

第1个简单java程序 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E7_AC_AC

1_E4_B8_AA_E7_AE_80_c104_253532.htm 既然Java 面向对象的基础已经被讨论过了，接下来让我们看一些实际的Java 程序。让我们从编译及运行下面这个简短示例程序开始。你将看到，这个程序的功能比你想像的要多。 /* This is a simple Java program. Call this file "Example.java". */ class Example { // Your program begins with a call to main(). public static void main(String args[]) { System.out.println("This is a simple Java program.").} } 注意：在下面的介绍中，将使用标准JDK（Java Developer ' s Kit，Java 开发工具包），它可从Sun Microsystems 公司得到。如果你正在使用其他的Java 开发环境，则Java 程序编译、运行过程可能有所不同。在这种情况下，请你查阅编译器的用户手册来获得具体指导。1 键入程序 对大多数计算机语言，包含程序源代码的文件名是任意的，但对于Java 就不行。关于Java，你需要知道的第一件事就是源文件的名字非常重要。对这个例子，源程序文件名应该是Example.java。下面我们将解释其中的原因。在Java 中，一个源程序文件被称为一个编译单元（compilation unit）。它是一个包含一个或多个类定义的文本文件。Java 编译器要求源程序文件使用.java 文件扩展名。请注意，文件扩展名长度是4个字符。所以，你的操作系统一定要有支持长文件名的能力。这意味着DOS 和Windows 3.1 是不支持Java 的（至少在目前是这样）。当然，它可在Windows 95/98 和Windows NT/2000 下正常工作。从上述示例程序中可以看出，程序中定义类名也是Example。这不是

巧合。在Java中，所有的代码都必须驻留在类中。按照约定，类名必须与源程序的文件名相同。你也应该确保文件名的大小写字母与类名一样，这是因为Java是区分大小写的。虽然文件名与类名必须一致的约定似乎有点专制，但是这个约定有助于你轻松地维护及组织程序。

2 编译程序

要编译示例程序Example，须运行编译器程序javac，并在命令行上指定源程序文件名，格式如下所示：`C:\>javac Example.java` 编译器javac产生了一个名为Example.class的文件，该文件包含程序的字节码。前面已讨论过，Java字节码中包含的是Java解释程序将要执行的指令码。因此，javac的输出并不是可以直接运行的代码。要真正运行该程序，你必须使用名叫java的Java解释器。具体做法是把类名Example作为一个命令行参数输入，格式如下所示：`C:\>java Example` 运行这个程序，将输出如下内容：`This is a simple Java program.` 当Java源代码被编译后，每个单独的类都被放入自己的输出文件中，并以类的名字加“.class”扩展名为其文件名。这就是为什么Java源程序文件必须与其中包含的类同名的原因源程序文件将与“.class”文件相同。运行Java解释器实际上是指定你想要解释器运行的类的名字，它会自动搜索包含该名字且带有.class扩展名的文件。如果找到，它将运行包含在该指定类中的代码。

2.2.3 详细讨论第1个示例程序

尽管Example.java很短，但它包括了所有Java程序具有的几个关键特性。让我们仔细分析该程序的每个部分。程序开始于以下几行：`/* This is a simple Java program. Call this file "Example.java". */` 这是一段注释（comment）。像大多数其他的编程语言一样，Java也允许你在源程序文件中加注释。注释中的内容将被编译器忽略。事实上，注

释是为了给任何阅读源代码程序的人说明或解释程序的操作。在本例中，注释对程序进行说明，并提醒你该源程序的名字叫做Example.java。当然，在真正的应用中，注释通常用来解释程序的某些部分如何工作或某部分的特殊功能。Java支持3种类型的注释。在示例程序顶部的注释称为多行注释（multiline comment）。这类注释开始于“/*”，结束于“*/”。这两个注释符间的任何内容都将被编译器忽略。正如“多行注释”名字所示，一个多行注释可以包含若干行文字。程序的下一行代码如下所示：`class Example {` 该行使用关键字class声明了一个新类，Example是类名标识符，整个类定义（包括其所有成员）都将位于一对花括号（{}）之间，花括号在Java中的使用方式与C或C++相同，目前，不必考虑类的细节，只是有一点要注意，在Java中，所有程序活动都发生在类内，这就是为什么Java程序是面向对象的。下面一程序是单行注释：`// Your program begins with a call to main().` 这是Java支持的第二种类型的注释。单行注释（single-line comment）始于“//”，在该行的末尾结束。通常情况下，程序员们对于较长的注释使用多行注释，而对于简短的、一行一行的注释则使用单行注释。下一行代码如下所示：`public static void main(String args[]) {` 该行开始于main()方法。正如它前面的注释所说，这是程序将要开始执行的第一行。所有的Java应用程序都通过调用main()开始执行（这一点同C/C++一样），我们在此还不能对该行的每一个部分作出精确的解释，因为这需要详细了解Java封装性的特点，但是，由于本书第1部分中的大多数例子都用到这一行代码，我们将对各部分作一个简单介绍。关键字public是一个访问说明符（access

specifier) , 它允许程序员控制类成员的可见性。如果一个类成员前面有public , 则说明该成员能够被声明它的类之外的代码访问 (与public相对的是private , 它禁止成员被所属类之外的代码访问) 。在本例中 , main() 必须被定义为public 类型 , 因为当程序开始执行时它需要被它的类之外的代码调用。关键字static 允许调用main() 而不必先实现该类的一个特殊实例。这是必要的 , 因为在任何对象被创建之前 , Java 解释器都会调用main() 。关键字void 仅通知编译器main() 不返回任何值。你将看到 , 方法也可以有返回值。如果这一切似乎有一点令人费解 , 别担心。所有这些概念都将在随后的章节中详细讨论。前面已经介绍过 , main() 是Java 程序开始时调用的方法。请记住 , Java 是区分大小写的。因此 , main 与Main是不同的。Java 编译器也可以编译不包含main() 方法的类 , 但是Java 解释程序没有办法运行这些类。因此 , 如果你输入了Main 而不是main , 编译器仍将编译你的程序 , 但Java 解释程序将报告一个错误 , 因为它找不到main() 方法。你要传递给方法的所有信息由方法名后面括号中指定的变量接收 , 这些变量被称为参数 (parameters) 。即使一个方法不需要参数 , 你仍然需要在方法名后面放置一对空括号。在main() 中 , 只有一个参数 , 即String args[] , 它声明了一个叫做args的参数 , 该参数是String 类的一个实例数组 (注 : 数组是简单对象的集合) 。字符串类型的对象存储字符的串。在本例中 , args接收程序运行时显示的任何命令行参数。本例中的这个程序并没有使用这些信息 , 但是本书后面讲到的其他一些程序将使用它们。该行的最后一个字符是“ { ”。它表示了main() 程序体的开始。一个方法中包含的所有代码都将

包括在这对花括号中间。另外，`main()` 仅是解释器开始工作的地方。一个复杂的程序可能包含几十个类，但这些类仅需要一个`main()` 方法以供解释器开始工作。当你开始引用被嵌入在浏览器中的Java 小应用程序时，你根本不用使用`main()` 方法，因为Web浏览器使用另一种不同的方法启动小应用程序。接下来的代码行如下所示。请注意，它出现在`main()` 内。

```
System.out.println("This is a simple Java program.");
```

本行在屏幕上输出字符串“ This is a simple Java program. ”，输出结果后面带一个空行。输出实际上是由内置方法`println()` 来实现的。在本例中，`println()` 显示传递给它的字符串。你将会看到，`println()` 方法也能用来显示其他类型的信息。该行代码开始于`System.out`，现在对它作详细说明为时尚早，需涉及很多复杂内容。简单的说，`System` 是一个预定义的可访问系统的类，`out`是连接到控制台的输出流。可能你已经猜到了，控制台输出（输入）在实际的Java 程序和小应用程序中并不经常使用。因为绝大多数现代计算环境从本质上讲都是窗口和图形界面的，控制台I/O 主要被用简单的实用工具程序和演示程序使用。在本书后面，你将会学到用Java 生成输出的其他方法。但是目前，我们将继续使用控制台I/O 方法。请注意，`println()` 语句以一个分号结束。在Java 中，所有的语句都以一个分号结束。该程序的其他行没有以分号结束，这是因为从技术上讲，它们并不是程序语句。程序中的第一个“`}`”号结束了`main()`，而最后一个“`}`”号结束类`Example` 的定义。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com