УРОК 1

Програма – сукупність команд (програмних кодів), що буде виконана процесором. *Android* програмується мовами *Java* або *Kotlin* (варіант *Java*, адаптований спеціально для *Android*). *Java*-команди упаковуються в методи, методи – в класи. Тому *Java*-програма – це сукупність класів.

Аналогічним чином відбувається збереження тексту у файлових системах:

Слова => речення => файл. Далі => каталог => кореневий каталог.

Команди => метод => клас. Далі => пакет => проєкт.

При створенні нового проєкту Android Studio формує шаблон проєкту, де міститься програма виведення тексту "Hello World!", яку наведено нижче на Рис.1:



Puc.1.

Ключовий елемент будь-якої програми — клас. Класи легко відрізняти від інших елементів програми — назви класів починаються зі строкових (великих) букв. В програмі-шаблоні задіяно три класи — створений нами клас MainActivity, а також два інших класи — Activity та Bundle (спробуйте знайти їх по перших строкових буквах). Це є нормальною практикою, коли класи використовують інші класи, в тому числі з інших пакетів (пакети - аналог каталогів). Але якщо задіяний клас знаходиться в іншому пакеті - його треба імпортувати (рядки 3, 4 на Рис.1).

По аналогії із тим, що каталоги можуть розміщатися всередині інших каталогів і відділятися символом / (*C:/Program Files/Intel/*), пакети також можуть розміщатися всередині інших пакетів і відділятися крапкою (android.app.Activity). Наприклад,

клас Activity розміщено всередині пакету арр, що, в свою чергу, розміщено всередині пакету android (рядок 3 на Рис.1).

Для нашого проєкту AndroidForKids1 оболонкою Android Studio створено пакет com.example.ism.androidforkids1 (рядок 1 на Рис.1), де ism – назва користувача у Windows (на різних комп'ютерах назви користувачів відрізняються).

Кожен клас містить фігурні дужки {}, всередині яких розміщено методи (та інші елементи) класу. Методи також містять фігурні дужки {}, всередині яких розміщено команди методу. Команди завершується символом ; (крапка з комою).

Якщо навести курсор на одну з фігурних дужок класу або методу, *Android Studio* виділить відповідну пару дужок кольором (рядки 9, 12 на Рис.1).

Символом // позначаються коментарі – пояснення, що не є частиною програми.

Створений Android Studio шаблон проєкту містить один клас MainActivity (рядки 6...13 на Рис.1), всередині якого міститься один метод onCreate (рядки 9...12 на Рис.1), що містить дві команди (рядки 10, 11 на Рис.1), серед яких ключовою є команда 2 setContentView (R.layout.activity main) (рядок 11 на Рис.1).

Назва цієї команди – "встановитиКонтентнеЗображення" (контент - зміст) – вказує, що в результаті її виконання до вікна буде занесене зображення, яке міститься у файлі, розміщеному в дужках, тобто у файлі *activity_main* каталогу layout всередині каталогу R (*res - resources*).



Puc.2.

Особливістю всіх *Android*-додатків, незалежно від їх складності, є обслуговування кожного вікна додатку парою файлів.

Один з файлів містить код, що написаний мовою *Java* та в назві обов'язково містить складову "Activity". Цей код представляє дії, що будуть відбуватись у вікні (рух

фігур, натискання кнопок, тощо). В шаблоні проєкту цей код представлений класом MainActivity і є розміщеним у файлі з такою ж назвою в пакеті com.example.ism.androidforkids1, що, в свою чергу, розміщено в пакеті java.

Назву файлу із другим кодом, що є написаним мовою XML, утворено переставлянням складових назви Java-класу — відповідно activity_main.xml. Цей код описує первинний дизайн вікна — зображення, що будуть відображатися на вікні безпосередньо після запуску Android-додатку, до виконання будь-яких дій.

В шаблоні проєкту файл activity_main.xml містить наступне зображення:



Рис.3.

Саме це зображення – текст "Hello World!" – команда setContentView (R.layout.*activity_main*) з рядку 11 на Рис.1 скопіює до вікна Android. Пересвідчитись в цьому можна, запустивши програму (Рис.1).

Дизайн вікна, представлений у файлі activity_main.xml, утворено відповідним кодом:



Puc.4.

Не пояснюючи інші деталі xml-кодів, відмітимо тільки три рядки - рядок 7:

tools:context=".MainActivity",

що ставить у взаємну відповідність коди файлів activity_main.xml та MainActivity, рядок 9:

TextView,

що розміщає на первинному дизайні вікна саме текстове зображення, та рядок 12:

android:text="Hello World!",

що задає текст у текстовому зображенні.

Рядки *xml*-кодів, що починаються із символів "android:", називаються атрибутами, де text – назва атрибуту, а "Hello world!" - значення атрибуту. Змінивши значення атрибуту в рядку 12 наступним чином:

android:text="Hello Android!",

отримаємо відповідні зміни в дизайні:

Ż	🙍 AndroidForKids1 [C:\Users\ISM\AndroidForKids1]\app\src\main\res\layout\activity_main.xml [app] - Android Studio 🦳 🗌									
<u>F</u> ile	<u>F</u> ile <u>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> indow <u>H</u> elp									
	H 🧕	🤤 💠 🔿 🔼 app	A # B 00 B 00 B 10	(2 1	Ĩ				
	AndroidForKids1 app arc main res arc activity main.xml									
t	activity_main.xml × © MainActivity.java ×									
1: Proje	Palette	Q, ‡- I∽	📚 🔻 🚫 👻 🔲 Nexus 4 👻 28 👻 💿 AppTheme 👻 🧼 💮 33% 🕂 🕲		Gradle	:				
2	Common	Ab TextView	⊙ ▼ 🔌 <u>8dp</u> , 🖍 🌾 🧵 ▼		dtribu					
tures	Text	Button ImageView			ites					
Cap	Buttons	I RecyclerView ±	AndroidForklus							
۲	Widgets	<> <fragment></fragment>								
	Layouts	ScrollView								
	Containers	 Switch 								
	Google									
0.	Legacy									
dun			le la							
Z: Stru	Componen	t Tree 🔅 ♦ 🗜								
	ာ္ Consti	raintLayout	Hello Android!	~~•						
	Ab Te	xtView- "Hello Android 🔺								
riants										
d Va										
Buil						1				
es					evice 1					
avorit			÷		ile E					
2 2										
*	Design	Text			er					
	🔮 TODO 📰 <u>6</u> : Logcat 🐙 Build 🔯 Terminal 🔍 Even									
	Hardcode	ed string "Hello !", should use `@	Ostring` resource Context: <no context<="" td=""><td>t></td><td>ъ</td><td></td></no>	t>	ъ					
				Р	uc.5	5.				

та на вікні :



Використовуючи атрибути

android:textSize, android:textColor, android:textStyle, android:typeface,

можна змінювати відповідно розміри, колір та тип шрифта букв тексту. Після додавання у рядки 13...16 коду activity main.xml атрибутів:

android:textSize="50sp" android:textStyle="italic" android:textColor="#990101"
android:typeface="serif"

текстове зображення на вікні суттєво зміниться:



Отриманої інформації по роботі з кодом activity_main.xml вже достатньо для створення вікна запуску гри *Flappy Bird* за допомогою атрибутів *xml*-кодів (Рис.8).

Для цього необхідно виконати наступні послідовні дії:

- підібрати в Інтернет у форматі .png (рекомендовано) або .jpg відповідні картинки-зображення заднього плану та кнопки запуску, а також занести їх до каталогу drawable каталогу res,
- встановити на вікно картинку заднього плану,
- усунути з вікна текстове зображення "Hello Android!",
- встановити на вікно зображення кнопки запуску гри.



Android Emulator - Nexus_5_API_24:5554



Puc.8.

Заносимо наперед підібрану картинку background.png до каталогу drawable:





Використання атрибуту:

android:background="@drawable/background"

(рядок 8 коду activity main.xml) сформує на вікні обрану картинку заднього плану:



Puc.10.

Усунення з вікна текстового зображення найбільш доцільно здійснити виділенням (сукупність клавіш *Cntrl-C*) та видаленням (сукупність клавіш *Cntrl-X*) рядків 10...21 коду activity main.xml, що представляють текстове зображення *TextView*:

Ż	👳 AndroidForKids1 [C:\Users\ISM\AndroidForKids1]\app\src\main\res\layout\activity_main.xml [app] - Android Studio — 🛛 🛛 🗙								
<u>F</u> ile	<u>E</u> dit <u>V</u> i	ew <u>N</u> av	<i>r</i> igate <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> indow <u>H</u> elp						
-	H 🧕	Q (⊨ ⇒ 🔨 🔩 app ∨ 🔈 🔸 🕸 🕼 🗥 🔓 📕 🖳 💺 🖙		Q				
	AndroidForKids1) app main res la lavout activity main.xml								
e activity main.xml × C MainActivity.java ×									
rojec	1 <2ml version="1"0" encoding="utf=8"2>								
÷	2 C a http://schemas.android.com/apk/res/android								
@	3	3 xmlns:app="http://schemas.android.com/apk/res-auto"							
	4	<pre>xmlns:tools="http://schemas.android.com/tools"</pre>							
nres	5	android:layout_width="match_parent"							
Capt	6	aı	ndroid:layout_height="match_parent"			iew			
1	7	to	ools:context=".MainActivity"						
	9	a	narora:background="@arawabre/background">						
	10 5	</td <td>TextView</td> <td></td> <td></td> <td></td>	TextView						
an	11	android:layout width="wrap content"							
g	12		android:layout_height="wrap_content"						
Z St	13		android:text="Hello Android!"						
	14		android:textSize="50sp"						
	15 🗖		android:textColor="#990101"						
lts	16		android:textStyle="italic"						
/aria	18		android:typerace="serif"						
-pi	19		app:layout constraintLeft toLeftOf="parent"						
Bu	20		app:layout constraintRight toRightOf="parent"						
	21		app:layout_constraintTop_toTopOf="parent" />			D			
s	22					evice			
orite	23	<td>roid.support.constraint.ConstraintLayout></td> <td></td> <td></td> <td>File</td>	roid.support.constraint.ConstraintLayout>			File			
Fav		I support constraint Constraint avout			Б				
ivi	Al androidsupport.constraint.constraint.ayout								
	Design	Text							
🔈 👍: Run 🔮 TODO 📰 <u>6</u> : Logcat 🗥 Profiler 🔱 Build 國 Terminal 🔍 Event Log									
	Can't bind	to local	8618 for debugger (35 minutes ago) 492 chars, 12 line breaks 22:1 CRLF UTF-8 Context: <no< td=""><td>context></td><td>6</td><td>⊕</td></no<>	context>	6	⊕			

Puc.11.

Android Emulator - Nexus_5_API_24:5554 👮 AndroidForKids1 [C:\Users\ISM\AndroidForKids1] - ...\app\src\main\res\Iayout\activity_main.xml [app] - Android Studio × <u>File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help</u> 늘 🔛 💿 😳 💠 🔶 🔨 🕞 app 🗸 🛼 🦊 🐞 🍈 🗥 🔓 🔳 🖳 🚣 🗔 Q 🔲 1 2 3:17 🚬 AndroidForKids1 🔪 📑 app 🔪 🖿 src 🔪 🖿 main 🔪 📑 res 🖓 🖿 layout 🛛 🚜 activity_main.xml AndroidForKids1 activity_main.xml × C MainActivity.java > ۲ 诸 1: Project Gradle <?xml version="1.0" encoding="utf-8"?> 2 C <android.support.constraint.ConstraintLayout</pre> 0 xmlns:android="http://schemas.android.com/apk/res/android" 3 Captures Preview \$ xmlns:app="http://schemas.android.com/apk/res-auto" 4 xmlns:tools="http://schemas.android.com/tools" android:layout width="match parent" 6 📰 Z: Structure android:layout height="match parent" tools:context=".MainActivity" 8 9 android:background="@drawable/background"> **Build Variants** Device File Explorer </android.support.constraint.ConstraintLayout> . android.support.constraint.ConstraintLayout \triangleleft 0 Favorites Design Text 属 <u>4</u>: Run 🍓 TODO 🛛 🖻 Logcat 🏿 🎝 Profiler 🔑 Build 🗖 Terminal 📵 Event Log Namespace declaration is never used 4:56 CRLF UTF-8 Context: <no context> ⇔ 'n.

(Зверніть увагу, що другу половину рядку 2 коду activity_main.xml з Puc.11 переведено на наступний рядок – Puc.12).

Заносимо до каталогу drawable наперед підібрану картинку playbutton.png:



Puc.13.

Для виведення на вікно зображення кнопки запуску гри необхідно внести зміни до рядків 2 та 11 коду activity_main.xml, пов'язані із зручністю компоновки зображень на вікні, наступним чином:



Puc.14.

Безпосереднє занесення зображення кнопки запуску на вікно здійснюється в режимі дизайну засобами Android Studio Builder шляхом перетягування (drag-and-drop) потрібного елемента на вікно мишкою із постійно натиснутою лівою клавішею. Кнопку запуску представимо зображенням типу ImageView, який треба знайти в палеті (Palette) доступних зображень, виділити та перетягнути на центр вікна:



Puc.15.

Android Studio запропонує обрати зображення для виведення на вікно:

👳 Resourc	es		×	
Q	:=	Add new resource	•	
Drawable	 Sample data 			
Color	avatars			
	backgrounds/scenic	No Preview		
	Project			
	▶ android			
	Theme attributes			
		OK		

Puc.16.

Оберемо пункт Project, в якому виділимо playbutton та натиснемо OK:



Puc.17.

Дизайн вікна для запуску гри Flappy Bird буде сформовано остаточно:



Внесення зображення кнопки запуску гри засобами Android Studio Builder зробить автоматичні зміни в коді activity main.xml (рядки 11...16) та на вікні додатку:



Puc.19.

ЗАВДАННЯ

Рівня складності 1 – обов'язкові.

1. Змінити текст, що виводиться, на *"Hello Ukraine!"* в атрибуті android:text. Переглянути текст дизайну вікна та на вікні після запуску додатку.

2. Скачати в Інтернет та занести до каталогу *drawable* каталогу *res* дві обрані за уподобанням картинки – для заднього плану вікна запуску гри *Flappy Bird* та для представлення кнопки запуску гри.

Рівня складності 2 – факультативні.

3. Змінити значення атрибутів android:textStyle та android:typeface. Апробувати вигляд тексту для всіх можливих значень цих атрибутів, визначених за посиланням <u>https://developer.android.com/reference/android/widget/TextView</u>.

4. За вищенаведеним посиланням самостійно розібратись із параметрами "sp", тощо, у значеннях атрибуту android:textSize. Апробувати вигляд тексту для всіх параметрів.

Рівня складності 3 – факультативні.

5. Самостійно розібратись із представленням натуральних чисел у двійковій та шістнадцятковій системах числення (Відео 2_Двійкова система числення).

6. Самостійно розібратись із параметрами "#RGB", тощо, у значеннях атрибуту android:textColor. Апробувати вигляд тексту для всіх параметрів.