第 1 章

Java 程序设计概述

1.1 情景导入

在目前程序开发语言中，Java 程序设计无论对于计算机专业的开发者还是对于计算机爱好者来说，都是十分受欢迎的语言之一。许多软件都是使用 Java 语言开发出来的。

Java 包括在以下场景运用：

1. 手机 APP，例如：订票，购物，打车等。
2. 办公软件，如 OA 系统，财务系统等。
3. 存在大量并发访问的软件，例如：淘宝，京东等。
4. 物联网，像快递柜，共享自行车等。
5. 大数据，Hadoop 框架等。

1.2 知识点概述

1.2.1 Java 简介

1. Java 语言的发展历程

SUN 公司在 1990 年 12 月开始研究的一个内部项目，试图修改和扩展以 C 的功能以方便程序的开发。经过了一些尝试，最后都以失败告终。SUN 公司一个叫做帕特里克·诺顿的工程师提议创造一种全新的语言。他和 JGosling（詹姆斯·高斯林）和麦克·舍林丹等工程师在该项目中使用基于 C 的基础，开发一种面向对象的环境。这种语言以他的办公室外的橡树命名被他称为“Oak”（橡树）。

1991 年，Oak 语言诞生。18 个月后，第一个成果是可触摸控制的手持家庭娱乐设备控制器，名为 start7（star seven）。Sun 公司为 Green 项目成立 First Person 公司。

1993 年，First Person 公司重定位 Oak。转向到网络应用领域。

1994 年，Oak 技术包括源代码全部免费公开。Oak 开发出网络浏览器 Webrunner，后更名为 HotJava。Oak 编写的 Applet 让网页由静态转成动态。

1995 年，Oak 注册登记时，发现该名已被占用，遂更名为 Java。First Person 公司解散。1995. 3. 23 是 Java 里程碑，Java 当年被评为十大优秀科技产品之一。JDK1. 0a2 版本正式对外发布。

1996 年，Sun 成立 Javasoft 分公司来发展 Java。Netscape 支持 Java。IBM、Symantec、Inprise、Microsoft IE 支持 Java。众多第三方的 Java 编译器被推出：VJ + +、caf 带 Builder 等。

1997 年，JDK 1. 1（Java Development Kit）发布。

1998 年，JDK 1. 2 发布，JDK 更名为 J2SDK（Java 2 Software Development Kit）。J2SDK 1. 2 又名 Java 2，并将 Java 的应用细分：

2000 年，J2SDK 1. 3 发布。

2002 年，J2SDK 1. 4 发布。

2004 年，J2SDK 5. 0 发布。

2005 年，J2SDK 6. 0 发布

1991 年 4 月，Sun 公司的 James Gosling 领导的绿色计划（Green Project）开始着力发展一种分布式系统结构，使其能够在各种消费性电子产品上运行。而 Green 项目组的成员一开始使用 C++语言来完成这个项目，由于 Green 项目组的成员都具有 C++背景，所以他们首先把目光锁定了 C++编译器，Gosling 首先改写了 C++编译器，但很快他就感到 C++的很多不足，需要研发一种新的语言 Java 来替代它，一杯飘香的咖啡成为了它的标志。

在 17 个月后，整个系统完成了，这个系统是更注重机顶盒式的操作系统，不过在当时市场不成熟的情况下，项目没有获得成功，但 Java 语言却得到了 Sun 总裁 McNealy 的赏识。

直至 1994 年下半年，由于 Internet 的迅猛发展和环球信息网 WWW 的快速增长，第一个全球信息网络浏览器 Mosaic 诞生了。此时，工业界对适合在网络异构环境下使用的语言有一种非常急迫的需求。Games Gosling 决定改变绿色计划的发展方向，他们对 Oak 进行了小规模的改造。Java 在 1995 年 3 月 23 日诞生了！Java 的诞生标志着互联网时代的开始，它能够被应用在全球信息网络的平台上编写互动性及强的 Applet 程序。

1995 年 5 月 23 日正式发布 Java 和 HotJava 浏览器。

在 1995 年 Sun 虽然推出了 Java，但这只是一种语言，而要想开发复杂的应用程序，必须要有一个的强大的开发库支持还行。因此，Sun 在 1996 年 1 月 23 日发布了 JDK1. 0。这个版本包括了两部分：运行环境（即 JRE）和开发环境（即 JDK）。在运行环境中包括了核心 API、集成 API、用户界面 API、发布技术、Java 虚拟机（JVM）五个部分。而开发环境还包括了编译 Java 程序的编译器（即 javac）。在 JDK1. 0 时代，JDK 除了 AWT（一种用于开发图形用户界面的 API）外，其它的库并不完整。

Sun 在推出 JDK1.0 后，紧跟着，Sun 在 1997 年 2 月 18 日发布了 JDK1.1。JDK1.1 相对于 JDK1.0 最大的改进就是为 JVM 增加了 JIT（即时编译）编译器。JIT 和传统的编译器不同，传统的编译器是编译一条，运行完后再将其扔掉，而 JIT 会将经常用到的指令保存在内容中，在下次调用时就不需要再编译了。这样 JDK 在效率上有了非常大的提升。

Sun 在推出 JDK1.1 后，接着又推出了数个 JDK1.x 版本。自从 Sun 推出 Java 后，JDK 的下载量不断彪升，在 1997 年，JDK 的下载量突破了 220 000，而在 1998 年，JDK 的下载量已经超过了 2 000 000。

从 1995 年 Java 诞生到 1998 年年底，Java 语言虽然成为了互联网上广泛使用的编程语言，但它并没有找到一个准确的定位，也没有找到它必须存在的理由：Java 语言可以编写 Applet，而 Flash 一样可以做到，而且更快，开发成本更低。

直到 1998 年 12 月，Sun 公司召开 JavaOne 大会，发布了 Java 历史上最重要的 JDK 版本：JDK1.2，伴随着 JDK1.2 一同发布的还有 JSP/Servlet、EJB 等规范，并将 Java 分成 J2EE、J2SE 和 J2ME 三个版本。

J2ME：主要用于控制移动设备和信息家电等有限存储的设备。

J2SE：整个 Java 技术的核心和基础，它是 J2ME、J2EE 编程的基础，

J2EE：Java 技术中应用最广泛的部分，J2EE 提供了企业应用开发相关的完整解决方案。这标志着 Java 已经吹响了向企业、桌面和移动三个领域进军的号角，标志着 Java 已经进入 Java2 时代，这个时期也是 Java 飞速发展的时期。

在 Java2 中，Java 发生了很多革命性的变化，而这些革命性的变化一直沿用到现在，对于 Java 的发展形成了深远的影响。知道今天还经常看到 J2EE、J2ME 等名称。

不仅如此，JDK1.2 还把它的 API 分成了三类。

核心 API：由 Sun 公司制定的基本的 API，所有的 Java 平台都应该提供。这就是平常所说的 Java 核心类库。

可选 API：这是 Sun 为 JDK 提供的扩充 API，这些 API 因平台的不同而不同。

特殊 API：用于满足特殊要求的 API。如用于 JCA 和 JCE 等第三方加密类库。

2002 年，Sun 发布了 JDK 历史上最成熟的版本：JDK1.4 成为发展最快的一个 JDK 版本。JDK1.4 已经可以使用 Java 实现大多数的应用了。

在此期间，Java 语言在企业应用领域大放异彩，涌现出大量基于 Java 语言的开源框架：Struts、WebWork、Hibernate、Spring 等；大量企业应用服务器也开始涌现：WebLogic、WebSphere、JBoss 等，这些都标志着 Java 语言进入了飞速发展时期。

2004 年 10 月，Sun 发布了万众期待的 JDK1.5，同时，Sun 将 JDK1.5 改名为 Java SE5.0，J2EE、J2ME 也相应地改名为 Java EE 和 Java ME。JDK1.5 增加了诸如泛型、增强的 for 语句、可变数量的形参、注释、自动拆箱和装箱等功能；同时，也发布了新的企业级平台规范，如通过注释等新特性来简化 EJB 的复杂性，并推出了自己的 MVC 框架规范：JSF，JSF 规范类似于 ASP.NET 的服务器端控件，通过它可以快速地构建复杂的 JSP 界面。SUN 发布 Java1.5 版，成为 Java 平台发展史上又一里程碑。为了表明该版本的重要性，

SUN 将之称为 Java5. 0。

2005 年 6 月，JavaOne 大会召开，SUN 公司公开 Java SE 6。此时，Java 的各种版本已经更名以取消其中的数字“2”：J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。一直以来，Sun 公司维持着大约 2 年发布一次 JDK 新版本的习惯。

2006 年 11 月，SUN 公司宣布 Java 全线采纳 GNU General Public License Version 2，从而公开了 Java 的源代码。

2009 年 4 月，Oracle 宣布将以 74 亿美元收购 Sun 公司，同时获取了 Java 的版权。

2011 年 7 月，Oracle 公司终于" 如约" 发布了 Java SE 7——这次版本的升级经过了将近 5 年时间。Java SE 7 也是 Oracle 发布的第一个 Java 版本，引入了二进制整数、支持字符串的 switch 语句、棱形语法、多异常捕抓、自动关闭资源的 try 语言等新特性。

2014 年 3 月，Oracle 公司发不了 Java SE8，这次版本升级为 Java 带来了全新的 Lambda 表达式、流式编程等大量新特性，这些新特性使得 Java 变得更加强大。

2017 年 9 月，Oracle 公司发布了 Java SE 9，这次版本升级强化了 Java 的模块化系统，让庞大的 Java 语言更轻量化，而且采用了更高效、更智能的 GI 垃圾回收器，并在核心类库上进行了大量更新，可以进一步简化编程，但对语法本身更新并不多。

2018 年 3 月，Oracle 公司发布了 Java SE 10，JDK 10 中引入了两个主要的增强功能：JEP 319 根证书：列表或根证书已公开给 JDK 的 cacerts 密钥库，一些与安全相关的 API 标记为要删除。

2018 年 9 月，Oracle 公司又发布了 Java SE 11。JDK 11 中引入的主要安全增强功能由 JEP 332：传输层安全性（TLS）1. 3 提供。这是 TLS 协议的新版本，与 TLS 规范的先前版本（1. 2）相比，提供了许多增强功能。此外，对 cacerts 密钥库中的根证书进行了更多增强（添加了一些根证书，并删除了一些根证书）。

之后 Oracle 公司便以每年发布两个版本的速度升级迭代。目前，JDK 版本已经发布到了 JDK 17。

2. Java 语言特点

（1）Java 是面向对象的语言。具有很强的设计功能，它可以促成明确的接口定义，并允许开发人员建立可重复使用的软件部件。

（2）Java 是分布式语言。Java 拥有广泛的能轻易地处理 TCP/IP 协议的运行库等。这使得在 Java 中比在 C 或 C++中更容易建立网络连接。Java 应用程序可以借助 URL 通过网络开启和存取对象，就如同存取一个本地文件系统一样简单。

（3）Java 是健壮的。Java 的目标是要协助开发人员建立各方面可靠的程序，

（4）Java 的安全性。Java 设计的目的是要能够使用于网络/分布式运算环境。为此，Java 非常强调安全性，以确保建立无病毒且不会被侵入的系统。

（5）Java 的中立性结构。Java 的设计目标是要支持网络应用程序。一般而言，网络是由许多不同的平台系统构成，包括各种 CPU 与操作系统结构。为了让 Java 应用程序能够在网络上任何地方执行，其编译器将会生成一种具备结构中立性的目标文件格式。编译后

的程序码可以在提供 Java 运行系统的多种不同处理器上面执行。

（6）Java 是高效能的程序。字节代码格式在设计上即已考虑了机器码的产生，因此实际的机器码生成程序相当简单。其生成的机器码是有效的，编译器自动分配寄存器，而在生成字节代码期间也会进行一些优化。

（7）Java 对多线程的支持。Java 拥有一组复杂的同步化基本单元，它们是以广泛使用的 C. A. R. Hoare 监视器与条件变量图为基础的。将这些概念融合到语言中之后，它们就变得更容易使用且更为健壮。

1.2.2　Java 开发环境搭建

开发运行 Java 程序，需要在电脑上安装相应的 JDK。

JDK：Java Development Kit 是 Java 开发运行环境。

JRE：Java Runtime Environment 是 Java 运行环境，如果不需要开发只需要运行 Java 程序，那么可以安装 JRE。例如程序员开发出的程序最终卖给了用户，用户不用开发，只需要运行程序，所以用户在电脑上安装 JRE 即可。

JDK 包含了 JRE，而 JRE 中包含虚拟机 JVM。首先安装 JDK（Java Development Kit 开发工具包）。

第一步：下载 JDK

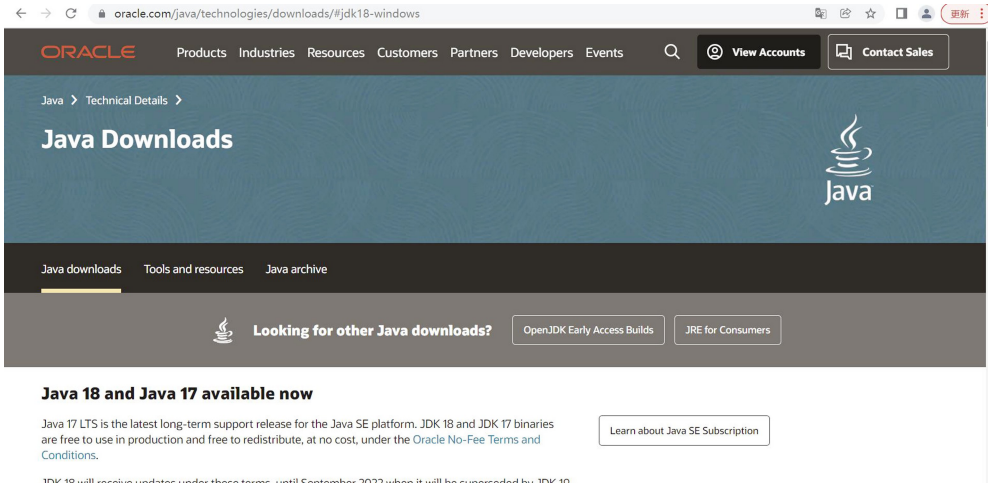

图 1.1　下载网址

首先需要打开 https：//www. oracle. com/java/technologies/downloads/，如下图所示。

拖拽右侧滚动条至下方，选择下图中“Java18”，并选择相应的系统，本书以“Windows”系统为例。点击下载路径下载即可。

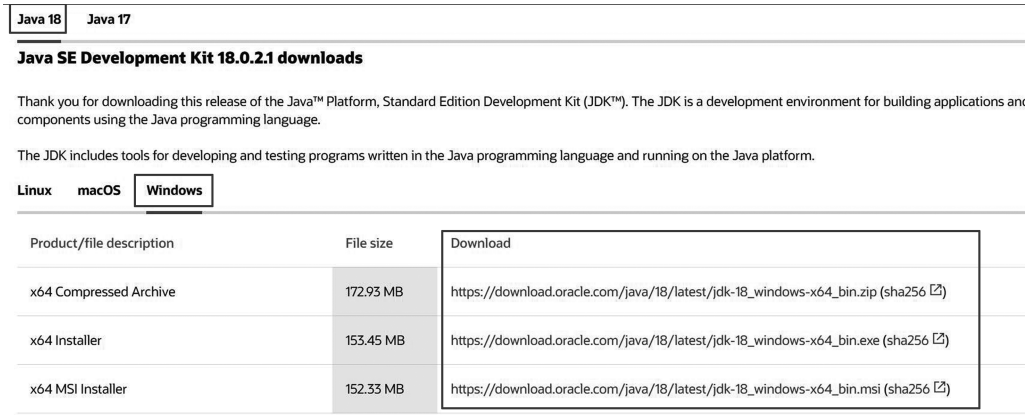

图 1.2　选择版本

下载好 JDK 后需要进行安装。

图 1.3　JDK 安装包

第二步：安装 JDK

1. 双击安装包后，出现如下图的界面，点击“下一步”。

图 1.4　安装初始界面

2. 出现下图时，可以更改安装路径，然后点击“下一步”。

图 1.5　更改目录

3. 选择好安装路径后，出现安装进程界面。

图 1.6　正在安装界面

4. 安装完成时，出现完成界面。

图 1.7　安装成功

第三步：配置环境

安装好 JDK 程序后，需要进一步的配置，才能正常使用。在 win10 环境配置步骤如下：

1. 右击桌面中“此电脑”图标，在快捷菜单中选择属性，如图 1.8 所示。

图 1.8　系统设置

2. 选择右侧“相关设置”中的“高级系统设置”，出现系统属性。

图 1.9　系统属性设置

3. 在系统属性中选择高级选项卡，点击下端的环境变量按钮，出现下图所示界面

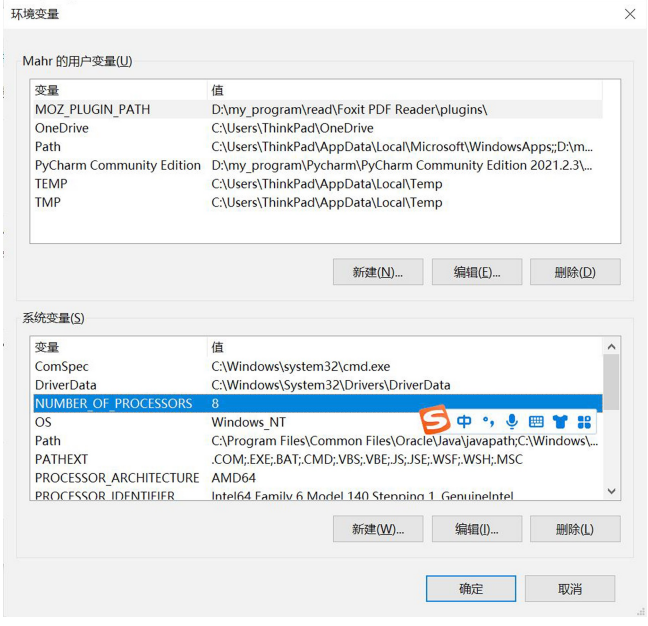

图 1.10　环境变量

4. 点击下面的新建按钮，添加系统变量。

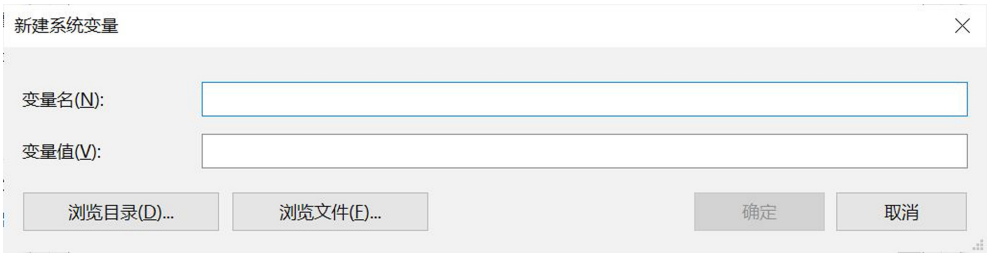

图 1.11　新建系统变量

5. 首先新建变量名 Java_HOME，变量值为安装 JDK 的路径。

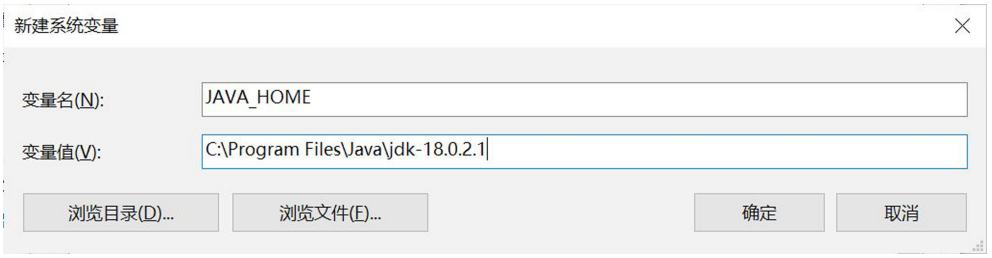

图 1.12　新建 Java_HOME 变量

6. 接着编辑或新建 PATH 变量，变量值为“%Java_HOME% \ bin”

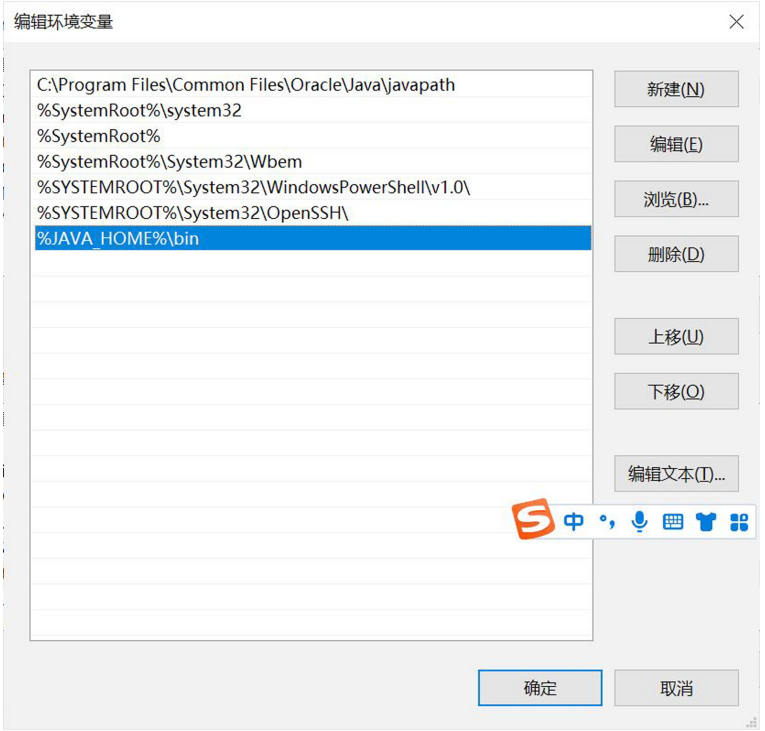

图 1.13　编辑或新建 PATH 变量

7. 最后设置 classpath 变量，变量值为“%Java_HOME% \ lib”.

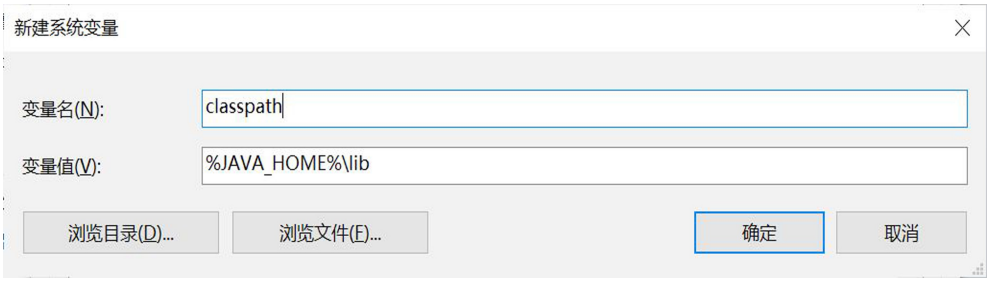

图 1.14　编辑 classpath 变量

第四步：测试 JDK 安装成功。

在左下方的“搜索”处，输入“cmd”命令。启动“命令提示符”窗口，在窗口中输入“java-version”然后按下回车，如下图所示，出现相关的版本信息，表示已经安装成功。

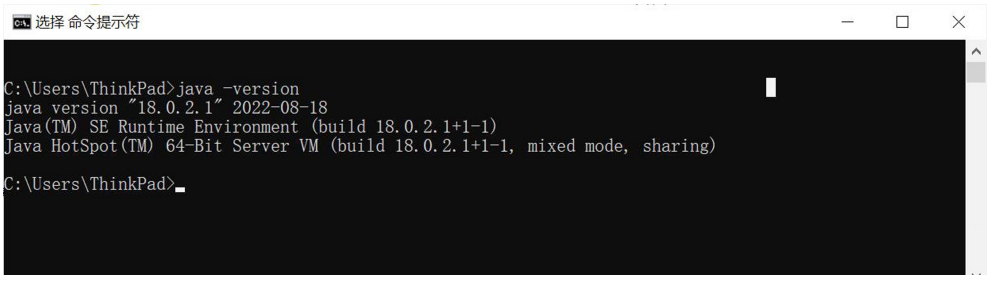

图 1.15　版本信息

1.2.3　Java 开发流程

1. 编写源程序

源程序/源代码：就是要给计算机下达的指令（使用 Java 语法编写的指令）。源代码要写在一个扩展名为”. java”的文件中。

2. 编译源程序

源代码需要使用编译器来编译，编译后会产生一个 . class 文件（ByteCode，字节码文件）。

JDK 中的 javac. exe 就是 Java 编译器。可以在命令提示行中使用以下命令完成编译。

javac 源程序名 . java

3. 运行程序

编译后生成的字节码文件（. class）文件使用 JDK 中的”java. exe”运行。在命令提示行中使用以下命令完成运行，并显示结果。

java 类名（注意此处不加扩展名）

Java 的基本组成单位为类，所以在源码编写时必须有类的设计。

类结构：类声明+类体。类的声明使用”class”关键字。类名后大括号内为类体部分。

源码编写最基本的组成部分如下所示：

```
public class 类名{
    //方法结构:方法声明+方法体
    // main 方法:Java 程序的入口
    //方法声明
    public static void main(String[] args){
        //方法体:编写让计算机执行的指令
        //实现让计算机在屏幕输出 Hello World 这句话!
        System.out.println("Hello World");
    }
}
```

其中：

public：公共的

class：类

static：静态的

void：空

main：主要的

String：字符串

args：arguments，参数

注意事项：

1. 注意单词大小写（完全一致）

2. 注意这些符号都是英文输入法输入的

( )小括号

{ }花括号，大括号

[ ]中括号

" 双引号

； 分号

3. 注意单词正确拼写

main 错误写成 mian

println（小写的 L）写成 printLn（错误写成大写的 I）

4. 注意下一级的代码缩进（规范）

5. 由 public 修饰的类名要和源文件名保持一致（命名不要出现空格等）。

1.2.4　第一个 Java 程序

1. 编写 Java 源程序

首先创建一个文件，保存为 .java 文件。再 Java 程序设计中，main 方法是程序的入口点，当创建一个 HelloWorld.java 源文件时，在该文件中 main 方法一般放置在主类中。一个 .java 文件中可以有多个类，但是这些类中只能有一个类用 public 关键字修饰，称为主类，主类的类名必须与文件同名。

在 HelloWorld.java 中代码如下：

```
public class HelloWorld{
    public static void main(String args[]){
        ……
    }
}
```

2. 编译 Java 源程序

编写好 .java 源代码后，用下列命令进行编译：

```
javac HelloWorld.java
```

如果编译器没有返回任何提示信息，表示编译成功，此时，会生成一个 HelloWorld.class 的文件。生成的 class 和对应的 java 源文件的文件名是同名的。该文件称为字节码文件。如果编译过程中遇到问题，则不会生成相应的字节码文件。必须把所有错误改正后重新执行编译命令，生成 .class 文件。

3. 运行程序

生成好的字节码文件后，用下列命令进行运行：

```
java HelloWorld
```

屏幕显示执行的结果，在执行 java 命令时，不需要写 .class，只写文件名即可。

至此使用命令行的方式，完成了运行第一个 Java 程序。但这种编辑方式比较麻烦，也不方便语法错误的检查。为了更方便编写及错误检查，可以使用 Eclipse 开发平台。

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。Eclipse 附带一个标准的插件集，包括 Java 开发工具（Java Development Kit，JDK）。Eclipse 是著名的跨平台的自由集成开发环境（IDE）。最初主要用来 Java 语言开发，通过安装不同的插件 Eclipse 可以支持不同的计算机语言，比如 C++和 Python 等开发工具。Eclipse 本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。

第一步：下载 Eclipse

1. 首先需要打开 https://www.eclipse.org，进入官网，如图 1.16 所示。

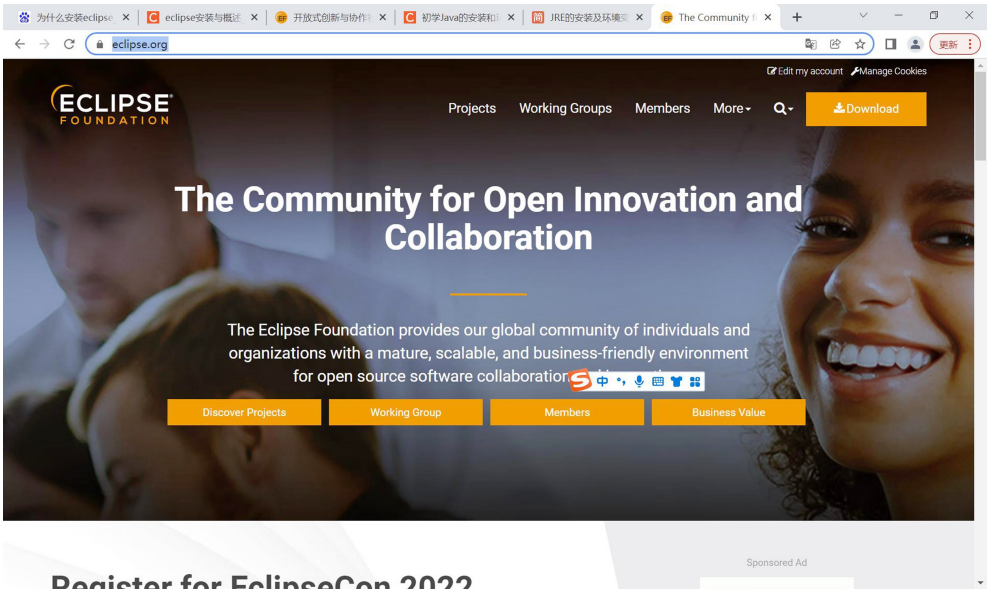

图 1.16　Eclipse 下载网址

2. 单击官网右上方的“Download”按钮，当弹出以下界面后点击红色框里的选项下载。

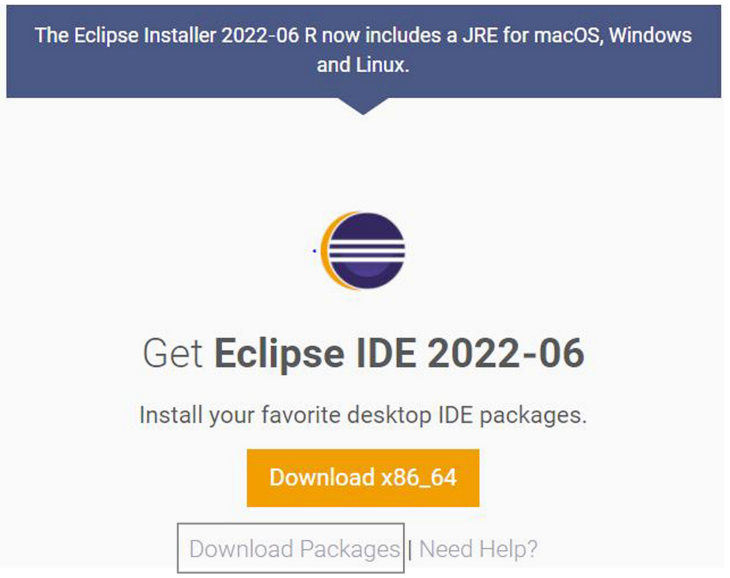

图 1.17　Eclipse 下载

3. 接下来，在下图红色框中的选项，选择对应自己电脑的版本下载。

图 1.18　Eclipse 版本

4. 接着会弹出下载页面，然后点击 Download 下载即可。

第二步：启动 Eclipse

1. 当下载完成后，会得到一个压缩包。
2. 点击下载好的 .zip 文件去解压，解压好以后会出现一个文件夹。
3. 打开解压好的文件夹，然后会出现 eclipse.exe 文件，双击启动 Eclipse。

图 1.19　Eclipse 压缩包

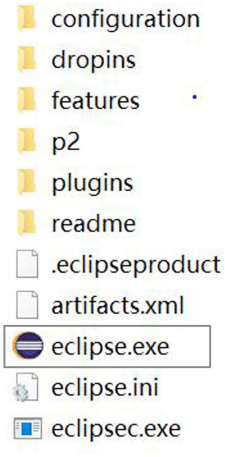

图 1.20　Eclipse 启动

第三步：编写运行“Hello World”程序

1. 在 Eclipse 窗口中点击“File”菜单中的“New”，再点击“Project”去创建第一个项目。

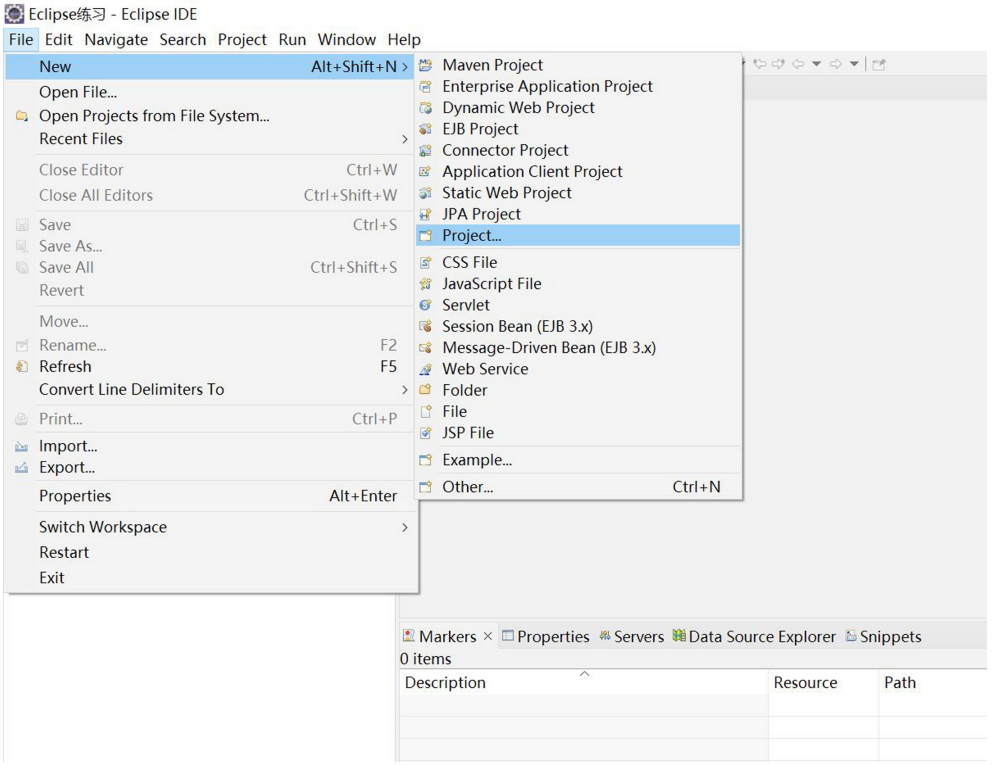

图 1.21　新建项目

2. 接着会弹出以下界面然后点击“Java Project”创建一个新项目，点击“Next”。

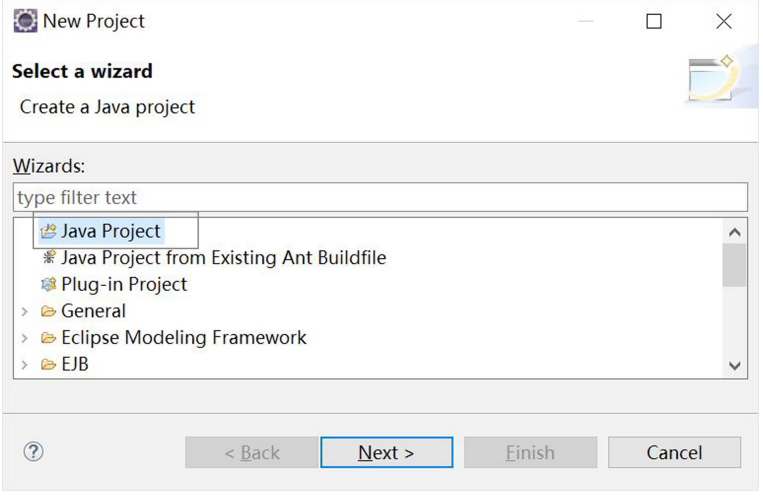

图 1.22　Java Project

3. 在红色框里为自己的第一个项目取名字，这里以“Demo”为例。单击“Finish”按钮。

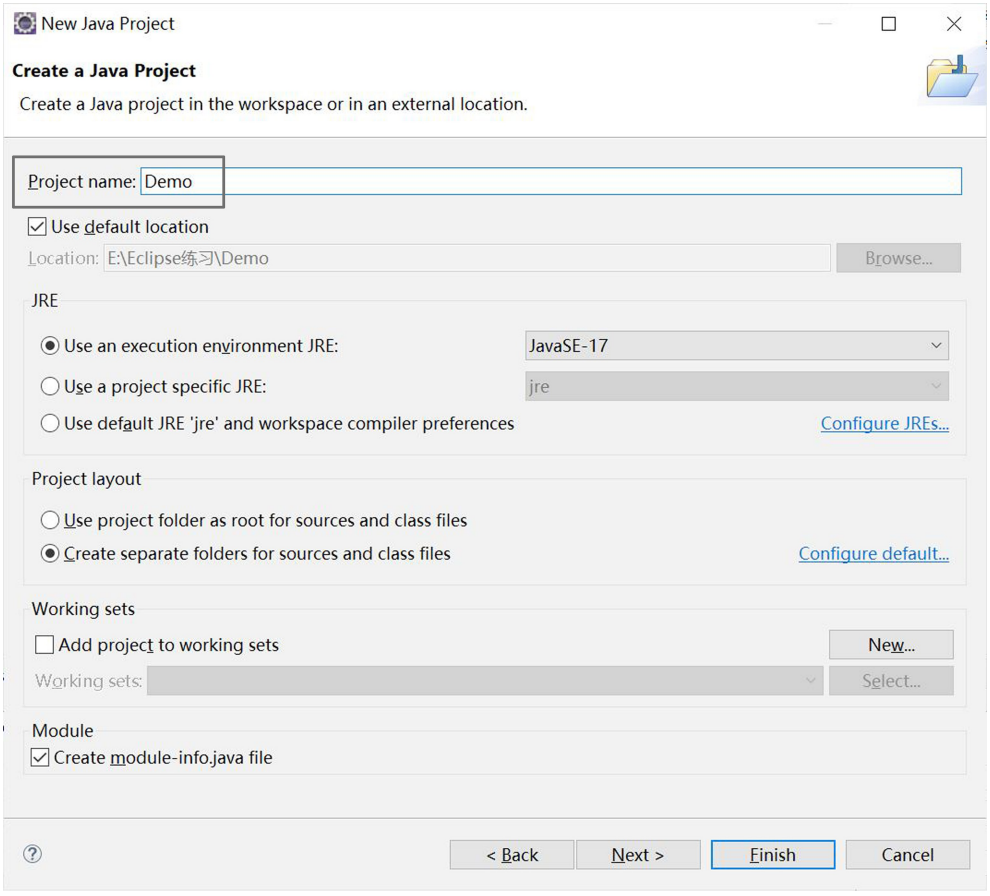

图 1.23　项目名

4. 回到主界面，可以看到创建好的“Demo”项目了。

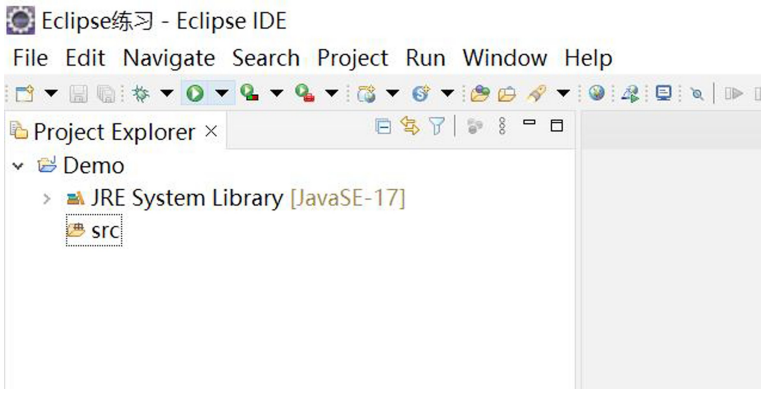

图 1.24　创建成功

5. 右击“src”选择“New”命令中的“Class”，创建一个 Class 类，这里以“Hello Woeld”为例。并勾选“public static void main（String［］args）”选项。单击”Finish”。

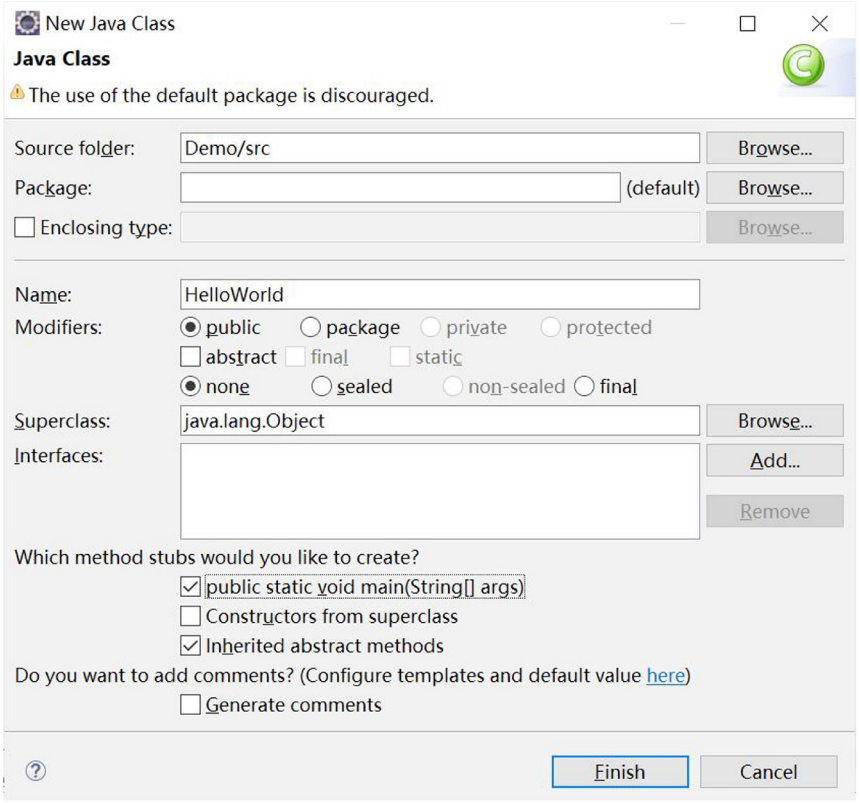

图 1.25　新建类

6. 接着在红色框方法里里输入，然后鼠标右键点击“Run As”中的“Java Application”。

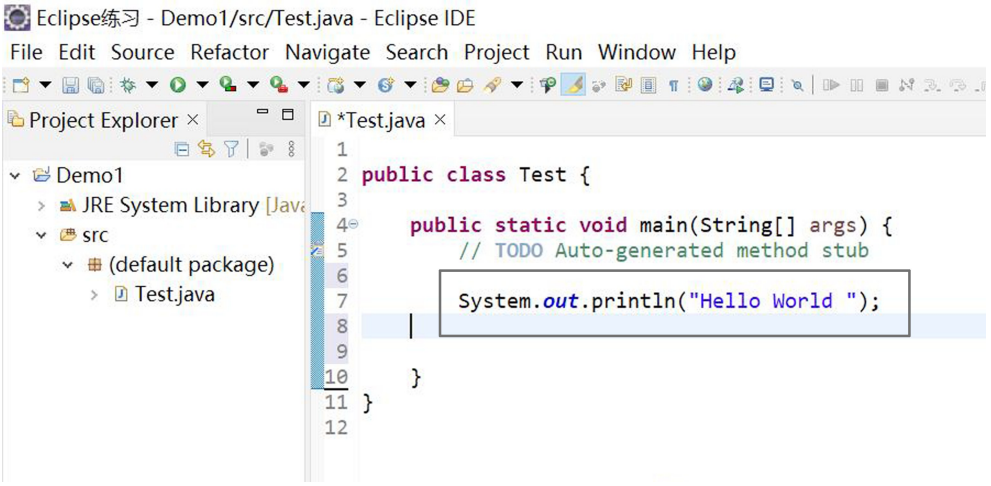

图 1.26　编写代码

7. 运行后显示“HelloWorld”。

图 1.27　运行效果

1.3　精选练习

一、填空题

1. Java 程序的运行环境简称之为_____。
2. 编译 Java 程序需要使用_____命令。
3. javac. exe 和 java. exe 两个可执行程序存放在 JDK 安装目录的_____目录下。

二、选择题

1. 以下选项中，哪些属于 JDK 工具？（多选）

   A. Java 编译器　　B. Java 运行工具

   C. Java 文档生成工具　　D. Java 打包工具

2. Java 属于以下哪种语言？

   A. 机器语言　　B. 汇编语言　　C. 高级语言　　D. 以上都不对

3. 下面哪种类型的文件可以在 Java 虚拟机中运行？

   A. . java　　B. . jre　　C. . exe　　D. . class

4. 面向对象编程的语言有哪些？

   A . C　　B. C++　　C. Java　　D. C#

三、问答题

1. 简述 Java 的特点。
2. 简述 JRE 与 JDK 的区别。

四、思考题

1. 请写出你的手机的主要属性和方法，并用 Java 语言写出此类手机的类示意代码。
2. 请找出 2 个你身边的对象，分析它们各自的属性和方法，写出 Java 类示意代码。

第 2 章

Java 基础

2.1 情景导入

本章学习常量和变量的概念，掌握 Java 中基本的数据类型的特点。程序的运行就像在厨房做菜一样，都是按照固定步骤具体执行的过程，最终把执行的结果（做出来的菜）呈现出来。

下面具体看一下这两个过程的联想对比，从中体会两者的关联。

表 2-1 情景导入对比

程序运行需要有源码	厨房做菜需要有菜谱（制作流程）
编写源码	编写菜谱（写制作流程）
首先明确要项目实现的功能 1. 确定要处理的数据。 2. 内存中需要分配什么样的空间存放，确定数据结构（数据类型、数据长度、数据关系）。 3. 数据的初始化。 4. 进行计算。 5. 显示计算处理的结果	先想好做什么菜？需要准备原材料 1. 确定都有哪些原材料。 2. 厨房中应该使用碗、盘、盆……来存放。 3. 先把原材料放入相应容器中。 4. 进行烹饪。 5. 把制作完成的菜装盘。

本章为完成在程序开发过程中对数据的识别、存储、初始化及计算过程。

2.2 知识点概述

Java 程序设计过程中如果对数据进行处理，首先考虑所使用数据的类型，然后再考虑内存空间的分配存储，最后根据数据的类型特点对数据进行计算处理。其中数据类型就是要确定在内存中开辟多大的存储空间及所放置数据的类型。变量就是存储数据所使用到的基本单元空间。运算符就是具体的运算符号，用于操作一个或多个数据得出具体的计算结

果。最后通过控制流语句来控制具体的执行顺序。

2.2.1　常量和变量

情景导入：变量就像做菜时用的碗，而常量就是能放在碗里面的实际物品（数据）。

图 2.1　容器

图 2.2　数据

碗可以理解为变量空间，碗中的西红柿、鸡蛋都为常量。碗根据所装物品种类不同，也有不同种类的碗；同样的种类又有大有小。即变量有不同的类型。碗中常量也有不同种类及多少的需求，所以常量也分类不同的数据类型。

一、常量

所谓常量就是其值固定不变的量。例如："HelloWorld"，123 等都是一些固定不变的数据，也可以理解为具体的数值，所以称这些数据都是常量。Java 中基本常量类型有整型常量、字符型常量、浮点型常量和布尔常量等。

在 Java 中还可以使用 final 关键字来定义一个常量。例如：int final S=0；此处声明了一个整型常量 S，它的值为 0。并习惯上把常量的名字定义为大写。

1. 整型常量

例 1：常量的应用，说说以下程序段中的常量有哪些？

```
public class Circle_Test
{
    final  static double PI=3.14;
    public static void main(String args[])
    {
        double r=10.0;
        double area=PI*r*r;
        System.out.println("圆的面积为:"+area);
    }
}
```

该程序的运行结果：

```
圆的面积为:314.0000
```

2. 整型常量

整型常量是整数类型的数据，有二进制、八进制、十进制和十六进制 4 种表示形式具体表示形式如下。

十进制表示方式：正常数字。如 13、25 等

二进制表示方式：以 0b（0B）开头。如 0b1011 、0B1001

十六进制表示方式：以 0x（0X）开头。数字以 0－9 及 A－F 组成如 0x23A2、0xa、0x10

八进制表示方式：以 0 开头。如 01、07、0721

3. 浮点型常量

浮点数常量就是在数学中用到的小数，也叫小数类型，分为 foat 单精度浮点数和 double 双精度浮点数两种类型。如 1.0、-3.15、3.168 等。

4. 字符型常量

字符常量用于表示一个字符，一个字符常量要用一对英文半角格式的单引''号引起来，它可以是英文字母、数字、标点符号以及由转义序列来表示的特殊字符。如'a'，'A'，'0'，'家'。

5. 布尔型常量

Java 的布尔型常量只有两个值，即 false（假）和 true（真）。

6. null 常量

null 常量只有一个值 null，表示对象的引用为空。

以上就是 java 常量中几种类型的介绍，只有掌握了常量的基础知识，才能下一步在 java 中使用常见。

二、变量

在 Java 程序设计中，使用的各种变量都应预先加以定义，即先定义，后使用。对变量的定义一定要准确的表达出所需要空间的类型和大小。所谓变量就是其值可以改变的量。变量是对内存一段空间的命名和指定。包括变量名和变量值这两部分信息。变量名是对内存中开辟的一段空间的名字的指定。而这段空间的大小，即所占的字节数由该变量的类型说明来确定。通过变量名可以对该段内存空间赋值和引用。

变量名的命名规则如下：

1. 必须以字母、汉字、“_”或“$”开头。
2. 变量名中只能包含字母、汉字、数字、“_”或“$”，不能含有其他符号。
3. 不能与系统关键字重名。

需要注意在 Java 中严格区分大小写。在变量使用过程中要“先声明，再使用”。

声明的形式如下：

类型说明符变量名=［变量初值］；

例如：

```
int x=0;
float f=1.24;
char c;
c='我'
```

例 2：写出下列程序的运行结果，并说说哪些数据为变量？

```
public class Test
{
    public static void main(String args[])
    {
        double r=10.0;
        System.out.println("圆的半径为:"+r);
    }
}
```

该程序运行结果为：圆的半径为：10.0

2.2.2　Java 中的数据类型

Java 是一种强类型语言，所以在 Java 中每一个变量在声明时必须指定数据类型。数据类型可以分为基本类型和引用类型两大类，本章只研究基本的数据类型，引用类型将在后面章节中具体讲解。

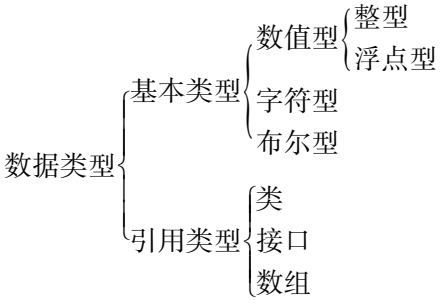

图 2.3　数据类型

情景导入：数据类型就是所使用的材料是米？是面？还是油？是蔬菜？还是水果？不同类型的原材料要选择不同类型的容器存放。同类型的原材料，根据用量多少还得进一步思考用小碗、中碗、还是大碗存放。

一、数值类型

1. 整数类型

Java 中的整数常量有三种具体的表示形式：十进制、二进制、八进制、十六进制。

十进制整数用 0 到 9 十个数字组成，计算过程逢十进一。直接书写的数字为十进制，例如：123，879 等。

二进制整数用 0 和 1 两个数字组成，计算过程逢二进一。在书写过程中必须以 0b 或 0B 开头，例如：0b101，0B11 等。

八进制整数用 0 到 7 八个数字组成，计算过程逢八进一。在书写过程中必须以 0 开头，例如：0123，067 等。

十六进制整数用 0 到 9，A 到 F，16 个符号组成，计算过程逢十六进一。在书写过程中必须以 0x 或 0X 开头，例如：0x123，0XAF 等。

整数类型：包括 byte、short、int、long 四种类型。

表 2-2　整数类型

类型	字节数	范围
Byte	1	$-2^{7}\ldots2^{7}-1$
Short	2	$-2^{15}\ldots2^{15}-1$
Int	4	$-2^{31}\ldots2^{31}-1$
Long	8	$-2^{63}\ldots2^{63}-1$

int 是整数类型中最常用的数据类型，一个整数例如 12 默认为 int 型数据，如果我们需要使用一个比较大的数据，由于 int 的数据范围有限，需要使用 long 类型。在整数后面加字母“L”或“l”表示 long 类型的数据。short 类型和 byte 类型则用来表示数据量比较小的数据，通过表 2-1 即可知道数据的表示范围。

2. 浮点数类型

浮点数类型用 float（单精度）或者 double（双精度）来声明。

float 型为单精度类型，占 4 个字节的空间，数据的范围-3.4E38～3.4E38。单精度浮点数必须在尾部加“F”或“f”后缀，否则浮点数默认为双精度型。

例如：123.22F 或 123.3f。

double 型为双精度类型，占 8 个字节的空间，数据的范围-1.798E308～1.798E308。双精度浮点数尾部可以加“D”或“d”后缀，也可以不加。

例如：143.2 或 143.2D。

二、字符类型

char 类型表示单个字符。一个 char 代表一个 16-bit 无符号的（不分正负的）Unicode 字符。char 类型数据必须放在一对单引号内，并且为单个字符。

例如：‘a’、‘我’、‘\t’等。

Java 同样可以使用转义字符。

表 2-3　Java 中的转义字符

转义序列	含义
\ b	退格
\ t	水平制表
\ n	换行
\ f	换页
\ r	回车
\ ”	双引号
\ ’	单引号
\ \	反斜杠

三、布尔类型

针对逻辑数据“真”和“假”，Java 使用了专门的布尔类型数据来表示。布尔类型的取值只有 true 和 false 两个值。使用关键字 boolean 来定义。在 Java 中，布尔型数据由于不是使用数值来表示，所以它不能与其他类型的数据进行转换。

例如：

```
boolean bz=true;
```

四、数据类型的转换

使用数值表示的数据之间可以相互转换。各类型之间的转换分为隐式转换和显式转换两种。当小空间向大空间数据类型转换时可以自动转换，即为隐式转换。大空间数据向小空间数据转换时需要强制转换，即为显式转换。

整数、字符型、浮点型数据在进行混合运算时相互转换，隐式转换具体转换规则如下图：

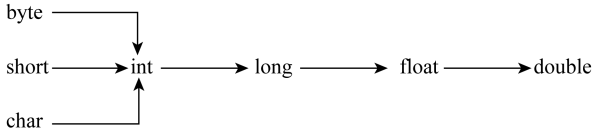

图 2.4　类型转换

其中 byte、short、char 类型数据运算时不会相互转换，这些数据首先转换为 int 然后

再进行运算。

例 3：写出下列程序运行结果

```
public class Test_3 {
    public static void main(String[] args){
        byte i1=20;
        char i2='a';
        int i3=i1;
        long i4=i2;
        float i5=50;
        double i6=i4;
        double i7=i5;
        System.out.println("byte i1="+i1);
        System.out.println("char i2="+i2);
        System.out.println("int i3="+i3);
        System.out.println("long i4="+i4);
        System.out.println("float i5="+i5);
        System.out.println("double i6="+i6);
        System.out.println("double i7="+i7);
    }
}
```

运行结果如下：

```
byte i1=20
char i2=a
int i3=20
long i4=97
float i5=50.0
double i6=97.0
double i7=50.0
```

显式转换为类型强制转换，具体格式如下：

变量=（目标类型）值；

例如：

```
    byte  b;
    int i=20;
    b=(byte)i;
```

例 4：请指出下列程序的错误

```
public class Test_4 {
    public static void main(String[] args){
        byte b1=10;
        byte b2=20;
        byte b3=b1+b2;
        System.out.println("byte b1="+b1);
        System.out.println("byte b2="+b2);
        System.out.println("byte b3="+b3);
    }
}
```

该程序编译时 byte b3=b1+b2；行会报错。因为在做 b1+b2 时，系统自动把 byte 类型转换成 int 类型。而 int 类型赋值给 byte 类型时不能自动转换，所以报错。需要强制类型转换。改行处理为：

byte b3=（byte）（b1+b2）；

2.2.3　Java 中的运算符

运算符指明对操作数的运算方式。组成表达式的 Java 操作符有很多种。运算符按照其要求的操作数数目来分，可以有单目运算符、双目运算符和三目运算符，它们分别对应于 1 个、2 个、3 个操作数。运算符按其功能来分，有算术运算符、赋值运算符、关系运算符、逻辑运算符、位运算符和其他运算符。

在 Java 运算符中，运算符一共几大常见的运算符

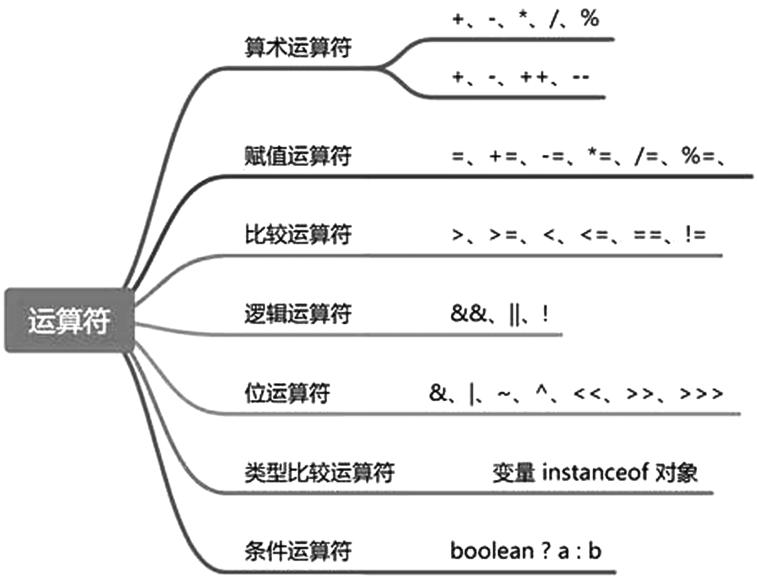

图 2.5　运算符

情景导入：运算可以理解为制作菜的方法，是炖、煮、炸还是炒……。Java 通过计算功能把原材料进行加工。每种加工结果不同，需要了解每种加工后的结果会是什么样子。

表达式：用运算符把常量或变量连接起来的式子。

表达式的类型：表达式的类型为表达式运算结果的数据类型。

1. 算术运算符

要就是用来进行算术运算的符号。算术运算符在平常生活中也是非常常见的，比如在+，-，*，/，<，>，而在 Java 中也有，而在 Java 中的运算符主要就是对常量和变量进行运算。

+：加法运算　　例：1+2　1+3　　int　a = 2；　a+"12345"

　　结果：3　　4　212345

注意：

（1）加号两边是数值型数据时，进行加法运算。

（2）加号两边有任意一边是字符串时，进行字符串的拼接。

-：减法运算　　例：2-1　3-1　　结果分别为：1　2

*：乘法运算　　例：5*2　3*5　　结果分别为：10　15

/：除法运算　　例：10/2　15/5　　结果分别为：5　3

注意：

（1）得到两个数据相除的商而不是余数（如果有浮点数的参入运算，则结果为浮点数）

（2）Java 中整数除以整数结果还是整数

：取模（取余）运算：例：5%2　11%3　结果：1　2

注意：

得到两个数据相除的余数，可以用来判断两个数是否能够整除。

++：自增 1　　例：int a=1　++a　　结果：2

--：自减 1　　例：int a=2　--a　　结果：1

注意：

（1）自增和自减在单独使用的时候，放在变量前或者变量后面都是一样的，都是自身加一或减一。

（2）在参与运算时，++或--在变量前，先自增，再以新值进行其他运算，而在变量后，先以原值进行其它运算，再自变化。

例 1：没有参与运算时：

```
int a=5
int b=6
a++
++b
```

结果：

```
a=6
b=6
```

参与运算：

++在后（a 先赋值给 b，a 在自增）

```
int b=a++
System.out.println(a)
System.out.println(b)
```

结果：

```
a=6
b=5
```

++在前（a 先自增在赋值给 b）

```
int b=++a
System.out.println(a)
System.out.println(b)
```

结果：

```
a=6
b=6
```

例 5：写出运行结果

```
public class Test_5{
    public static void main(String[] args){
        int x=10;
        int y=20;
        int sum=x+y;
        int sub=x-y;
        int ch=x*y;
        int div=x/y;
        int re=x%y;
        System.out.println("x="+x+" y="+y);
        System.out.println("x+y="+sum);
        System.out.println("x-y="+sub);
        System.out.println("x*y="+ch);
        System.out.println("x/y="+div);
        System.out.println("x%y="+re);
    }
}
```

运行结果为：

```
x=10  y=20
x+y=30
x-y=-10
x*y=200
x/y=0
x%y=10
```

例 6：写出运行结果

```
public class Test_6 {
  public static void main(String[] args){
    int x=10;
    int y=20;
    int i1=x++;
    int i2=++x;
    int i3=y--;
    int i4=--y;
    System.out.println("x++="+i1);
    System.out.println("++x="+i2);
    System.out.println("y--="+i3);
    System.out.println("--y="+i4);
  }
}
```

运算结果为：

```
x++=10
++x=12
y--=20
--y=18
```

2. 关系运算符

关系运算符也叫做比较运算符，是用来描述两个数据之间的关系，判断结果为 true 或 false。

常见的关系运算符：<（小于），>（大于），==（等于），<=（小等于），>=（大等于），!=（不等于）。

```
public class d {
    public static void main(String[] args){
    int a=10;
```

```
        int b=20;
            System.out.println(a<b);    //如果成立则是 true
        }
    }
```

结果：true

例 7：写出运行结果

```
public class Test_7 {
    public static void main(String[] args){
        int i=10;
        int j=20;
        char c1='A';
        char c2='B';
        String s1="abc";
        String s2="abc";
        System.out.println("10==20:"+(i==j));
        System.out.println("10==20:"+(i!=j));
        System.out.println("A==B:"+(c1==c2));
        System.out.println("A==B:"+(c1>=c2));
        System.out.println("abc==abc:"+(s1.equals(s2)));
    }
}
```

运行结果如下：

```
10==20:false
10==20:true
A==B:false
A==B:false
abc==abc:true
```

3. 位运算符

位运算符有：与（&）、非（~）、或（|）、异或（^）、右移（>>）、左移（<<）。

&：双目运算符，运算时均把运算数转换为二进制再做比较，规则：当相同的位上均为1时结果为1，否则结果为0. 如：1010&1101，转为二进制：1111110010&10001001101，比较结果为：1000000转为十进制：64。所以1010&1101=64；

| ：当两边操作数的位有一边为1时，结果为1，否则为0。如1100 | 1010=1110

~：0变1，1变0

^：两边的位不同时，结果为1，否则为0. 如1100^1010=0110

>>：a>>b，将a的二进制数表示形式右移b位。

<<：a<<b，将 a 的二进制数表示形式左移 b 位。

例 8：写出运行结果

```
public class Test_8 {
    public static void main(String[] args){
        int x=10;
        int y=20;
        int z=-15;
        System.out.println("x<<2"+(x<<2));
        System.out.println("y>>2"+(y>>2));
        System.out.println("x&y"+(x&y));
    }
}
```

运行结果如下：

```
x<<240
y>>25
x&y0
```

4. 逻辑运算符

用于判断“并且”，“或者”．“除非”等逻辑关系，逻辑运算符两端一般连接值为布尔类型的关系表达式。

常见的逻辑运算符：

逻辑与：&& 并且的关系，要求所有条件都满足，即有 false 则整体为 false

逻辑或：|| 或者的关系，要求只要满足任意一个条件即可，即有一个 true 则整体为 true

逻辑非：！表示否定，取反的意思，以前为 false，现在为 true，以前为 true，现在为 false

例 9：写出运行结果

```
public class Test_9 {
    public static void main(String[] args){
        boolean b1=true;
        boolean b2=false;
        System.out.println("b1&&b2="+(b1&&b2));
        System.out.println("b1&b2="+(b1&b2));
        System.out.println("b1||b2="+(b1||b2));
        System.out.println("b1|b2="+(b1|b2));
        System.out.println("b1^b2="+(b1^b2));
        System.out.println("! b1="+(! b1));
```

```
    }
}
```

运算结果如下：

```
b1&&b2=false
b1&b2=false
b1||b2=true
b1|b2=true
b1^b2=true
!b1=false
```

5. 赋值运算符

用于给变量赋值的运算符。

常见的赋值运算符

基本赋值运算符：=给变量赋值，不是相等，==用来表示相等。

扩展赋值运算符：比如：+=，-=，*=，/=，%=，&=，^=，|=，<<=，>>=。

例如：　x+=3　　x*=3　　x%=3

等价于：x=x+3　　x=x*3　　x=x%3

例 10： 写出运行结果

```
public class Test_10 {
    public static void main(String[] args){
        int b1=2;
        int b2=4;
        System.out.println("b1+=b2="+(b1+=b2));
        System.out.println("b1-=b2="+(b1-=b2));
        System.out.println("b1*=b2="+(b1*=b2));
        System.out.println("b1/=b2="+(b1/=b2));
        System.out.println("b1%=b2="+(b1%=b2));
    }
}
```

运算结果如下：

```
b1+=b2=6
b1-=b2=2
b1*=b2=8
b1/=b2=2
b1%=b2=2
```

6. 三元运算符

又叫“三目运算符”，即由三部分组成，格式：(关系表达式)？表达式1：表达式2。

运算流程：如果关系表达式结果为true，执行表达式1，如果关系表达式结果为false，则执行表达式2

```
public class d {
    public static void main(String[] args){
    int a=10;
    int b=20;
    int max=a>b? a:b;  //如果a>b成立为true,则执行a,如果为false,则执行表达式b
        System.out.println(max);
    }
}
```

结果：1

例11：写出运行结果

```
public class Test_11 {
    public static void main(String[] args){
        int b1=2;
        int b2=4;
        int b=b1>b2? b1:b2;
        System.out.println("b1和b2中比较大的是"+b);
    }
}
```

运算结果为：

```
b1和b2中比较大的是4
```

7. instanceof

该运算符是双目运算符，左面的操作元是一个对象，右面是一个类或接口。当左面的对象是右面的类（或右边类的子孙类）创建的对象、或者是右边接口的实现类（或实现类的子孙类）的对象时，该运算符运算结果是true，否则是false。

8. 运算符优先级

Java的表达式就是用运算符连接起来的符合Java规则的式子。运算符的优先级决定了表达式中运算执行的先后顺序。例如x<y&&!z相当于（x<y）&&（！z），没有必要去记忆运算符号的优先级别，在编写程序时可尽量的使用括号来实现你想要的运算次序，以免产生难以阅读或含糊不清的计算顺序。运算符的结合性决定了并列相同级别的运算符的先后顺序。例如：加减的结合性是从左到右，8-5+3相当于（8-5）+3。具体优先级如下表

所示：

表 2-4　优先级

优先级	运算符	类	结合性
1	( )	括号运算符	由左至右
1	[ ]	方括号运算符	由左至右
2	!、+（正号）、-（符号）	一元运算符	由右至左
2	~	位逻辑运算符	由右至左
2	++、--	递增与递减运算符	由右至左
3	*、/、%	算术运算符	由左至右
4	+、-	算术运算符	由左至右
5	<<、>>	位左移、右移运算符	由左至右
6	>、>=、<、<=	关系运算符	由左至右
7	==、!=	关系运算符	由左至右
8	&（位运算符号 AND）	位逻辑运算符	由左至右
9	^（位运算符号 XOR）	位逻辑运算符	由左至右
10	|（位运算符号 OR）	位逻辑运算符	由左至右
11	&&	逻辑运算符	由左至右
12	||	逻辑运算符	由左至右
13	?：	条件运算符	由右至左
14	=	赋值运算符	由右至左

2.4　Java 中的标识符、关键字

情景导入：注释就是在做菜时加上步骤说明。标识符就是给做好的菜或原材料起的名字，比如西红柿、鸡蛋、西红柿炒鸡蛋等。关键字是事先约定好的名字，比如做菜中“炒”、“炖”、“煮”……。

一、注释

注释就是对代码的解释和说明，其目的是让人们能够更加轻松地了解代码。注释是编写程序时，写程序的人给一个语句、程序段、函数等的解释或提示，能提高程序代码的可读性。注释只是为了提高可读性，不会被计算机编译，不会影响到代码执行。

在 Java 中注释的语法：写在要解释的代码行或代码段上方或右方。一个好的程序，应该在源代码中有合适的注释。

Java 语言中定义了三种注释形式：单行注释、多行注释、文档注释。

单行注释：//解释的内容

多行注释：/* 解释的内容 */

文档注释：javadoc 注释 /** 解释的内容 */

代码规范：好的程序员写的代码不只是给机器看，还要给人看。

1. 编写合适的注释，单行注释后要写一个空格
2. 注意下一级的代码要缩进，同一级的代码要注意左对齐
3. 左大括号前不换行，左大括号后换行，右大括号独占一行，且和这行代码声明对齐
4. 类名命名注意不要出现空格、特殊符号

二、标识符

标识符主要用于为变量命名。具体命名要求如下：

- 以字母、汉字、“$”符或者下划线“_”开头
- 首字符除外，可以包含字母、汉字、“$”符或者下划线“_”和数字
- 不能与系统关键字重名。

三、Java 中关键字、保留字

Java 中有一些特定含义、并用做特殊用途的单词称为关键字（keyword）。所有关键字都是小写的。goto 和 const 虽然从未被使用，但也作为 Java 关键字保留。

Java 中一共有 51 个关键字，如下表所示：

表 2-5 表 Java 关键字

abstract	assert	boolean	break	byte	continue
case	catch	char	class	const	double
default	do	extends	else	final	float
for	goto	long	if	implements	import
native	new	null	instanceof	int	interface
package	private	protected	public	return	short
static	strictfp	Super	switch	synchronized	this
while	void	throw	throws	transient	try
volatile					

2.3 项目实践

下面结合简单的数据类型处理器来实现几种数据的“+”运算。本项目首先运行显示

“数据类型处理”选择界面。如图 2.6 所示。

```
请输入要计算的数据类型
1---整数
2---单精度
3---字符串
4---字节
5---退出
```

图 2.6　数据类型选择界面

根据所选择的数据类型，输入相应类型的数据后，显示运算结果。例如输入“1”，进行整数的计算，如图 2.7 所示。

```
1
请输入要计算的整数:
3
4
结果为：7
```

图 2.7　选择整数类型求和

输入 3 针对字符串进行“+”运算，如图 2.8 所示。

```
3
请输入要计算的字符串
Hello
World
结果为：HelloWorld
```

图 2.8　选择字符串类型，连接运算

本项目是针对各种基本数据类型的计算。该项目设计思路如下：

1. 首先屏幕显示类型选择界面。
2. 根据所选择的类型，执行相应的类型输入和处理

1. 使用输出，显示类型选择界面。

```
System.out.println("请输入要计算的数据类型");
System.out.println("1---整数");
System.out.println("2---浮点数 ");
System.out.println("3---字符串");
System.out.println("4---字节");
System.out.println("5---退出");
```

2. 使用输入，输入类型选择。

```
Scanner scan=new Scanner(System.in);
int lx=scan.nextInt();
```

3. 选择整数类型。

```
if(lx==1)
{
    int x,y,z;
    System.out.println("请输入要计算的整数:");
    x=scan.nextInt();
    y=scan.nextInt();
    z=x+y;
    System.out.println("结果为:"+z);
}
```

4. 选择浮点数类型。

```
else if(lx==2)
{
    float x,y,z;
    System.out.println("请输入要计算的单精度数:");
    x=scan.nextFloat();
    y=scan.nextFloat();
    z=x+y;
    System.out.println("结果为:"+z);
}
```

5. 选择字符串类型,进行字符串连接运算。

```
else if(lx==3)
{
    String x,y,z;
    System.out.println("请输入要计算的字符串:");
    x=scan.next();
    y=scan.next();
    z=x+y;
    System.out.println("结果为:"+z);
}
```

6. 选择字节类型。

```
else if(lx==4)
{  byte x,y;
    int z;
    System.out.println("请输入要计算的字节(注意不要超范围):");
    x=scan.nextByte();
    y=scan.nextByte();
    z=(int)x+(int)y;
    System.out.println("结果为:"+z);
}
```

7. 选择退出。

```
else if(lx==5)
    {
        System.exit(0);
    }
```

完整的程序源代码如下：

```
import java.util.Scanner;
public class JSQ {
    public static void main(String[] args) {
        System.out.println("请输入要计算的数据类型");
        System.out.println("1---整数");
        System.out.println("2---浮点数 ");
        System.out.println("3---字符串");
        System.out.println("4---字节");
        System.out.println("5---退出");
        Scanner scan=new Scanner(System.in);
        int lx=scan.nextInt();
        if(lx==1)
        {
            int x,y,z;
            System.out.println("请输入要计算的整数:");
            x=scan.nextInt();
            y=scan.nextInt();
            z=x+y;
            System.out.println("结果为:"+z);
```

```
            }
            else if(lx==2)
            {
                float x,y,z;
                System.out.println("请输入要计算的单精度数:");
                x=scan.nextFloat();
                y=scan.nextFloat();
                z=x+y;
                System.out.println("结果为:"+z);
            }
            else if(lx==3)
            {
                String x,y,z;
                System.out.println("请输入要计算的字符串:");
                x=scan.next();
                y=scan.next();
                z=x+y;
                System.out.println("结果为:"+z);
            }
            else if(lx==4)
            {
                byte x,y;
                int z;
                System.out.println("请输入要计算的字节(注意不要超范围):");
                x=scan.nextByte();
                y=scan.nextByte();
                z=(int)x+(int)y;
                System.out.println("结果为:"+z);
            }
            else if(lx==5)
            {
                System.exit(0);
            }
        }
    }
```

2.4 项目强化

题目1：写出运行结果

```
1  public class Test_11 {
2    public static void main(String[] args){
3        int x=4;
4        int y=6;
5        int i1=x++;
6        int i2=++x;
7        int i3=y--;
8        int i4=--y;
9        System.out.println("x++="+i1);
10       System.out.println("++x="+i2);
11       System.out.println("y--="+i3);
12       System.out.println("--y="+i4);
13    }
14 }
```

该程序运行结果如下：

```
x++=4
++x=6
y--=6
--y=4
```

笔记：该程序中第5行i1的值为x++的值，先用x，再加1。第6行i2的值为++x的值，先加1，再使用x的值。第7，8行执行“--”操作，同“++”。

题目2：写出运行结果

```
1  public class Test_12 {
2    public static void main(String[] args){
3      int x=19;
4      int y=2;
5      int z=15;
6      System.out.println("x<<y="+(x<<y));
7      System.out.println("z>>y="+(z>>y));
8      System.out.println("x&y="+(x&y));
```

```
9  }
10 }
```

运行结果如下：

```
x<<y=76
z>>y=3
x&y=2
```

笔记：该程序第 6 行显示（x<<y）的值，“<<”为左移操作，即把 x 的二进制表示形式左移 y 位。x 的值 19 的二进制表示形式为 10011，左移两位二进制表示形式为 1001100，转换为十进制为 76。第 7 行显示（z>>y）的值，“>>”为右移操作，把 z 的二进制表示形式右移 y 位。z 的值 15 的二进制为 1111，右移两位的二进制为 11。转换为十进制为 3。第 8 行显示（x&y）的值，把 x 和 y 的二进制形式执行按位与运算。即二进制形式（10011&00010），得二进制表示 00010，转换为十进制为 2。

2.5 精选练习

一、选择题

1. 下列变量定义错误的是（　　）。

A. int a;　　B. double b=4.5;

C. boolean b=true;　　D. float f=9.8;

2. 下列数据类型的精度由高到低的顺序是（　　）。

A. float，double，int，long　　B. double，float，int，byte

C. byte，long，double，float　　D. double，int，float，long

3. 对于一个三位的正整数 n，取出它的十位数字 k（k 为整型）的表达式是（　　）。

A. k=n/10%10　　B. k=（n-n/100*100）%10

C. k=n%10　　D. k=n/10

4. 执行下列代码后，c 的值是（　　）。

```
int a=3;
char b='5';
char c=(char)(a+b);
```

A. ‘8’　　B. 53　　C. 8　　D. 56

5. 6+5%3+2 的值是（　　）。

A. 2　　B. 1　　C. 9　　D. 10

6. 指出下列正确的语句

A. byte x1=389　　B. long x2=4.5

C. int x2=87L　　D. long x4=-20

7. 指出下列类型转换中正确的是（　　）。

A. int　x1 = ‘A’　　B. long　x2 = 8. 4f

C. int　x3 = (boolean) 8. 9　　D. int　x4 = 8. 3

8. 现有一变量声明为 boolean aa；下面赋值语句中正确的是（　　）。

A. aa = false;　　B. aa = False;

C. aa = "true"　　D. aa = 0

9. 设有定义 int　i = 6；执行语句：i += i - 1；后，i 的值为（　　）。

A. 10　　B. 121　　C. 11　　D. 100

10. 设有定义 float x = 3. 5f，y = 4. 6f，z = 5. 7f；则以下的表达式中，值为 true 的是（　　）。

A. x>y || x>z　　B. x! = y

C. z > (y+x)　　D. x<y && !(x<z)

11. 设有定义 int i = 123；long j = 456；下面赋值不正确的语句是（　　）。

A. j = i;　　B. j = (long) i;

C. i = (int) j;　　D. i = j;

12. 以下错误的字符常量是（　　）。

A. '| '　　B. '\ ''　　C. "\ n"　　D. '我'

13. 变量命名规范说法正确的是（　　）。

A. 变量由字母、下划线、数字、$ 符号随意组成；

B. 变量不能以数字作为开头；

C. A 和 a 在 java 中是同一个变量；

D. 不同类型的变量，可以起相同的名字；

14. 执行下列程序段后 b，x，y 的值分别是（　　）。

```
int x =6,y=8;
boolean b;
b=x>y &&++x == --y;
```

A. true，6，8　　B. false，7，7

C. true，7，7　　D. false，6，8

15. 执行下列程序段后，b，x，y 的值分别是（　　）。

```
int x =6,y=8;
boolean b;
b =x>y ||++x == --y;
```

A. true，6，8　　B. false，7，7

C. true，7，7　　D. false，6，8

16. Java 中，byte　数据类型的取值范围是（　　）。

A. -256　至 255　　B. -255　至 255

C. -128　至 127　　D. -127　至 128

17. 分析如下的 Java 代码段，（　　）编译没有异常和错误。

byte b=257;

boolean b=null;

float f=1. 3;

int i=12;

18. 表达式（11+3*8）/4%3 的值是（　　）。

A. 31　　B. 0　　C. 1　　D. 2

19. 请先阅读下面的代码。

```
int x=1;
int y=2;
if (x%2 == 0){
    y++;
} else {
    y--;
}
```

上面一段程序运行结束时，变量 y 的值为下列哪一项？A

A. 1　　B. 2　　C. 3　　D. switch 语句

20. 如下那些表达式是合法的？

A：int w=（int）888. 8;

B：byte x=（byte）1000L;

C：long y=（byte）100;

D：byte z=（byte）100L;

E：short s=（int）100;

F：以上都不合法。

二、填空题

1. 以下代码的输出结果是_____。

```
int i=1;
char c='a';
char d=(char)(c+i);
System.out.println(d);
```

2. 以下代码的输出结果是_____。

```
int i=2;
char c='A';
int d=(char )(c+i);
System.out.println(d);
```

3. 下面代码执行完后的输出是_____。

```
int x=3;
int y=4;
```

```
boolean b=true;
System.out.println("b="+(b==(y<x )  )  );
```

4. 设：int x=2，y=4，z=3；

表达式“x>y&&z>y”的结果是_____。

5. 程序设计中使用三种控制结构分别是___【1】___结构，___【2】___结构，___【3】___结构。

6. Java 中的实型数据按精度的不同，可分为单精度和双精度。

使用的类型符号分别是___【1】___，___【2】___。

7. 表示“n 不能被 19 整除尽”的 Java 语言表达式是_____。

8. 表示“字符变量 ch 中的内容是大写字母”的的 Java 语言表达式是_____。

9. 若定义：int a=2，b=4；表达式：++a！=b--的值是_____。

10. 若定义：int a=2，b=6；表达式（a++）+（++b）+a＊b 的值是_____。

三、读程序

1. 若运行程序是输入 13　4，下列程序的输出结果是：

```
class  T1{
  public static void main(String[] args )  {
      Scanner key=new Scanner(System.in );
      boolean  y;
      int  a,b;
      a=key.nextInt( );
      b=key.nextInt( );
      System.out.println("a/b="+ (a/b )  );
      y=a<b;
      System.out.println("y="+y  );
  }
}
```

2. 运行程序后，下列程序的输出结果是：

```
class  T2{
  public static void main(String[] args )  {
      int  a=52,b=42;
      System.out.println("a/b="+ (a/b )  );
      System.out.println("a% b="+ (a% b )  );
  }
}
```

3. 运行程序后，下列程序的输出结果是：

```
class  T3{
  public static void main(String[] args )  {
      int  a=6,b=7,c=8;
      System.out.println("a+b>=c:  "+ (a+b>=c )  );
      System.out.println("a+b==c:  "+ (a+b==c )  );
  }
}
```

四、编程序

1. 输入直角三角形的两个条直角边，求斜边的长度输出。
2. 输入一个整数，分别求出其各位、十位数输出。
3. 编写程序计算并输出半径为 5 的圆的周长，
4. 计算公式为：周长=2＊半径＊圆周率。