Android 应用采用 Java 编程语言编写。Android SDK 工具将您的代码 — 连同任何数据和资源文件 — 编译到一个 APK：*Android 软件包*，即带有 .apk 后缀的存档文件中。一个 APK 文件包含 Android 应用的所有内容，它是基于 Android 系统的设备用来安装应用的文件。

安装到设备后，每个 Android 应用都运行在自己的安全沙箱内：

* Android 操作系统是一种多用户 Linux 系统，其中的每个应用都是一个不同的用户；
* 默认情况下，系统会为每个应用分配一个唯一的 Linux 用户 ID（该 ID 仅由系统使用，应用并不知晓）。系统为应用中的所有文件设置权限，使得只有分配给该应用的用户 ID 才能访问这些文件；
* 每个进程都具有自己的虚拟机 (VM)，因此应用代码是在与其他应用隔离的环境中运行；
* 默认情况下，每个应用都在其自己的 Linux 进程内运行。Android 会在需要执行任何应用组件时启动该进程，然后在不再需要该进程或系统必须为其他应用恢复内存时关闭该进程。

Android 系统可以通过这种方式实现*最小权限原则*。也就是说，默认情况下，每个应用都只能访问执行其工作所需的组件，而不能访问其他组件。 这样便营造出一个非常安全的环境，在这个环境中，应用无法访问系统中其未获得权限的部分。

不过，应用仍然可以通过一些途径与其他应用共享数据以及访问系统服务：

* 可以安排两个应用共享同一 Linux 用户 ID，在这种情况下，它们能够相互访问彼此的文件。 为了节省系统资源，可以安排具有相同用户 ID 的应用在同一 Linux 进程中运行，并共享同一 VM（应用还必须使用相同的证书签署）。
* 应用可以请求访问设备数据（如用户的联系人、短信、可装载存储装置 [SD 卡]、相机、蓝牙等）的权限。 用户必须明确授予这些权限。 如需了解详细信息，请参阅 [使用系统权限](https://developer.android.com/training/permissions/index.html)。

以上内容阐述了有关 Android 应用在系统内存在方式的基础知识。本文的其余部分将向您介绍以下内容：

* 用于定义应用的核心框架组件
* 您用来声明组件和应用必需设备功能的清单文件
* 与应用代码分离并允许您的应用针对各种设备配置适当优化其行为的资源