## 初识JAVA

1. java由sum公司开发的一门编程语言，现在被oracle公司收购。
2. Java的跨平台：只需要下载相应版本的JDK。就可以直接运行java程序。不需要针对每个平台开发一套程序。
3. Java的注释有三种：

（1）单行注释 // （2）多行注释 /\* \*/ （3）文档注释/\*\* \*/

（4）打印：

\* System.out.println() 打印并换行。

\* System.out.print() 打印，但不换行

\* System.out.print() 使用\n转义符， 让其能换行打印

\* 使用\t，让光标移动到下一个水平制表位置。

1. java编译运行过程：

程序员编写的java源程序（xxx.java），经过javac命令编译为字节码文件（.xxx。.class）。并在虚拟机上运行

## 变量

1、 变量的命名

\* 在java中，标识符用来为程序的常量，变量，方法，类，接口和包命名

\* 标识符规则：

\* 1、由字母，数字，下划线(\_),美元符号($)组成。

\* 2、不能以数字开头

\* 3、不能使用关键字,布尔值(true,false),null 命名

\* 4、区分大小写，没有长度限制

\* 5、可以写中文，但不建议。

\* 建议：

\* 1、见名知意。

\* 2、驼峰命名法：多个单词连接一起，第一个单词首字母小写，其余的单词首字母大写（fileName）

1. java的八大基本类型：

取值范围 大小

boolean true /false 1个字节

byte -2的7次方~2的7次方-1 1个字节

short -2的15次方~2的15次方-1 2个字节

int -2的31次方~2的31次方-1 4个字节

long -2的63次方~2的63次方-1 8个字节

char 0~2的16次方 2个字节

double -1.7E的38次方~1.7 E的38次方 4个字节

float -3.4 E的308次方~3.4 E的308次方 8个字节

3变量的命名和赋值规制：

（1）先声明变量，后赋值 （2）声明变量同时赋值

4long类型的数值，应该在后面加L，float类型的值，应该在后面加F，char类型的是要加单引号；

5 转义字符：\* \n:换行

\* \t:相当于Tab键

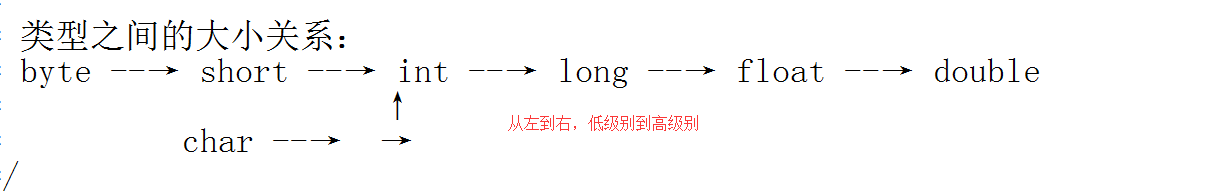
\* \b:相当于BackSpace键

\* \\:反斜杠 \

\* \' :表示单引号

\* \" :表示双引号

1. 基本类型转换：自动转换，强制转换



7 byte,short,char之间发生运算时，会统一全部变成int在参与运算。

8 算术运算符：

+ - \* / %(取模) ++(自增1) --(自减1)

注意：对于除法运算符"/",如果两个操作数为整数，结果会舍弃小数部分而保留整数部分。

如果操作数中有一个是浮点数，将会自动类型转换，结果会保留小数部分。

## 常用运算符

1 赋值运算：+= -= \*= /= %=

2 算术运算：+ - \* / % ++ --

3 逻运算：& | ^ ! && ||

4 条件运算： 条件？表达式1：表达式2

## 选择结构与循环结构

#### 选择结构

### 单分支

语法： if(条件){

代码块

}

先判断条件是否成立，成立则执行代码块。不成立就不执行代码块

### 双分支

if的双分支结构：

\* 语法:

\* if(条件){

\* 代码块1

\* }else{

\* 代码2

\* }

\*

\* 当条件成立，执行代码块1；如果条件不成立，则执行代码块2

\* 注意:else子句不能单独使用，它是if语句的一部分。

### 多分支

\* 语法:

\* if(条件1){

\* 代码块1

\* }else if(条件2){

\* 代码块2

\* }else if(条件3){

\* 代码块3;

\* }else if(....){

\* .....

\* }else{

\* 代码块n

\* }

\* 首先对if语句的条件1判断，为true，则执行代码块1。之后分支结构结束。

\* 若为false，则判断条件2，为true则执行代码块2。之后分支结束。

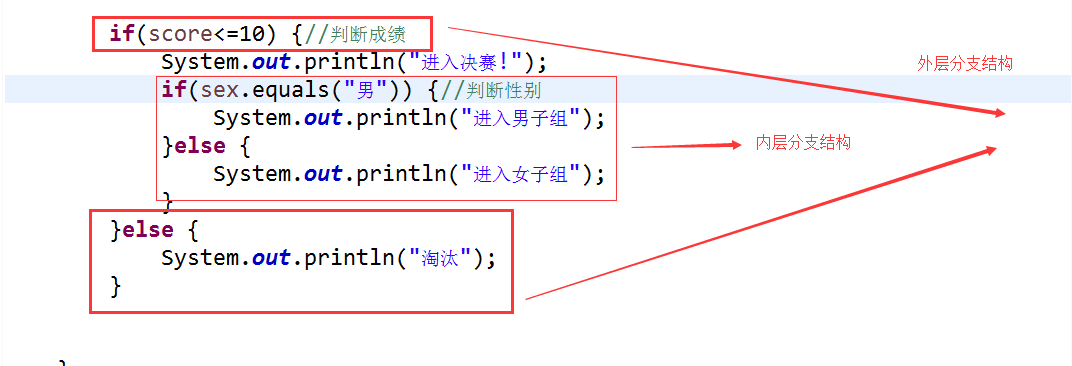
\* 依次如此...

\* 最后，如果前面的条件都不成立，则执行else的代码块

\*

\* 注意:多分支if结构中，各个条件的顺序是不能任意排列的。

### 分支嵌套



### Switch

Switch(表达式){

case 常量 1:

语句;

break;(注：非必输项)

case 常量 2:

语句;

break;

…

default:(注：非必选项)

语句;

}

常见错误：

1. 如果每个case执行完后都需要跳出来，则不要忘记写break。
2. case后面的常量值不能重复。
3. default的位置顺序可以变动，但要符合逻辑顺序

Switch结构只能处理等值判断条件。针对于判断条件处在一个值得区间，这使用if结构或者三目运算符

### 循环结构

## Whle

\* while(循环条件，结果是boolean){

\* 循环体

\* }

\* 先判断循环条件，如果为ture，则执行循环体。

\* 如果为false，结束循环

## do\_while

\* while(循环条件，结果是boolean){

\* 循环体

\* }

\* 先判断循环条件，如果为ture，则执行循环体。

\* 如果为false，结束循环

While与do\_while的区别在于，do\_while不管循环条件是否成立，至少有一次执行循环体。

## For

\* for(表达式1;表达式2;表达式3){

\* 循环体

\* }

\*

\* 首先执行表达1，一般进行变量的初始化。

\* 然后执行表达式2，判断循环条件。

\* 如果为true，则走循环体。

\* 循环体执行完毕后，执行表达式3，改变循环变量的值。

\*

\* 再次执行表达式2，判断循环条件.....后续依次如此。

1循环次数确定用for

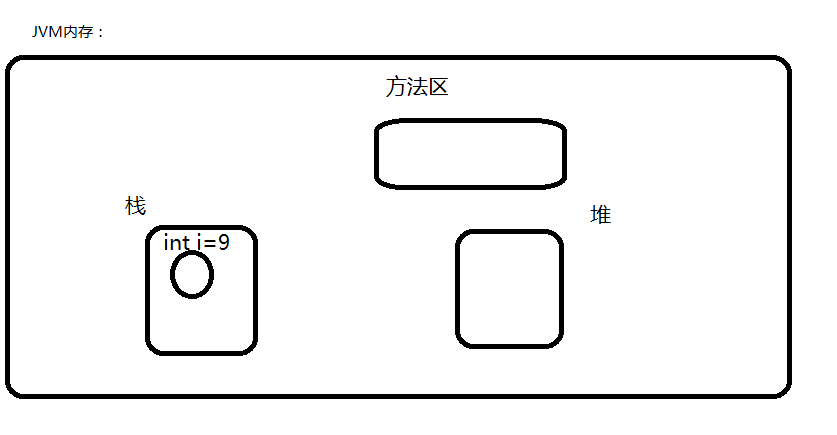
不确定用while或do while

2 Jvm内存分区：栈，堆，方法区！

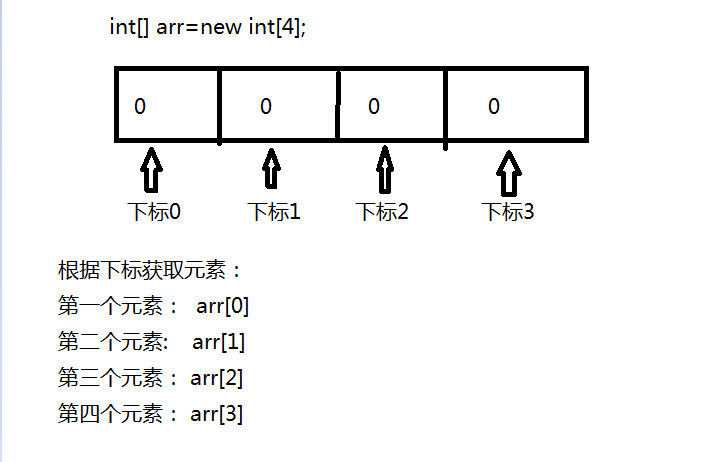
# 数组

## JVM内存分区

Jvm内存分为:栈 ,堆，方法区



## 获取数组元素

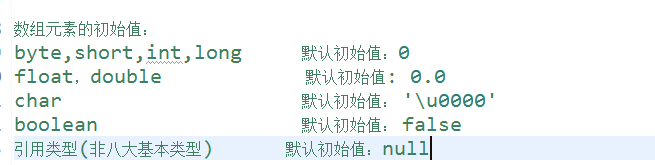


## 数组定义

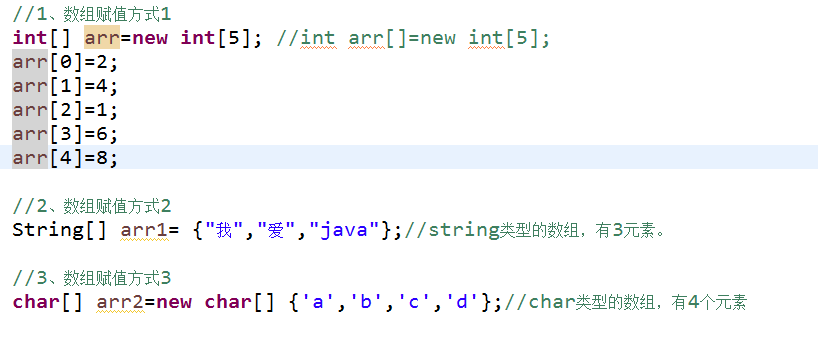
数组：存储相同数据类型的一组数据。

当数组初始化完毕后，java为数组在内存中，分配一段连续的空间，其在内存开辟的空间也将随之固定下来。此时数组长度就不在发生变化。即使数组没有保存任何数据，数组所占的空间依然存在(默认初始值)

1. 数组默认的初始值

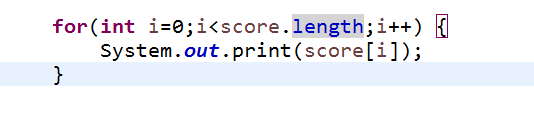


5、3种数组赋值方式



6、2种数组遍历(循环)

1、普通for遍历



2、增强型for



##### 计算数组长度

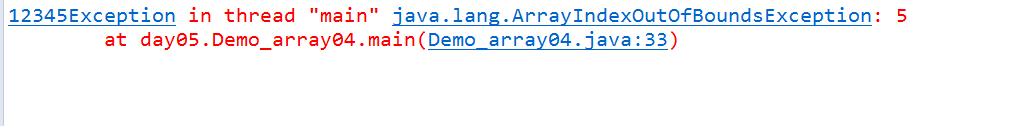
数组名.length

比如：int[] arr={1,2,3,4}；

数组长度为： arr.length

数组常见异常

数组下标越界异常： java.lang.ArrayIndexOutOfBoundsException



### 二维数组

## Arrays类

java.util包下提供了一些工具类,而java.util.Arrays类专门用于操作数组的而工具类。

Arrays.equals(array1,array2) 比较两个数组是否相等。只有当两个数组的长度相等， 对应位置的元素值也一一相等时，该方法返回true，否则返回false。

Arrays.sort(array) 对数组array进行升序排列

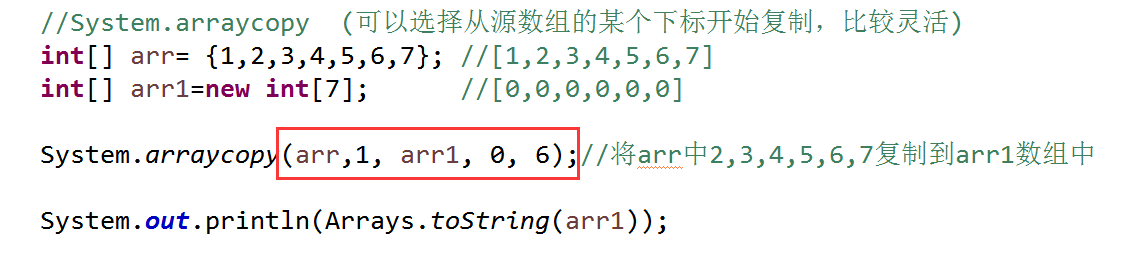
Arrays.toString(array) 将一个数组array转换为字符串

Arrays.fill(array,val) 把数值array的所有元素都赋值为val。

Arrays.copyOf(array,length) 把数组复制成一个长度为length的新数组，返回类型 与array数组类型一致

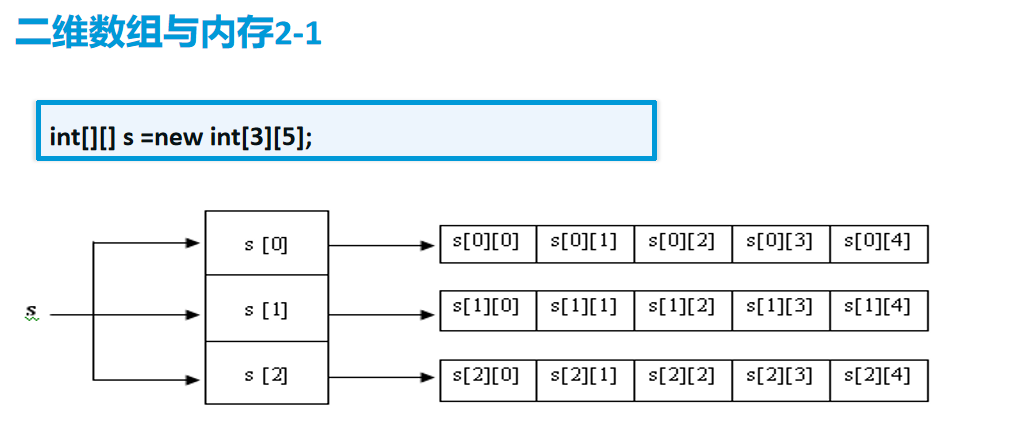
补充点：

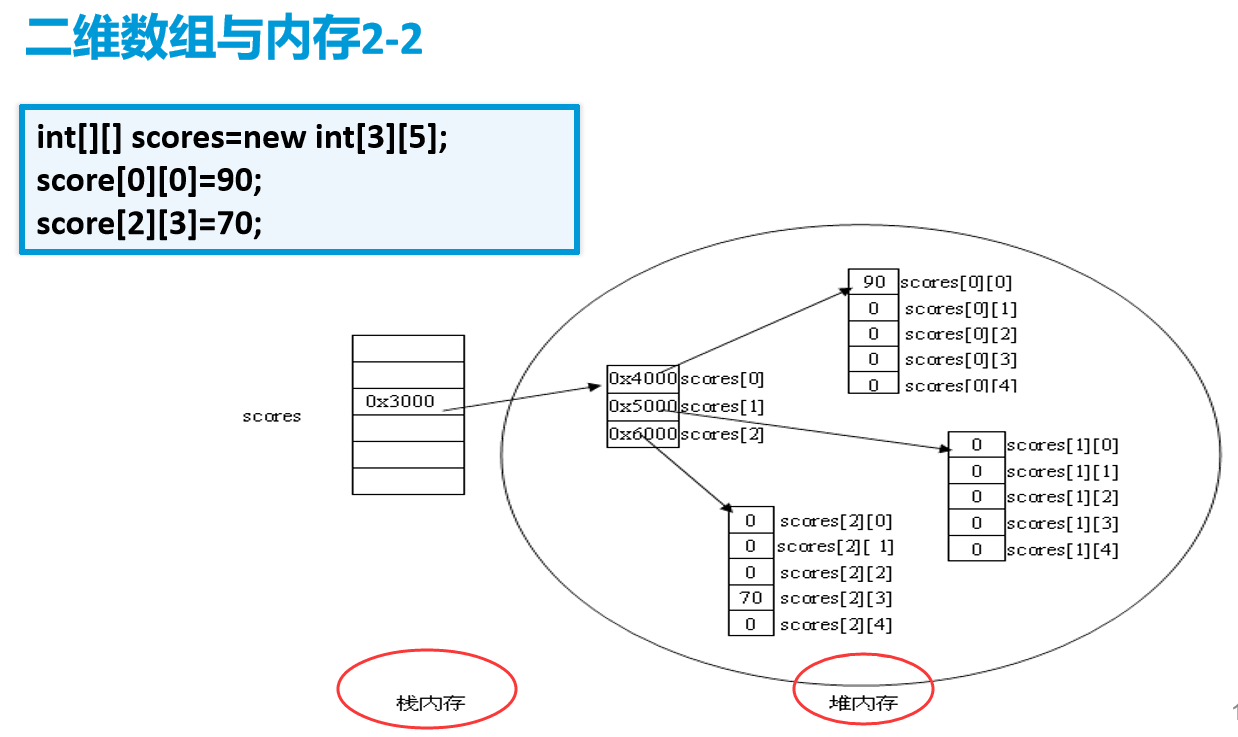
System.arraycopy(源数组,源数组的起始位置,目标数组,目标数组的起始位置,要复制的数组元素个数);



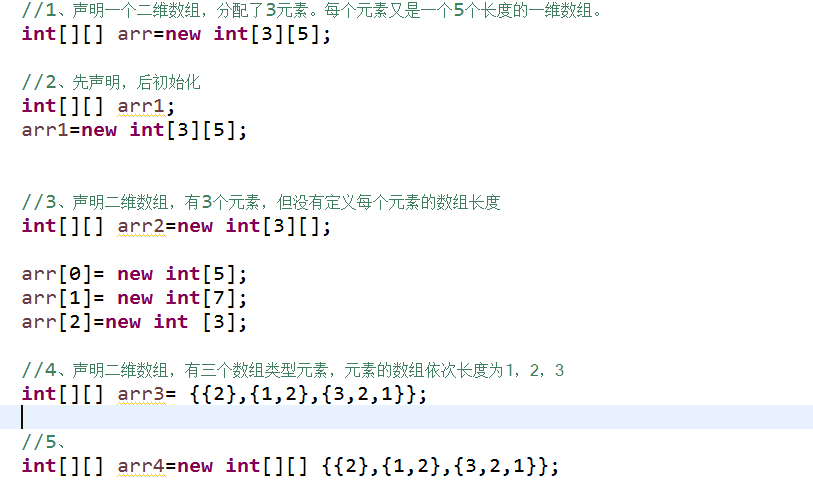
## 二维数组与内存

二维数组实际是一个一维数组，他的每一个元素又是一个一维数组





## 二维数组的声明与初始化



# Linux操作常用的命令

cd ：切换 cd huhua

cd .. :返回上级目录

pwd:查看路径

ls:查看当前目录下的所有文件

find:文件查找 find -name hello.txt

grep:内容查找 grep -lr 我们 \*

view:仅查看文件内容 view hello.txt

mkdir:创建文件夹 mkdir huhua

vi:创建查看并修改 创建文件 加后缀 vi study.xml

:wq :保存并退出

:q :不保存退出

tar :打包和压缩

tar -cvf 加包名 建包 tar -cfv huhua2018.tar

tar -rvf 添加文件 tar -rfv (你的文件路径+文件名)

tar -xvf 解压（注意目录） tar -cfv huhua2018.tar

ps -ef |grep java 查看含有"java"的活动进程

kill -9 pid 杀死进程

i : 插入

cp：复制文件

mv:移动文件

rm:删除文件

rm -r: 删除文件夹

zip –r 压缩文件 unzip 解压文件

# Git的安装及基本命令

### Git简介

1. Git(读音为/gɪt/。)是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。
2. Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。
3. Git是目前世界上最先进的分布式版本控制系统。

### 安装Git

在官方网站下载：<https://git-scm.com/>

因为Git是分布式版本控制系统，所以，每个机器都必须自报家门：你的名字和Email地址。

git config –global user.name “姓名”

git config –global user.email “邮箱”

###### 创建版本库

什么是版本库呢？版本库又名仓库，英文名***repository***，你可以简单理解成一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改、删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻可以“还原”。

创建版本库：选择一个合适的位置，创建一个空目录

第一步. 通过 cd 切到该目录下

1.cd +目录路径 切换到该目录

2.pwd 查看路径

3.git init把这个普通的目录变成Git可以管理的仓库：

##### 操作版本库

1. 在版本库添加一个文件
2. git add 文件名 添加到仓库
3. git commit –m “备注” 提交到仓库
4. git status 查看版本库状态

#### 修改版本库中文件

修改文件之后：

1. git status 查看版本库状态
2. git add 提交
3. git commit –m “”上传

#### 本部回退

1. git log 查看版本号
2. git log --pretty=oneline 查看版本号
3. git reset –hard HEAD^ 回退到之前一个版本 ~100回退之前的100个
4. git reset –hard 版本号前8位

注意：在git中，你新增了一行，这就是一个修改，删除了一行，也是一个修改，更改了某些字符，也是一个修改，删了一些又加了一些，也是一个修改，甚至创建一个新文件，也算一个修改。

git rm –rf 文件名 ：删除文件

## 分支管理

1. git branch :查看所有分支
2. git branch 分支名：创建分支
3. git checkout 分支名：切换分支

一定要先切换分支之后，然后在该分支下创建就行操作。

## 远程仓库

git remote add origin 上传网址 ： 将本地库关联远程库

git push origin master 推送内容

git clone 网站：克隆文件