**广东省高等教育自学考试《移动应用开发技术》课程考试大纲**

**（课程代码：14567）**

**Ⅰ 课程性质与课程目标**

**一、课程性质和特点**

课程以Android开发为例进行移动应用开发教学，Android开发是移动应用领域被广泛使用的一门移动应用开发技术。《移动应用开发技术》是软件工程专业、计算机应用技术等相关专业的一门重要的应用开发类课程，是面向社会岗位需求的一门课程。课程涉及Android 基础知识、UI界面、数据存储、SQLite数据库、四大组件、网络编程、图形图像处理、多媒体应用等。通过本课程的学习，学生能够掌握Android基础知识，学会编写移动应用程序。

通过该课程的学习，使学生获得基本全面的Android开发知识，掌握基于Android的移动应用开发的科学方法；提高学生软件需求分析、设计与开发能力，使学生能够运用Android针对特定需求完成移动app的分析、设计与开发，为将来从事软件设计与开发打下扎实的基础，以适应当代社会对移动应用软件开发的需求。

**二、课程目标**

1. 熟悉Android开发环境，熟悉Android项目结构、各种资源的管理与使用；熟练掌握Android程序单元的生命周期、创建、跳转与数据传递。
2. 熟练掌握Android界面设计，包括布局以及常用界面控件的设计，能设计基本的移动前端界面。
3. 掌握Android端数据存储方法、数据共享方法；了解Android的广播机制、服务实现方法。
4. 掌握Android访问网络资源方法，能够实现获取网络数据并在前端展示。
5. 了解常用的绘图类及动画设计，能进行图形图像处理。
6. 了解音频、视频的播放方法，能开发简单的音频视频播放app；
7. 通过综合案例的分析、设计与开发，学习移动应用开发的流程、规范和方法。

**三、与相关课程的联系与区别**

移动应用开发与软件技术基础课程密切相关。

软件技术基础课程以java语言为例进行教学，学习数据类型、表达式、流程控制、数组等基本语法的使用，掌握面对对象编程方法以及综合软件分析、设计与开发实践，使学生具备以程序开发解决实际问题能力，是移动应用开发课程的先修课程。

移动应用开发课程以android开发为例进行授课，android应用开发针对手机、智能手表等移动设备进行应用开发，涉及界面设计和代码编写，前端采用xml语言，但一般采用可视化开发，代码编写采用java语言，因此软件技术基础（java）是移动应用开发课程的先修课程。

**四、课程的重点和难点**

本课程的重点难点如下：

1. **课程重点**

本课程重点是培养学生的移动应用开发能力，理解和掌握基于Android的移动应用开发方法，能够利用Android设计移动应用app。重点内容包括：

1. Android项目结构、各类资源的创建、管理与使用。
2. Android前端界面的设计，包括常用布局和控件设计。
3. Android的程序活动单元的管理和事件处理机制。
4. **课程难点**

本课程难点包括：

1. 高级控件RecyclerView的使用；
2. SQLLite数据库操作访问。
3. 网络编程。

**Ⅱ 考核目标**

本大纲在考核目标中，按照识记、领会、应用三个层次规定其应达到的能力层次要求。三个能力层次是递进关系，各能力层次的含义是：

**识记**：要求考生熟悉并理解大纲各章中知识点基本概念。本课程主要包括Android项目结构、各类资源、常见界面布局、常见控件及事件处理机制； Activity创建、配置以及跳转管理；服务、数据存储、数据共享的基本概念；访问网络资源方法以及数据交换格式；图形图像及多媒体应用开发相关技术。

**领会：**指考生能够领悟和理解大纲知识点内涵和外延，涉及的概念及其之间的关系，并能正确使用大纲知识点完成验证性实验，考察考生对相关技术的理解和实践能力。要求考生在开发实验环境下（Android Studio），运用相关技术完成验证性实验，深入理解相关技术、语法的使用方法，比如可以在界面上创建一个按钮，并能正确地设置相关属性。具体包括：创建Android项目、使用相关资源、完成基本的界面设计、实现事件处理；能进行数据存储、网络访问、图形图像以及音频视频多媒体应用设计等Android应用独立功能实现。

**应用：**指考生能够运用大纲知识点解决实际应用问题。本课程要求考生能够针对特定任务，基于Android设计一个独立的功能模块，一般针对某个具体章节知识模块，考察考生对该知识模块中相应知识点的应用能力情况。比如，设计一个用户登录界面，并编写相应的事件处理代码实现用户交互。

**Ⅲ 课程内容与考核要求**

第一章 Android基础入门

**一、学习目的与要求**

1、了解通信技术、Android特点及体系结构。

2、掌握开发环境的搭建，学会使用Android Studio开发工具。

3、掌握Android程序的开发，并动手开发HelloWorld程序。

**二、考核知识点与考核要求**

1. Android简介

识记： 通信技术；Android四层体系结构； Dalvik虚拟机。

2. Android开发环境

识记：Android Studio；SDK;模拟器。

领会：Android Studio安装；模拟器的创建配置；SDK管理。

3. Android项目结构

识记：android主要目录：app、main\java、res；主要配置文件：AndroidManifest.xml、app/build.gradle。

领会： Android项目创建过程；Android项目结构及主要配置文件，包括drawable、values、mipmap、layout等目录和AndroidManifest.xml配置文件的作用。

应用：基于Android Studio创建一个android项目。

4. 资源管理

识记：res/drawable: 图片资源；res/mipmap ：图标资源；res/values：样式文件、字符串资源、颜色资源、尺寸资源；res/layout：布局资源。

领会：各类资源的定义与使用方法。

应用：在项目相关目录中放置相关资源文件，并在项目中使用。

5. 程序调试

识记：Android单元测试；Junit单元测试；Logcat。

领会：单元测试的作用；log的六个级别及对应Log类的静态方法。

应用：能够利用单元测试进行测试；通过Logcat窗口查看调试信息。

**三、本章重点、难点**

1.本章重点

Android开发环境搭建；Android项目创建； Android项目结构；模拟器的创建；DDMS的使用；Android程序打包过程; 资源文件的编写。

2.本章难点

资源文件的设计与使用（尤其样式文件）、AndroidManifest.xml配置文件的理解。

第二章 Android常见布局

**一、学习目的与要求**

1. 了解Android界面开发的常用的几种布局。
2. 掌握常用布局的设计。

**二、考核知识点与考核要求**

1. View控件及界面布局编写方法

识记：View控件关系；两种布局实现方法。

2. 布局的通用属性

识记：理解布局常见属性：id、layout\_width、layout\_height、background、margin、padding。

应用：能针对界面布局进行常用属性设置。

3. 常见布局

识记：约束布局、线性布局、相对布局、表格布局、帧布局中子控件排列方式；各类布局关键属性设置。

领会：能够使用相应布局完成界面控件排列设置，能够正确地设置相关属性。

应用：能够根据界面设计需求，综合利用一个或多个布局实现界面设计。

**三、本章重点、难点**

1.本章重点

约束布局、线性布局的使用与属性设置。

2.本章难点

布局的合理选择以及控件相互关系设置。

第三章 Android 常见控件

**一、学习目的与要求**

掌握常见控件的设计与属性设置。

**二、考核知识点与考核要求**

1. 简单控件

识记： TextView、EditView、Button、ImageView、RadioButton、CheckBox控件及常用属性；Toast类。

领会：创建控件；控件主要属性设置；控件的事件处理。

应用：能够根据界面设计需求，选择合适的控件以及合适的属性设置。能够设置控件事件监听以及事件处理，可以通过代码获取或改变控件属性。

2. 列表控件

识记：RecyclerView。

领会：RecyclerView控件设计步骤：包括界面布局设计、适配器类的设计以及数据与控件绑定。

应用：能够根据界面设计需求，利用RecyclerView实现数据列表展示。

3. 自定义控件

识记：自定义控件概念。自定义控件类结构及主要方法：onMeasure、onDraw、onLayout方法。

**三、本章重点、难点**

1.本章重点

简单控件的使用，包括TextView、EditView、Button、ImageView；RecyclerView控件的使用。

2.本章难点

RecyclerView控件的使用；自定义控件的使用。

第四章 程序活动单元Activity

**一、学习目的与要求**

1. 掌握Activity生命周期；
2. 掌握Activity的创建、配置、启动和关闭；
3. 掌握Intent、Activity跳转；
4. 掌握Activity任务栈和启动模式；
5. 了解Fragment的使用。

**二、考核知识点与考核要求**

1. Activity生命周期

识记：Avtivity生命周期。

领会：理解Avtivity生命周期中不同阶段所执行的方法。

应用：能够针对Avtivity生命周期不同阶段编写相关代码。

2. Avtivity的创建、配置、启动和关闭

识记： Avtivity创建、配置、启动和关闭的作用。

领会：Avtivity创建和配置方法。

应用：创建Avtivity，在AndroidManifest.xml配置Avtivity。

3. Intent和IntentFilter

识记：Intent作用、分类和属性设置；IntentFilter作用和属性。

领会：显式Intent实现跳转代码；隐式Intent属性设置和在代码中引用。

4. Avtivity跳转和数据传递

识记：Avtivity跳转和数据传递。

领会：Intent跳转方法、参数传递方法，Activity数据回传方法。

应用：能够实现多个Activity通过Intent跳转、参数传递及数据回传。

5. 任务栈和启动模式

识记：Avtivity管理任务栈；四种启动模式standard、singleTop、singleTask和singleInstance。。

领会：Avtivity启动模式设置方法。

6. Fragment

识记：Fragment的作用、生命周期，与Activity关系。

领会：Fragment创建与管理。

应用：能够实现Activity管理多个Fragment片段移动程序。

**三、本章重点、难点**

1.本章重点

Avtivity创建和配置方法；Intent跳转方法和参数传递方法。

2.本章难点

Intent参数传递和数据回传、Fragment。

第五章 数据存储

**一、学习目的与要求**

1. 了解5种数据存储方式，掌握文件和SharedPreferences存储方式。
2. 掌握SQLite数据库的使用，能够实现增删改查操作功能。

**二、考核知识点与考核要求**

1. 文件存储

识记：两种文件存储方式；文件存储原理。

领会：两种文件存储方式的读写操作步骤及相关的权限设置。

应用：能实现文件的内部存储和外部存储的读写操作功能。

2. SharedPreferences存储

识记：SharedPreferences存储原理、应用场景、XML文件格式。

领会：SharedPreferences存储的读写删除方法。

应用：利用SharedPreferences存储保存、读取、修改数据。

3. SQLite数据库

识记：SQLite数据库概念、作用；SQLite数据库及表的创建方法。

领会：SQLite数据库操作数据相关语法。

应用：能编写SQLite数据库操作类，实现数据库及表的创建，实现数据的增删改查等功能。

**三、本章重点、难点**

1.本章重点

SharedPreferences存储；SQLite数据库的编程。

2.本章难点

SQLite数据的增删改查等功能实现。

第六章 内容提供者和内容观察者

**一、学习目的与要求**

1、要求学生掌握内容提供者的创建，并能使用内容提供者操作数据；

2、要求学生了解内容观察者的使用，学会使用内容观察者观察其他程序的数据变化

**二、考核知识点与考核要求**

1. 内容提供者

识记：ContentProvider的作用及应用场景。

领会：ContentProvider的创建方式及在AndroidManifest.xml中配置。

应用：能根据项目需要创建ContentProvider。

2. 访问内容提供者

识记：ContentResolver的作用及应用场景。

领会：ContentResolver访问其他程序的一般过程；Uri.parse、query方法的使用。

应用：能够利用ContentResolver访问其他应用程序数据。

3. 内容观察者

识记：内容观察者ContentObserver的工作原理及用法。

领会：内容观察者ContentObserver实现方法；ContentObserver、onChange方法的使用。

应用：能够利用内容观察者ContentObserver观察指定Uri的数据。

**三、本章重点、难点**

1.本章重点

内容提供者的使用、对数据进行增删改查、内容观察者的使用。。

2.本章难点

内容提供者的使用、内容观察者的使用。

第七章 广播机制

**一、学习目的与要求**

1、要求学生掌握广播接收者的创建，以及如何自定义广播；

2、要求学生掌握有序广播和无序广播的使用，能够对有序广播进行拦截。

**二、考核知识点与考核要求**

1. 广播机制概述

识记：广播机制原理及实现流程。

领会：广播机制的应用场景。

2. 广播接收者

识记：广播接收者BroadcastReceiver作用及过程。

领会：广播接收者BroadcastReceiver创建方法。

应用：能够利用BroadcastReceiver实现广播消息的接收和处理。

3. 自定义广播和广播类型

识记：自定义广播发送和接收过程；有序广播和无序广播的工程流程。

领会：自定义广播程序结构、创建方法；有序广播和无序广播的动态注册、广播发送等方法。

应用：能实现自定义广播程序创建及使用；能根据实际需求选择实现序广播或无序广播实现广播消息的发送和接收。

**三、本章重点、难点**

1.本章重点

自定义广播的发送与接收、广播的类型（有序广播和无序广播）。

2.本章难点

自定义广播和广播类型。

第八章 服务

**一、学习目的与要求**

1、要求学生掌握服务的生命周期，以及启动服务的两种方式；

2、要求学生学会使用服务与Activity通信，并且能够完成音乐播放器案例。

**二、考核知识点与考核要求**

1. 服务概述

识记：服务（Service）的作用；后台运行、跨进程访问概念。

2. 服务的创建

识记：服务的创建方法。

领会：服务程序结构；服务配置；

3. 服务的生命周期

识记：理解服务的生命周期顺序及对应的方法。

4. 服务的启动方式

识记：两种服务启动方式StartService、BindService。

领会：服务启动方法，StartService和BindService服务各生命周期对应方法的代码编写。

应用：能针对特定需求，创建服务程序，实现后台运行功能。

5. 服务的通信

识记：两种服务的通信方式：本地服务通信和远程服务通信。

领会：两种通信方式服务的创建、流程及所涉及的主要方法。

应用：能结合实际需求，如音乐的后台播放，实现服务发创建以及通信管理。

**三、本章重点、难点**

1.本章重点

服务的创建、服务的生命周期、服务的启动方式。

2.本章难点

服务的启动方式。

第九章 网络编程

**一、学习目的与要求**

1、了解HTTP协议，学会使用HttpURLConnection访问网络；

2、了解WebView的使用；

3、掌握Handler原理，会使用Handler进行线程间通信；

4、掌握Json数据的解析；

**二、考核知识点与考核要求**

1. 通过HTTP访问网络

识记：HTTP请求和响应过程；HttpURLConnection访问网络的GET与POST请求方式。

领会：HttpURLConnection访问网络的流程及所涉及的关键代码；

应用：能用HttpURLConnection通过get和post方法访问网络数据。

2. 使用WebView访问网络

识记：WebView控件的作用、代码及常用方法。

领会：利用WebView显示网页的流程，包括权限设置、界面设计、代码编写；

应用：能针对需求，利用WebView控件实现网页的访问，包括执行Html代码、支持JavaScript代码等。

3. JSON数据解析

识记：JSON数据格式及在网络访问中的作用。

领会：JSON格式的对象结构、数组结构；掌握两种JSON解析方法，包括使用JSONObject、JSONArray解析和Gson解析。

应用：能针对需求，利用WebView控件实现网页的访问，包括执行Html代码、支持JavaScript代码等。

4. Handler消息机制

识记：Handler消息机制，Message、Handler、MessageQueue、Looper四个关键对象作用。

领会：Handler消息机制处理流程及四个关键对象之间关系。

**三、本章重点、难点**

1.本章重点

使用HttpURLConnection访问网络、JSON数据的解析。

2.本章难点

HttpURLConnection访问网络。

第十章 图形图形处理

**一、学习目的与要求**

1、掌握常用绘图类，能够绘制图形图像；

2、掌握常见几种动画效果的实现方法。

**二、考核知识点与考核要求**

1. 常用绘图类

识记：Bitmap类、BitmapFactory类、Paint类、Canvas类、Matrix类等几种绘图类的作用。

领会：绘图类的程序结构；几种绘图类主要方法。

2. 动画

识记：补间动画、逐帧动画、属性动画三种动画效果。

领会：动画xml文件配置；补间动画平移、缩放、旋转、改变透明度的实现过程；逐帧动画的原理和实现方法；属性动画Animator类创建、动画效果添加及监听等

**三、本章重点、难点**

1.本章重点

图形图像绘制、补间动画、逐帧动画。

2.本章难点

图形绘制。

第十一章 多媒体应用开发

**一、学习目的与要求**

1、掌握使用MediaPlayer、SoundPool播放音频；

2、掌握使用VideoView播放视频。

3、掌握使用MediaPlayer、SurfaceView播放视频；

**二、考核知识点与考核要求**

1. 音频播放

识记：MediaPlayer类、SoundPool类作用、异同及主要方法。

领会：MediaPlayer类和SoundPool类播放音频文件程序设计主要步骤和关键方法；

2. 视频播放

识记：VideoView和 SurfaceView的作用和差异。

领会：VideoView和 SurfaceView组件的主要方法，实现播放视频文件的开发流程；

**三、本章重点、难点**

1.本章重点

利用MediaPlayer类、SoundPool类实现音频播放；SurfaceView和MediaPlayer类实现视频播放。

2.本章难点

利用SurfaceView和MediaPlayer类实现视频播放。

**Ⅳ 关于大纲的说明与考核实施要求**

**一、自学考试大纲的目的和作用**

课程自学考试大纲是根据专业自学考试计划的要求，结合自学考试的特点而确定。其目的是对个人自学、社会助学和课程考试命题进行指导和规定。

课程自学考试大纲明确了课程学习的内容以及深广度，规定了课程自学考试的范围和标准。因此，它是编写自学考试教材和辅导书的依据，是社会助学组织进行自学辅导的依据，是自学者学习教材、掌握课程内容知识范围和程度的依据，也是进行自学考试命题的依据。

**二、课程自学考试大纲与教材的关系**

课程自学考试大纲是进行学习和考核的依据，教材是学习掌握课程知识的基本内容与范围，教材的内容是大纲所规定的课程知识和内容的扩展与发挥。

**三、关于自学教材**

《Android移动开发基础案例教程》，黑马程序员主编，人民邮电出版社，2021年第2版。

教材第12章适合作为课程设计的参考样例，不纳入本大纲考核范围，考生可根据个人兴趣学习。

**四、关于自学要求和自学方法的指导**

本大纲的课程基本要求是依据专业考试计划和专业培养目标而确定的。课程基本要求还明确了课程的基本内容，以及对基本内容掌握的程度。基本要求中的知识点构成了课程内容的主体部分。因此，课程基本内容掌握程度、课程考核知识点是高等教育自学考试考核的主要内容。

为有效地指导个人自学和社会助学，本大纲已指明了课程的重点和难点，在章节的基本要求中一般也指明了章节内容的重点和难点。

**（一）多动手实验，理论与实践相结合**

移动应用开发技术是一门动手编程能力培养课程，自学者应理论和实践相结合学习本门课程。搭建好实验环境，多动手进行编码实验，在学习过程中参照教程中语法代码、案例，在电脑上调试运行，通过理论介绍和实验结果对比，能更深入理解相关语法含义，通过理解能更容易记忆相关知识点。

**（二）****梳理知识脉络，多做理论题练习，加深知识点理解和记忆**

移动应用开发技术是重点培养动手实践开发能力课程，但课程包括理论考试和实践考试，实践考试往往针对基础的重要的实践性强的知识点，在实验过程中也可以查看帮助或集成环境的提示，并不需要记忆太多的语法细节。理论考试题涉及面广，考察的较全面和系统的知识点，需要系统的学习和归纳，自学者通过思维导图等形式总结知识脉络。另外需要多进行理论习题练习，加深对知识点的理解和记忆，同时要特别注意该题所涉及的相关知识点。

**（三）复现案例，提高程序开发能力**

移动应用开发同java语言等基础编程语言不同，基础编程语言往往一条命令也可以执行，很容易验证某条命令执行效果。Android开发是面向实际社会需求岗位的一门技术，项目结构较为复杂，每一个知识点的实现需要才能完成，往往需要对整体知识体系有一定的理解才能实现一个功能。复现案例是学习移动开发的一个好方法，能够快速帮助初学者建立系统的知识体系。

**（四）基于项目驱动方法学习，提升软件开发综合能力**

以团队或个人基于某个特定应用场景，开发一个移动应用项目，解决特定实际问题。可以参考综合案例，按照系统分析、功能设计、系统实现等软件开发步骤，开发一个小型应用系统。通过项目的设计与开发，能够提高综合分析、设计、编码以及创新应用能力，能够更全面训练移动应用开发能力。

**五、对社会助学的要求**

**（一）移动开发入门指导**

移动应用开发技术课程以编程为主，自学者往往在刚入门时在开发环境、编码过程、程序编写规范、运行调试等方面遇到各种问题。助学者可就以上几个方面进行指导，帮助自学者能够独立的进行完整的编码实验，并给出编程中遇到问题如何解决的自主学习方法建议。

**（二）注意培养自学者自主学习能力**

程序编写总会遇到各种问题，包括语法错误、逻辑错误、环境错误等各种问题。助学者应在指导过程中注重培养学生自主学习解决问题的能力。首先指导学生学会调试程序，看懂程序报错信息，根据报错信息去互联网搜索解决问题办法；其次，培养学生养成良好的编码习惯，学会通过集成工具查看各种帮助信息；最后要学会充分利用各种网络学习资源培养自主学习能力，善于利用人工智能进行辅助学习，例如利用大语言模型文心一言进行java学习规划、答疑、代码生成以及优化等。

**（三）帮助自学者对重点难点的辅导**

Android界面设计是课程重点，是整个课程的基础，助学者应帮助自学者理解UI设计，深入理解UI设计布局及控件设计，各种资源的管理以及在界面中使用，列表数据RecyclerView控件是移动开发广泛使用的控件，也是难点，帮助自学者深入理解适配器的使用以及与数据的绑定。

结合案例分析是帮助自学者全面理解移动应用开发的重要手段，应针对一个综合案例，按照软件开发流程，从分析、设计到实现整个过程，帮助自学者掌握移动应用开发流程和技能要求。

**（四）建议每学分2-3个助学学时。**

**六、对考核内容的说明**

本课程要求考生学习和掌握的知识点内容都作为考核的内容。课程中各章的内容均由若干知识点组成，在自学考试中成为考核知识点。因此，课程自学考试大纲中所规定的考试内容是以分解为考核知识点的方式给出的。由于各知识点在课程中的地位、作用以及知识自身的特点不同，自学考试将对各知识点分别按三个认知层次确定其考核要求。

**七、关于考试命题的若干规定**

1、本课程考试采用闭卷笔试方式考核，考试时间150分钟，按百分制记分,60 分为及格。

2、本大纲各章所规定的基本要求、知识点及知识点下的知识细目，都属于考核的内容。考试命题既要覆盖到章，又要避免面面俱到。要注意突出课程的重点、章节重点，加大重点内容的覆盖度。

3、命题不应有超出大纲中考核知识点范围的题，考核目标不得高于大纲中所规定的相应的最高能力层次要求。命题应着重考核自学者对基本概念、基本知识和基本理论是否了解或掌握，对基本方法是否会用或熟练。不应出与基本要求不符的偏题或怪题。

4、本课程在试卷中对不同能力层次要求的分数比例大致为：识记占30%，领会占40%，应用占30%。

5、要合理安排试题的难易程度，试题的难度可分为：易、较易、较难和难四个等级。每份试卷中不同难度试题的分数比例一般为：2:3:3:2。

6、课程考试命题的主要题型一般有单项选择题、填空题、简答题、程序设计题、程序填空题等题型。

**附录 题型举例**

1. **单项选择题**

在下列选项中，Android中的五大存储数据方式不包括（ ）。

A.文件 B.SQLite数据库 C.SharedPreferences D.Map

1. **填空题**

Android中的布局文件位于【 】目录中。

1. **简答题**

请简要介绍一下Android中内容提供者的作用。

1. **程序设计(填空)题**

开发一个“标准体重计算”app，功能要求如下：

布局文件activity\_main.xml，界面设计如图所示。

事件处理：选择性别并输入身高，单击“计算”按钮触发事件，事件过程根据性别、身高数据，计算标准体重，在体重对应文本框中显示标准体重。

男性：标准体重 =(身高cm－80)×70﹪ 女性：标准体重=(身高cm－70)×60﹪

根据界面和功能，补充布局文件和Activity代码，请在“\_\_\_\_\_\_”处补充完整(填写行数不一定与“\_\_”行数量一致)。

参考界面布局如下



布局界面代码activity\_main.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/LinearLayout1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/skyblue"

android:orientation="vertical"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

tools:context="com.example.kaoshi2016.MainActivity">

<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

android:id="@+id/textView1"

style="@style/CustomTitleFontStyle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:text="标准体重计算"

android:textAppearance="?android:attr/textAppearanceLarge" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:text="性别" />

<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

android:id="@+id/checksex"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="男" />

</LinearLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="身高" />

<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

android:id="@+id/high"

android:layout\_width="252dp"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="60至210cm之间"

android:inputType="number" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="体重" />

<EditText

android:id="@+id/weight"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="0.40"

android:hint="10到200kg之间" />

</LinearLayout>

<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:onClick="submit"

android:text="计算" />

</LinearLayout>

代码MainActivity.java

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private EditText high, weight;

private CheckBox sex;

private Button btn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

补充代码

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

public void submit(View view) {

补充代码

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

}