第 1 章　建立 Java 程序开发环境

Java 是一门重要的程序设计语言。使用 Java 语言既可以开发桌面应用程序，也可以开发基于网络的应用程序。一些知名的电子商务系统、电子政务系统、自动化办公系统等，很多是使用 Java 语言开发的。本章首先简单介绍 Java 语言的特点，然后介绍如何安装 Java 语言程序开发环境，最后使用 Java 语言开发一个简单的程序。

1.1　Java 语言概述

计算机是 20 世纪 40 年代人类最伟大的发明创造之一。计算机从其诞生之日起，为人类的文明进步和发展做出了巨大贡献。近年来，以高性能计算机为基础的大数据和人工智能的发展和应用，使计算机相关技术再次成为技术发展的重要方向。但是，要使计算机按照人们的意愿工作，人们必须使用计算机能够理解的称为“计算机程序设计语言”的工具来完成。

1.1.1　程序设计语言

计算机程序设计语言，简称程序设计语言，是人们对计算机“发号施令”进而控制计算机工作的重要工具。程序设计语言包括高级语言和低级语言：C/C++、Java、Python、C# 等，都是常用的高级语言；机器语言、汇编语言，则是低级语言。这里的“高级语言”“低级语言”是针对程序设计语言的语义丰富方面而言的，并不具有日常概念上的层次高低的含义。简单来说，高级语言语义丰富，容易学习和使用，是开发应用程序的主要语言；而低级语言则语义单一，不易学习也不易使用，低级语言是计算机出现初期被使用的主要语言。

1.1.2　Java 语言的特点

Java 语言是高级程序设计语言，是当今非常流行和广为使用的程序设计语言之一。Java 语言功能丰富、容易学习也容易使用。概括起来，Java 语言具有以下特点。

（1）简单：相对于其他高级程序设计语言，Java 语言出现的时间比较晚，因此，Java 从语言的设计上吸纳了其他高级语言的优良性质，并克服了其他语言的缺点和不足，所以，Java 语言简单且易学易用。

（2）面向对象：Java 语言是一种面向对象的编程语言。“面向对象”是当代程序设计语言的一大进步和创举。通过面向对象编程技术，Java 语言将程序的各个部分看作一个个

的“对象”，通过对象之间的交互达成程序的功能、性能目标。

（3）解释执行或实时执行：Java 编译程序编译 Java 源代码并生成字节码，所生成的字节码既可以通过 Java 虚拟机直接解释执行，也可以通过 Java 的 JIT（just in time）技术将字节码转换成计算机机器码，从而提高执行效率。

（4）可移植：由于 Java 编译器将 Java 源代码编译生成 Java 字节码，并通过 Java 虚拟机解释执行，所以，Java 程序并不依赖具体执行平台，用 Java 编写的程序可以在任何操作系统上运行。

（5）支持多线程：Java 语言从设计上就支持多线程，并提供了优秀的多线程并发控制机制，因此，相比于其他高级语言，使用 Java 编写多线程程序更容易也更可靠。

1.2 建立 Java 开发环境

有句老话：工欲善其事必先利其器。要学习 Java 语言并使用 Java 语言开发计算机应用程序，首先需要建立 Java 开发环境。为了建立 Java 开发环境，需要执行以下两个基本步骤：①安装 JDK；②安装 IntelliJ IDEA 工具。本书使用在业界广泛使用的 IntelliJ IDEA 作为编程工具环境。

安装 JDK 和 IntelliJ IDEA

安装 JDK 和 IntelliJ IDEA 说明

1.3 第一个“Hello world!”程序

现在，已经安装了开发 Java 程序所必需的工具，可以开发 Java 程序了。在开发 Java 程序之前，需要在 IDEA 中创建一个 Java 程序工程。为什么需要创建“程序工程”呢？因为一个 Java 程序会包括多个文件，同时，还会包括一些以管理为目的的辅助性文件，IDEA 工具需要对这些文件需要进行统一集中管理。使用“程序工程”这项技术就可以达到这个管理目标。

1.3.1 创建 Java 程序工程

下面创建一个最简单的 Java 程序工程，运行这个程序后，将在显示器上显示“Hello world!”，因此，这个程序称为“Hello world!”程序。启动 IDEA，单击界面上的“+”按钮，创建一个 Java 程序工程，如图 1-1 所示。

创建 Java 程序工程

在如图 1-1 所示的界面中，在 Name 输入框中输入工程的名称；在 Location 选择框中选择存放工程的文件夹；在 Language 选项中选择 Java；在 Build system 选项中选择 Maven；在 JDK 选项中选择所安装的 JDK 工具，由于之前安装的 JDK 工具是 JDK 17，所以这里选择 17 Oracle OpenJDK version 17.0.5；选中 Add sample code 单选框；单击 Create 按钮创建这个名称为 ch01-01 的工程。成功创建

Java 程序工程后，将显示如图 1-2 所示的界面。

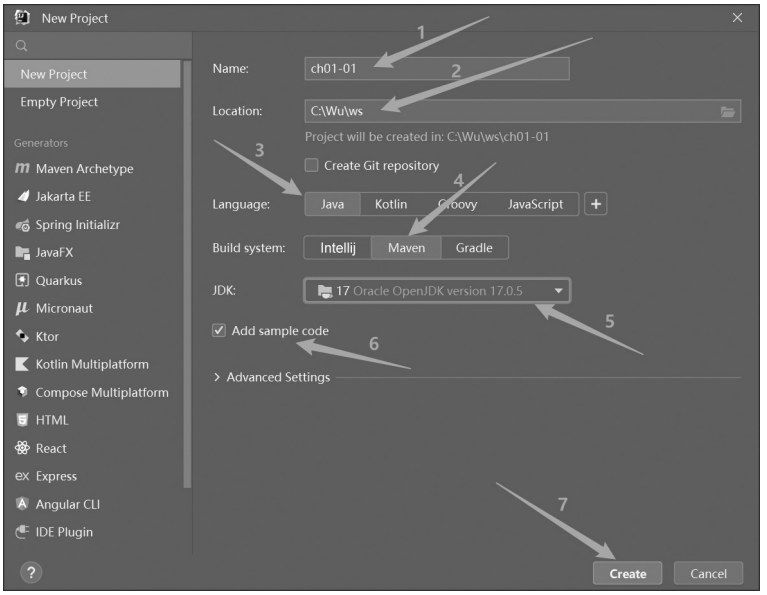

图 1-1　创建程序工程

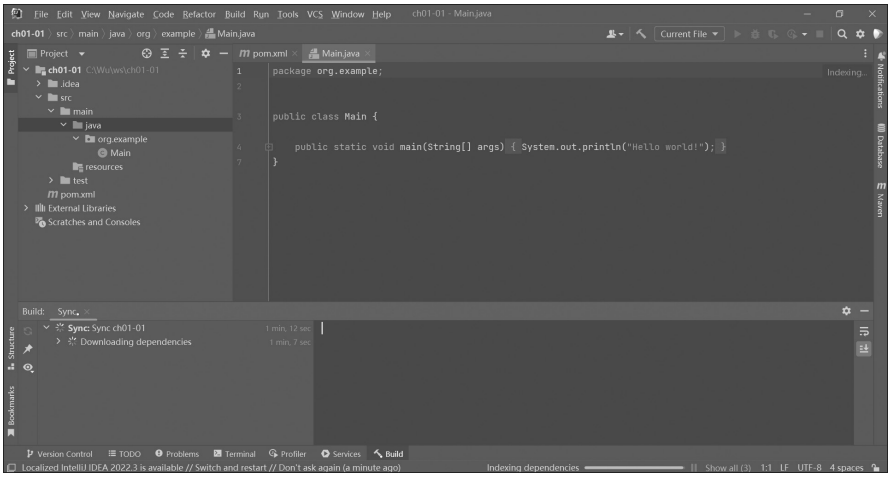

图 1-2　成功创建 Java 程序工程后的界面

由于这是第一次创建 Java 程序工程，IDEA 需要下载必要的软件支持插件，这需要一定的时间，请耐心等待这个过程完成。当图 1-2 中最下方的进度条消失后，表示下载完成，将显示如图 1-3 所示的界面。

在如图 1-2 和图 1-3 所示的界面中，界面的背景颜色太黑，同时，文字字号太小，可以根据需要修改界面的主题和文字大小。为此，在如图 1-3 所示的界面中，选择 File → Settings 命令，如图 1-4 所示。

在如图 1-5 所示的界面中，在 Theme 选择框中选择 IntelliJ Light，通过修改 Size 值设置文字大小。例如，在这里设置为 14。

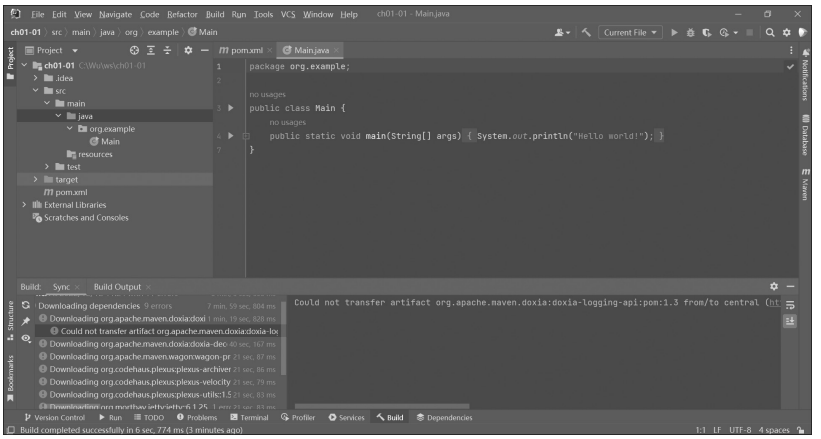

图 1-3　IDEA 完成支持工具下载

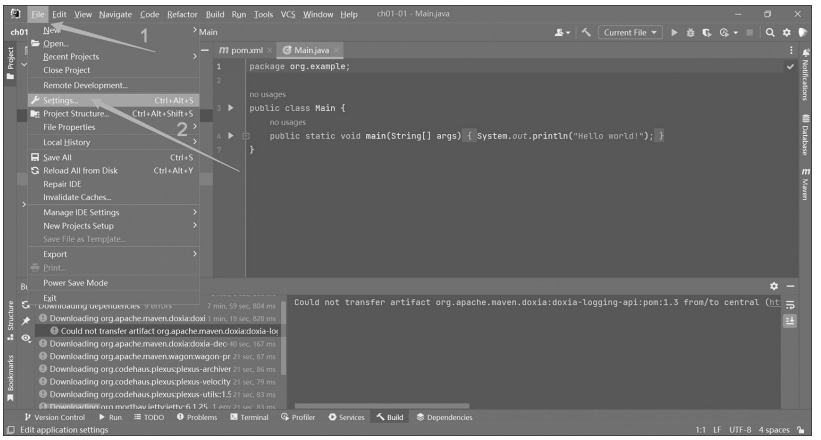

图 1-4　修改 IntelliJ IDEA 界面主题和文字大小

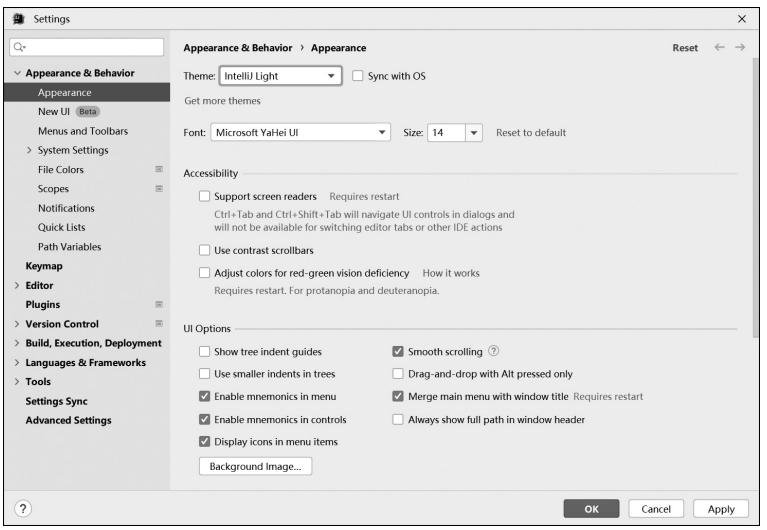

图 1-5　IDEA 界面主题

在如图 1-5 所示的界面中，选择左边的 Editor → Font 命令，如图 1-6 所示。

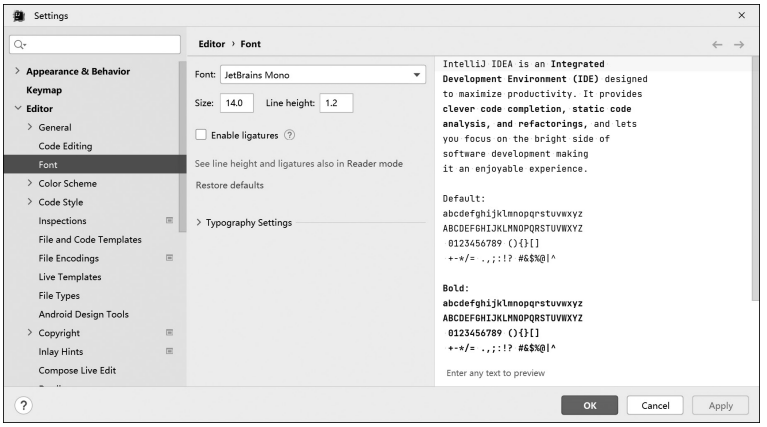

图 1-6　代码编辑文字大小设置

在图 1-6 中，通过修改 Size 值设置文字大小。例如，在这里设置为 14。完成设置后，单击 OK 按钮，返回主界面，如图 1-7 所示。

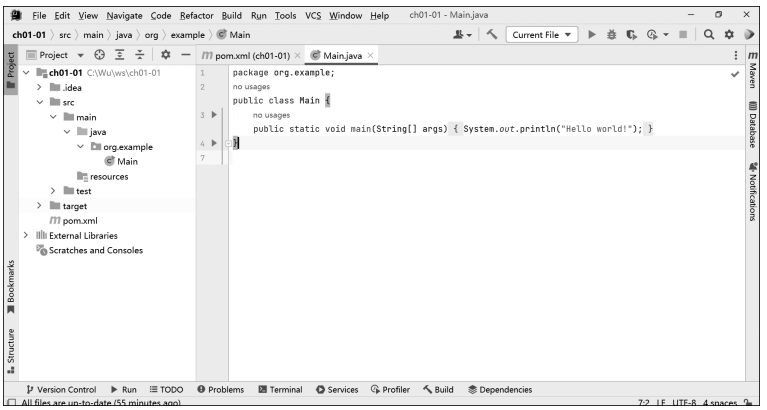

图 1-7　IDEA 主界面

如图 1-7 所示是程序开发人员使用 IDEA 开发应用程序时的工作界面：在这个界面中，既可以编写 Java 程序代码，也可以运行 Java 程序，还可以调试 Java 程序等。下面先运行这个刚刚创建的 Java 程序看看效果。

1.3.2　运行 Java 程序

由于在如图 1-1 所示的界面中选中了 Add sample code 单选框，所以 IDEA 为这个程序工程创建了一个必需的称为“Java 主类”的代码（包含 main( ) 入口函数的类称为主类）。这个主类代码如图 1-8 所示。矩形框中的代码就是 Java 主类代码，保存在名称为 Main.java 的文件中。矩形框中的代码和保存在 Main.java 中的内容是一致的，它们都被称为“Java 源代码”。任何一个可运行的 Java 程序都必须至少有一个主类代码。

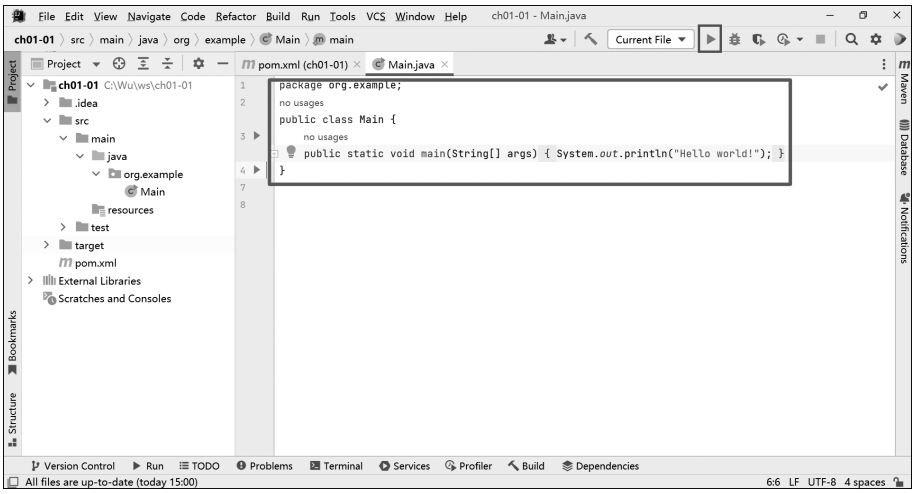

图 1-8　Java 程序主类代码

由于 ch01-01 程序工程已经有了 Java 主类代码，因此，这个程序是可以直接运行的。为此，在如图 1-8 所示的界面中单击▶按钮，IDEA 经过必要的处理后，将显示如图 1-9 所示的运行结果。

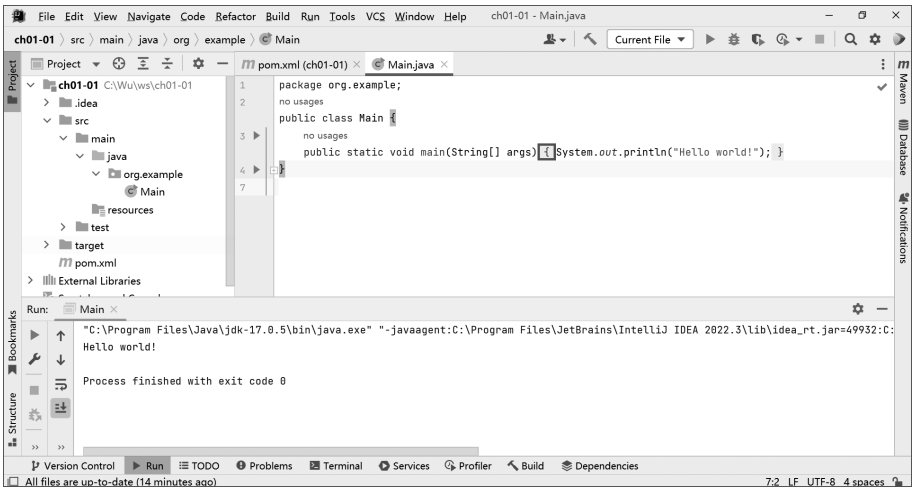

图 1-9　ch01-01 程序工程的运行结果

在如图 1-9 所示的界面中，“Hello world!”就是 ch01-01 程序工程运行后显示的一段文字信息。那么，这个程序为什么会显示这句话呢？这需要了解 Java 程序的运行过程。

1.4　Java 程序的运行过程

在 IDEA 中运行一个 Java 程序非常简单：只需要在如图 1-8 所示的界面中单击▶按钮即可启动 Java 程序运行。其实，在 IDEA 内部，为了运行一个 Java 程序，会经过以下两

个阶段：第一个阶段称为编译或者构建阶段，在这个阶段，IDEA 会使用 JDK 中的 Java 编译器将 Java 源代码编译（也称为“构建”，或 Build）成 Java 的字节码；第二个阶段称为执行阶段，IDEA 会使用 JDK 中的 Java 虚拟机执行在第一个阶段所生成的 Java 字节码，进而启动 Java 程序运行。

1.4.1　编译代码

如果源代码没有错误，编译成功的 Java 代码即可被 Java 虚拟机启动运行；如果 Java 源代码有错误，如将某个对象的名字输错了，那么，Java 编译器是无法正确编译 Java 源代码的。例如，在如图 1-10 所示的 Main.java 的源代码中，如果将其中的 out 对象的名字错误地输入成 outt，会发现该程序不能运行，将显示如图 1-10 所示的错误信息。

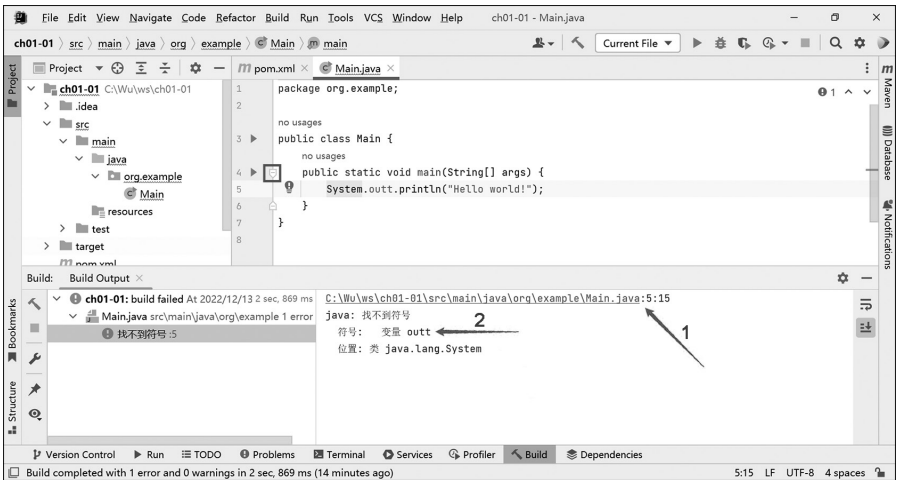

图 1-10　程序代码错误导致不能正确运行

在图 1-10 中，IDEA 将显示如箭头 1 所指示的源代码中的错误，并在箭头 2 所指示的地方给出错误的原因。

细心的读者会发现，如图 1-10 所示的代码与如图 1-9 所示的代码有细微不同：在如图 1-10 所示的代码中，main( ) 函数的两个大括号和程序语句在不同的行中；而在如图 1-9 所示的代码中，main( ) 函数的两个大括号和程序语句在同一行。本质上，如图 1-9 所示的代码和如图 1-10 所示的代码是一致的。在如图 1-10 所示的代码中，单击矩形框中的按钮后，代码的显示效果将与如图 1-9 所示的代码一致；或者，在如图 1-9 所示的代码中，单击矩形框中的大括号“{”后，代码的显示效果将与如图 1-10 所示的代码一致。

另外，如果希望关闭如图 1-10 所示的编译错误信息，可以单击 Build 选项卡关闭它，再次单击 Build 选项卡又可以将编译错误信息显示出来。

还有一种能够直观发现代码错误的方法：将光标放在代码中出现红色文字的地方，即可看到代码错误的原因，如图 1-11 所示。

在图 1-11 中，矩形框内的数字表示代码中存在错误的个数；IDEA 通过小框提示错误信息，例如，此处的错误表示 IDEA 不能解析 outt 这个符号。

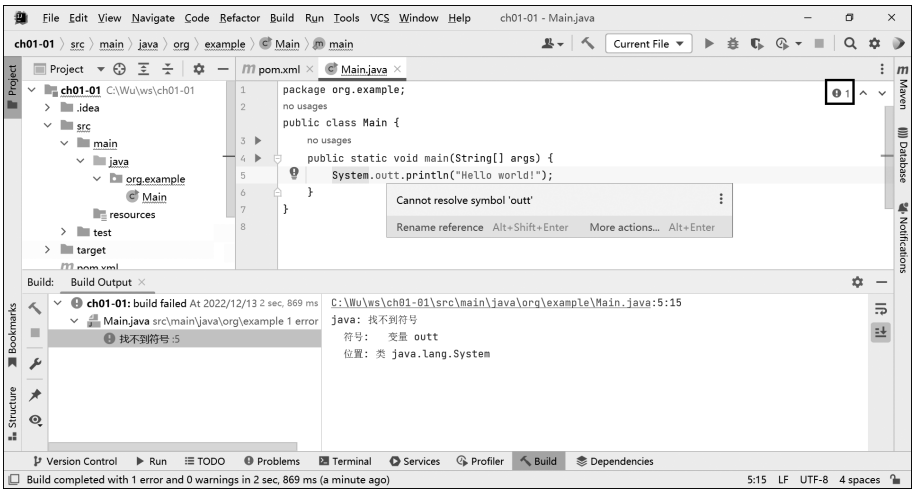

图 1-11　直观发现代码错误的方法

1.4.2　Java 程序的运行机理

当 IDEA 使用 Java 编译器成功将 Java 源代码编译为 Java 字节码后，IDEA 会使用 Java 虚拟机执行 Java 程序。为了执行 Java 程序，Java 虚拟机会在 Java 源代码中寻找名称为 main 的入口函数。例如，在如图 1-10 所示的 ch01-01 的 Java 工程中，包含 main( ) 入口函数的代码如下所示（源代码为 01-01.java）。

```
package org.example;
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

Java 虚拟机找到 main( ) 入口函数后，会逐条语句地执行由一对大括号所括住的语句。在这个例子中，Java 虚拟机会执行语句：

```
System.out.println("Hello world!");
```

这条语句的作用就是在显示器上显示“Hello world!”文字。这就是为什么会在图 1-9 中显示“Hello world!”。现在，对 Main 主类中的 main( ) 函数做以下修改（源代码为 01-02.java）。

```
package org.example;
public class Main {
    public static void main(String[] args) {
```

```
        System.out.println("Hello world!");
        System.out.println("我在学习 Java 面向对象程序设计！");
    }
}
```

当 Java 虚拟机执行这个程序时，会逐条执行 main( ) 函数的大括号所括住的语句，也就是先执行语句：

```
System.out.println("Hello world!");
```

在显示器上显示“Hello world!”文字，然后执行语句：

```
System.out.println("我在学习 Java 面向对象程序设计！");
```

在显示器上显示“我在学习 Java 面向对象程序设计！”文字。修改后程序的执行结果如图 1-12 所示。

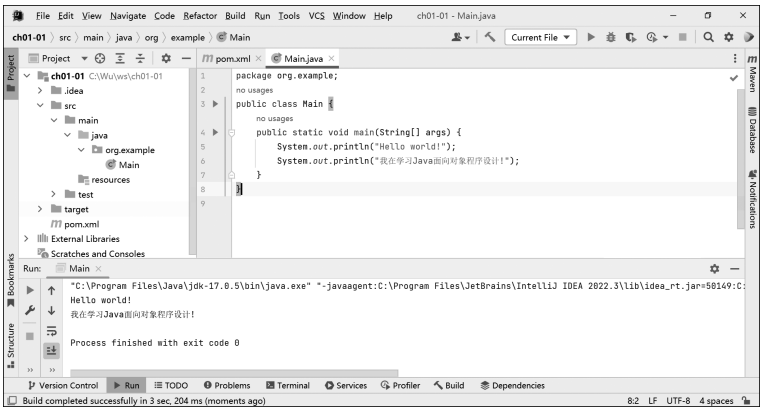

图 1-12　修改后程序的执行结果

1.5　练习：安装 Java 开发环境

请在自己的计算机上安装 Java 程序开发环境，完成后编写一个能够显示如图 1-13 所示四边形的 Java 程序。

```
* * * * * * * * * * * *
*                     *
*                     *
*                     *
*                     *
*                     *
* * * * * * * * * * * *
```

图 1-13　四边形

第 2 章　Java 基本运算和输入 / 输出

近年来，计算机在电子商务、电子政务、人工智能和大数据方面得到了广泛应用。但是本质上，计算机就是做数学运算的机器。因此，作为控制计算机工作的 Java 程序，必须具备基本的运算功能，这些功能包括：加减乘除数学运算、大小关系运算、是非（真假）逻辑运算等。本章首先介绍 Java 程序的组成，然后介绍 Java 程序的基本数据类型和如何使用变量保存程序运算数据，最后介绍如何在 Java 程序中进行数据的输入 / 输出。

2.1　Java 程序的组成

Java 程序工程由一个或多个程序包构成，在每个 Java 程序包下，可以存放 0 个或多个 Java 程序类文件（或简称为类）。每个 Java 程序类又是由一个或多个成员属性及函数组成的，每个函数包含 0 条或多条 Java 语句。为了清晰地看到 Java 程序的基本组成结构，下面新建一个名称为 ch02-01 的 Java 程序工程。

为了在已经打开项目工程的 IDEA 中新建一个程序工程，选择 File → New → Project 命令，如图 2-1 所示。

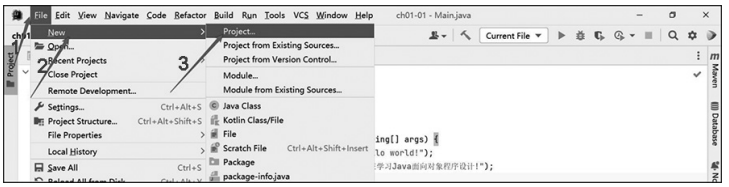

图 2-1　新建一个程序工程

出现的界面与第 1.3 节介绍的创建程序工程的过程一致，只是要注意这里将程序工程的 Name 修改为 ch02-01。创建完成的工程如图 2-2 所示。

图 2-2　创建完成的工程