

第一章 Java 概述

考点一：Java 语言简介

- 1、Java 语言的前身是 Oak 语言,是美国 Sun Microsystems 公司于 1991 年推出的, 仅限于公司内部使用的语言。
- 2、Java 是一种功能强大的程序设计语言, 既是开发环境, 又是应用环境, 它代表一种新的计算模式。
- 3、Java 语言的特点:
 - (1) 语法简单,功能强大,安全可靠。
 - (2) 与平台无关。
 - (3) 解释编译两种运行方式。
 - (4) 多线程。
 - (5) 动态执行兼有丰富的 API 文档及类库。
- 4、Java 的运行系统和链接需要执行的类, 并作必要的优化后, 解释执行字节码程序。
- 5、Java 程序文件的扩展名是.java, 编译后生成的字节码文件的扩展名是.class。
- 6、Java 语言的一个重要特点是与平台的无关性, 实现这一特点的关键是使用 Java 虚拟机 (JVM)

考点二：Java 开发环境的安装与配置

- 1、JDK (Java SE Development Kit, Java 语言软件开发工具包) 是原 Sun 公司 (现已被 Oracle 公司收购) 提供的软件包, 其中含有编写和运行 Java 程序的所有工具, 包括组成 Java 环境的基本构件。
- 2、下载完毕, 找到下载文件所在的目录, 双击 idk8u131-windows-x64.exe 直接运行, 开始安装 JDK。按照安装向导进行安装即可。
- 3、安装完成后, 需要设置环境变量。环境变量设置完毕。重启计算机让这些设置生效。
- 4、在 Windows 操作系统下安装 JDK 并设置系统环境变量后, 可以在命令行窗口输入 javac 命令进行测试。
- 5、在 Windows 操作系统下安装 JDK 后, 需要正确设置系统的环境变量 path 和 CLASSPATH。

考点三：Java 程序示例

- 1、Java 程序分为两种, 一种是 Java 应用程序 (Java Application), 另一种是 Java 小应用程序 (Java Applet), 或叫 Java 小程序。
- 2、Java 程序由类构成, 含有一个 main () 方法, 称为主方法或主函数。程序是通过 Java 解释器来执行的独立程序, 可以使用命令行命令直接运行。整个程序的运行入口是 main () 方法, main () 方法执行完毕, 整个程序也即结束。
- 3、一个程序可以包含一个或多个 .java 文件。不论文件个数有多少, 其中只能有一个 main () 方法。
- 4、源文件是文本形式的文件, Java 的执行系统是不能识别的, 它必须经过编译, 生成字节码的类文件后才能运行。类文件是二进制格式的, 它有统一的格式, JVM 可以识别类文件并执行它。
- 5、编译一个程序的命令格式是:
javac 「选项」 源文件名

如果在命令行中输入 javac，则系统会显示所有选项。这些选项都是可选的。如果源文件不在当前目录，则需要在文件名的前面加上目录。

运行一个 Java 程序的命令格式是：

java 「选项」程序名[参数列表]

java 是解释器的名字，表示要运行一个由“程序名”指定的程序。程序名也就是类的名字，后面的参数列表是可选的。如果想向程序传送参数值，则可以把这些参数依次列在程序名的后面，参数个数不限。

IDE 是集成开发环境的缩写，这是一个提供给开发人员使用的程序开发环境，通常包括了代码编辑器、编译器、调试器和图形用户界面等工具。

考点四：使用 Java 核心 API 文档

- 1、JDK 文档中有许多 HTML 文件，这些是 JDK 提供的应用程序编程接口（Application Programming Interface, API）文档，可使用浏览器查看。API 是原 Sun 公司提供的使用 Java 语言开发的类集合，用来帮助程序员开发自己的类和程序。最基本的是 Java 核心 API。
- 2、核心 API 文档是按层设计的，以主页方式提供给用户。

考点五：Java 中的面向对象技术

- 1、所谓面向对象的方法学，就是使分析、设计和实现一个系统的方法尽可能地接近人们认识一个系统的方法。通常包括 3 个方面：面向对象的分析（Object-Oriented Analysis, OOA）、面向对象的设计（Object-Oriented Design, OOD）和面向对象的程序设计（Object-Oriented Programming, OOP）。
- 2、OOP 技术把问题看成是相互作用的事物的集合，也就是对象的集合。对象具有两个特性，一是状态，二是行为。状态是指对象本身的信息，行为是实现对对象的操作。
- 3、OOP 中采用了三大技术：封装、继承和多态。封装体现的特点是将对象的属性及实现细节隐藏起来；只给出如何使用的信息。将数据及对数据的操作捆绑在一起成为类，这就是封装技术。

第二章 数据和表达式

考点一：基本语法元素

1、在 Java 程序中,换行符及回车符都可以表示一行的结束,它们可被看作是空白。另外,空格键、水平定位键 (Tab) 亦是空白。为了增加程序的可读性,Java 程序的元素之间可以插入任意数量的空白,编译器会忽略掉多余的空白。

2、Java 中的 3 种注释形式:

(1) 表示 “//” 开始一直到行尾均为注释,一般用它对声明的变量,一程序的作用进行简短说明。“//” 是注释的开始,行尾表示注释结束。

(2) 可用于多行注释,“/*” 是注释的开始,“*/” 表示注释结束,“/*” 和 “*/” 之间的所有内容均是注释内容。这种注释多用来说明方法的功能、设计逻辑及基本思想等。

(3) 文档注释,以 “/**” 开头,以 “*/” 结束。一般地,在公有类定义或公有方法头的前面会使用这样的注释。

3、语包是 Java 程序的最小执行单位,程序的各语句间以分号 “;” 分隔。

4、Java 语言定义了许多关键字,关键字也称为保留字。

5、在 Java 语言中,标识符是由字母、数字、下划线 () 或美元符 (\$) 组成的字符串,其中数字不能作为标识符的开头。标识符区分大小写,长度没有限制。除以上所列几项之外,标识符中不能含有其他符号,例如+、=、*及%等,当然也不允许插入空白。在程序中,标识符可用作变量名、方法名、接口名和类名等。

6、类中声明的变量的作用域是整个类,方法中声明的变量的作用域是从该变量的声明处开始到包含该声明的语句块结束处。

考点二：基本数据类型

1、Java 的数据类型共分为两大类,一类是基本数据类型,另一类是复合数据类型。基本数据类型共有 8 种,分为 4 小类,分别是整型、浮点型、字符型和布尔型。整型和浮点型有时也合称为数值型。复合数据类型包括数组、类和接口。其中,数组是一个很特殊的概念,它是对象,而不是一个类,一般地把它归为复合数据类型。

2、Java 语言提供了 4 种整型量,对应的关键字分别是: byte、short、int 和 long。下表列出了 4 种整数类型的字节大小和可表示的范围。

表 Java 整数类型的表示范围

整数类型	整数长度	字节数	表示范围
byte	8 位	1	$-2^7 \sim 2^7 - 1$ (-128 ~ 127)
short	16 位	2	$-2^{15} \sim 2^{15} - 1$ (-32768 ~ 32767)
int	32 位	4	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
long	64 位	8	$-2^{63} \sim 2^{63} - 1$ (-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807)

3、整型量可用十进制、八进制或十六进制形式表示,以 1~9 开头的数为十进制数,以 0 开头的数为八进制数,以 0x 或 0X 开头的数为十六进制数。

- 4、Java 浮点数类型遵从标准的浮点规则。浮点数类型有两种：一种是单精度浮点数，用 float 关键字说明；另一种是双精度浮点数，用 double 关键字说明，它们都是有符号数。
- 5、单个字符用 char 类型表示。一个 char 表示一个 Unicode 字符，其值用 16 位无符号整数表示，范围为 0~65535。char 类型的常量值必须用一对单引号（'）括起来，分为普通字符常量和转义字符常量两种。
- 6、逻辑值有两个状态，它们常被写作 on 和 off、true 和 false、yes 和 no 等。在 Java 中，这样的值用 boolean（布尔）类型表示，布尔类型也称作逻辑类型。boolean 类型有两个常量值：true 和 false，它们全是小写，计算机内部使用 8 位二进制数表示。

考点三：表达式

- 1、表达式由运算符和操作数组成，对操作数进行运算符指定的操作，并得出运算结果。
- 2、常量操作数很简单，只有简单数据类型和 String 类型才有相应的常量形式。
- 3、变量是存储数据的基本单元，它可以用作表达式中的操作数。变量在使用之前要先声明。变量声明的基本格式为：
类型 变量名 1 [=初值 1][,变量名 2 [=初值 2]]...;
其中，类型是变量所属的类型，既可以是简单数据类型，如 int 和 float 等，又可以是类类型。有时也把类类型的变量称为引用。方括号中的初值是可选的。如果没有，则表明仅是声明了一个变量，否则是在声明变量的同时，给变量赋了初值,称为对变量进行了初始化。
- 4、具有 null 值的引用不指向任何对象。如果使用它指向的对象，则将导致一个异常。异常是程序运行时发生的一个错误。
- 5、变量的作用域是指可访问该变量的代码范围。类中定义的成员变量的作用域是整个类。方法中定义的局部变量的作用域是从该变量的声明处开始到包含该声明的语句块结束处，块外是不可使用的。

6、运算符：

- (1) 算术运算符。算术运算符包括通常的加 (+)、减 (-)、乘 (*)、除 (/)、取模 (%)，完成整型和浮点型数据的算术运算。许多语言中的取模运算只能用于整型数，Java 对此有所扩展，它允许对浮点数进行取模操作。
- (2) 关系运算符。关系运算符用来比较两个值,包括大于 (>)、大于等于 (>=)、小于 (<)、小于等于 (<=)、等于 (==) 和 不等于 (!=) 6 种。关系运算符都是二元运算符，也就是每个运算符都带有两个操作数。运算的结果是一个逻辑值。Java 允许 “==” 和 “!=” 两种运算用于任何数据类型。
- (3) 逻辑运算符。逻辑运算符包括逻辑与 (&&)、逻辑或 (||) 和逻辑非 (!)。前两个是二元运算符，后一个是 unary 运算符。Java 对逻辑与和逻辑或提供“短路”操作功能。
- (4) 位运算符。位运算符用来对二进制位进行操作，包括按位取反 (~)、按位与 (&) 按位或 (|)、异或 (^)、右移 (>>)、左移 (<<) 及无符号右移 (>>>)。位运算符只能对整型和字符型数据进行操作。
- (5) 其他运算符。Java 中的运算符还包括扩展赋值运算符 (++=、-=、*=、/=、%=、&=、|=、^=、>>=、<<= 和 >>>=)、条件运算符 (? :)、点运算符 (.)、实例运算符 (instanceof)、new 运算符及数组下标运算符 ([]) 等。

7、Java 是一种强类型语言，不支持变量类型间的自动任意转换，有时必须显式地进行变量类型的转换。每个数据都与特定的类型相关，允许整型、浮点型、字符型数据进行混合运算。运算时，不同类型的数据先转换为同一种类型，然后再进行运算。转换的一般原则是位数少的类型转换为位数多的类型，这称作自动类型转换。这样做的目的是保证转换时不丢失有用信息。

获取完整资料 请下载 APP 或关注公众号



扫码下载 app



扫码关注公众号