千锋教育
www.mobiletrain.org

千锋Java学院出品

Java语言基础

Java Platform Standard Edition

良心教育，匠心品质

课程目标

CONTENTS

千锋教育
www.mobiletrain.org

ITEMS 1 变量

ITEMS 2 数据类型

ITEMS 3 运算符

ITEMS 4 类型转换

ITEMS 5 类型提升

ITEMS 6 控制台录入

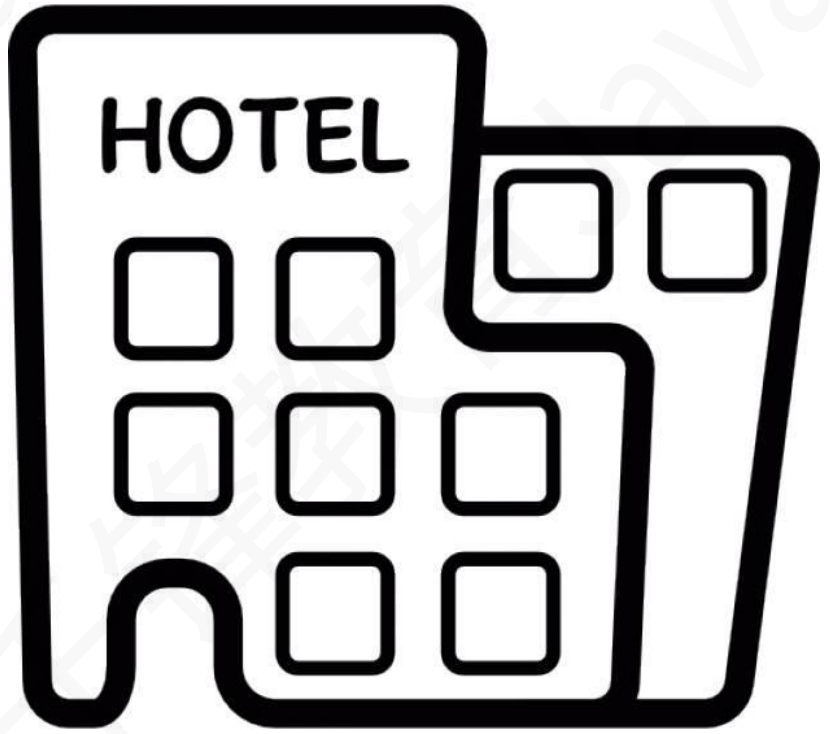
变量的概念

千锋教育
www.mobiletrain.org

- 概念：计算机内存中的一块存储空间，是存储数据的基本单元。
  - 整个内存就好像是酒店，当中包含了多个房间。
  - 房间的容量（大小）不同（单人间、两人间...）
  - 每个房间都有一个唯一的门牌号。
  - 每个房间的住客（类型）也不同。

- 酒店的房间 — 变量
  - 房间的类型 — 数据类型
  - 房间门牌号 — 变量名
  - 房间的住客 — 值

变量的组成



变量的定义流程

千锋教育
www.mobiletrain.org

- 声明：

  数据类型 变量名;

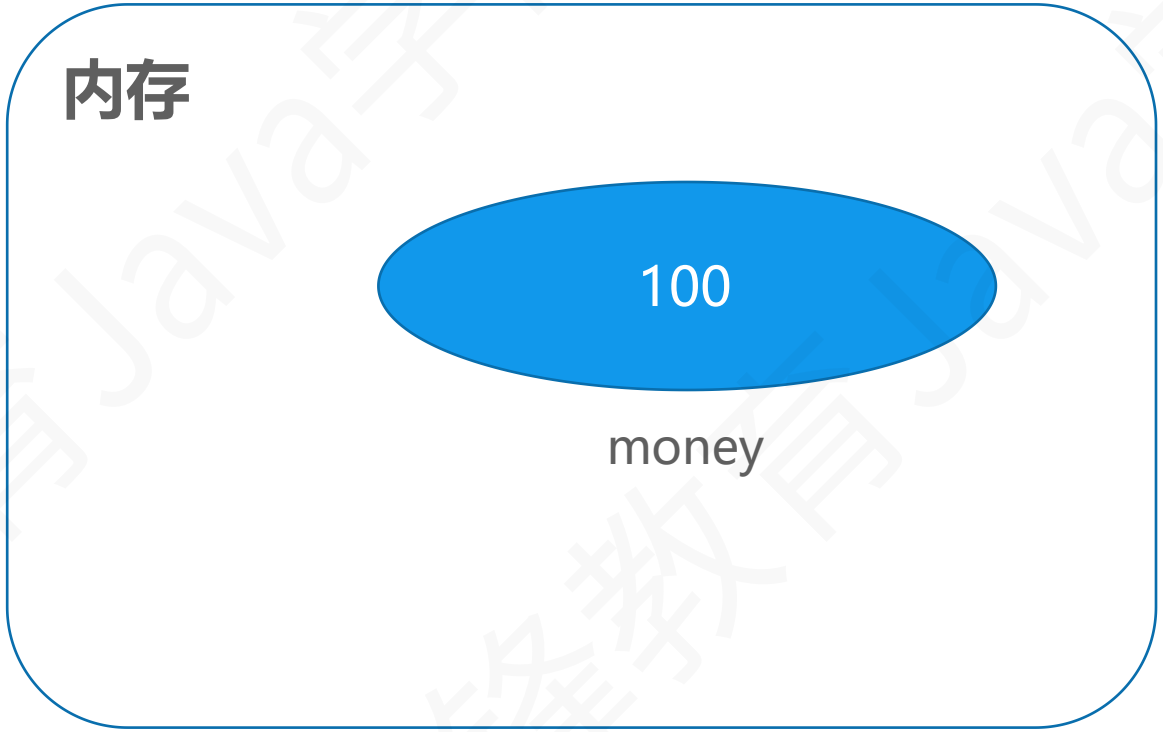
  int money; //开辟整数变量空间

- 赋值：

  变量名 = 值;

  money = 100; //将整数值赋给变量

- 应用：
  - System.out.print(money);



- 注意：Java是强类型语言，变量的类型必须与数据的类型一致。

变量的定义方式

- 声明变量的3种方式：
  - 先声明，再赋值：【常用】

    数据类型 变量名;

    变量名 = 值;

  - 声明并赋值：【常用】

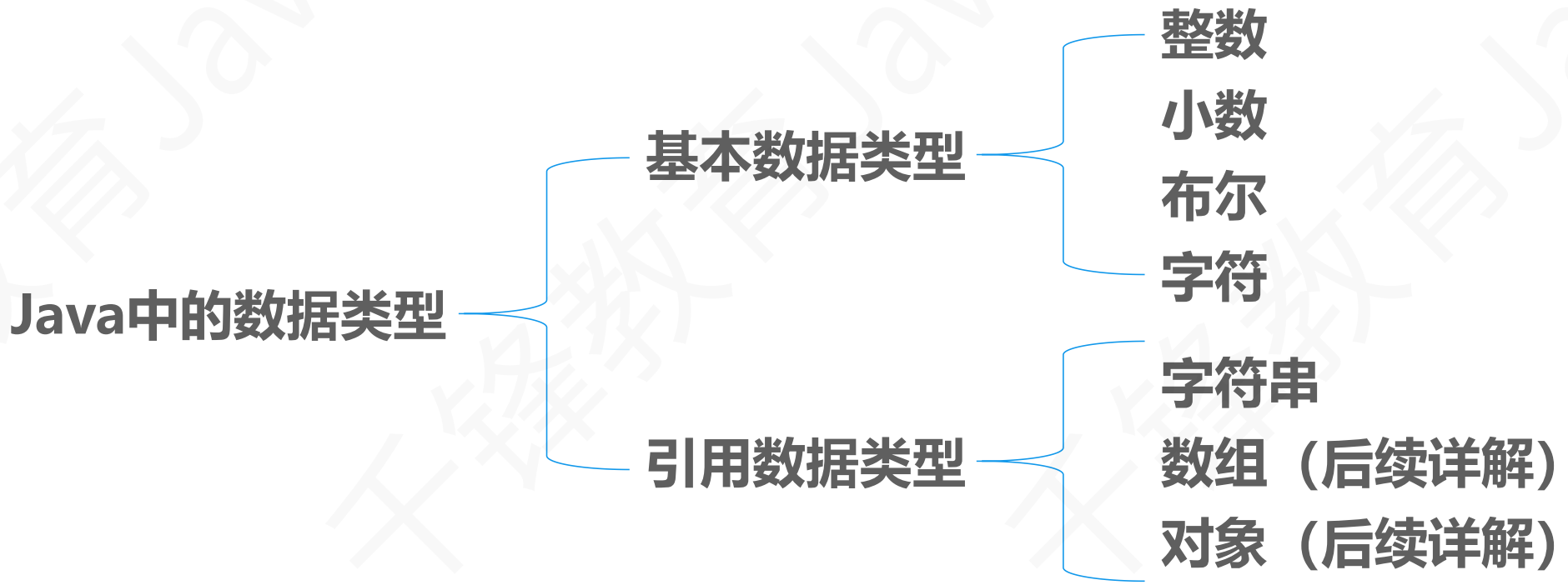
    数据类型 变量名 = 值;

  - 多个同类型变量的声明与赋值：【了解】

    数据类型 变量1，变量2，变量3 = 值3，变量4，变量5 = 值5;

数据类型

- Java中的变量具有严格的数据类型区分。（强类型语言）
- 在Java语言中，任何一个值，都有其对应类型的变量。



基本数据类型（整数）

千锋教育
www.mobiletrain.org

类型	字节	取值范围（二进制）	取值范围（十进制）
byte	1字节	-2^7 ~ 2^7-1	-128 ~ 127
short	2字节	-2^15 ~ 2^15-1	-32768 ~ 32767
int	4字节	-2^31 ~ 2^31-1	-2147483648 ~2147483647
long	8字节	-2^63 ~ 2^63-1	-9223372036854775808 ~9223372036854775807

- **注意：int为整数的默认类型，如需为long类型赋值较大整数时，需在值的后面追加“L”**

基本数据类型（小数/浮点数）

千锋教育
www.mobiletrain.org

类型	字节	负数取值范围	正数取值范围
float	4字节	-3.4E+38 ~ -1.4E-45	1.4E-45 ~ 3.4E+38
double	8字节	-1.7E+308 ~ -4.9E-324	4.9E-324 ~ 1.7E+308

- **浮点型数值采用科学计数法表示：**
- **2E3 等价于 2 * 10 ^ 3（结果：2000.0）**
- **3E5 等价于 3 * 10 ^ 5（结果：300000.0）**

  - **注意：double为浮点数的默认类型，如需为float类型赋值时，需要在值的后面追加“F”**

基本数据类型（布尔）

类型	字节	取值范围	描述
boolean	1字节	true / false	仅可描述“真”或者“假”

- 可直接赋值true / false
- 也可赋值一个结果为true / false的表达式
- 注意：Java中的boolean不能参与算数运算

基本数据类型（字符-1）

千锋教育 www.mobiletrain.org

- 前置知识：
  - ASCII(American Standard Code for Information Interchange)美国信息交换标准码。
  - 基于拉丁字母的一套电脑编码系统，主要用于显示现代英语和其他西欧语言。
  - ASCII是最通用的信息交换标准，为英文字符设定了统一并且唯一的二进制编码。

ASCII值	控制字	ASCII值	控制字	ASCII值	控制字	ASCII值	控制字	ASCII值	控制字	ASCII值	控制字	ASCII值	控制字	ASCII值	控制字
0	NUT	16	DLE	32	(space)	48	0	64	@	80	P	96	、	112	p
1	SOH	17	DCI	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	TB	39	,	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	/	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	`
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

基本数据类型（字符-2）

千锋教育 www.mobiletrain.org

- **前置知识：**
  - Unicode(万国码)是计算机科学领域里的一项业界标准，包括字符集、编码方案等。
  - 它**为每种语言中的每个字符设定了统一并且唯一的二进制编码，**
  - 以满足跨语言、跨平台进行文本转换、处理的要求，（其中包含了ASCII编码）。

U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4f00	伀	企	伂	伃	伄	伅	伆	伇	伈	伉	伊	伋	伌	伍	伎	伏
4f10	伐	休	伒	伓	伔	伕	伖	众	优	伙	会	伛	伜	伝	伞	伟
4f20	传	伡	伢	伣	伤	伥	伦	伧	伨	伩	伪	伫	伬	伭	伮	伯
4f30	估	伱	伲	伳	伴	伵	伶	伷	伸	伹	伺	伻	似	伽	伾	伿
4f40	佀	佁	佂	佃	佄	佅	但	佇	佈	佉	佊	佋	佌	位	低	住
4f50	佐	佑	佒	体	佔	何	佖	佗	佘	余	佚	佛	作	佝	佞	佟
4f60	你	佡	佢	佣	佤	佥	佦	佧	佨	佩	佪	佫	佬	佭	佮	佯
4f70	佰	佱	佲	佳	佴	併	佶	佷	佸	佹	佺	佻	佼	佽	佾	使
4f80	侀	侁	侂	侃	侄	侅	來	侇	侈	侉	侊	例	侌	侍	侎	侏
4f90	侐	侑	侒	侓	侔	侕	侖	侗	侘	侙	侚	供	侜	依	侞	侟
4fa0	侠	価	侢	侣	侤	侥	侦	侧	侨	侩	侪	侫	侬	侭	侮	侯
4fb0	侰	侱	侲	侳	侴	侵	侶	侷	侸	侹	侺	侻	侼	侽	侾	便
4fc0	俀	俁	係	促	俄	俅	俆	俇	俈	俉	俊	俋	俌	俍	俎	俏
4fd0	俐	俑	俒	俓	俔	俕	俖	俗	俘	俙	俚	俛	俜	保	俞	俟
4fe0	俠	信	俢	俣	俤	俥	俦	俧	俨	俩	俪	俫	俬	俭	修	俯
4ff0	俰	俱	俲	俳	俴	俵	俶	俷	俸	俹	俺	俻	俼	俽	俾	俿
......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......

基本数据类型（字符-3）

类型	字节	取值范围	字符编码
char	2字节	0 ~ 65535	Unicode字符集（万国码）

- Unicode中每个字符都对应一个二进制整数，可以使用多种方式赋值。
  - 字符赋值：char c1 = 'A';（通过''描述为字符赋值）
  - 整数赋值：char c2 = 65;（通过十进制数65在字符集中对应的字符赋值）
  - 进制赋值：char c3 = '\u0041';（通过十六进制数41在字符集中所对应的字符赋值）

转义字符（1）

- 如果需要在程序中输出一个单引号字符，该如何完成?

```
package demo;

public class TestChar {
    public static void main(String[] args) {
        char c = ''';
    }
}
```

编译错误：
未结束的字符文字

- 为了解决这一问题，Java采用了转义字符来表示单引号和一些特殊符号。

转义字符（2）

转义字符	描述
\n	换行符
\t	缩进（制表位）
\\	反斜线
\'	单引号
\''	双引号

引用数据类型（字符串）

千锋教育
www.mobiletrain.org

类型	取值范围	字符编码
String	任何” “之间的字面值	Unicode字符序列

- **String类型的字面取值：**
  - **String str1 = "Hello";**
  - **String str2 = "您好";**
  - **String str3 = "Java Engineer";**
  - **String str4 = "微服务架构师";**

思考

- 以下哪些赋值语句可以通过编译?
  - byte a = 128; × 超出取值范围
  - short b = 65; √
  - short c = -32000; √
  - float d = 12.34; × 浮点数默认为double类型，此时应加F或f
  - char e = '65'; × 字符只能有一个
  - char f = 65; √ 可直接赋值为整数
  - short g = 65; char h = g; × short g可能为负数，无法直接赋值给char h
  - boolean i = ''true''; × true / false 为Java保留字，直接使用
  - String j = '' 123''; √ 空格也是字符

类型转换（1）

千锋教育 www.mobiletrain.org

- 自动类型转换：
  - 两种类型相互兼容。
  - 目标类型大于源类型。

```
package demo;

public class TestAutoConvert {
    public static void main(String[] args) {
        short s = 123;
        int i = s;
    }
}
```

自动转换成功，编译通过

类型转换（2）

- 强制类型转换：
  - 两种类型相互兼容。
  - 目标类型小于源类型。

```
package demo;

public class TestForceConvert {
    public static void main(String[] args) {
        short s = 123;
        byte b = s;
    }
}
```

自动转换失败，编译错误

```
package demo;

public class TestForceConvert {
    public static void main(String[] args) {
        short s = 123;
        byte b = (byte)s;
    }
}
```

强制转换：(目标类型)值

类型转换（3）

- 强制类型转换规则：
  - 整数长度足够，数据完整。

    例：int i = 100; byte b = (byte)i;　　//b = 100
  - 整数长度不够，数据截断。

    例：int i = 10000;　　byte b = (byte)i;　　//b = 16（符号位变化，可能变为负数）
  - 小数强转整数，数据截断。

    例：double d = 2.5;　　int i = (int)d;　　//i = 2（小数位舍掉）
  - 字符整数互转，数据完整。

    例：char c = 65; int i = c;　　//i = 65
  - boolean的取值为true/false，不可与其他类型转换。

运算符

- **算数运算符：两个操作数进行计算**

操作符	描述
+	加、求和
-	减、求差
*	乘、求积
/	除、求商
%	模、求余

- **算数运算符：一元运算符（只有一个操作数）**

操作符	描述
++	递增，变量值+1
--	递减，变量值-1

运算符

- 赋值运算符：等号右边赋值给等号左边

操作符	描述
=	直接赋值
+=	求和后赋值
-=	求差后赋值
*=	求积后赋值
/=	求商后赋值
%=	求余后赋值

运算符

- **关系运算符：两个操作数进行比较**

操作符	描述
>	大于
<	小于
>=	大于等于
<=	小于等于
==	等于
!=	不等于

运算符

- **逻辑运算符：两个boolean类型的操作数或表达式进行逻辑比较**

操作符	语义	描述
&&	与（并且）	两个操作数，同时为真，结果为真
||	或（或者）	两个操作数，有一个为真，结果为真
!	非（取反）	意为“不是”，真即是假，假即是真

- **三元运算符：将判断后的结果赋值给变量**

操作符	语义	描述
？：	布尔表达式?结果1:结果2	当表达式结果为真，获得结果1 当表达式结果为假，获得结果2

表达式

- 使用运算符连接的变量或字面值，并可以得到一个最终结果。
- 例如：
  - `1 + 2;`
  - `int a = 3;`　　`a - 2;`
  - `int b = 10;`　　`int c = 20;`　　`b * c;`　　`c / b;`
  - `short d = 100;`　　`int e = 200;`　　`d > e;`　　`d <= e;`
  - ……

自动类型提升

- 进行算数运算时：
  - 两个操作数有一个为double，计算结果提升为double。
  - 如果操作数中没有double，有一个为float，计算结果提升为float。
  - 如果操作数中没有float，有一个为long，计算结果提升为long。
  - 如果操作数中没有long，有一个为int，计算结果提升为int。
  - 如果操作数中没有int，均为short或byte，计算结果仍旧提升为int。

- 特殊：任何类型与String相加（+）时，实为拼接，其结果自动提升为String。

控制台输入

- 程序运行中，可在控制台（终端）手动录入数据，再让程序继续运行。
- 导包语法：import 包名.类名;//将外部class文件的功能引入到自身文件中。
- 使用顺序：
  - 导入 java.util.Scanner。
  - 声明 Scanner 类型的变量。
  - 使用Scanner类中对应的方法（区分类型）：

```
.nextInt();          //获得整数
.nextDouble();       //获得小数
.next();             //获得字符串
.next().charAt(0);//获得单个字符
```

  - 注：如果输入了不匹配的数据，则会产生 java.util.InputMismatchException

总结

- 变量：
  - 计算机内存中的一块存储空间，是存储数据的基本单元。
- 数据类型：
  - 基本数据类型(8种)、引用数据类型(String、数组、对象)。
- 运算符：
  - 算数运算符、赋值运算符、关系运算符、逻辑运算符。
- 类型转换：
  - 自动类型转换、强制类型转换。
- 类型提升：
  - 数字间的常规类型提升，字符串的特殊类型提升。
- 控制台录入：
  - 引入工具包、声明Scanner、调用对应方法接收控制台录入数据。

THANK YOU

做真实的自己，用良心做教育

Author：Aaron
Version：9.0.2