

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™

移动与嵌入式开发技术



Beginning Flash, Flex, and AIR Development for Mobile Devices

Flash+Flex+AIR

移动开发入门经典

——适用于Android、iOS和BlackBerry

[英] Jermaine G. Anderson 著

陈武 陈俊 译



清华大学出版社

移动与嵌入式开发技术

Flash+Flex+AIR 移动开发入门经典

——适用于 Android、iOS 和 BlackBerry

[英] Jermaine G. Anderson 著

陈 武 陈 俊 译

清华大学出版社

北 京

Jermaine G. Anderson
Beginning Flash, Flex, and AIR Development for Mobile Devices
EISBN: 978-0-470-94815-6
Copyright © 2012 by Wiley Publishing, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2012-0333

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Flash+Flex+AIR 移动开发入门经典——适用于 Android, iOS 和 BlackBerry/(英)安德森(Anderson, J. G.)
著; 陈武, 陈俊 译. —北京: 清华大学出版社, 2012.9

(移动与嵌入式开发技术)

书名原文: Beginning Flash, Flex, and AIR Development for Mobile Devices

ISBN 978-7-302-29544-0

I. ①F… II. ①安… ②陈… ③陈… III. ①移动电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2012)第 170532 号

责任编辑: 王 军 于 平

装帧设计: 牛艳敏

责任校对: 蔡 娟

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 22.5

字 数: 548 千字

版 次: 2012 年 9 月第 1 版

印 次: 2012 年 9 月第 1 次印刷

印 数: 1~3500

定 价: 49.00 元

产品编号: 042794-01

前 言

当今对移动内容和应用程序存在巨大的需求。许多世界领先的设备制造商和平台都开始支持 Adobe Flash Player，并且其中的许多厂商还直接在移动设备的操作系统中集成 Adobe AIR，因此当前是学习如何为小屏幕设计并制作内容的最好时机。

随着移动行业快速前进的步伐，跟进了解 Adobe Flash 平台的最新发展是非常重要的，因此本书包括了自从笔者前一本书 *Professional Flash Lite Mobile Development* (Wrox, 2010) 出版之后移动平台 Flash 的所有关键性发展。

本书将介绍如何利用行业领先软件来制作移动内容。通过学习，你将充分了解为移动设备优化的 Adobe Flex 框架和 MXML 组件，并且学习如何利用 Flash Player 10.3 和 AIR ActionScript 3.0 API。

本书介绍的内容适合于各种层次的开发人员。它将帮助你从一个基础水平开始创建基于 Flash 的移动应用程序。本书还包含大量程序代码示例，并进行了详细的解释，基本上涵盖了如何从零开始使用 ActionScript 3.0 创建适合于 Flash Player 10.1 和 AIR 2.5 目标平台的移动应用程序。本书还适合于那些希望跨多种平台来展示移动内容的开发人员。

本书读者对象

本书的目标读者定位于那些希望创建和分发新移动应用程序的移动开发人员。

有各种经验的程序员和开发人员也可以参考本书来了解如何使用 Adobe Flash、Flex 和 AIR 来制作适用于手机和设备的内容。

本书的目的是帮助有经验的移动开发人员和新手综合了解 Flash、Flex 和 AIR。就此而言，第 1 章和第 2 章主要定位于 Adobe Flash 平台的新手，介绍 Flash、Flex 和 AIR 的背景，以及用于创建移动应用程序的工具——即 Flash Builder 和 Device Central CS 5.5。

本书主要内容

本书为读者介绍的许多主题涵盖了针对 Adobe Flash 平台当前交互进行开发的关键方面，通过如下章节引导读者：

- **第 1 章——Flash、Flex 和 AIR 简介：**该章概述了 Adobe Flash 平台，包括 Flash Player、Flex 框架以及 Adobe 集成运行时。其中还为读者提供了关于使用 AS3 编程的基础知识和 MXML 的简要介绍。
- **第 2 章——入门：**该章重点介绍了使用工具进行移动开发的入门知识，其中介绍

了用于开发和测试 Flash 内容的 3 个基本工具：Adobe Flash Builder、Adobe Flash Professional CS5 和 Adobe Device Central CS5。

- **第 3 章——为 Android、BlackBerry 和 iOS 设备构建 AIR 应用程序：**该章介绍了如何为 Google Android 移动平台构建 AIR 2.5 应用程序，并且使用了大量篇幅重点介绍 Adobe AIR Application Descriptor 设置。该章还介绍了如何为 Google Android 平台更新 AIR 应用程序。
- **第 4 章——触摸、多点触摸和手势：**该章介绍了 Flash Player 10.1 中引入的用户输入功能，并且提供了大量可以跟随练习的程序代码清单，使用 Adobe Flash Builder 4.5 和 Adobe Device Central CS5 来创建和测试示例。
- **第 5 章——为多种屏幕尺寸进行开发：**该章介绍了为具有不同屏幕尺寸的多种设备创建内容的最佳实践。
- **第 6 章——调试应用程序：**该章介绍了如何在 Adobe Flash Builder 中使用 Flash Debug Perspective。在这一章中还介绍了错误处理，特别是全局错误处理，它是 Flash Player 10.1 中引入的一个新功能。
- **第 7 章——使用文件系统：**该章详细介绍了如何使用 AIR 文件系统 API，并介绍了在 Adobe Flash Builder 中使用 MXML 和 Flex 框架创建 Files Explorer 移动应用程序的详细步骤。
- **第 8 章——操作数据：**该章介绍了一些在移动应用程序中使用数据的方法。它还重点介绍了 SQLite，并逐步骤介绍了如何创建一个 Teams 数据库应用程序。
- **第 9 章——使用音频和视频：**该章重点介绍了在移动应用程序中添加声音和视频的多种方法，并且还介绍了开放源代码媒体框架(Open Source Media Framework, OSMF)。
- **第 10 章——利用设备功能：**该章将注意力转到 AIR 2.7 中所引入的特定依赖于设备支持的 API，包括利用设备的摄像头、麦克风、Web 浏览器和地理位置功能。

本书编排方式

本书以这样一种方式编排，因此读者可以从任何一章开始阅读。

根据设计，第 1 章至第 3 章包含的程序代码与其他章相比较少；从第 4 章开始将会显著增加示例程序的数量，在阅读时可以跟随这些步骤操作并实现相应的应用程序。

本书每一章的开头都会提供这一章的目标列表和简介，末尾则会提供本章内容小结、练习和本章中学习的概念。

第 10 章将会讲解更多需要用户使用移动设备来测试内容的任务。

本书适用的环境

使用本书需要安装如下操作系统中的一种：

- Mac OS
- Windows
- Linux

为了使用程序代码示例并运行本书中的示例应用程序，需要用到如下的工具：

- Adobe Flash Builder 4.5
- Adobe Device Central CS 5.5
- Adobe AIR 2.7 SDK

虽然完成本书中的全部任务不一定明确需要一台支持 Flash 或 AIR 的移动设备，但是推荐使用一台运行 Gingerbread 2.3.4 的 Google Android 设备。书中介绍的大多数示例，除了能够在 Google Android 上运行之外，还可以在版本 4.x 及以上的 iOS 设备上运行，包括 iPad、iPhone 和 iPod Touch 设备。所有示例还能够在运行 BlackBerry Tablet OS 上的 BlackBerry PlayBook 设备上运行。

源代码

在读者学习本书中的示例时，可以手动输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 <http://www.tupwk.com.cn/download> 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击 Download Code 链接，就可以获得所有的源代码。



由于许多图书的标题都很类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-94815-6。

在下载代码后，只需要用解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

目 录

第 1 章	Flash、Flex 和 AIR 简介	1
1.1	Adobe Flash	1
1.2	ActionScript 3.0	2
1.2.1	ECMAScript	2
1.2.2	关键概念	3
1.3	Flex 框架	11
1.3.1	Flex 4.5.1	11
1.3.2	MXML	12
1.3.3	Spark 库组件	14
1.3.4	数据绑定	21
1.3.5	Flex 移动应用程序结构	22
1.3.6	移动开发的考虑事项	31
1.4	Adobe AIR	31
1.5	小结	32
1.5.1	练习	32
1.5.2	本章所学内容	33
第 2 章	入门	35
2.1	使用 Flash Builder 4.5.1	35
2.1.1	使用工作空间	36
2.1.2	使用 Flash Perspective	37
2.1.3	使用 Flash Debug Perspective	38
2.1.4	使用 Source 和 Design 视图	39
2.2	使用 Flash Builder 创建移动项目	41
2.3	定义运行配置	52
2.3.1	在桌面上运行移动应用程序	52
2.3.2	在设备上运行移动应用程序	55
2.4	小结	62
2.4.1	练习	63
2.4.2	本章所学内容	63
第 3 章	为 Android、BlackBerry 和 iOS 设备构建 AIR 应用程序	65
3.1	AIR 应用程序描述符文件	65
3.1.1	在 AIR 应用程序描述符文件中设置属性	66
3.1.2	手动编辑 Hello World App 的应用程序描述符文件	66
3.1.3	BlackBerry Tablet OS 配置	76
3.1.4	为 Google Android 打包	79
3.1.5	为 Apple iOS 打包	85
3.1.6	为 BlackBerry Tablet OS 打包	90
3.2	更新 AIR 应用程序	92
3.2.1	从应用程序描述符文件获取详细信息	93
3.2.2	使用版本号	93
3.3	小结	94
3.3.1	练习	94
3.3.2	本章所学内容	95
第 4 章	触摸、多点触摸和手势	97
4.1	多点触摸交互	98
4.1.1	确定触摸输入支持	98
4.1.2	创建多点触摸和手势应用程序示例	99
4.1.3	触摸事件处理	102
4.1.4	在交互对象上注册触摸事件	110
4.1.5	确定支持触摸点数量	112

4.2 手势交互	113	6.5 单步执行代码	185
4.2.1 确定设备支持哪些手势	113	6.6 小结	188
4.2.2 手势事件和事件处理	114	6.6.1 练习	189
4.2.3 在交互对象上注册手势 事件	115	6.6.2 本章所学内容	189
4.2.4 处理手势事件	120	第 7 章 使用文件系统	191
4.3 在 Device Central 中利用多 点触摸面板	123	7.1 从文件系统读取	191
4.4 小结	123	7.1.1 File 和 FileStream 类	192
4.4.1 练习	124	7.1.2 创建 Files Explorer App 示例	195
4.4.2 本章所学内容	124	7.2 修改文件和文件系统	207
第 5 章 为多种屏幕尺寸进行开发	125	7.3 利用浏览对话框	220
5.1 多种屏幕尺寸的考虑	125	7.3.1 打开单个文件	220
5.1.1 像素密度	126	7.3.2 打开多个文件	221
5.1.2 利用设备 DPI	127	7.3.3 将单个文件保存到某个 位置	227
5.2 使内容适应舞台尺寸	131	7.4 小结	227
5.2.1 使用 StageScaleMode 和 StageAlign 类	131	7.4.1 练习	228
5.2.2 处理舞台尺寸变化事件	132	7.4.2 本章所学内容	228
5.2.3 创建 Sprite Layout App 示例	133	第 8 章 操作数据	229
5.3 处理设备朝向	144	8.1 检测网络可用性变化	229
5.3.1 使用 StageOrientation 类	145	8.1.1 使用 URLRequest 获取 数据	229
5.3.2 使用 StageOrientation Event 类	145	8.1.2 监视 URLRequest 对象	230
5.4 在 Flex 中使用布局	148	8.1.3 创建 Maintaining Data App 示例	231
5.5 小结	167	8.2 使用 SQLite 存储数据	238
5.5.1 练习	167	8.3 小结	276
5.5.2 本章所学内容	168	8.3.1 练习	276
第 6 章 调试应用程序	169	8.3.2 本章所学内容	277
6.1 设置断点	169	第 9 章 使用音频和视频	279
6.2 全局错误处理	180	9.1 Open Source Media Framework 介绍	279
6.3 处理未捕获错误	180	9.1.1 创建 URLResource 对象	280
6.4 Try...Catch 语句	183	9.1.2 创建 MediaElement 对象	280

9.1.3	媒体特质	282	10.1.2	创建 Camera App 示例	306
9.1.4	使用 MediaTraitType 类来 标识特质	283	10.2	使用设备的麦克风捕获声音	313
9.1.5	使用 MediaPlayer 来播放 媒体元素	284	10.2.1	使用 Microphone 类	313
9.1.6	使用 MediaPlayerSprite 类 播放媒体资源	286	10.2.2	使用 SampleDataEvent 类	314
9.1.7	处理特质事件	287	10.2.3	创建 Microphone App 示例	316
9.2	使用 Flex OSMF 封装器	289	10.3	利用设备的 Web 控制器	326
9.2.1	使用 VideoPlayer 组件	289	10.3.1	使用 StageWebView 类	326
9.2.2	创建 MediaPlayer 示例	290	10.3.2	创建 Browser App 示例	328
9.3	小结	302	10.4	利用设备的地理位置传感器	336
9.3.1	练习	302	10.4.1	使用 Geolocation 类	336
9.3.2	本章所学内容	303	10.4.2	使用 GeolocationEvent 类	337
第 10 章	利用设备功能	305	10.4.3	创建 Geolocation App 示例	337
10.1	使用设备的摄像头	305	10.5	小结	345
10.1.1	使用 CameraUI 类	306	10.5.1	练习	345
			10.5.2	本章所学内容	346

第 1 章

Flash、Flex 和 AIR 简介

本章主要内容

- Adobe Flash 平台概述
- ActionScript 3.0 语言的关键概念
- Flex 框架和 MXML 组件介绍
- Adobe AIR 简介

本章将会介绍本书的核心元素：Flash、Flex 和 AIR。

首先介绍 Flash 的核心内容以及本书所使用的编程语言 ActionScript 3.0。

然后介绍 Flex 的关键功能和 MXML 组件，通过一些代码片段来介绍一些示例。

最后介绍 AIR(Adobe Integrated Runtime)的功能。

1.1 Adobe Flash

Adobe Flash 平台由几个基于 Adobe Flash 的运行时客户端组成：Flash Player、Flash Lite 和 Adobe AIR。每个运行时都有自己的函数集和仅适用于该运行时的 API。

Flash 平台还包括一个框架组件：Flex。所有这些组件、运行时客户端和组件框架，都支持并利用 SWF 格式。

Flash 主要用于富内容和富应用程序的 Web 部署。它作为一个 Web 浏览器插件安装，并且能够以独立模式运行其内容。Adobe Flash 的标志如图 1-1 所示。



图 1-1 Adobe Flash 的标志

移动设备上的 Flash

当前，移动设备有两种专门支持 Flash 的方式。它们分别使用 Flash Lite 和 Flash Player。

1. Flash Lite 4.0

Flash Lite 用于运行那些希望在性能有限的移动设备上运行的 Flash 内容和应用程序。Flash Lite 提供了与 Flash Player 不同的功能集。直到最近, Flash Lite 一直只支持 ActionScript 2.0(AS2)。当前 Flash Lite 的版本是 4.0, 并且该运行时现在支持 ActionScript 3.0(AS3)。

若要了解关于 Flash Lite 的更详细信息, 以及学习如何使用该技术创建移动应用程序, 请参阅 Jermaine Anderson 撰著的 *Professional Flash Lite Mobile Application Development*(Wrox, 2010)一书。

2. Flash Player 10.x

Flash Player 10.1 是第一个定位于同时支持在传统 Web 浏览器和移动设备上进行内容开发和 SWF 格式部署的 Flash Player 版本。

在本书编写时, Adobe Flash Player 11 已经发布了 Beta 测试版本, 它支持开发人员预览该运行时下一版本的新功能和增强功能。由于这些新的 API 有可能会被去除, 所以在本书中没有介绍其中的任何新功能, 而是预期它会发布一个新版本。更详细信息请访问 Adobe Labs 网站(labs.adobe.com/technologies/), 并搜索 Adobe Flash Player 11。

本书的重点是介绍如何使用 AS3 开发用于在其最新版本 Adobe Flash Player 10.3 上运行的 Flash 内容。

Flash 在 Google Android 和 BlackBerry Tablet OS 移动平台上得到完全支持。除非您是最近才从火星回来, 否则应该知道在苹果公司的 iOS 平台上不支持 Flash。但是, 使用 AS3 和 AIR, 可以通过独立应用程序的方式, 让应用程序在该平台上运行。

1.2 ActionScript 3.0

AS3 是一种面向对象的语言, 用于创建多媒体内容, 以便在 Flash 运行时客户端的 Flash Player、Flash Lite 和 Adobe AIR 上运行。

1.2.1 ECMAScript

AS3 语言的核心基于 ECMAScript 第 4 版语言规范, 可以在 ECMA 国际网站 www.ecmascript.org/上查看该规范。可以在网站 <http://livedocs.adobe.com/specs/actionscript/3> 查看 AS3 规范。

AS3 具有与 Java 非常类似的句法; 它也非常类似于广泛使用的 JavaScript, 后者是 Web 浏览器中最流行的一种语言。如果读者已经熟悉这些编程语言中的某一种, 则也会非常熟悉 AS3。

ECMAScript 4 定义了许多如何用 AS3 编写代码的规则, 包括语法和句法。这些将会使得代码按照一种特定方式编写, 意味着开发人员可以很容易阅读其他人编写的代码。

许多核心编程概念, 例如变量、函数、类、对象、表达式和语句, 都继承于 AS3, 它

们构成了 ECMAScript4 的主要部分。

ECMAScript4 还为 AS3 定义了几个内置的数据类型，以允许开发人员更好地利用常用数据类型，例如数组、布尔值、数值和字符串。

1.2.2 关键概念

使用 AS3 编程时需要理解许多重要的关键概念；这些概念将有助于本书后面部分的理解。下面将介绍这些概念。

1. 类、对象和实例

类(class)用于为对象提供属性和功能，使得对象可以通过其方法和函数来完成特定任务。类实质是对象的蓝图。

AS3 类是用.as 文件后缀名标识的基于文本的文件。类使用关键字 `class` 定义，后面跟随类名称，该名称需要使用大写字母开头。这个类名称必须与用于创建该类的文件名相同。下面的代码片段显示了一个用于定义名称为 `Mobile` 的类的示例。

```
class Mobile {}
```

对象(object)代表用于描述一个特定事物的程序部分。该事物可以是一个具有三条边并使用蓝色填充的形状；也可以是一个具有 PG 评级的电影；或者是一个允许存储联系人的移动设备。对象可以是你能想到的任何事物。

对象具有属性，用于为它们提供特征；对象还具有方法，允许它们执行特定任务。

在 AS3 中，对象由类创建，方法是：实例化类，在类名称和括号()之前调用 `new` 关键字。下面的代码片段显示了一个实例化的 `Mobile` 类示例。

```
new Mobile();
```

2. 包

包(package)定义了类的路径，可以相对其他可能具有相同名称的类来唯一标识类。包还反映了文件夹结构。

最后，包用于避免类之间的冲突。

为包提供有意义的名称是一种良好的编程习惯。将类似的类组合到一个包中则是一种常见的编程方式，这样做可以更加容易搜索并利用不同的类。

`Mobile` 类可以被放入到一个名为 `devices` 的包内，与另一个有可能要创建的名为 `Tablet` 的类并列放置在一起。

```
package devices
{
    class Mobile {}
}
```

虽然这个示例向我们展示的是一个完全限定的包，但是包的命名规则通常会更加细化，包名称定义通常使用句点分隔的反向 DNS 风格的字符串。例如，`devices` 包声明可以很容易地引用为 `wrox.com` 域的一部分，并使用第 1 章名称 `com.wrox.ch1.devices`，如下面的代码片段所示：

```
package com.wrox.ch1.devices

{
    class Mobile {}
}
```

包还可以用于将类导入到 AS 文档中，这时需要使用 `import` 关键字标识，如下面的代码片段所示，其中再次使用了 `com.wrox.ch1.devices` 示例：

```
import com.wrox.ch1.devices.Mobile;
```

类在使用之前，需要通过它的完整的包引用来引用。



说明：在 AS3 中需要指定包声明，以确认类声明位于包的花括号内部。包名称可以省略为空，但是应该避免这样做。在 AS2 中创建类时不需要包声明。

3. 函数、方法和作用域

函数用于让对象执行一项特定任务，并执行一个可重用的指令集。函数使用 `function` 关键字后跟任务名称来定义。

类作用域修饰符

可以使用 4 个关键字为对象的属性和函数设置作用域：`public`、`private`、`protected` 和 `internal`。

`public` 作用域修饰符表示一个变量属性或函数可以从任何位置访问。`protected` 作用域修饰符表示只有当前类和子类可以使用该变量或函数。`private` 作用域修饰符将访问限制在类内部，而 `internal` 作用域修饰符则将作用域限制在它所在的包内部。

类构造函数

类的构造函数(constructor)用于初始化一个对象并创建它。构造函数实质上是类的一个方法，并且它必须声明为 `public` 作用域。下面的代码段显示了 `Mobile` 类的构造函数，它仅仅是一个简单的 `Mobile()`。

```
package devices
{
    public class Mobile
    {
        public function Mobile()
        {

```

```

        // Code to initialize Mobile
    }
}

```

4. 返回类型

根据函数中所包含的指令集，返回类型提供了函数调用所生成值的类型指示。在下面的代码段中，定义了一个名称为 `launchApp()` 的公共函数，其返回类型指定为 `void`。

```

package devices
{
    public class Mobile
    {
        public function Mobile()
        {
            // Code to initialize Mobile
        }

        public function launchApp():void
        {
            // Code to launch an app on the mobile.
        }
    }
}

```

这个 `void` 关键字表示函数将不会返回任何值。另外还要注意类构造函数没有指定返回类型。

5. 变量

变量(variable)是一个与对象相关联的唯一标识符，用于保存对一个值的引用。在 AS3 中，可以为变量指定数据类型，以便在编译时执行类型检查。变量使用 `var` 关键字后面跟随变量名称来定义。

在类中，在函数外部增加的变量可以在类作用域内引用；它们也称为实例变量(instance variable)。它们用于描述对象的特征，用于为对象提供属性。

下面的代码片段中声明了两个私有变量：`contacts` 为 `Array` 数据类型，`phoneNumber` 为 `Number` 数据类型。

```

package devices
{
    public class Mobile
    {
        private var contacts:Array;
        private var phoneNumber:Number;

        public function Mobile()
        {

```

```

        // Code to initialize Mobile
    }

    public function launchApp():void
    {
        // Code to launch an app on the mobile.
    }
}
}

```

下面的代码演示了如何创建一个 **Mobile** 对象并将它赋值给变量 **mobileObj**。

```
var mobileObj:Mobile = new Mobile();
```

注意这里赋值给对象的数据类型是 **Mobile**。任何时候按这种方式实例化对象时，都需要定义变量的数据类型，否则将会得到“变量没有类型声明(variable has no type declaration)”的警告。

在方法或函数内部，实例变量可以用于设置或引用数据值。下面的代码片段使用 **this** 关键字，在 **Mobile** 的构造函数方法内部对 **phoneNumber** 变量赋值。

```

package devices
{
    public class Mobile
    {
        private var contacts:Array;
        private var phoneNumber:Number;

        public function Mobile()
        {
            this.phoneNumber = 011003637;
        }

        public function launchApp():void
        {
            // Code to launch an app on the mobile.
        }
    }
}

```

6. 静态变量和方法

静态引用是指可以直接引用而不需要实例化类并创建对象。

在下面的代码片段中，为 **Mobile** 类提供了一个静态变量 **deviceType**，其类型定义为字符串，并赋值为 **Smartphone**。其中还定义了一个名称为 **switchOn()** 的静态函数。

```

package devices
{
    public class Mobile
    {

```

```

public static var deviceType:String = "Smartphone";

private var contacts:Array;
private var phoneNumber:Number;

public function Mobile()
{
    this.phoneNumber = 011003637;
}

public function launchApp():void
{
    // Code to launch an app on the mobile.
}

public static function switchOn():void
{
    // Code to switch on the device.
}
}

```

下面的代码演示了如何调用 `switchOn()` 函数：

```
Mobile.switchOn();
```

7. 形参和实参

形参(parameter)是指在函数声明的括号中定义并指定其数据类型的局部变量。

实例方法允许为对象定义呈现其功能和可执行哪些操作的函数。还可以在实例方法中定义形参以便给对象传递值。

下面的代码片段显示了 `Mobile` 类定义了一个新实例方法，名称为 `addContact()`。该方法具有公共作用域，并且包含两个形参：`cName` 为 `String` 数据类型，代表联系人的姓名；`cNumber` 为 `Number` 数据类型，代表联系人的手机号码。

```

package devices
{
    public class Mobile
    {
        public static var deviceType:String = "Smartphone";

        private var contacts:Array;
        private var phoneNumber:Number;

        public function Mobile()
        {
            this.phoneNumber = 011003637;
        }
    }
}

```



```

        public function addContact(cName:String, cNumber:Number):void
        {
            // Code to add a new contact
        }

        public function launchApp():void
        {
            // Code to launch an app on the mobile.
        }

        public static function switchOn():void
        {
            // Code to switch on the device.
        }
    }
}

```

注意局部变量只是临时性的。

为了调用 `addContact()` 方法，需要使用 `Mobile` 类的一个实例，并为该方法提供实参 (argument)。对于这个方法需要两个实参，如下面的代码片段所示。

```

var mobileObj:Mobile = new Mobile();
mobileObj.addContact("Olivia", 736300110);

```

这里 `Olivia` 和 `736300110` 都是实参。每个实参都必须与为方法指定的形参的数据类型相匹配。

8. 条件和循环语句

条件语句是在满足特定条件时才会执行的语句。

在下面的代码片段中，为 `Mobile` 类添加了另一个名称为 `switchOff()` 的静态函数。其中可以看到一个条件 `if` 语句检查变量 `isOn`，在类中，这个变量是一个初始值设置为 `false` 的布尔值。在 `switchOn()` 中，有一个 `if` 语句用于检查 `isOn` 的值是否为 `false`；这通过布尔值前面的惊叹号 (即 `!isOn`) 来指定。函数 `switchOff()` 通过查询 `isOn` 变量是否等于 `false`，然后通过将 `isOn` 设置为 `false` 以执行 `if` 语句的 `else` 程序块中的代码，演示了另一种编写代码实现相同功能的方法。

```

package devices
{
    public class Mobile
    {
        public static var deviceType:String = "Smartphone";

        private var contacts:Array;
        private var phoneNumber:Number;
        private var isOn:Boolean = false;
    }
}

```