

2018

计算机软件技术基础

Android 项目描述

程源

广东机电职业技术学院

2018/9/1

目录

如何学习 Android 开发.....	- 2 -
本课程所需的基础.....	- 3 -
参考书	- 3 -
本课程内容	- 3 -
开发工具	- 4 -
类的概念	- 4 -
什么是类	- 4 -
使用 new 实例化对象.....	- 7 -

0 引言

如何学习 **Android** 开发

学习 **Android** 开发，对于每个新手都是一个巨大的挑战。**Android** 有自己的语言文化，即 **Java** 语言。但仅掌握 **Java** 还远远不够，还需要学习很多新的理论和技术知识。本课程旨在理清头绪，引领同学们穿越 **Android** 这个陌生的领域。

要成为一名合格的 **Android** 开发人员，需要做到：

- 着手开发一些 **Android** 应用；
- 彻底理解你的 **Android** 应用。

本课程将通过指导同学们完成两个 **Android** 应用开发，协助同学们完成上述两件事情，并根据需要逐步介绍计算机软件开发的各种概念、理论和技术知识。

本课程的教学方法是：在着手开发实际应用的同时学习理论知识，而不是先学习一堆理论，再考虑如何将理论应用于实践。

学完本课程，同学们将具备必要的开发经验和理论知识，成为一名 **Android** 开发者。以此为起点，能够进行实际开发并继续深入学习。

本课程所需的基础

学习本课程，同学们需要了解 Java 语言，包括类、对象、接口、监听器、包、内部类、匿名内部类、泛型等基本概念。由于多数同学目前对这些概念还比较陌生，建议找本 Java 入门书用一两周时间学习一下。

对于接口、匿名内部类等重要的 Java 语言点，我会做必要的简短回顾。

参考书

《Android Programming (Android 编程权威指南)》。

本课程内容

本课程通过一个精心安排的商业及完整的 Android 应用，使同学们在开发实践过程中学到重要的理论知识和开发技巧，并从中获得最直接的开发经验。

Vocabulary（词汇学习）——本课程贯穿始终的一个应用。通过它，我们将学习 Android 应用的基本组成、activity、界面布局（layout）以及显式 intents。以及类、监听器、接口与内部类和设计模式等软件技术基础类容。

同时我们会通过对 Vocabulary 的不断升级，学习片段（fragment）、菜单选项、相机调用、隐式意图（implicit intent），数据持久化、语音识别（AI）、后台服务和 HTTP 等高级内容。

开发工具

最好有 4G 以上内存的笔记本电脑。

学习本课程前，需要准备 Android Studio 作为开发工具。

面向对象的概念

面向对象程序设计（英语：Object-oriented programming，缩写：OOP）是一种程序设计范型，同时也是一种程序开发的方法。对象指的是类的实例。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性。

什么是类

类（**class**）。我们可以将手机（Mobile）想象成一个类。用 java 语言表述，即

```
public class Mobile {.....}
```

但是 Mobile 是个很抽象的概念，市面上没有 Mobile 这个具体的实物，只有诸如华为荣耀 8、小米 note2 或 Iphone6s 等具体的手机。对于抽象的类型，在 java 语言中，是在 class 前面加上 **abstract** 描述，即

```
public abstract class Mobile {.....}
```

事实上，我们说自己有手机，通常是说自己有电话等联系方式，即

Mobile 这类事物至少有一个共同的功能，即“打电话”。如果没有这个功能，就不能称其为手机。用 java 语言表述为：

```
public abstract class Mobile {  
    private Button mCallButton;//至少有一个打电话的按钮，成员变量  
    private String mPhoneNumber;//至少有能显示电话号码拨号盘成员变量  
    protected double mPrise;//价格，成员变量，这个可以有  
    protected void call(String phoneNumber) {.....}//要有个拨打电话号码的方法  
}
```

但是，如何“拨打电话号码”，不同类型的手机各不相同。**Mobile** 这个类没有告诉我们，因此“拨打电话号码”是个抽象的方法，同样，我们也在其前面加上 **abstract**：

```
public abstract class Mobile {//一个抽象类  
    protected Button callButton;//至少有一个打电话的按钮  
    protected String mPhoneNumber;//至少有能显示电话号码拨号盘  
    protected double mPrise;//价格，成员变量  
    protected abstract boolean call(String phoneNumber);//没有实现电话号码的抽象方法  
}
```

注释：**protected** 什么意思？只有儿子能继承，其他类型不能继承。当然还有 **private**，自己私有的，谁也不能继承；**public**，共有的，谁都可以使用。**call** 方法前面的 **boolean** 是什么？一种数据类型，只有：**true** 和 **false** 两种可能。拨通了，**true**；没拨通，**false**。**call** 方法后面括号内的 **call(String phoneNumber)** 是参数，即传给 **call** 方法的电话号码。

那么 **iphone8** 呢？当然是一款具体类型 **Mobile**。先有 **Mobile**，然后才有 **iphone8**，也就是 **Mobile** 的儿子。我们称其为 **Mobile** 的子类。**Mobile** 所有的财产都被它继承了，用 java 语言表述为：

```
public class IPHONE8 extends Mobile {//一个具体的类  
    protected Button callButton;//因为继承自 Mobile，这里就不需要写出来了。  
    protected String mPhoneNumber;//同上  
    protected double mPrise;//价格，成员变量
```

```
.....//其他 iphone6s 特有的按钮
protected boolean call(String phone number) {.....} //有实现拨电话号码的方法
.....//其他 iphone6s 特有的方法
}
```

当然，还有很多其他类型的手机，我们可以表述为：

```
public class MInote2 extends Mobile { //小米 note2 一个具体的类
..... //其他小米 note2 特有的按钮
protected boolean call(String phone number) {.....} //小米 note2 实现拨电话号码的方法
..... //其他小米 note2 特有的方法
}
```

```
public class Honor8 extends Mobile { //华为荣耀 8 一个具体的类
.....//其他华为荣耀 8 特有的按钮
protected boolean call(String phone number) {.....} //华为荣耀 8 拨电话号码的方法
.....//其他 iphone6s 特有的方法
}
```

当然，世上没有凭空的手机，都是手机厂商生产，更确切地说是构造（OEM）出来的，并在商店里或网上销售，并标出价格、性能等等。用 java 语言表述为：

```
public class IPHONE8 extends Mobile { //一个具体的类
    public IPHONE6s(double price){
        mPrise = price;
        .....
    } //构造函数，这里仅标明价格
    .....//其他 iphone8 特有的按钮
protected boolean call(String phone number) {.....} //有实现拨电话号码的方法
    .....//其他 iphone6s 特有的方法
}
```

```
public class MInote2 extends Mobile { //小米 note2 一个具体的类
    ..... //其他小米 note2 特有的按钮
    public class MInote2 (double price) {
        mPrise = price;
        .....
    } //构造函数，这里仅标明价格;
```

```

        protected boolean call(String phone number) {.....} //小米 note2 实现拨电话号码的方法
        ..... //其他小米 note2 特有的方法
    }

    public class Honor8 extends Mobile { //华为荣耀 8 一个具体的类
        ..... //其他华为荣耀 8 特有的按钮
        public Honor8 (double price) {
            mPrise = price;
            .....
        } //构造函数，这里仅标明价格;
        protected boolean call(String phone number) {.....} //华为荣耀 8 拨电话号码的方法
        ..... //其他 iphone6s 特有的方法
    }

```

可以看到，构造函数与类同名，在构造具体手机对象时，构造函数会运行，以便将实例对象初始化。何为“实例对象初始化”？

使用 **new** 实例化对象

前面我们创建了三款手机类型。注意，这里指的是手机类型，而不是你的手机。不能说“iPhone8 是我的”，只能说：“我有一款 iPhone8 手机”，即我买的那款具体的 MyIphone8 手机。这里，具体的 MyIphone8 手机就是 iPhone8 类的对象的实例。它是一款实实在在的手机，拥有 iPhone8 手机的全部功能。那么，这款手机何时是我的？当我在手机商店交钱的那一刻，就具体实例化了我的 MyIphone8 手机。用 java 语言表述为：

```

iPhone8 MyIphone8 = new iPhone6s(6000); //我买了一款 iPhone6s 手机,6000 元
Honor8 MyHonor8 = new Honor8(4000);
MInote2 MyMInote2 = new MInote2(3800);

```

有了实例化的 MyIphone8 手机，就可以拨打电话了：

```

boolean calling = MyIphone8.call("13902088888"); //用 iPhone6s 手机打 13902088888。
If(calling){.....} //通话，即 calling=true
else {.....} //挂断,即 calling=false

```

注意 MyIphone8 后面那个点“.”，这个必须有。

关于类，先介绍这么多，因为对于本讲，这些知识已经足够了。下面我们将开启我们的 Android 旅程。