

BASIC JAVA 课程内容总结

刘苍松

email: liucs@tarena.com.cn

liucangsong@gmail.com

Mobile: 15601035818

一、认知层次

- 1 了解、理解 (棒槌)
- 2 掌握、会 (菜鸟)
- 3 熟悉 (真正的会)
- 4 精通 (骨灰级)

二、动力定型——硬技能专家 需要 10000 小时的动力定型过程

三、Linux——山寨版 Unix

四、当前目录:

目录=文件夹

路径=系列文件夹 /home/soft02

绝对路径: 以"/" 为开头的都是绝对路径

相对路径: 与上面相反, 相对于当前目录的路径

五、文件系统 File System

(1) Windows 文件系统—— 有多个跟目录: C:\ D:\ E:\

(2) Linux 文件系统:

1、 只有一个根 / , 就象只有 C:\

2、 会把其他介质 (USB, CD) 卷映射目录

如: /media/cdrom, /media/usb0

这个目录叫“挂载点”

3、 个人主目录(Home), 用户对自己的个人主目录拥有最高管理权限。

如: /home/soft01

4、 root 跟用户，管理员用户

## 六、linux 命令

- 1、 pwd 命令，查看当前工作目录。
- 2、 cd [目录] 改变当前工作目录。
- 3、 ls 查看目录内容。
- 4、 mkdir 创建目录  
——要注意绝对路径，相对路径
- 5、 cat filename 查看文本文件内容
- 6、 cp file1 file2

练习任务：漫游以下目录：

个人主目录： /home/soft01

```
    /home
    /
    /opt
    /opt/eclipse
```

常用命令：

```
cd /opt/eclipse
pwd
ls
eclipse
./eclipse
./myeclipse
```

```
*****
*****
```

一、开发 java 的工具具有：

(1) Eclipse （日食，月食）

可以从 [www.eclipse.org](http://www.eclipse.org) 下载

IBM 和 SUN 公司各自拥有一个产品，即 eclipse 和 myeclipse。

(2) Console 控制台

(3) SUN NetBeans

二、关于 Eclipse

(1) IBM 提供的开源软件，十分主流。

(2) 工作空间：是一个目录，是开发结果默认的保存目录。

如： /home/soft01/workspace

Java 通过 JVM 实现了跨平台技术

Java 源代码 (\*.java) 经过 Java 的编译器编译成

Java 字节码 (\*.class), 执行 Java 字节码, Java 字节码

经过 JVM 的解释为具体平台的具体指令, 并执行。

不同平台有不同的 JVM, 主流平台都提供了 JVM。所以 JAVA 字节码可以在所有平台上能够解释执行。在这个意义上 Java 是跨平台的。也就是说: Java 的字节码是跨平台的。

### (3) 平台的创建和配置

JDK---java development kit;

JRE---java runtime environment;

JVM---java virtual machines;

### (4) 一个工程的结构:

project

```
|--src
|  |--basic.day01
|  |          |-- HelloWorld.java
|  |--basic.day02
|  |          |-- JFrameHelloWorld.java
|--bin
|  |--basic.day01
|  |          |-- HelloWorld.class
|  |--basic.day02
|  |          |-- JFrameHelloWorld.class
```

### (5) Java JDK (java development kit)

JDK java 开发 工具包 /opt/jdk or /opt/jdk\_1.6.01

|-- bin 开发工具(命令: java,javac,jar,javad,jar,rmic,jps...)

|-- jre JRE Java 运行环境

    |-- lib (rt.jar) Java API(应用编程接口, 系统类库)

    |    (java 提供的现成组件)

    |-- bin (java\_vm)JVM Java 虚拟机

### (6) Java 的注释, 被 Java 忽略的部分

1 单行注释 // ...

2 多行注释 /\* \*/

3 文档注释 /\*\* \*/

预习: java 变量类型, 运算符等

\*\*\*\*\*  
\*\*\*\*\*

## 一、变量

1、他 她 我 你 某人 佚名 旺财 X-man x = 1

您好! 它

2、int age = 15;// 00000000 00000000 00000000 00001111

### 3、Java 中的变量

a—— Java 是强类型语言，

b—— 变量必须声明，并且初始化以后使用

c—— 变量必须有明确的类型

d—— 变量不能重复定义

### 4、变量的作用域

a—— 在声明的地方开始，到块结束为止

b—— 离开变量的作用域结束，变量将回收。

## 二、二进制

### 1、计算机内部只有二进制数据!

int i = 15; char c = 'A';

'A' -> 00000000 01000001

15 -> 00000000 00000000 00000000 00001111

$$115 = 1*100 + 1*10 + 5*1$$

$$= 1 \times 10^2 + 1 \times 10^1 + 5 \times 10^0$$

基数: 10

权: 基数<sup>n</sup>

基数:2

权: 128 64 32 16 8 4 2 1

### 2、二进制: 1111 = 1\*2<sup>3</sup> + 1\*2<sup>2</sup> + 1\*2<sup>1</sup> + 1\*2<sup>0</sup>

$$= 1*8 + 1*4 + 1*2 + 1*1$$

$$= 15(10)$$

8421

1001 = 9

$$01000001 = 1*64 + 1*1$$

$$= 65(10)$$

128 64 32 16 8 4 2 1  
1 1 0 0 0 0 0 0 = 192(10)

3、十进制:  $65 = 6 \times 10 + 5 \times 1$   
 $= 6 \times 10^1 + 5 \times 10^0$

256 128 64 32 16 8 4 2 1  
95(10)= 0 1 0 1 1 1 1 1 (2)

```
int age = 15;  
age = 00000000 00000000 00000000 00001111  
int age = 0xf;
```

#### 4、16进制

0 1 2 3 4 5 6 7 8 9 a b c d e f  
10 11 12 13 14 15

$41(16) = 4 \times 16 + 1 = 65(10)$   
4 1

0100 0001(2) = 65(10)

\* 将16进制作作为2进制的简写形式.

4e2d(16)=0100 1110 0010 1101(2)

#### 5、八进制是二进制的简写,3位对应转换

练习:

243(10)-> 1111 0011(2) -> f3(16)

\*\*\*\*\*  
\*\*\*\*\*

### 一、Java的数据类型

1、Java数据分为基本类型和引用类型. Person someone

2、基本类型分为八种:

byte short int long float

double char boolean

a—— 整数类型 byte short int long

整数都是有符号整数

```

byte 8bit -128 ~ +127
short 16bit -32768 ~ 32767
int 32bit -2G ~ 2G-1 -2^31 ~ 2^31-1
long 64bit -2^63 ~ 2^63-1

```

```
int a = 5;
```

建议使用 int, 很少使用 short, 使用 int 要注意取值范围, int 的值不大!

b——直接量-字面量: 直接给出的常量值如: 1

- (1) 整数的直接量的类型默认是 int 类型
- (2) 以 L, l 为结尾的字面量是 long 类型

1024

1024L

(3) 可以使用整数直接量给 byte, short 类型赋值, 但是不要超过 byte 和 short 的范围.

c——浮点数: 就是小数, float, double

34642323.44 = 3.464232344 \* 10^7  
尾数                      指数

10001000.(2)

1.0001000 \* 2^11(2)

尾数              指数

- (1) float 32 位

0      00000111 100010000000000000000000

符号位      指数                      尾数

```
int a = 0x7fffffff;
```

```
float f = a;
```

- (2) double

1 符号位

11 指数

52 尾数

double 是 64 位, float 32 位, double 比 float 精确

- (3) 默认的浮点数字面量是: double 类型的  
字面量后缀: d/D f/F

```
double d = 1d;//1.0
```

1 LL 1D 1F

(4) 只使用 double 计算浮点数,float 类型基本不用!  
浮点数不能精确运算

```
int:10001000 -> float:1.0001000*2111
```

```
0 00000111 10001000 00000000 00000000
```

3、 字符类型:char, 字符类型是一个 16 位无符号整数!

a—— 字符类型的值是对应字符的编码, 是 unicode 编码. Java 支持国际化.  
英文部分的值与 ascII 编码一致.

char 是定长编码, 所以的字符都是 16 位.

```
'A' -> 0x0041
```

```
'中' -> 0x4e2d
```

在编码中: '0'~'9', 'a'~'z', 'A'~'Z' 都是连续编码的!

```
'0' == '\u0000' '\u0000' == 0
```

b—— 最小值:0, 最大值:65535 = 2<sup>16</sup>-1

c—— 字符字面量使用单引号为定界符号:'中'

字面量也是整数常量!

特殊字符采用转义字符表示:

```
如: '\n' '\t' '\\ ' '\b' '\r'
```

```
'\ ' '\" ' '\u4e2d'
```

```
char c = '\\';
```

4、 布尔类型: boolean

字面量: true false

```
float f = (float)2.6D;
```

```
char c = 65;
```

二、数据类型转换:

1、 自动类型转换(隐式类型转换)

从小类型到大类型, 自动完成

```
如: int n = 'A';
```

```
long l = 'A';
```

```
char ->
```

```
byte -> short -> int -> long -> float -> double
```

2、 强制类型转换, 从大类型到小类型需要强制转换

强制类型转换有风险, 会造成精度损失或者溢出

如:

```
long l = 1024L*1024*1024*4;//0x100000000
int i = (int)l;//0

double pi = 3.1415926535897932384626;
float f = (float)pi;
```

### 三、Java 中的方法

1、方法就是函数:  $y=f(x)=3x+6$ ;

2、方法的语法

```
(修饰词)(返回值类型)(方法名)(参数列表){
    //方法体
}
```

3、方法需要有 return 语句, 返回和返回值类型兼容的数据

4、方法的参数和方法内部的变量 都是局部变量, 这些变量作用域只在方法内部!

5、实参: 方法调用的参数

形参: 方法的参数定义,是临时变量.

作业: 1 复习并且完成 全部 课堂演示代码, 添加适当的输出语句, 研究输出结果, 添加适当的注释.

2 尝试 计算 10 个以上 255 以内的整数 2 进制,16 进制,相互转换. 并且编程验证计算结果

3 定义方法: 参数是一个整数类型, 没有返回值,在方法中输出这个整数的 10 进制,2 进制,16 进制字符串

预习: Java 运算符, 流程控制语句.

```
*****
*****
```

### 一、Java 运算符

1、数学运算符: + - \* / %

a——运算的封闭性:

- 1) 同种类型参与运算
- 2) 返回同种类型(可能发生:上溢出和下溢出)
- 3) byte, short, char 按照 int 运算

b——Java 中字面量的运算被 Javac 优化了, 优化为一个固定的常量.

```
int d1 = 2;
double d2 = 3;
double dx = d1/d2;//0.666666
```

```

int i1 = 2;
int i2 = 3;
int ix = i1/i2;//0 (下溢出)

char c = 'A'+1+1+1;//68
char c2 = (char)('A' + n);

int i = 1024*1024*1024*4;//(上溢出)
long l = 1024*1024*1024*4;//(上溢出)
long ll = 1024L*1024*1024*4;

```

c—— % 取余数运算

0%5 -> 0, 2%5->2, 5%5->0, -1%5->-1

```

----->
n      -6-5-4-3-2-1 0 1 2 3 4 5 6 7 8 9 10
n%5    -6 0-4-3-2-1 0 1 2 3 4 0 1 2 3 4 0

```

d——字符串的连接运算: +

任何数据与字符串连接(+) 都会生成新字符串

","+2 -> ",2"

''+2 -> ?整数

2、 关系运算符 && || ! | &

1) boolean b = '\u0000'==0;//true

```

int age = 12;
char sex = '男';
boolean isChild = age<16;
boolean isMan = sex=='男';
boolean isBoy = isChild && isMan;
boolean isGirl = isChild && ! isMan;
boolean isKid = isBoy || isGirl;

```

2) && || 是短路的逻辑运算.

& | 是非短路运算

大多数使用 短路运算.

3、 自增运算: ++, --

- a—— i++, 后++, 先将 i 的值作为整个表达的值, 然后将 i 增加 1.
- b—— ++i, 先++, 先将 i 增加 1, 然后将 i 的值作为整个表达的值.

#### 4、 分支流程控制语句

```
1) if(布尔表达式){
    //...
}else{
    //...
}
2) While 循环控制
while(布尔表达式){

}
```

```
while(true){
    break;
}
```

作业:

1、 复习并且完成 全部 课堂演示代码, 添加适当的输出语句, 研究输出结果, 添加适当的注释.

2、 (选做)案例练习 输出等额本息还款计划表

如(示例):

月份	本金	利息	归还
第1月	930	712	1642
第2月	940	701	1642
.....			

预习: 流程控制语句 do...while, Java 数组.

\*\*\*\*\*  
\*\*\*\*\*

#### 一、 Switch case 语句

```
switch(整数表达式){
    case 整数(int)常量:
        // ...
```

```
        break;
    case 整数常量:
        // ...
        break;
    default:
        //...
}
```

例如:

```
Scanner console = new Scanner(System.in);
char sex = console.nextLine().charAt(0);
switch(sex){
    case '男':
        System.out.println("先生您好");
        break;
    case '女':
        System.out.println("女士您好");
        break;
    default:
        System.out.println("有某搞错");
        break;
}
```

## 二、循环控制

```
for(初始化表达式;布尔表达式; 递增表达式){
    //循环体
}
```

do ... while() 循环控制

```
do{
    //循环体
}while(布尔表达式);
```

## 三、break continue

- 1) break 结束本次循环
- 2) continue 继续当前循环

## 四、原码与补码

原码	二进制	十进制		二进制	十进制
	0000	0		1000	8
	0001	1		1001	9
	0010	2		1010	10
	0011	3		1011	11
	0100	4		1100	12
	0101	5		1101	13
	0110	6		1110	14
	0111	7		1111	15

补码	二进制	十进制		二进制	十进制
	1000	-8		0000	0
	1001	-7		0001	1
	1010	-6		0010	2
	1011	-5		0011	3
	1100	-4		0100	4
	1101	-3		0101	5
	1110	-2		0110	6
	1111	-1		0111	7

```

1110  -2
+0010  2
0000  0

```

```

int max = 0x7fffffff;
byte b = (byte)max;

```

## 五、 数组

a—— 数组的声明:

```
int[] ary;
```

```
int ary[]; //不推荐的写法!
```

b—— 数组的初始化:

- 1) 动态初始化: 必须指明数组的长度

```
int[] ary = new int[10];  
int[] ary = new int[]{3,1,4,1,5,9};
```

- 2) 静态初始化: 只能在声明数组时候使用!

```
int[] ary = {3,1,4,1,5,9};  
//ary = {2,7,1,8};//编译错误  
ary = new int[]{2,7,1,8};
```

- 3) 数组成员自动初始化为"零"值.

"零"值: 0, 0.0, '\u0000', false, null

c—— 数组的访问

- 1) 使用 `ary.length` 属性可以获取数组的长度
- 2) 数组的下标(index)从 0 开始, 最大是 `length-1` 一共 `length` 个元素.

- 3) 可以使用下标访问指定的元素:

```
int a = ary[2];//读取  
ary[2] = 8;//写入
```

- 4) 如果越界访问会出现异常(错误)

`ArrayIndexOutOfBoundsException`

区别: 数组变量 是引用数组的代词  
数组实例 是具体的数组个体

```
int[] a = new int[3]{2,3,5};  
//复制  
a = new int[6];  
//复制  
a = new int[9];
```

## 六、 数组的扩展问题

- 1) Java 数组的大小是固定的
- 2) 利用数组复制到方法可以变通的实现数组扩展
- 3) `System.arraycopy()` 可以复制数组
- 4) `Arrays.copyOf()` 可以简便的创建数组副本
- 5) 创建数组副本的同时将数组长度增加就变通的实现了数组的扩展

案例 1 : 统计一个字符在字符串中的所有位置.

字符串: 统计一个字符在字符串中的所有位置

字符: '字'

返回: {4,7}

案例2 :实现文字的竖排输出

字符串: 统计一个字符在字符串中的所有位置

折行文字数量: 4

- 。 所符字统
- ~有串符计
- ~位中在一
- ~置的字个

提示: `chs[row*cols+col]=str.charAt(i++)`

作业:

- 1 复习并且完成 全部 课堂演示代码, 添加适当的输出语句, 研究输出结果, 添加适当的注释.
- 2 实现案例:

案例1 : 统计一个字符在字符串中的所有位置.

字符串: 统计一个字符在字符串中的所有位置

字符: '字'

返回: {4,7}

```
char key = '字';
String str = "统计一个字符在字符串中的所有位置";
int[] count = {};
for(int i=0; i<str.length();i++){
    char c = str.charAt(i);
    if(c==key){
        //扩展数组,
        //添加序号 i
    }
}
```

案例2 : 统计一个字符在字符数组中的所有位置.

字符数组: 统计一个字符在字符串中的所有位置

字符: '字'

返回: {4,7}

案例3: 字符串统计案例:

字符串: "123456789012"

将字符串的奇数位数字相加为  $c1 = 1 + 3 + 5 + 7 + 9 + 1$

将字符串的偶数位数字相加为  $c2 = 2 + 4 + 6 + 8 + 0 + 2$

案例 4: (选做) 将一个整数数位翻转

如: 整数 56123

返回结果为整数: 32165

提示: 使用  $\%10$  获取最后一位

使用  $/10$  去除处理完的最后一位

预习: 数组的排序.

```
*****  
*****
```

/\* 条码应用

694821332904 1

\* 前 12 位的奇数位的和  $c1: 6 + 0 + 2 + 4 + 6 + 8 = 26$

前 12 位的偶数位和  $c2: 9 + 1 + 3 + 5 + 7 + 9 = 34$

将奇数和与偶数和的三倍相加  $c1+c2*3: 26 + 34 * 3 = 128$

取结果的个位数: 128 的个位数为 8 ( $\%10$ )

用 10 减去这个个位数:  $10 - 8 = 2$

所以校验码为 2

( 注: 如果取结果的个位数为 0, 那么校验码不是为 10 ( $10 - 0 = 10$ ), 而是 0。 )

\*/

## 一、数组排序

### 1) 选择排序

原理:a 将数组中的每个元素,与第一个元素比较,

如果这个元素小于第一个元素,就将这个两个元素交换.

b 每轮使用 a 的规则,可以选择出一个最小元素放到第一个位置.

c 经过  $n-1$  轮比较完成排序

简单说: 每轮选择最小的放到前面.

### 2) 冒泡排序

原理: a 逐一比较数组中相邻的两个元素,如果后面的数字小于前面的数字,就交换先后

元素.

b 经过一个轮次的比较, 一定有一个最大的排在最后的位置.

c 每次比较剩下的元素, 经过  $n-1$  次比较, 可以实现排序

简单说: 比较交换相邻元素, 每次最大的漂移到最后

### 3) 插入排序

原理: a 将数组分为两部分, 将后部分的第一张逐一与前部分每一张比较, 如果当前元素小, 就一点被比较元素.

b 找到合理位置插入.

### 4) 方法的递归调用

案例:  $1+2+3+\dots+n=f(n)=n+f(n-1)$  && ( $n>0$ )

解决问题简练, 性能很差。一定给出结束条件。

作业:

1 复习并且实现全部案例, 找出全部的疑问, 并解决。

2 实现递归代码:

```
//案例:  $n!=1*2*3*\dots*n=f(n)$   
//       $=n*f(n-1)$  &&  $f(1)=1$ ;
```

```
//案例:  $1+3+5+\dots+n=f(n)$   
//       $=n+f(n-2)$  &&  $f(1)=1$ ;
```

```
//案例:  $1/1+1/2+1/3+\dots+1/n=f(n)$   
//       $=1/n+f(n-1)$  &&  $f(1)=1$ ;
```

### 3 案例

实现随机生成双色球号码: [02 22 13 16 18 12] [12]

红球 33 个球 (01~33) 取 六

蓝球 16 个球 (01~16) 取 一

提示:

蓝球池 {"01", "02", "03", "04", ... "16"}

红球池 {"01", "02", "03", "04", ... "33"}

使用标记{ f, f, f, f, ... f}

结果采用一个数组存储, 数组可以利用数组扩容追加新的"球号"  
处理逻辑参考如下过程:

#### 1 随机生成红球序号

- 2 检查"红球序号"是否使用过(取出过) , 如果使用过 返回 1
- 3 取出一个红球, 设置使用标记为 true
- 4 是否取出了 6 个红球, 如果没有到 6 个, 返回 1
- 5 对红球结果排序
- 6 取出一个篮球到结果中

4 案例: 生成 4 位网站验证码, 使用字符数组扩容实现

- 1) 不能重复
- 2) 数字和字符, 但是不能包含 1,0,o,0,1,L,Z,2,9,g 等

5 身份证第 18 位算法 身份证第 18 位 (校验码) 的计算方法

34052419800101001X。

1、将前面的身份证号码 17 位数分别乘以不同的系数。

从第一位到第十七位的系数分别为:

7-9-10-5-8-4-2-1-6-3-7-9-10-5-8-4-2。

2、将这 17 位数字和系数相乘的结果相加。

3、用加出来和除以 11, 看余数是多少?

4、余数只可能有 0-1-2-3-4-5-6-7-8-9-10 这 11 个数字。

其分别对应的最后一位身份证的号码为 1-0-X-9-8-7-6-5-4-3-2。

例如: 某男性的身份证号码是 34052419800101001X。

我们要看看这个身份证是不是合法的身份证。

首先我们得出前 17 位的乘积和是 189, 然后用 189 除以 11 得出的余数是 2, 最后通过对对应规则就可以知道余数 2 对应的数字是 x。

所以, 可以判定这是一个合格的身份证号码。

java 书写规范

类名 首字母大写 类名不要用中文

方法名 开头字母小写

对课程内容的全体呈现, 如有错误, 请多指正!