**.NET程序集入门-初识程序集**

**.NET程序集包括.exe和.dll程序集，程序集是类的代码库，它们可以相互被引用。一个应用程序可以由多个程序集拼装而成（一个.exe + 多个.dll）。**

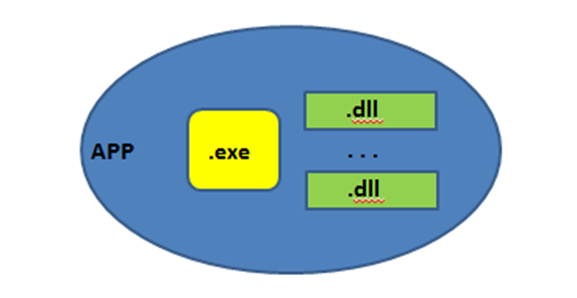
****

图1.1 应用程序组成

* **.exe有什么作用？**.exe，全名execute，是执行应用程序的二进制文件，它包含全

部或者部分程序逻辑。

* **. dll有什么作用？**

.dll，全名dynamic linked library，动态链接库，习惯上称应用程序扩展，它包含部分程序逻辑。

* **为什么应用程序要分成两种程序集？**

将应用程序分为.exe和.dll程序集的好处在于，.dll程序集可以配置成为共享程序集，使得多个应用程序共同使用一个.dll的内部逻辑，优势在于：

* 共享程序集可以瘦身应用程序，减小程序体积。
* 使得程序的应用扩展变得更为灵活。
* **.NET程序集的优点**
* 代码重用：多个应用程序可以使用同一个共享程序集。
* 语言无关：C#程序集可以继承VB程序集中的类。
* 类型边界明确：在不同的程序集（如a.dll和b.dll）中在同一个命名空间中定义相同名称的类，.NET将它们视为不同的类。
* 版本化：每个程序集在创建时被分配一个格式为<major>.<minor>.<build>.<version>的版本号，默认为1.0.0.0。版本号+ 公钥值 = 强命名的程序集可使得程序集的不同版本在同一台机器上共存不冲突。通过使用强命名，CRL可以确保客户端调用程序加载正确版本的程序集。
* 可配置：程序集可以被配置为“私有”和“共享”。使用XML配置文件，指示CRL在指定的路径查找正确的程序集。规范的“私有程序集”存放在与客户端同一目录上或者其子目录中，而“共享程序集”存放在全局程序集缓存GAC的特定目录。
* **.NET程序集的包含什么数据？**
* Windows文件首部：标识应用程序基于控制台、图形用户界面还是.dll代码库驻留在Windows操作系统上，应用于开发编译器或者内存加载二进制图像时。
* CRL文件首部：标识多个数据块，如元数据和资源的位置、构建程序集的运行库版本、（可选）公钥值，它被CRL作为映像数据加载到内存。
* CIL代码：程序集的核心之一，独立于平台和CPU的中间语言。程序运行时，实时JIT编译器将其编译成特定平台和CPU的指令。它是.NET程序集语言无关的关键所在。
* 元数据：程序集的核心之一，完整描述程序集包含的类型、类型成员和外部类型的格式。CRL利用元数据在内存中解析类型的位置，它是对象序列化、延迟绑定和可扩展应用程序的脊柱。
* 程序集清单：程序集的核心之一，包括描述当前程序集依赖的所有外部程序集、当前程序集自定义的特性（版本化、公司信息等，可通过AssemblyInfo.cs文件修改）。
* 可选的程序集嵌入资源：如程序图标、图像文件、声音片段和字符串表。同时，.NET支持本地化资源的卫星程序集，应用于国际化的软件。