enter a new result" & CRLF & "and clickOK"
    End If
End Sub
```

در شرط اول بررسی می‌کنیم که آیا نتیجه وارد شده درست است یا خیر.
پروژه را ذخیره می‌کنیم و روی دکمه Run کلیک می‌کنیم.



با مشاهده دستگاہ، باید چیزی شبیه به تصویر زیر، با شماره‌های مختلف مشاهده کنید.



حال می‌توانید نتیجه جمع خواسته شده را انجام داده و در کادر متن بنویسید و با زدن دکمه نتیجه کار خود را ببینید.