

进一步研究字面量 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_\\_E8\\_BF\\_9B\\_E4\\_B8\\_80\\_E6\\_AD\\_A5\\_E7\\_c97\\_138599.htm](https://www.100test.com/kao_ti2020/138/2021_2022__E8_BF_9B_E4_B8_80_E6_AD_A5_E7_c97_138599.htm)

### 3.7.1 整数字面量

整数可能是在典型的程序中最常用的类型。任何一个数字的值就是一个整数字面量。例如1, 2, 3和42。这些都是十进制的值，这意味着对他们的描述基于数字10。还有另外2种进制被整数字面量使用，八进制（octal，基数是8）和十六进制

（hexadecimal，基数是16）。在Java中对八进制的值通过在它的前面加一个前导0来表示。正常的十进制的数字不用前导零。这样，看起来有效的值09将从编译器产生一个错误，因为9超出了八进制的范围0~7。程序员对数字更常用的是十六进制，它清楚地与8的大小相匹配，如8, 16, 32, 和64位。通过前导的0x或0X表示一个十六进制的字面量。十六进制数的范围是0~15，这样用A~F（或a~f）来替代10~15。整数字面量产生int值，在Java中它是32位的整数值。既然Java对类型要求严格，你可能会纳闷，将一个整数字面量赋给Java的其他整数类型如byte或long而没有产生类型不匹配的错误，怎么可能呢。庆幸的是，这个问题很好解决。当一个字面量的值被赋给一个byte或short型的变量时，如果字面量的值没有超过对应类型的范围时不会产生错误，所以，一个在第2章中曾简要地提及字面量，现在已经讲述了内置的类型，让我们进一步研究它们。

### 3.7.1 整数字面量

整数可能是在典型的程序中最常用的类型。任何一个数字的值就是一个整数字面量。例如1, 2, 3和42。这些都是十进制的值，这意味着对他们的描述基于数字10。还有另外2种进制被整数字面量使用，八

进制 (octal, 基数是8) 和十六进制 (hexadecimal, 基数是16)。在Java中对八进制的值通过在它的前面加一个前导0来表示。正常的十进制的数字不用前导零。这样, 看起来有效的值09将从编译器产生一个错误, 因为9超出了八进制的范围0~7。程序员对数字更常用的是十六进制, 它清楚地与8的大小相匹配, 如8, 16, 32, 和64位。通过前导的0x或0X表示一个十六进制的字面量。十六进制数的范围是0~15, 这样用A~F (或a~f) 来替代10~15。整数字面量产生int值, 在Java中它是32位的整数值。既然Java对类型要求严格, 你可能会纳闷, 将一个整数字面量赋给Java的其他整数类型如byte或long而没有产生类型不匹配的错误, 怎么可能呢。庆幸的是, 这个问题很好解决。当一个字面量的值被赋给一个byte或short型的变量时, 如果字面量的值没有超过对应类型的范围时不会产生错误, 所以, 一个转义序列说明续表

\f 换页  
\t 水平制表符  
\b 退格

### 3.7.5 字符串字面量

Java中的字符串字面量和其他大多数语言一样将一系列字符用双引号括起来。字符串的例子如: "Hello World" "two\nlines" "\"This is in quotes\""

为字符串定义的字符转义序列和八进制/十六进制记法在字符串内的工作方法一样。关于Java字符串应注意的一件重要的事情是它们必须在同一行开始和结束。不像其他语言有换行连接转义序列。注意: 你可能知道, 在大多数其他语言中, 包括C/C++, 字符串作为字符的数组被实现。然而, 在Java中并非如此。在Java中, 字符串实际上是对象类型。在这本书的后面你将看到, 因为Java对字符串是作为对象实现的, 因此, 它有广泛的字符串处理能力, 而且功能既强又好用。

100Test  
下载频道开通, 各类考试题目直接下载。详细请访问

