

了解JAVA

Java能做什么

Java作为一门编程语言，目前在以下领域有广泛使用：

1、做大型应用（企业级）

企业级应用是指那些为商业组织、大型企业而创建并部署的解决方案及应用。这些大型企业级应用的结构复杂，涉及的外部资源众多、事务密集、数据量大、用户数多，有较强的安全性考虑。总之就是大而复杂的应用。

2. 安卓

用来开发安卓的APP。

3. 应用软件

应用软件是指运行在个人计算机上的软件，如excel、qq等。开发java广泛使用的编辑器eclipse就是使用java开发的。

Java的特点

1. 简单性

简单性是相对于C++语言而说的，在Java之前，C++语言是最流行的编程语言。Java剔除了C++中许多很少使用且难以理解的特性。

2. 面向对象

面向对象是一种程序设计技术。与其相对的是面向过程。

3. 健壮性

健壮性指的是程序运行可靠，不易出错。一方面，Java的编译器能够发现许多其他语言在运行时才能发现的错误，另一方面，取消了C++的指针概念，从而避免了重写内存和损坏数据的可能性。

4. 可移植性

JAVA语言是基于虚拟机的，屏蔽掉了硬件的差异，从而可以实现一次编写，随处运行的目的。

5. 解释型

编程语言按其编译和执行的方式，可以分为编译型语言和解释型语言。由于计算机的底层采用二进制的方式表示，因此，计算机只能识别和执行由二进制表示的机器代码。因此我们需要将编写的源程序翻译成计算机可以识别和执行的机器代码。一共有两种方式，一种是将所有的源代码，一次性的翻译成机器代码，然后每次运行时都执行机器代码。另一种是，翻译一句执行一句。前者称为编译，后者称为解释。

6. 高性能

最初硬件水平较低的时候，因为java需要在虚拟机上运行，因此给人留下了速度较慢的印象。但随着硬件的水平提高，两者的差距逐渐缩小。并且，现在的解释器也非常出色，在某些

情况下甚至超过了传统的编译器。因为解释器包含更多的代码信息，可以对常用代码进行优化。

Java当前的地位

Jun 2018	Jun 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.368%	+0.88%
2	2		C	14.936%	+8.09%
3	3		C++	8.337%	+2.61%
4	4		Python	5.761%	+1.43%
5	5		C#	4.314%	+0.78%
6	6		Visual Basic .NET	3.762%	+0.65%
7	8	▲	PHP	2.881%	+0.11%
8	7	▼	JavaScript	2.495%	-0.53%
9	-	▲▲	SQL	2.339%	+2.34%
10	14	▲▲	R	1.452%	-0.70%
11	11		Ruby	1.253%	-0.97%
12	18	▲▲	Objective-C	1.181%	-0.78%
13	16	▲	Visual Basic	1.154%	-0.86%
14	9	▼▼	Perl	1.147%	-1.16%
15	12	▼	Swift	1.145%	-1.06%
16	10	▼▼	Assembly language	0.915%	-1.34%

Java的工作原理



开发前的准备

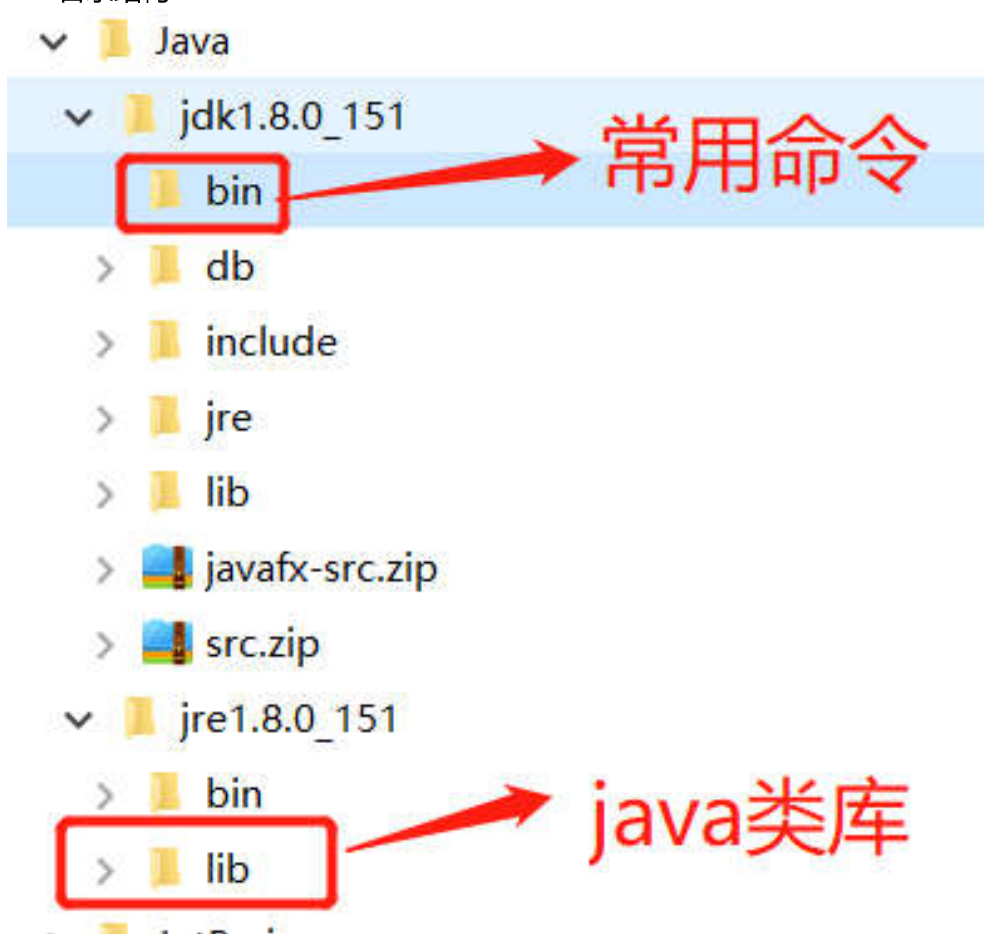
准备开发环境

了解JRE、JDK的区别

JRE(java running environment) java运行环境，包含了JVM和所有的类库。
JDK (Java Development Kit) Java开发工具集。包含了JRE和常见的开发工具。



JDK目录结构



了解PATH

Path是一个操作系统定义的环境变量，里面定义了一些常用命令的存放路径。当执行某个命令时，系统会在path变量指定的目录中搜索应用程序。

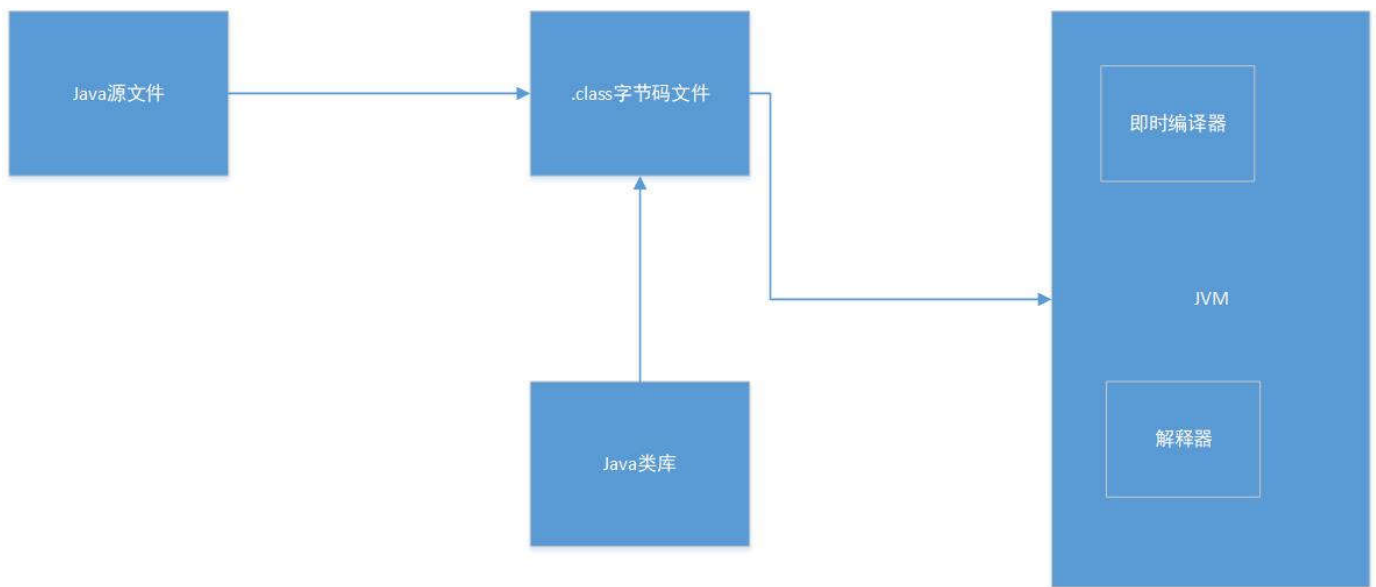
Path: 路径设置为jdk下的bin目录;

将classpath定义为jre的lib目录;

Classpath= .jre的lib目录

Classpath:指定了系统运行时找类库的位置;

了解java虚拟机工作原理



第一个java程序

用记事本新建Hello.java，并写入下列代码后保存。

```
public class Hello{
    public static void main(String[] args) {
        System.out.println("Hello Word");
    }
}
```

打开DOC，切换到Hello.java所在目录，执行javac Hello.java 得到Hello.class字节码文件。

执行java Hello，运行java程序。

注意：

1. Java区分大小写；
2. Public 称为访问控制符，控制其他程序对这段代码的访问级别，后继将详细讲。
3. 关键字class代表定义一个类，可以将类看作是组织代码的一种方式。
4. 类名必须以字母开头，后面可以跟字母或数字，长度基本没有限制，不能用java的关键字，类名必须以驼峰结构命名。
5. 文件名要和public的那个类的名称相同，。
6. Java区分大小写。
7. Java虚拟机将从指定类中的main方法开始执行。
8. System.out.println("Hello Word"); 是一个输出语句。
9. 将 “{” 和 “}” 称为块，java程序中用块来分割程序的各个部分。

Eclipse的安装

在生产环境中，一般不会真的使用记事本来开发代码，而是使用专用的编辑器来编写代码，提高编码的效率。常见的java编辑器Eclipse和 IDEA。因为eclipse是开源免费的，所以在这里介绍eclipse的使用。

在安装eclipse之前，需要先安装JDK，注意：JDK的版本要和Eclipse的版本一致。例如：JDK是32位版则Eclipse也要是32位版，二者保持一致。登录<https://www.eclipse.org/downloads/eclipse-packages/>后，下载Eclipse IDE for Java EE Developers 版本。

关键概念解释：

工作空间：工作空间是eclipse存放项目代码的地方，一个eclipse可以有多个工作空间。

输入/输出

读取输入：//吃啥饭

1. 首先构建Scanner 对象。

```
Scanner scan = new Scanner(System.in);
System.out.print("你叫什么名字？");
```

```
String name = scan.nextLine();
//读取整数用nextInt
Int a = scan.nextInt();
```

JAVA基础

注释

注释不会出现在可执行程序中。因此可以在程序中添加任意多的注释，而不用担心代码会膨胀。Java中的注释有三种：

1. 单行注释：//
2. 多行注释：以 /*开始，以 */结尾。
3. 文档注释，这种注释可以用来生成文档。以 /**开始，以 */结尾；
注意：/* */注释不能嵌套。

计算机基础知识

1. 计算机的世界只有二进制数。
二进制与10进制的转换。

2. 计算机内的单位。
1bit 用来存储0或1
8bit = 1B;
1024B = 1KB;
1024KB = 1MB;
1024MB = 1GB;
1024GB = 1TB;

数据类型

Java是一种强类型语言，这意味着要为每个变量声明一种变量类型。变量是存储数据的地方，可以理解为存储变量的容器。

变量：存储内容可变的量。变量的声明方式：

Int age;//声明变量

赋值：age = 13;

初始化：声明并赋值。 Int age = 13;

常量：存储内容不可变的量。

Final int age = 13;

字面量：以人类可读的语言形式出现的固定值。

变量类型：用来说明变量可以存储什么样的数据。

Java种的数据类型一共有8种基本类型的变量。

类型	存储需求	取值范围
Int	4字节	±21亿
Short	2字节	-32768 ~ 32767
Long	8字节	-9 223 372 036 854 775 808 ~9 223 372 036 854 775 807
Byte	1字节	-128 ~ 127
Float	4字节	±3.402 823 47E+38F（有效位6-7 位）
Double	8字节	±1.797 693 134 862 315 70E+308（有效位15位）
Boolean	1位	True/false
Char	2字节	字符

整数

在别的语言中，int在不同字长的机器上，有不同的表示范围。Java中，所有的数据类型的长度是不变的。

0x 16进制；

0 8进制；

JDK7开始，0b 表示2进制

JDK7开始，整数或浮点型字面可以嵌入一个或多个下划线，以方便阅读。例如：123_456；

默认情况下，整数的字面量是int类型。若想定义long类型的字面量，使用L或l后缀。

默认情况下，浮点字面量是double类型，若想定义float类型的字面量，使用f或F后缀。

长整数：

Long a = 123L;

- 1. 二进制相关知识—进制转换
- 2. 负数的表示方式

浮点数

Float f = 123.12f;

Float d = 1234.32d;

double num1 = 50123.12E25; //科学记数法

默认情况下，浮点数的字面量是double类型。

浮点数有三个特殊的值：

Double.POSITIVE_INFINITY; //正无穷大

Double.NEGATIVE_INFINITY; //负无穷大

Double.NaN; //不是一个数

注意：浮点数在存储时会存在误差，例如：2.0 - 1.9，会打印出0.10000000000000009，这种误差是由2进制无法精确的表示分数1/10，就像十进制无法精确的表示1/3一样。当对小数有很高的精度要求时，可以使用BigDecimal类。

Char类型

Java中，使用Unicode编码来表示一个字符，每个字符占两个字节的长度。字符使用单引号来界定。例如：'A'。

注意：字符本质上是一个数字；

Unicode 字符的取值范围是\u0000到\uffff; \ u03c0表示字母π。

在表示字符时，有些字符无法看到或者无法直接表达时，常使用转义字符的方式来表示。下表为常见的转义字符。

转义序列	名称	Unicode 值	转义序列	名称	Unicode 值
\b	退格	\u0008	\"	双引号	\u0022
\t	制表	\u0009	\'	单引号	\u0027
\n	换行	\u000a	\\	反斜杠	\u005c
\r	回车	\u000d			

Boolean类型

Boolean用来表示真和假:true/false。

注意:java中，整数和boolean值之间不能相互转换。

常量

Final 修饰的变量称为常量。常量一旦定义就不能再修改。

运算符

运算符的作用是对相关数据进行操作。

算术运算符：+、-、*、/、%

注意：

整数除以0会抛出一个异常。
浮点数除以0会得到NaN或无穷大

赋值运算符：

```
Int a = 12;  
A += 12;  
A * = 12;  
A -= 12;  
A /= 12;
```

自增/自减运算符

分为前置和后置两种情况：
A++,A--,++A,--A;
前置：先加（减）后使用。后置：先使用，后加（减）

```
Int a = 2 * ++m;
```

此处有练习

关系运算符：

关系运算符的运算结果是一个boolean值。
== 相等、!= 不相等、> 大于、>=大于等于、< 小于、<= 小于等于
&& 与运算、|| 或运算；//注意短路效果
三元操作符：表达式1? 表达式2: 表达式3;
案例：找出3个数中的最大数。

位运算符

& and、| or、^ xor、~ not
注意：& 和 | 也可以用到布尔值上，但不会有短路效果。
>>、<<位移运算符;
>>> 无符号位移运算，此操作使用0来填充高位。

运算符的优先级

运 算 符	结 合 性
[] . () (方法调用)	从左向右
! ~ ++ -- + (一元运算) - (一元运算) () (强制类型转换) new	从右向左
*/ %	从左向右
+ -	从左向右
<< >> >>>	从左向右
< <= > >= instanceof	从左向右
= = !=	从左向右
&	从左向右
^	从左向右
	从左向右
&&	从左向右
	从左向右
?:	从右向左
= += -= *= /= %= &= = ^= <<= >>= >>>=	从右向左

可以使用（）来改变运算的级别。

数学相关函数：

```
Math.ceil(double d);//向上取正  
Math.floor(double d);//向下取正  
Math.round(double d);//四舍五入  
Math.pow(x,a);//x的q次幂  
Math.sqrt(x);//x的平方根
```

数据类型的转换

1. 自动转换，在数的四则运算中，表达式中的数据会自动进行转换，一般来说，转换的原则是：短转长。
2. 强制转换。需要在（数据类型）的方式进行强制数据转换。注意：强制转换会损失数据的精度。

```
Double x = 1.234;
```

```
Int y = (int)x;
```

Char 类型本质上是一个数字。

案例：

将大写字母转换为小写字母：

```
Scanner c = new Scanner(System.in);
String a = c.next();
char c1 = a.charAt(0);
System.out.println((char)(c1 + 32));
```

生成随机数0-10之间：

```
Random random = new Random();
int a = random.nextInt(10);
或使用 Math.random(); 返回0到1之间的一个随机数
随机生成4个字母
```

字符串

字符串顾名思义就是字符的序列。Java中没有内置的字符串类型，而是提供了一个叫String的类。字符串用双引号来表示。

```
String name="zhang san";
```

字符串相关操作：

1. 字符串截取 substring
 2. 字符串拼接 "abcd" + "efg";
- a. 当字符串跟一个非字符串的值进行拼接时，后者会被转换成字符串。
 - b. Join方法将字符串用指定分隔符拼接起来。

```
String all = String.join(" / ", "S", "M", "L", "XL"); // S/M/L/XL
```

字符串的不变性：为了提高字符串的效率，java使用了字符串常量池的概念。

为了提高拼接字符串的效率，java引入了StringBuilder和StringBuffer

```
StringBuilder sb = new StringBuilder();
```

```
Sb.append();
```

```
Sb.toString();
```

StringBuffer是StringBuilder的线程安全版本，允许多个线程对其进行字符串的拼接。StringBuilder的效率会比StringBuffer效率高。

检查字符串是否相等：s.equals(b);

```
"Hello".equals(s);
```

注意：不要使用==来比较字符串是否相等。==可以比较两个字符串是不是同一个字符串。

空串与null

""代表长度为0的字符串对象，而null表示变量还没有指向任何对象。

检查一个字符串是否为空，可以使用以下方法：

```
If(string == null ){//检查字符串为null
```

```
If(string != null && string.length != 0){//不为空
```

String 常用方法：

截取某个字符：charAt;

获取第一个匹配字符串的索引值：indexOf

获取最后一个匹配字符串的索引值：lastIndexOf()

去两边空格：trim()

拼接字符串：join()

转换大小写：toLowerCase()/toUpperCase();

截取字符串：substring(int begin);

替换字符串：replace

以XX开始/结尾: `startsWith()/endsWith()`

比较两个字符串是否相等: `equals()`

大数值

如果整数和浮点数的精度不能够满足需求, 可以使用`BigDecimal`和`BigInteger`两个类。这两个类可以使用任意长度的整数和任意精度的浮点数。

1. 使用`valueOf`方法将普通数值转换为大数值。

```
BigInteger a = BigInteger.valueOf(100);
```

```
    a.add(b); //a +b
```

```
    a.subtract(b) //a-b
```

```
    b.multiply(b); //a*b
```

```
    divide() //除法
```

```
    mod() //求余数
```

```
    valueOf(); //获取整数
```

```
    compareTo(b); //a和b相比较
```

语句结构

在了解语句结构之前, 先了解“**块**”的概念, 块儿定义了java中变量的作用范围。

块 (即复合语句) 是指由一对大括号括起来的若干条简单的Java 语句。块确定了**变量的作用域**。一个块可以嵌套在另一个块中。

注意: 不能在嵌套的两个块中声明同名的变量。

```
{
    int n ;
    {
        int n; //Error
    }
}
```

顺序

选择

语句1: `if()`

语句2: `if()` else {}

语句3: `if()` else `if()`

语句4: `switch () {}`

注意: `case` 后可以是`char`、`byte`、`short`、`int`。从jdk7开始, 后面可以跟字符串。

案例1: 求3个数中的最大值;

循环

```
While(){} //求一个数各个数位的和
```

```
Do{} while()
```

```
For(){} 
```

```
Break与continue;
```

```
带标签的循环。
```

案例1: 乘法口诀表

案例2: 猜数字

数组

数组是一种数据结构, 表示一组相同类型的变量的集合。

声明数组：

类型[] 数组名; 或 类型 数组名[];

Int[] a; 或 Int b[];

创建数组：

Int[] a = new Int[100];

创建一个数组时，数字类型所有的元素都初始化为0，boolean初始化为false,对象都初始化为null。

遍历数组

补充for-each循环

数组的初始化：创建数组并为数组赋值的简化形式。

Int[] s = {1,2,3,4,5,6,7};

还可以：

S = new int[]{1,2,3,4,5,6,7};

注意：java中允许存在长度为0的数组。

New int[0];

数组的拷贝：

在java中可以将一个数组变量拷贝给另一个数组变量。此时，两个变量引用同一个数组。

Int[] c = s;

C[i] == s[i]; //true.

如果希望将一个数组拷贝到另一个数组中去：

Int[] n = Arrays.copyOf(arrays,arrays.length); //第二个参数是新数组的长度，可以使用这个方法扩展数组。

数组的排序：

Arrays.sort () ; //数组的快速排序

案例：出题程序，从题目池里选择5个题目不重复的试题，最后再对题目进行排序。

Java.util.Arrays中的常用方法：

Sort();//排序

BinarySearch(type[], type v); //二分查找

多维数组

多维数组可以理解成：元素是数组的数组。

多维数组的声明：

Int[][] mutArrays;

多维数组的初始化：

mutArrays = new int[5][10];

知道元素的情况下，也可以不调用new

Int[][] arrs = {

{1,2,3}

{4,5,6}

}

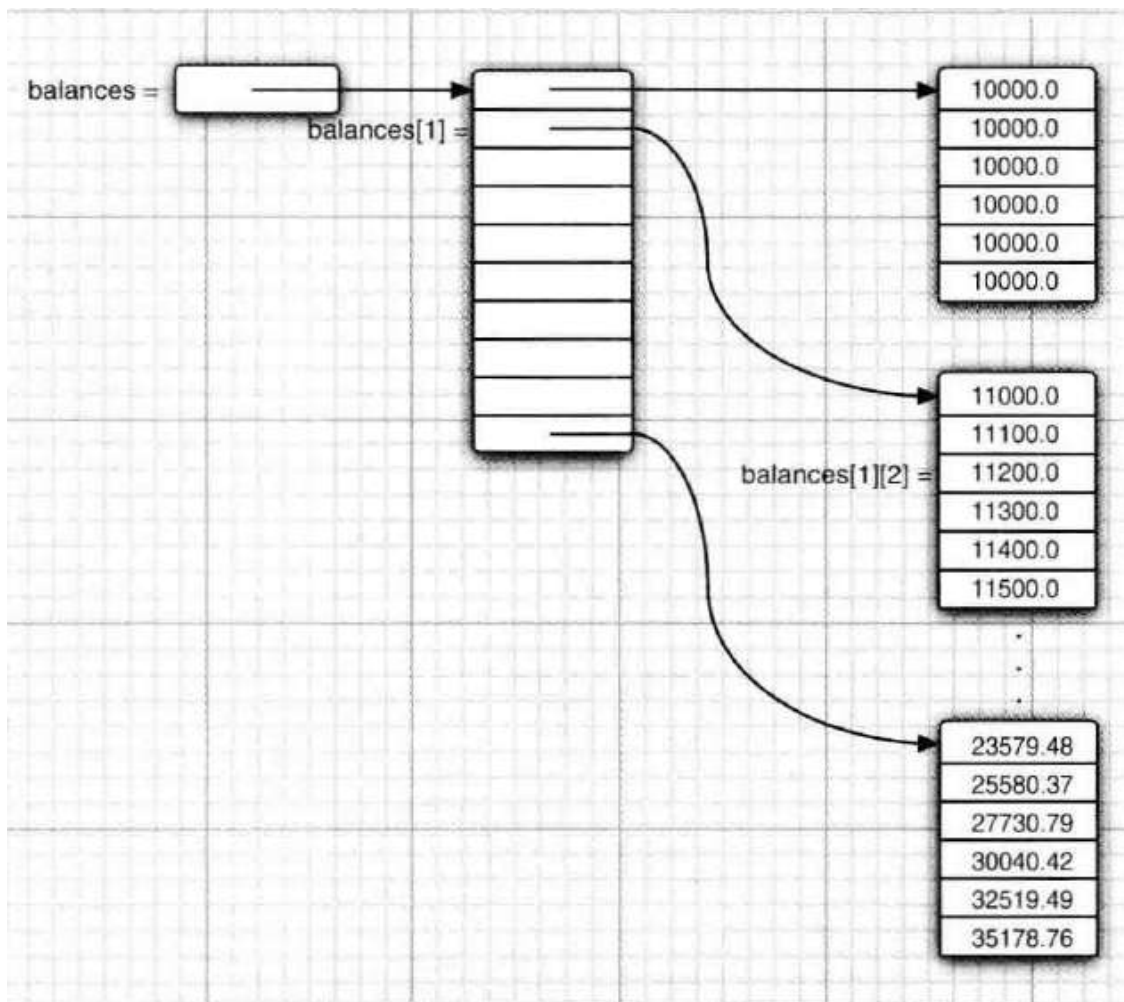
一旦确定后，就可以通过下标来进行访问。Arrs[i][j];

注意：打印二维数组的值可以使用for循环，也可以使用Arrays.deepToString(arr);

常见算法

1. 冒泡排序

2. 二分查找



二维数组示意图

二维数组可以整维的交换:

```
int[][] arr = new int[][] {
    {1,2},
    {3,4}
};

System.out.println(Arrays.deepToString(arr));
int[] tmp = arr[1];
arr[1] = arr[0];
arr[0] = tmp;

System.out.println(Arrays.deepToString(arr));
```

不规则数组：多维数组中，元素的数量不必相等。

```
int[][] arr = new int[][] {
    {1,2,5},
    {3,4}
};
```