1. とにかく Android アプリを作ってみよう

1.1 Android Studio の起動

Android アプリの開発言語は Java です. Android アプリの開発では統合開発環境 Android Studio が最も良く用いられており、この実習でも Android Studio を使います. この実習で用いる Android Studio は Android アプリの開発用の設定を終えています. デスクトップ上の以下のアイ コンをクリックして Android Studio を起動して下さい.



1.2 Android アプリケーションプロジェクトの作成

Android Studio 上では 1 つのソフトウェアは 1 つのプロジェクトとして管理されます. 「Start a new Android Studio project」を選んで,新しい Android Studio プロジェクトを 作成します.





Android では画面を Activity と呼びます. ここでは, Empty Activity を選びます.

👳 Create New Project		×
Configure your project		
	<u>N</u> ame HelloAndroid	
÷	Package name cosmosclient.example.com.helloandroid	
	Save location Java を選択	
	C.*Users*Nitta*StudioProjects*HelloAndroid	
Empty Activity	Minimum API level API 15: Android 4.0.3 (IceCreamSandwich) Your app will run on approximately 100% of devices. Help me choose This project will support instant apps Use androidx.* artifacts	
Creates a new empty activity		
	Previous Next Cancel	



1.3 Android アプリの起動

作成したアプリは、「プロジェクトのルートを選択 → Run メニュー → Run `app`」でエ ミュレータ上に起動することができます.

アプリケーションを実行する AVD を選ぶ(ここでは Pixel 3a XL API28)

:¥Users¥Nitta¥StudioProjects¥HelloAndroid2] - ...¥app¥src¥main¥java¥cosmosclient¥example¥com¥h



sers¥Nitta¥StudioProjects¥HelloAndroid] - ...¥app¥src¥main¥java¥cosmosclient¥example¥com¥helloandroid¥MainActivityjava [app] - A



エミュレータが起動して以下のようにアプリの画面が表示されたら成功です.



2. アプリの画面レイアウトを変更してみよう

2.1 画面上のテキストの変更

Android アプリの画面レイアウトは簡単に変更することができます. プロジェクト内の res フォルダ の中の layout フォルダの中に画面ごとに xml ファイルが配置されています. この xml ファイルを 編集することによって画面レイアウトを変更することができます. 以下の xml ファイルをダブルクリッ クして開いてください.





エディタ上で GUI 部品を移動・編集したり,新しい GUI 部品を配置したりすることが できます.



まず、画面上に配置されている文字列(テキストビュー)を選択して、右側の Attributes ビューを使ってその属性を変更してみましょう.以下のように text 属性を変更して表示さ れている文字列を変更します.また、一番上の id 属性はプログラム中から GUI 部品を操作 するときに必要となるものなので、適当な文字列を割り当てておきます.ここでは、id と して text を割り当てます.

以下のようにエディタ上に画面レイアウトが表示されれば OK です.



マウスでドラッグしてテキストビューを画面中央上部に移動します.





左側のパレットからボタンを選んで画面上に配置してみましょう.



Text 属性を「OK」に変更して、OK ボタンにします.



3. ボタンを押したときの動作をプログラミングしてみよう

ここでは OK ボタンを押したときに、テキストビューに表示されている文字列を変更するプログラ ムを書いていきます.

3.1 プログラム中で OK ボタンを取ってくる方法

まずは、OK ボタンの id を確認しておきます. (button)

Attributes		Q 🕈	—	jid を確認し
button		В	utton	ておく
id	button			
► Declared Attribu	ıtes	+ ·	- 1	
► Layout				
▼ Common Attrib	utes			
style	@android:style/V	Vic 🔻	0	
onClick		•	0	
background	🚾 @android:dra	wable/b	0	
text	OK		0	
🗲 text			0	
contentDescription			0	
► textAppearance	@android:style/T	ex 🔻	0	
alpha			0	
▼ All Attributes				
alpha			0	
			-	

次に画面を制御する Java プログラムを開きます. Android アプリでは, Java プログラ ムはすべて java フォルダの中に格納されます. Java プログラムの拡張子は「.java」です. 画面毎にそれを制御するための Java プログラムが 1 つ作成されます. 例えば Main 画面に 対して, 画面レイアウトは activity_main.xml で定義されますが, 画面の制御は MainActivity.java に記述されます. それでは,以下のように MainActivity.java をクリッ クして開いてみましょう.



MainActivity.java には MainActivity というクラスが定義されています. クラスという のは Java プログラムを構成する基本要素で、基本的に1つの.java ファイルに1つのクラ スが定義されます. MainActivity クラスは, Main 画面を制御するクラスということにな ります.なお, Android では画面のことをアクティビティと呼ぶので覚えておいてください.



MainActivity クラスの中にはさらにいくつかの処理を含めることができます. 上の例で は、1 つの処理が含まれています. Java では、このような処理のことをメソッドと言いま す. 例えば, onCreate()というメソッドがありますが, これは Main 画面(アクティビティ) が作られたときに実行される処理です.画面(アクティビティ)を制御するクラスの中でもっ とも重要なのは、この onCreate()メソッドです. Android ではアクティビティを制御する クラスには必ず onCreate()メソッドが含まれるようになっています.

OKボタンを取ってくるプログラムはこのonCreate()メソッドの中に記述します. OKボタンは以下 の1行で取ってくることができます.

GUI 部品の部品の id

Button okButton = (Button)findViewById(R.id.buttor);

ここで、赤枠で囲った部分が事前に確認しておいた OK ボタンの id になっていることに注 意してください. この id の部分を変えることで画面上の任意の GUI 部品をプログラム中で 取ってくることができます. ここで取ってきたボタンはプログラム中の「okButton」とい う変数に格納されます. onCreate()メソッドの中に OK ボタンを取ってくるプログラムを

追加した結果は以下のようになります.



ここで赤文字が 2 箇所出ていますが、これはこの部分でコンパイルエラーが発生している ことを示しています.



この赤文字をクリックすると, Android Studio がエラーの解消方法を提示してくれます. ここでは,「Alt+Enter」キーを押してコンパイルエラーを解消します.

3.2 OK ボタンを押したときに実行されるプログラム

プログラム中でOKボタン取ってきたのは、取ってきたOKボタンに、ボタンを押したときの処理を 設定するためです.先ほど追加したプログラムの下に以下の赤枠で囲った部分を追加してください.

Button okButton = (Button)findViewById(R.id.button);



ちょっと複雑なプログラムですね.実際このプログラムをきちんと読めるようになるため には Java をかなり勉強しなければなりません.この時点でそこまで Java を深く理解して いる必要はありませんので、今のところはコピペで結構です.ただし、上のプログラムの 中に onClick()メソッドが含まれていることに注意して下さい.この <u>onClick()メソッド</u> <u>が OK ボタンを押されたときに実行される処理に相当</u>します.上のプログラムのままでは、 onClick()メソッドの中に何の処理も書かれていないため、このままでは OK ボタンが押 されても何も起こりません.そこで、onClick()メソッドの中に以下の2 行の処理を付け 加えることにします.



この2行のうちの1行目を見てください.先ほど書いたOKボタンを取ってくる処理に似ているのがわかるでしょうか?これは,画面上に配置されたもう一つのGUI部品であるテキストビューを取ってくるプログラムです.idを指定する部分が,テキストビューのidである「text」となっていることに注意して下さい.また,ここで取ってくるGUI部品はボタンではなくテキストビューなので,先ほど「Button」となっていたところが「TextView」となっていることが分かると思います.2行目で取ってきたテキストビューに"OK Button Clicked!!"という文字列を設定しています.

最終的に onCreate()メソッド内に追加するプログラムは以下のようになります. これで, OK ボタンを押すとテキストビューの表示が"OK Button Clicked!!"に変わるというプログ ラムが書けました.

```
Button okButton = (Button)findViewById(R.id.button);
okButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        TextView textView1 = (TextView)findViewById(R.id.text);
        textView1.setText("OK Button Clicked!!");
    }
});
```

Android Studio では編集中のファイルは自動保存されるので, ユーザが明示的にファイル を保存する必要はありません. アプリケーションを起動し、OKボタンを押してテキストビューの表示が変われば成功です.



4. 新規画面(アクティビティ)の追加と画面遷移

4.1 新規アクティビティの追加

Android アプリのアクティビティは「プロジェクトのルートを選択 \rightarrow File メニュー \rightarrow New \rightarrow Activity $\rightarrow \dots$ 」で追加します. ここでは, Empty Activity を追加します.

HelloAndroid [C:¥Users¥Nitta¥StudioProjec	ts¥HelloAndroid]¥app¥src¥main¥res¥layo	out¥activity_main.xml [app] - Android Studio
<u>File E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e	<u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S W</u> in	idow <u>H</u> elp
New 🕨	New Project	🖌 r. 🖉 🐇 🗆 🔊 R. 🕸 🕨 🖛 🖙 Q.
늘 Open	Import Project	
🚰 Profile or debug APK	Project from Version Control	
Open <u>R</u> ecent	New Module	in.xml 🔀 🖸 MainActivity.java 🛛
Close Project	Import Module	Q ‡ — ◇▼ 🛇▼ 🛛 Pixel▼ 🛎 29▼ (
Link C++ Project with Gradle	Module	Ab TextView
✗ Settings Ctrl+Alt+S	🛃 Android Resource File	Button
Project Structure Ctrl+Alt+Shift+S	Android Resource Directory	ImageView
Other Settings	Sample Data Directory	≔ RecyclerView 🖄
Import Settings	🖆 File	<> <fragment></fragment>
<u>Export Settings</u>	🗳 Scratch File Ctrl+Alt+Shift+Insert	Scrollview
Settings Repository	Directory	Test
Export to Zip File	S C++ Class	
Save All Ctrl+S	提 C/C++ Source File	
Sync Project with Gradle Files	📕 C/C++ Header File	
Sync with File System Ctrl+Alt+Y	🐳 Image Asset	
Invalidate Caches / Restart	🖷 Vector Asset	• • •
Export to <u>H</u> TML	http://www.communec	ayout
🖶 Print	Gradle Kotlin DSL Build Script	est"
Add to F <u>a</u> vorites	G Gradle Kotlin DSL Settings	"ОК" 🔺
File Encoding	Edit File Templates	ок
Remove BOM	🛱 AIDL 🕨 🕨	
Associate with File Type	🖗 Activity 🔹 🕨	📫 Gallery
Line Separators	🖷 Android Auto	The Navigation Drawer Activity
Make Directory Read-only	🖷 Folder 🕨 🕨	Master/Detail Flow
Power Save Mode	🖷 Fragment 🕨 🕨	Basic Activity
E <u>x</u> it	👘 Google 🔹 🕨	Fullscreen Activity
<i>1</i> 2	🖷 Other 🕨 🕨	Fragment + ViewModel
	Service	Bottom Navigation Activity
Build: Build Output $ imes$ Sync $ imes$	Ul Component	Tabbed Act moty Activity Crist
S < ▼ ✓ Build: completed successfully	Wear •	Scrolling Activity
	Widget	
★ I ► ✓ Configure build		Android TV Activity (Requires minSdk >= 21)
🖌 🗸 Calculate task graph	Resource Bundle	Login Activity /
Run tasks		Empty Activity
d Va		
		Android Things Peripheral Activity (Requires minSdk >= 24)
×		Settings Activity

👧 New And	droid Activity			×
R	Configure Activ	vity		
		Creates a new e	empty activity	
		Activity Name:	SubActivity	
	÷		Generate Layout File	
		Layout Name:	activity_sub	
			Launcher Activity	
		Package name:	cosmosclient.example.com.helloandroid	
		Source Language:	Java	
		Target Source Set:		
		The name of the activity	r class to create	
			Previous Next Cancel	<u> </u>





Attributes	٥	¢ –	
Ab textView		TextView	
id	textView		
Declared Attribute	ites	+ -	
► Layout			
▼ Common Attribu	ıtes		
text	Next Activity	0	— text 属性を変更
🗲 text		0	
contentDescription		0	
► textAppearance	@android:style/Tex	- 0	
alpha		0	
▼ All Attributes			
alpha		O	
▶ autoLink	P	0	
autoText		0	
background	Ø	0	
bufferType		-	
capitalize		•	
clickable		0	
contentDescription		0	
cursorVisible		0	

4.2 画面遷移のプログラム

画面遷移をするためのプログラムの書き方を説明します.まず Main アクティビティに次のアクティビティに進むための Next ボタンを追加します. id の確認



MainActivity.java を開いて下さい.



onCreate()メソッドの最後に以下のプログラムを付け加えてください.



Next ボタンを取ってきてボタンを押したときの処理を設定する部分は,OK ボタンのとき と同じです.ただ,押したときの処理内容(onClick()メソッドの中の2行、赤枠内)が 異なっています.onClick()メソッドの中の1行目で現在のアクティビティと遷移先のア クティビティを設定して画面遷移の準備を行います.2行目で実際に画面遷移を行います.



アプリケーションを起動し, Next ボタンをクリックします.

次の画面に遷移すれば成功です.



5. 位置情報(GPS)を取得してみよう

5.1 位置情報取得のための準備(プログラム編)

Sub アクティビティで GPS から取得した位置情報を表示するようにしてみましょう. 位置情報を取得するには LocationManager という標準クラスを使用します. 具体的には以下の2行をSubAvticity.javaのonCreate()メソッドの最後に追加します. onCreate()に **i** ののについてのです。

LocationManager manager = (LocationManager)getSystemService(LOCATION_SERVICE);
manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);

protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentV @android.location.LocationManager? Alt+Enter
 LocationManager manager = (Loc ronmanager)getSystemService(LOCATION_SERVICE);
 manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);

LocationManager が赤色になっていると思いますので、クリックして「Alt+Enter」キーを押してく ださい. エラーを解消してもまだ別のエラーが残っていると思います. SubActivity クラス の最初の行に以下の赤枠の部分を付け加えてください.

public class SubActivity extends AppCompatActivity implements LocationListener{

追加部分の LocationListener が赤色になっているので,先ほどと同様に「Alt+Enter」キーを押してエラーを解消します.そうすると、今度はこの行の「SubActivity」の部分にエラーが出るようになります. クリック 選択

E		
	Public class SubActivity extends AppCompat	tActivity implements LocationListener {
	🥊 Implement methods	
	🔮 Make 'SubActivity' abstract	
Ē	💎 Create Test	▶ ceState) {
	🔊 Create subclass	•
	♥ Unimplement Interface	•
	Interface 'LocationListener' as @DeprecationListener' as @DeprecationListener'	ed ► ager)getSystemService(LOCATION_SERVICE);
	manager.requestLocationUpdates(Loc	cationManager.GPS_PROVIDER, minTime:0, minD
Ģ		

次にこの部分をクリックし、「implement methods」を選んで再びエラーの解消をしましょう.

👳 Select Methods to Implement 🛛 🕹 🗙	
↓2 ○ 조 ·	
android.location.LocationListener	
👼 🖕 onLocationChanged(location:Location):void	
💼 🍯 onStatusChanged(provider:String, status:int, extras:Bundle):	
💼 🍯 onProviderEnabled(provider:String):void	
💼 🐂 onProviderDisabled(provider:String):void	
android.view.Window.Callback	
🎯 ゛ on Pointer Capture Changed (has Capture: boolean): void	
✓ Insert @ <u>O</u> verride OK Cancel	

OK を押すと, SubActivity クラスの下の方に新たに 4 つのメソッドが自動で追加された と思います. これで, すべてのエラーが解消されたはずです.

プログラムについて説明しましょう.onCreate()メソッドに追加した2行はあまり深く 考えずに,基本的にコピペで構わないと思います.ただ,この2行を付け加えることで, SubActivity クラスに位置情報が送信されるようになるということだけ知っておいてくだ さい.次に SubActivity クラスの最初の行に追加した「implements LocationListener」 の部分ですが,これは SubActivity クラスが送られてきた位置情報を受け取れるようにす るためのものです.実際に位置情報を受け取るのは,後から自動で追加された以下の onLocationChanged()メソッドになります. 位置情報が変わるたびにこのメソッドが実行 されるようになります.

public void onLocationChanged(Location location) {
}

5.2 位置情報取得のための準備(マニフェスト編)

位置情報を取得できるようにするためにはプログラムの準備だけでは足りません. 実際に上のプ ログラムを動かしても位置情報は取得することができません. Android 端末にはいろんなセンサー が用意されていますが, アプリからそれらのセンサーを使うためにはマニフェストファイルの修正を 行わなければなりません. マニフェストファイルとは, 以下の場所にある AndroidManifest.xml フ ァイルのことです. このファイルをダブルクリックして開いてみましょう.



AndroidManifest.xml の全体がテキスト表示されるのでこれを直接編集していきます.



マニフェストファイルの下の方に以下の2行を付け加えてください.



これは、このアプリが GPS センサーにアクセスする許可を与えることを示しています.

5.3 位置情報取得と表示

以上で位置情報を取得できるようにするための準備が完了しました.それでは,再び SubActivity.javaを修正して位置情報を取得して画面に表示できるようにしましょう.先ほども説明 したように,位置情報は以下のメソッドで受け取れるようになっています.

public void onLocationChanged(Location location) {

}

このメソッドの引数「location」に位置情報が渡ってきています.具体的には,緯度と経 度の情報が入っています.まずは,このメソッドの中で緯度と経度を受け取るための変数 を用意しましょう.

public void onLocationChanged(Location location) { double latitude, longitude; ____ 追加

Java では C 言語と同様に, int 型, float 型, double 型などの変数を使うことができます. また変数を使う場合は事前に宣言をしておかなければならないのも同様です. なお, latitude は緯度, は longitude 経度を表します. これらの変数に実際の緯度と経度を代入 するために以下の 2 行を付け加えてください. 追加

	,讵
<pre>latitude = location.getLatitude();</pre>	1
<pre>longitude = location.getLongitude();</pre>	

引数 location の中から緯度情報と経度情報を取ってきて,それぞれ latitude と longitude という変数に代入しています.次にこれらの情報をテキストビューに表示するようにして みましょう.そのためにテキストビューの id を確認します.



「textView」という id が確認できたと思います.

さて, SubActivity.java の修正の続きです. 現時点で onLocationChanged()メソッドの内容 は以下のようになっているはずです.

```
public void onLocationChanged(Location location) {
    double latitude, longitude;
    latitude = location.getLatitude();
    longitude = location.getLongitude();
}
```

このメソッドの最後に以下の2行を付け加えてください.

テキストビューのid

TextView textView2 = (TextView)findViewById(R.id.textView); textView2.setText("緯度:" + latitude + ", 経度:" + longitude);

1 行目はもう分かりますね. 2 行目で緯度と経度の値をテキストビュー(textView)に設定しているのですが,1 つ注意して欲しいのがこの行で行っている文字列の足し算です. Java では文字列同士や文字列に数字を足し算を使って足すことができ,これは文字列の連結を意味します.C 言語では文字列の連結に strcat()関数を使っていたと思いますが,足し算の方がはるかに簡単ですね.

5.4 エミュレータ上での実行

アプリを実行してエミュレータ上で Sub アクティビティを表示します.



しかし緯度も経度も表示されません.ここで,エミュレータの右側にあるコントローラの「・・・」 をクリックして下さい.

そうすると、以下のようなダイアログが表示されます.

Exten	ded controls							x
0	Location	GPS data point						
4	Cellular	 Decimal Sexagesimal 			Latitude 37.422			
- ii	Battery	Occurrent			Longitude			
e.	Phone				Altitude (meters)			ノクリック
0	Directional pad				0.0			
Ô	Fingerprint	GPS data playback					SEND	
\$	Settings							
?	Help	Delay (sec)	Latitude	Longitude	Elevation	Name	Description	
		> Sp	eed 1X 🔍			L	DAD GPX/KML	

ここで Location タブを選んで SEND ボタンをクリックして下さい. これはエミュレータ上の仮想の GPS に指定した緯度と経度の値(仮の値)を送ることを意味しています.

以下のように Sub アクティビティに緯度と経度の仮の値が表示されれば成功です.

