

Java 零基础从入门到就业

Miels Herro

第二章——数据类型

0.1 标识 (zhi) 符

包名、类名、变量名、方法名这些都是标识符。

(只要是其名字的地方，那个名字就是标识符)

标识符定义规则：

- 1、四个可以：数字、字母 (一般是英文字母)、下划线 _、美元符号 \$；
- 2、两个不可：数字不能开头；不能用 java 中的关键字；
- 3、见到名字就明白意思 (增加可读性)；
- 4、大小写敏感；
- 5、驼峰命名 (比如：HiWorld)
 - 类名：首字母大写，其余遵循驼峰命名；
 - 方法名、变量名：首字母小写，其余遵循驼峰命名；
 - 包名：全部小写；
- 6、长度无限制，但不建议过长，因为后面调用会很麻烦。

0.2 关键字

定义：被 Java 语言赋予了特殊含义，用作专门用途的单词。

特点：Java 中所有关键字都为小写。

0.3 变量和常量

举例：丽丽在去年是 18 岁，在今年是 19 岁；

年龄从 18→19 不断变化 → “年龄” 是变量；

去年一年是 18，今年一年是 19 → “18”，“19” 是常量，并且是常量中的字面常量。

字面常量有：

整型常量是指整数，如 64269858；

实型常量指小数，如 3.1415926；

字符常量指单引号 + 单个字符，如 'a'；

字符串常量指双引号 + 字符串，如 "HelloWorld"；

逻辑常量指 true+false 两个常量，可以表示真假、正反、黑白。

0.4 变量的声明

变量本质上就是代表一个“可操作的存储空间”，空间位置是确定的，但是里面放置什么值是不确定的。我们可以通过变量名来访问“对应的存储空间”，从而操纵这个“存储空间”内存储的值。Java 是一种强类型语言，每个变量必须声明其数据类型。变量的数据类型决定了变量占据存储空间的大小。

比如 `int a=3` 就表示 `a` 变量的空间大小是 4 个字节。

变量作为程序中最基本的存储单元，其要素包括变量名、变量类型和作用域。变量在使用前必须对其声明，只有在变量声明后，才能对其分配对应长度的存储空间。

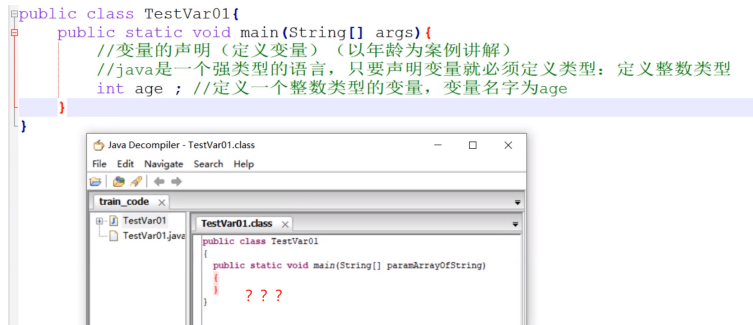
0.4.1 变量的声明、赋值、使用

```
public class TestVar01{
    public static void main(String[] args){
        int age;
        age=24;
    }
}
```

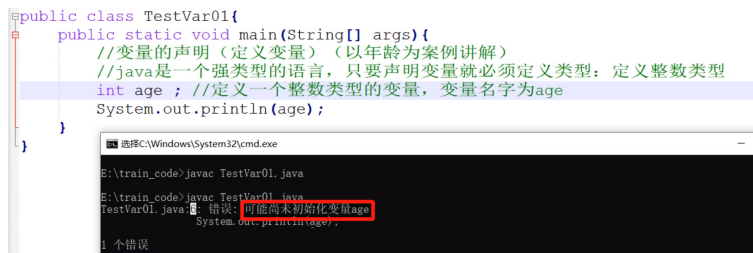
line2: `main` 方法一旦写完，程序在执行的时候就从这里开始执行，否则虚拟机不知道从何处开始；

line3: 定义一个整数类型的变量，变量名字叫 `age`。只要声明变量，就要定义类型，因为 Java 是强类型语言；此处定义整数类型，因为年龄总是 22、24 这样的整数，不会说 22.4；

反编译：如果只定义一个变量，但是没有对变量赋值的话，该变量相当于没有被定义，如下图所示；

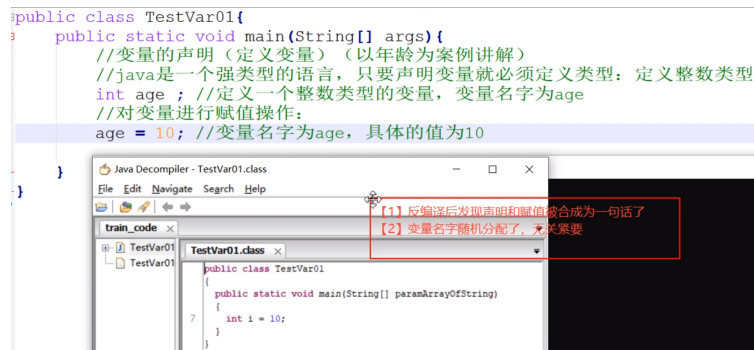


line4-(`System.out.println(age)`): 如果变量没有进行赋值的话，使用的时候会报错：尚未初始化变量，如下图所示；



line4: 变量名字为 `age`，具体的值为 24，并且变量的值可以任意改变；

0.4.2 小拓展



```

public class TestVar02{
    public static void main(String [] args){
        int a=10;
        int b=20;
        int c=a+b;
    }
}

```

操作命令：javac TestVar02.java 产生.class 字节码文件

javap -v TestVar02.class(-v 表示 verbose，查看详情，如下图)

```

descriptor: ()V
flags: ACC_PUBLIC
Code:
    stack=1, locals=1, args_size=1
    0: aload_0
    1: invokespecial #1                  // Method java/lang/Object.<init>:()V
    4: return
LineNumberTable:
    line 1: 0

public static void main(java.lang.String[]):
    descriptor: ([Ljava/lang/String;)V
    flags: ACC_PUBLIC, ACC_STATIC
    Code:
        stack=2, locals=4, args_size=1
        0: bipush        10
        2: istore_1
        3: bipush        20
        5: istore_2
        6: iload_1
        7: iload_2
        8: iadd
        9: istore_3
        10: return
    LineNumberTable:
        line 3: 0
        line 4: 3
        line 5: 6
        line 6: 10

```

字节码指令

0.5 变量的内存

内存只占用一块空间，以“int age=18”为例

0.6 变量的作用域

作用域指的就是作用范围，即变量在什么范围内有效（作用范围就是在定义变量之后，离变量最近的花括号）

```

public class TestVar04{
    public static void main(String [] args){

```

```

        int a = 10;
        System.out.println(a);
    }
    public void eat(){
        System.out.println(b);
    }
}

```

备注：line4 可以运行；line4 复制到 line3 之前就不能运行，变量一定要赋值之后才有效；

局部变量：定义在方法中的变量；

成员变量：定义在类中但是在方法外（比如在 line1 和 line2 之间插入 “int b=10;”）；

（那么在 line4 下添加一行 “System.out.println(b);” 是否可以访问呢？）

（可以的，因为 b 是成员变量，在 line1 花括号到 line6 花括号之间，任何对 b 的操作都可以访问，包括 line7 的访问也可以；但是在 line7 后加 “System.out.println(a);” 就不能访问；但是在 line7 后加 “int a=40;”，此动作不属于重复定义，所以可以对新定义的 a 进行访问）

（代码块就是指一堆花括号之间的区域，出了花括号以后就不能访问；花括号之内的变量不允许再次定义，花括号内的花括号也不可以）

0.7 基本数据类型量

Java 中的数据类型有以下几种：



除了基本数据类型以外的所有类型，都属于引用数据类型，本章重点：基本数据类型。

0.7.1 整数类型常量

整数型常量就是数字，常见为十进制，以 0 开头的八进制，以 0X 或 0x 开头的十六进制，以 0B 或者 0b 开头的二进制（电脑系统给出的计算器就可以直接进行进制之间的转换）

0.7.2 整数类型变量

例题：byte 右侧表数范围 127 怎么算出来的？

byte：1 字节 = 8 位 0/1

二进制：01111111=127(十进制)

类型	占用存储空间	表数范围
byte	1字节	$-2^7 \sim 2^7-1$ (-128~127)
short	2字节	$-2^{15} \sim 2^{15}-1$ (-32768~32767)
int	4字节	$-2^{31} \sim 2^{31}-1$ (-2147483648~2147483647) 约21亿
long	8字节	$-2^{63} \sim 2^{63}-1$

代码:

```
public class TestVar04{
    public static void main(String [] args){
        int num1=12;
        System.out.println(num1);
        int num2=012;
        System.out.println(num2);
        int num3=0x12;
        System.out.println(num3);
        int num4=0b10;
        System.out.println(num4);
        byte b=12;
        System.out.println(b);
        short s=30000;
        System.out.println(s);
    }
}
```

备注：默认情况下的赋值都是十进制的；给变量赋值的时候，可以根据不同进制进行赋值，但是最后的输出全是十进制形式。

备注：“byte b=12;”定义了一个 byte 类型的变量，名字叫 b，赋值为 12，此时可以正常输出；

“byte b=129;”属于超范围定义，会报错，因为 byte 类型最大只能到 127；

“short s=30000;”没有超范围，可以正常输出；

“int i=12345678910;”达到百亿，超过了最大范围 21 亿，不能输出结果；

“long num5=12345678910”可以实现输出；

(执行的时候还是会报错，那是因为定义变量时，整数类型默认就是 int 类型，对于 int 类型来说，它超出范围了)

(要想把一个长的整数赋值给 long 类型变量，那么在后面加上 L 或者 L 即可)

注意：只有当这个数超出了 int 类型的范围后才需要加上 L，fu'ze'wu'xu'guan

0.7.3 浮点类型常量

浮点就是对应的小数。

182.5 就是十进制下的浮点数；31415926e-7 是科学计数法下的圆周率

0.7.4 浮点类型变量

float 类型又称作是单精度类型，可以表示 6 7 位有效数字；

double 类型又称作是双精度类型，可以表示 15 16 位有效数字。

float 类型的数值后面默认带有后缀 F 或者 f，没有后缀的浮点数默认为 double 类型，也可以加后缀 D 或者 d 以明确其确实是 double 类型。

备注：有效数字是从左边第一个不为 0 的数到最后一个数。

float 4 个字节 32 位的分配是：1 符号位、2-9 指数位、其余尾数位；

“2 乘 10 的 4 次方”中 2 是尾数位、10 是底数、4 是指数。

```
public class TestVar06{
    public static void main(String[] args){
        double num1=3.14;
        System.out.println(num1);
        double num1=314E-2
        System.out.println(num2);
        float f1=3.146789416513156498465;
        System.out.println(f1);
    }
}
```

问题：编译最后是“float f1=3.146789416513156498465;”时候发生报错，显示“从 Double 转为 float 可能不兼容，会发生数据丢失”。

解析：浮点型常量默认是 double 类型的，要想将 double 类型数赋值给 float 类型，在最后添加 F 或者 f。

备注：double 类型之后可以加 D 或者 d，但是一般都省略不写。

注意：最好不要进行浮点类型的比较；“=”表示“赋值”；“==”的运算符的结果是逻辑常量，即“true”或者“false”

0.8 编码和字符集

编码：信息从一种形式变为另一种形式的过程；

字符集/编码表：编码和解码过程的根据 (有很多很多种类)；

(由权威机构形成的编码表才可以称之为字符集/编码表，如 ASCII、ISO8859-1、GB2312、GBK、Unicode)

ASCII 是英文字符集，用一个字节的 7 位表示，ISO8859-1 是西欧字符集，用一个字节的 8 位表示，GB2312 和 GBK 分别是简体中文字符集和繁体中文字符集，最多用两个字节表示中文，so 问题来了：当有两个字节分别表示英文或者西欧字符时，怎样能让计算机知道是两个字符，还是两个字节合起来是一个中文字符呢？

字节第一位如果是 0，则表示一个字节就足够表示该字符；如果第一位是 1 的话，则表示一个字节就不够表示该字符，需要和后面合起来当作一个字符来看。

Unicode 是国际通用字符集，融合了人类所使用的所有字符。为每个字符分配唯一的字符码。

Unicode 按照最多两个字节存储信息，则有 2 的 15 次方即 65532 种字符可以被存储，但是出现和上述一样的问题，如果第一位通过 0/1 进行区分一个字节还是两个字节，则可表示信息就不足以包含所有字符，于是 Unicode 在出现后的很长时间内都没有得以实现。

直到后面互联网推出了 UTF 标准，它有三种编码方案：UTF-8、UTF-16、UTF-32

Unicode符号范围(十六进制)	UTF-8编码方式
0000 0000 - 0000 007F	0xxxxxxx(兼容原来的ASCII)
0000 0080 - 0000 07FF	110xxxxx 10xxxxxx
0000 0800 - 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000 - 0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

以 UTF-8 为例进行讲解：中文“苗”的 UTF-8 码在十进制下是 33495(二进制下为 1000 0010 1101 0111，八进制下为 101 327，十六进制下为 82D7)

对照上图可以看到，十六进制下的 82D7 落在了第三行，所以底层的二进制存储就是把“苗”的二进制码塞到第三行的框架即“1110xxxx 10xxxxxx 10xxxxxx”中去，得到“1110 1000 10 001011 10 010111”

(后期对 UTF-8 标准进行了更改，现在最多可以用 6 个字节表示)

0.9 字符类型

Java 中用单引号来表示字符变量

```
public class TestVar07{
    public static void main(String[] args){
        char ch1='8';
        System.out.println(ch1);
        char ch1='a';
        System.out.println(ch2);
        char ch1='A';
        System.out.println(ch3);
        char ch1='zhongdehanzi';
        System.out.println(ch4);
        char ch1='?';
        System.out.println(ch5);
        char ch6=' ';
        System.out.println(ch6);
        System.out.println("_____");
        char ch7='\n';
        System.out.println("aaa"+ch1+"bbb");
        System.out.println("aaa_\nbbb");
        System.out.println("=====");
        System.out.println("\ Java\n");
    }
}
```

转义字符：\ 将后面的普通字符转换为特殊含义。

备注：ch6 一样也可以被输出，就算是空格，也得是单个空格；

——下的两句话可以理解为：上面是“字符串 + 字符 + 字符串”，下面整体是一个字符串，可以说明 a+a+a 成为一个字符串，\n 同样也是一个单一字符；

\t 表示“需要空的格子 + 前面的字符串 =8 位”；

\b 表示退格，实现过程为“输出 aaa，检测到 \b 后删除前一个字符，继续输出后面的 bbb”；

\r 的输出结果是 bbb，实现过程为“输出 aaa，检测到 \r 之后，把光标移动到本行开头重新输出 bbb”；

\" 表示将” 原样输出；\' 表示将’ 原样输出；\\ 表示将 \ 原样输出。

以上代码全部都可以被运行，故 Java 中无论字母数字还是中文还有符号，都是字符类型的常量，都占用两个字节 (char 是用 UTF-16 标准进行编码的)

```
public class TestVar07{
    public static void main(String[] args){
        char ch1='A';
        System.out.println(ch1);
        System.out.println(ch1+90);
        System.out.println(155-ch1);
        char ch2='zhongdehanzi';
        System.out.println(ch2);
        System.out.println(ch1+90);
        System.out.println(20103-ch1);

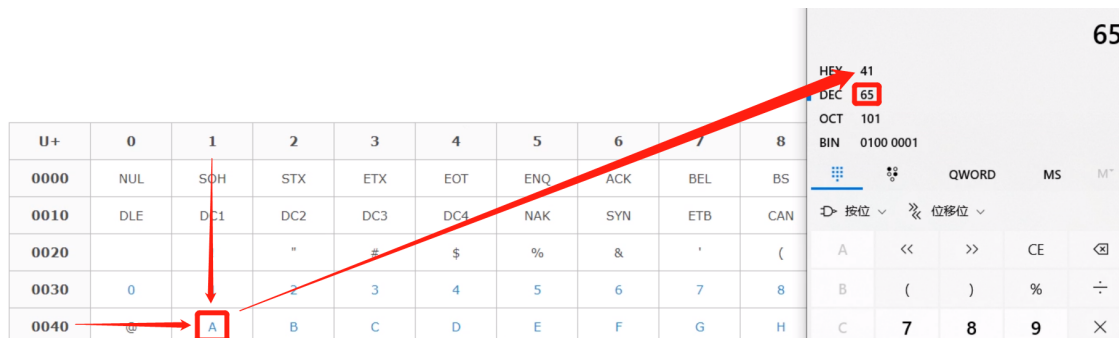
        int num=(int)ch2;
        System.out.println(num);

        int num2='zhongdehanzi';
        char ch3=20013;
    }
}
```

备注：输出结果是 A，换行后 155，换行后 90.

“A” → 我们看到的 char 类型的样子就是它的字面常量本身，但是底层在进行计算的时候，实际上是按照 ASCII 码进行计算的。

但是之前又说 char 类型是按照 Unicode 码表进行存储的，怎么解释呢？Unicode 码表是兼容 ASCII 码表的，前 128 位是一样的，具体的实现过程如下图所示 (Unicode 码表网址为：<https://www.cnblogs.com/csguo/p/7401874>).



U+	0	1	2	3	4	5	6	7	8
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS
0010	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN
0020		"	#	\$	%	&	'	(
0030	0	1	2	3	4	5	6	7	8
0040	@	A	B	C	D	E	F	G	H

HEX 41

DEC 65

OCT 101

BIN 0100 0001

QWORD MS M

按位 位移位

A << >> CE <

B () % ÷

C 7 8 9 ×

小面试题:

```
public class TestVar07{
    public static void main(String [] args){
        char ch='2'+2;
        System.out.println(ch);
    }
}
```

输出结果应该是 4, 实现过程: 2 对应的码为 50, 50 参与运算, $50+2=54$, 54 在 Unicode 码表中对应的字符为 4.

0.10 解释乱码问题

在【编码】中选择【UTF-8】后输入以下代码:

```
public class TestVar09{
    public static void main(String [] args){
        char ch1="你好☐我是苗";
        System.out.println(ch1);
    }
}
```

在命令控制台编译时会出现乱码是为什么呢?

答: 将编码格式改为 UTF-8 之后, 表示程序中的文字是用 UTF-8 进行编码的, 但是最终打在控制台编译时用的是 GBK 进行解码, 所以会解成乱码。

那么就需要将编码格式改为 GBK, 同时 ANSI 会获取当前操作系统的格式 (现在的操作系统是中文的, 使用的就是 GBK 编码格式), 所以将编码格式改为 ANSI 后就默认使用 GBK 进行编码, 最后命令行编译和执行代码都不会出错。

备注: 用记事本选择编码格式时一般要选择 ANSI→ 获取当前系统的操作格式, 一般都是 GBK

0.11 布尔类型

Boolean 类型有两个常量值, 分别是 true 和 false, 在内存中占一位, 不是一个字节 (但是不能用 0 和)

```
public class TestVar10{
    public static void main(String [] args){
        boolean flag1=true;
        System.out.println(flag1);
        boolean flag2=false;
        System.out.println(flag2);
        boolean flag3=4==8;
        System.out.println(flag3);
        boolean flag4=4<8;
        System.out.println(flag4);
    }
}
```

```
}
```

备注：输出结果为“true”

0.12 基本数据类型转换问题

输入以下代码查看输出结果：

```
public class TestVar10{
    public static void main(String[] args){
        double d=6;
        System.out.println(d);
        int i=6.5;
        System.out.println(i);
        int a=(int)6.5;
        System.out.println(a);
    }
}
```

备注：输出结果为“6.0”\n” 从 double 转换到 int 可能会有损失”\n”6”

什么是类型转换？

在赋值运算或者算术运算的时候，要求数据类型一致，就要进行类型的转换。并且转换的种类分为自动转换（如 double a=6）和强制转换（如 int i=6.5，简称为强转）。

在同一个表达式中，有多个数据类型的时候，应该如何处理：

```
double d=12+1256L+8.5F+3.14+'a'+true;
```

以上代码在编译的时候出错。多种数据类型参与从运算的时候，整数类型、浮点类型还有字符类型都可以参与运算，唯独布尔类型不能参与运算。

```
double d=12+1256L+8.5F+3.14+'a';
```

备注：类型级别：byte、short、char→int→long→float→double

级别的作用：当一个表达式中有多个数据类型的时候，要找出当前表达式中级别最高的那个类型，然后其余的类型都转换为当前表达式中最高类型进行计算。

```
故 d=12+1256L+8.5F+3.14+'a'
    =12.0+1256.0+8.5+3.14+97.0
    =3168.14
```

在进行运算的时候，左边 = 右边，直接赋值；左边 < 右边，强制转换；左边 > 右边，自动转换。

以下几种属于特殊情形：对于 byte、short、char 类型来说，只要在他们表述的范围内，赋值时不用强转（如 byte a=12 不需要强转）

0.13 习题 1——初识 Scanner

实现功能：求圆的周长和面积？

```

public class TestVar11{
    public static void main(String [] args){
        double c=2*3.14*5;
        System.out.println("周长为："+c);
        int s=3.14*5*5;
        System.out.println(s);
    }
}

```

备注：输出为“周长为：31.40000000000002”，“面积为78.5”

想要精度更加提高的话， π 的值从 3.14 变化到 3.1415925，但是为了避免多次输入数值，则在之前定义变量 `pi=3.1415926`，这样就可以一劳永逸，以后只需要改变变量的值，那接下来所有用到变量的地方，也都发生变化；但是对于 π 这种确定的数值，不想让他人修改，则需要将程序赋值一程序改为如下：

```

final double pi=3.1415926;

```

如果后面对 π 再次进行修改的话，在编译的时候就会报错。

一个变量被 `final` 修饰之后，这个变量就变成了常量，这个常量就是字符常量，其值不可以再次发生改变。（根据约定俗成的规定：字符常量的名字全部大写）

```

public class TestVar11{
    public static void main(String [] args){
        final double PI=3.1415926;
        //拿来一个扫描器：
        Scanner input=new Scanner(System.in);
        //让扫描器扫描键盘录入的int类型的数据：
        int r=input.nextInt();
        int c=2*PI*r;
        System.out.println(c);
    }
}

```

备注：以上程序编译会报错：找不到符号 `Scanner`（需要在第一行上面插入“`import java.util.Scanner;`”，可理解为在 `java.util` 下将 `Scanner` 拿过来使用）；此时编译和执行都没问题，命令控制台的光标在新的一行跳动，此时应该输入半径值并且按动回车结束；`ctrl+C` 强制结束；为了不让光标闪烁的很突然，可以再添加“`int input=sc.nextInt();`”之前添加“`System.out.println("请输入一个半径")`”；也可以把添加的改为“`System.out.print("请输入一个半径");`”；“`int input=sc.nextInt();`”表示只能输入 `int` 类型，输入其他类型会报错。

0.14 习题 2——加深对 `Scanner` 的使用

```

import java.util.Scanner;
public class TestVar11{
    public static void main(String [] args){
        //键盘录入学生的信息：年龄、身高、姓名、性别：
        //键盘录入年龄：（接收int型数据）

```

```

Scanner sc=new Scanner(System.in);
System.out.print("请录入年龄：");//友好性提示
int age=sc.nextInt();//"nextInt"中的System.out.println()
//键盘录入身高：(接收double型数据)
System.out.print("请录入身高：");//友好性提示
double height=sc.nextDouble();
//键盘录入姓名：(接收string型数据)
System.out.print("请录入姓名：");//友好性提示
String name=sc.next();
//键盘录入性别：(接收char类型)
String genderStr=sc.next();
char gender=genderStr.charAt(0);
//以上两句可合二为一：char gender=sc.next().charAt(0);
System.out.println("该学生的信息为：姓名是"+name+",
年龄是"+age+", 身高是"+height+", 性别是"+gender);
}
}

```

备注：nextInt 和 nextDouble 的"I" 和"D" 要大写。