北京邮电大学软件学院

2020-2021 学年第二学期记录文档

课程名称： 移动终端软件开发技术

项目名称： 实验 1：第一次个人 Android 开发实践

项目完成人：

姓名： 马瑞遥 **学号：** **2018211915**

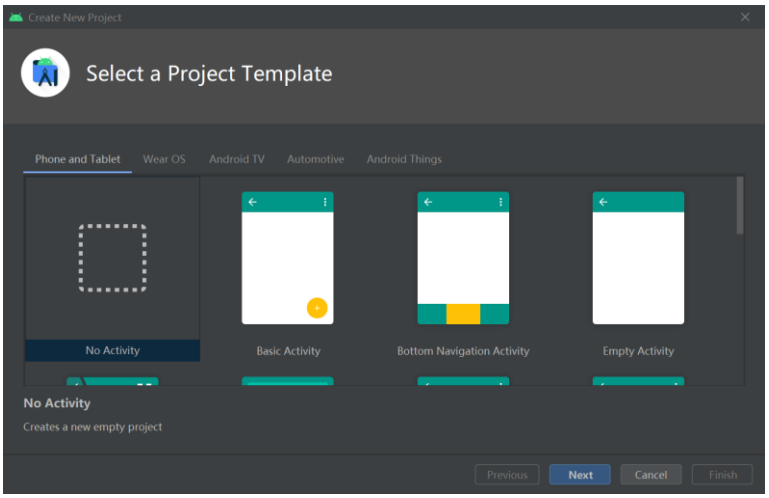
指导教师： 谢坤

日　　期： 2021 年 4 月 1 日

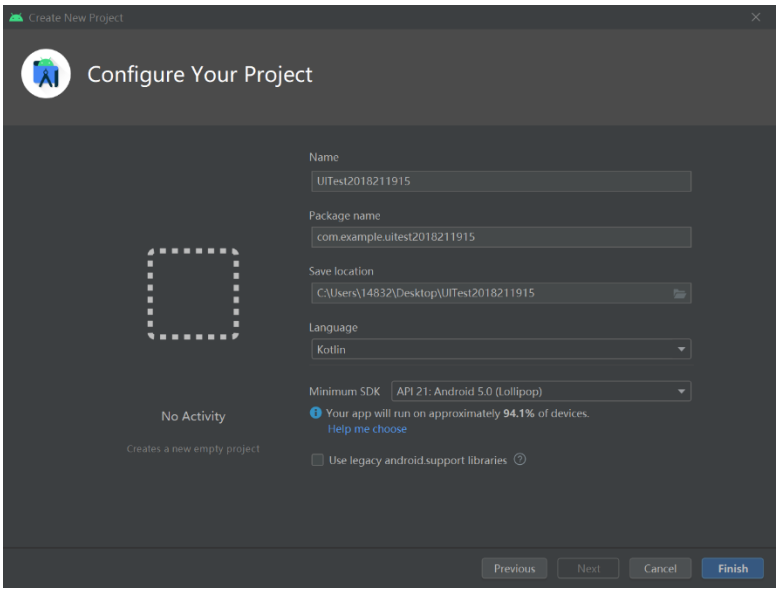
一、 关键步骤描述

1.1 创建项目

1) 打开 Android Studio，选择创建一个 No Activity 的项目。



2) 将项目命名为 UITest2018211915（包含学号），选择存放路径（图中所示为桌面），选择项目所使用的语言为 Kotlin。点击 Finish 按钮，完成项目的创建。



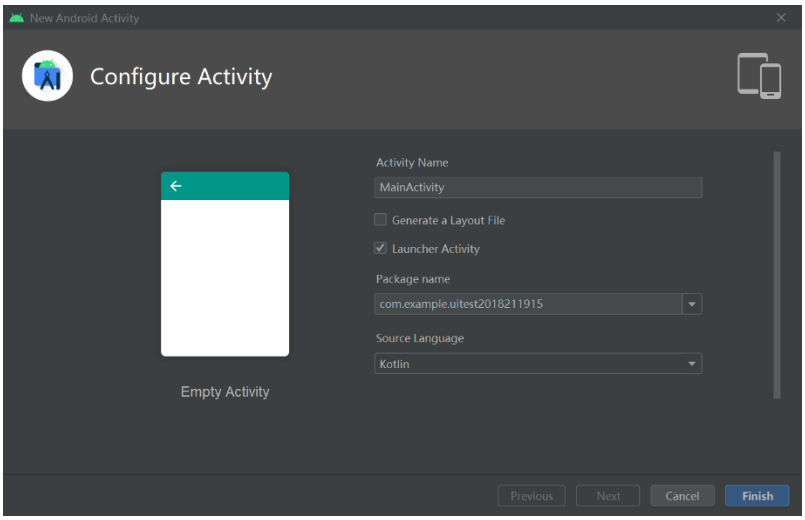
3) 特别地，如果之后项目的存放路径包含中文字符，则需在 gradle.properties 文件中，添加 android.overridePathCheck=true。

```
android.overridePathCheck=true
```

1.2 创建并注册 Activity

1.2.1 创建并注册 MainActivity

1) 打开之前创建的 UITest2018211915 项目，新建一个 Empty Activity。将该 Activity 命名为 MainActivity，并勾选 Launcher Activity 选项，以将该 Activtiy 作为项目启动时所显示的主 Activity。这里暂不勾选 Generate a layout file（生成布局文件）这一选项。



2) 创建好 MainActivity 后，Android Studio 会自动帮我们完成 Activity 的注册与创建。由于勾选了 Launcher Activity 选项，Android Studio 还会将 MainActivity 设置成主 Activity。这之后，在 AndroidManifest.xml 文件中，手动设置 MainActivity 的 label（标题栏、应用等名称）为 My UI Test。

```
//主Activity
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) //创建Activity
```

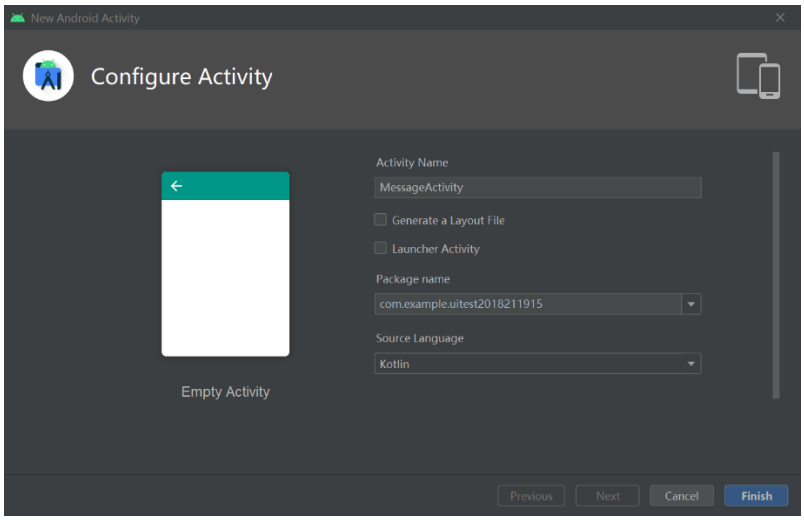
```
<activity
    android:name=".MainActivity"
    android:label="My UI Test">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

至此，MainActivity 的创建与注册便已完成。

1.2.2 创建并注册 MessageActivity

MessageActivity 的创建注册与 MainActivtiy 类似，只是在创建时不勾选 Launcher Activity 选项，即不作为项目启动时所显示的主 Activity。之后，在 AndroidManifest.xml 文件中，手动设置 MessageActivity 的 label（标题栏、应用等名称）为 Message Show 即可。



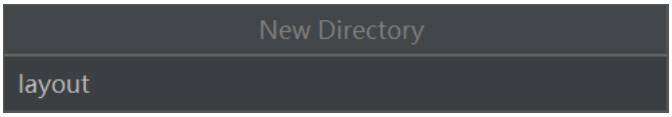

```
//显示信息的Activity
class MessageActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) //创建Activity
```

```
<activity
    android:name=".MessageActivity"
    android:label="Message Show">
</activity>
```

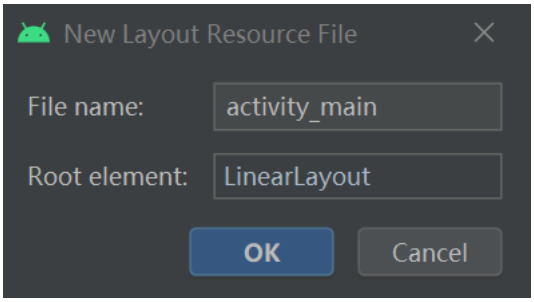
1.3 创建并加载布局

1.3.1 创建并加载 activity_main.xml 布局

1) 新建一个 Directory，并将其命名为 layout。



2) 在 layout 目录下，新建一个 Layout Resource File，将其命名为 activity_main，选择类型为 LinearLayout（线性布局），用于作为 MainActivity 的布局。



3) 为 MainActivity 编写并设计 activity_main.xml 布局。

a) 将整体布局设置为纵向线性排列（vertical）

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

b) 对于每一行的输入信息提示文本及输入框，将其布局设置为局部的横向线性排列（horizontal）。下图为姓名输入信息提示文本及输入框布局的例子展示，其余输入信息提示文本及输入框的布局与其类似。

```
<!--姓名文本及输入框-->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_marginTop="150dp"
    android:orientation="horizontal">
```

c) 设置相应的外边距（margin），并将输入信息提示文本的宽度固定（wrap_content），其余的横向位置均留给输入框占用（即 android:layout_weight="1"，android:layout_width="0dp"）。设置控件的对齐方式为 center_vertical（垂直居中），并为输入框设置相应的提示信息（hint）。下图为姓名输入信息提示文本及输入框布局的例子展示，其余输入信息提示文本及输入框的布局与其类似。

```
<!--姓名文本-->
<TextView
    android:id="@+id/textViewName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="30dp"
    android:text="姓名:"
    android:textSize="20sp" />

<!--姓名输入框-->
<EditText
    android:id="@+id/inputName"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="30dp"
    android:layout_weight="1"
    android:hint="请输入您的姓名" />
```

d) 特别地，对于年龄输入框，将其输入框类型设置为数字格式（number）；对于身高输入框，将其输入框类型设置为带小数点的浮点格式（numberDecimal）。

```
android:inputType="number" />
```

```
android:inputType="numberDecimal" />
```

e) 对于最后一行的显示信息按钮和删除显示按钮，将其布局设置为局部的横向线性排列（horizontal），并设置按钮的对齐方式为 center_vertical（垂直居中）。

```
<!--显示信息与删除显示按钮-->
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="100dp"
    android:orientation="horizontal">
```

```
<!--显示信息按钮-->
<Button
    android:id="@+id/buttonShow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:text="显示信息" />

<!--删除显示按钮-->
<Button
    android:id="@+id/buttonDelete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="60dp"
    android:text="删除显示" />
```

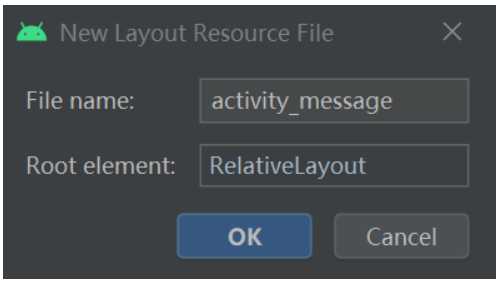
4) 在 MainActivity 的 onCreate()方法中进行 activity_main.xml 布局的加载。

```
//主Activity
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) //创建Activity
        setContentView(R.layout.activity_main) //加载布局
```

1.3.2 创建并加载 activity_message.xml 布局

1) 在 layout 目录下，新建一个 Layout Resource File，将其命名为 activity_message，选择类型为 RelativeLayout（相对布局），用于作为 MessageActivity 的布局。



2) 为 MessageActivity 编写并设计 activity_message.xml 布局。

a) 对于用于显示信息的文本区域，将该控件置于父控件的中心位置，并为其设置左右外边距及字号大小、宽高等。

```
<!--信息显示文本-->
<TextView
    android:id="@+id/textViewMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:text=""
    android:textSize="20sp" />
```

b) 对于返回按钮，将该控件置于父控件的底部位置，并设置相应的下边距，设置该控件的对齐方式为 centerHorizontal（水平居中）。

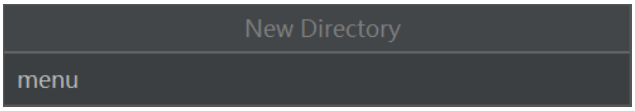
```
<!--返回按钮-->
<Button
    android:id="@+id/buttonBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="60dp"
    android:text="返回" />
```

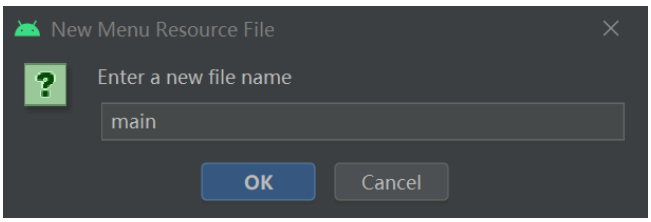
3) 在 MessageActivity 的 onCreate()方法中进行 activity_message.xml 布局的加载。

```
//显示信息的Activity
class MessageActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) //创建Activity
        setContentView(R.layout.activity_message) //加载布局
```

1.4 在 MainActivity 中使用 Menu

1) 新建一个 Directory，并将其命名为 menu。之后，在 menu 目录下新建一个 Menu Resource file，并将其命名为 main。





2) 在 main.xml 中新建四个菜单项，分别是：设置、新建、打印和邮件。

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <!--设置菜单项-->
    <item
        android:id="@+id/itemSet"
        android:title="设置" />

    <!--新建菜单项-->
    <item
        android:id="@+id/itemNew"
        android:title="新建" />

    <!--打印菜单项-->
    <item
        android:id="@+id/itemPrint"
        android:title="打印" />

    <!--邮件菜单项-->
    <item
        android:id="@+id/itemMail"
        android:title="邮件" />

</menu>
```

3) 在 MainActivity 中创建菜单栏，让界面中的菜单项在默认情况下不显示，当用户主动点击菜单按钮时再弹出具体菜单内容。当用户点击菜单栏中的某一项时，使用 Toast 方法弹出相应弹窗信息以提示用户。

```
//创建并加载菜单栏
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.main, menu)
    return true
}
```

```
//点击菜单栏的不同选项，弹出相应弹窗信息以提示用户
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item?.itemId) {
        R.id.itemSet -> Toast.makeText( context: this,  text: "您点击了设置菜单项", Toast.LENGTH_SHORT).show()
        R.id.itemNew -> Toast.makeText( context: this,  text: "您点击了新建菜单项", Toast.LENGTH_SHORT).show()
        R.id.itemPrint -> Toast.makeText( context: this,  text: "您点击了打印菜单项", Toast.LENGTH_SHORT).show()
        R.id.itemMail -> Toast.makeText( context: this,  text: "您点击了邮件菜单项", Toast.LENGTH_SHORT).show()
    }
    return true
}
```

1.5 按钮的监听事件

1.5.1 显示信息按钮的监听事件

显示信息按钮的监听事件写在了 MainActivity 的 onCreate()方法中。在显示信息按钮的 setOnClickListener()方法中，首先，需通过 EditText 的.text.toString()

方法，来获取用户在输入框中输入的姓名、年龄和身高的字符串信息。

之后，对用户的输入信息进行检查，如果用户未填写某项信息，则用 Toast 方法弹出弹窗信息以提示用户。

当用户完整地输入了信息后，则通过 Intent 的 putExtra()方法，将用户信息暂存在 Intent 中。之后，使用显示 Intent，通过 startActivity()方法，启动 MessageActivity，并将用户信息传递给 MessageActivity 以进行显示。

```
//点击显示信息按钮，获取用户输入，启动MessageActivity并进行信息传递与显示
buttonShow.setOnClickListener { it: View!
    //获取用户输入
    val name = nameEditText.text.toString()
    val age = ageEditText.text.toString()
    val height = heightEditText.text.toString()

    if (name == "") { //如果用户未输入姓名，弹出弹窗以提醒用户
        Toast.makeText( context: this, text: "您还未输入姓名", Toast.LENGTH_SHORT).show()
    } else if (age == "") { //如果用户未输入年龄，弹出弹窗以提醒用户
        Toast.makeText( context: this, text: "您还未输入年龄", Toast.LENGTH_SHORT).show()
    } else if (height == "") { //如果用户未输入身高，弹出弹窗以提醒用户
        Toast.makeText( context: this, text: "您还未输入身高", Toast.LENGTH_SHORT).show()
    } else { //如果用户按要求输入信息，启动MessageActivity并进行信息传递与显示
        val intent = Intent( packageContext: this, MessageActivity::class.java)
        val message = name + age + "岁了，身高是" + height + "米"
        intent.putExtra( name: "message", message)
        startActivity(intent)
    }
}
```

相应地，在 MessageActivity 的 onCreate()方法中，需通过 Intent 的 getStringExtra()方法，来获取从 MainActivity 中传递过来的用户信息，并将其转换为字符串类型。之后，需将 MessageActivity 中 TextView 的 text 内容设置为之前从 MainActivity 获取到的用户信息，以便在 MessageActivity 中进行用户信息的展示。

```
val message = intent.getStringExtra( name: "message") //获取从MainActivity中传递过来的用户信息

messageView.text = message //显示信息
```

1.5.2 删除显示按钮的监听事件

删除显示按钮的监听事件写在了 MainActivity 的 onCreate()方法中。在删除显示按钮的 setOnClickListener()方法中，需将所有的 EditText 的 text 内容设置为

null，以对姓名、年龄和身高的输入信息进行清空。

```
//点击删除显示按钮，清空上方输入框中的内容
buttonDelete.setOnClickListener { it: View!
    nameEditText.text = null
    ageEditText.text = null
    heightEditText.text = null
}
```

1.5.3 返回按钮的监听事件

返回按钮的监听事件写在了 MessageActivity 的 onCreate()方法中。在返回按钮的 setOnClickListener()方法中，需调用 finish()方法，以对 MessageActivity 进行销毁。

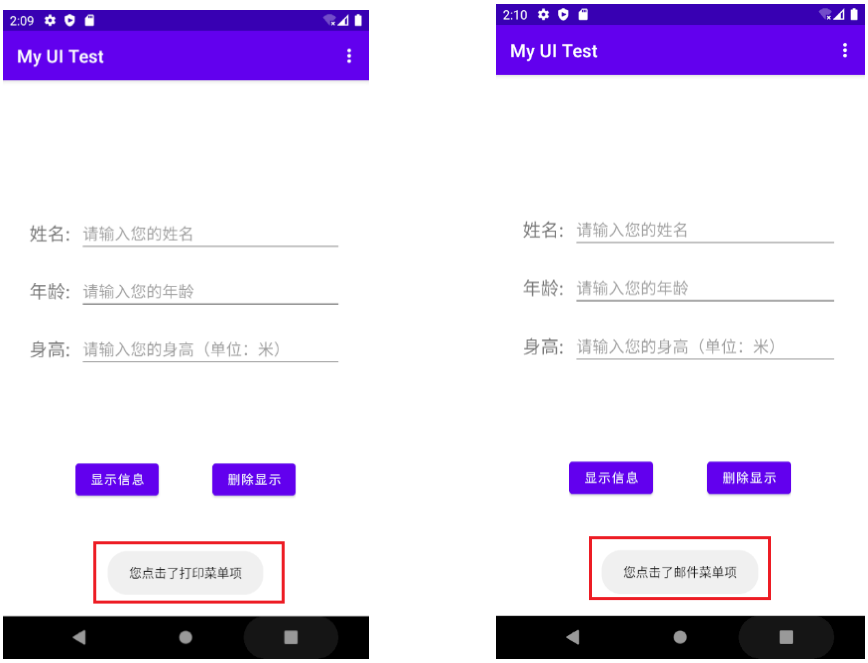
```
//点击返回按钮，返回MainActivity
buttonBack.setOnClickListener { it: View!
    finish()
}
```

二、 运行结果

1) 点击运行按钮，程序便开始运行，其界面展示如下。

2) 点击位于界面右上角的菜单栏，弹出的菜单项有：设置、新建、打印和

邮件。点击某个菜单项，会弹出相应的 Toast 提醒，提醒内容为：您点击了 XX 菜单项。



3) 当用户在界面上方的输入框中完整地输入了信息（姓名、年龄和身高）后，点击显示信息按钮，会弹出另一个界面以显示用户所输入的信息。在另一个界面中，点击返回按钮，便会返回到上一界面（主界面）。

4) 在主界面中，点击删除显示按钮，上方输入框中输入的信息便会被全部清空。

5) 一些健壮性处理。

a) 当用户在输入年龄信息时，程序所弹出的键盘是数字（number）类型的，即用户只能在年龄输入框中输入无符号整数。

b) 当用户在输入身高（单位：米）信息时，程序所弹出的键盘是带小数点的浮点数（numberDecimal）类型的，即用户只能在身高输入框中输入无符号浮点数。

c) 当用户输入的信息不完整、有空项时，如果用户点击了显示信息按钮，程序便会弹出 Toast 提醒以提示用户。此时，程序不再弹出另一个界面以显示用户信息。


2:38
My UI Test
姓名: mry
年龄: 20
身高: 请输入您的身高（单位：米）
显示信息
删除显示
您还未输入身高