**数组开始了**

1.数组开始（图中的黄色字体代表定义数组 ）

**public** **class** Shuzu {

**public** **static** **void** main(String args[]){

**int[] a = new int[3]**; //定义一个数组

a[0] = 1;

a[1] = 2;

a[2] = 3;

**for**(**int** i=0;i<a.length;i++){

System.*out*.println(" "+a[i]);

}

}

}

2>一位数组数组的声明和初始化

**public** **class** Shengmingshuzu {

**public** **static** **void** main(){

//声明

**int**[] a;

**int** b[];

//创建数组对象

a = **new** **int**[3];

b = **new** **int**[4];

//初始化的方式，现在我们有3种方法

//1.数组默认初始化的方式和成员变量相同 数字0，布尔false

//2.1动态初始化

a[0] = 4;

a[1] = 5;

//2.2循环动态初始化

**for**(**int** i = 0;i < a.length;i++){

a[i] = i \* 4 ;

}

//3.1静态初始化

**int** c[] = {1,4,2,3}; 这里是直接就进行了初始化

}

}

2.String

1.charAt的使用（输入索引值将字符输出（超过的按照最后一个字符输出））

定义一个数组，如下就可以了

String str1 = **new** String("这是我的第一个字符串");//相当于是一个数组

String str2 = "我是一个兵，";

System.*out*.println(str1.charAt(2)); // 里面的字符是 我

System.*out*.println(str2.charAt(2)); 代表的是索引号是2的字符



2.equals的使用（或者直接利用 == 也是可以的(但是情况会不同)）

\* equals是直接比较的数组中的东西，

\*但是我们直接等于（ = = ）的时候是我们看他们所指向的对象是否是相同的 《 new 出来的必定不等于 直接String 出来的》

String str1 = **new** String("这是我的第一个字符串");//相当于是一个数组

String str2 = "我是一个兵，";

String str3 = "这是我的第一个字符串";

String str4 = "我是一个兵，";

System.*out*.println(str1.charAt(2)); // 里面的字符是 我

System.*out*.println(str2.charAt(2));

System.*out*.println(str1 == str3); //false

System.*out*.println(str1.equals(str3)); //ture

System.*out*.println(str1.equals(str2)); false// str1是否等于str2 等于输出ture，

System.*out*.println(str1 == str2); false

System.*out*.println(str2 == str4); ture

System.*out*.println(str2.equals(str4)); ture

3.indexOf 代表的东西（输入一个字符串中有的字符,输出索引号）

\* indexOf（‘’）中间只能填写一个字符，否则会报错，

\* 如果输入的字符在该字符串中没有那么他将会输出 – 1

String str2 = "我是一个兵，";

System.*out*.println(str2.indexOf('我'));

System.*out*.println(str2.indexOf('s'));

输出的结果是 

4. substring

\* (提供一个索引值，将该索引值代表的字符以及之后的字符一同输出)

\* 如果索引值不在该字符串之间，那么系统将会报错

String str2 = "我是一个兵，";

System.*out*.println(str2.substring(1));

输出的结果是 是一个兵

5 .replase

\* 将字符串中的某个字符换掉

\* 如果没有提供的字符（则不会变化已经有的字符串）

\* 只能提供一个字符，否则会报错

String str2 = "我是一个兵，";

System.*out*.println(str2.replace('我','是'));

System.*out*.println(str2.replace('哈 ','是'))

输出结果是 是是一个兵，

我是一个兵。

6.split（拆分字符串）

String str5 = "张宇晋，刘利，从前是";

String[] a = str5.split("，"); //输入的是字符串

**for**(**int** i = 0;i<a.length;i++){

System.*out*.println(a[i]);



3.String 的补充

1.equalsIgnoreCase （忽略大小写看看是否相等）

System.*out*.println("WOSHIyigebing".equalsIgnoreCase("woshiyigebing"));

输出 ture

2.lastIndexOf (从后面往前面找)

System.*out*.println("woshiwoderen".lastIndexOf('o'));

输出结果是 6

3.startsWith(判断是不是以……开头) 注意输入的是字符串

endswith （判断是不是以……结尾）

System.*out*.println("woshiwoderen".startsWith("w"));

输出结果是 ture

4.toLowerCase 大写变小写

toUpperCase 小写变大写

System.*out*.println("WOSHI".toLowerCase());

5.添加字符串 (和下面的StringBuilder可以更改的序列进行比较)

String a = "张宇晋";

**for**(**int** j = 0;j < 11;j++){

a +=j;

}

System.*out*.println(a);

}

输出 张宇晋012345678910

4.length的用法

1\* 数组中 a.length 代表的是属性

2\* 变量中 例如String a = “”; a.length()代表的是方法

6.StringBuilder

StringBuffer

\* 可变字符序列StringBuilder 不安全效率高

\* StringBuffer 安全但是效率低 一般不用

\*不可以变换的的是 String

《》使用append进行追加

**public** **static** **void** mian(String args[]){

StringBuilder a = **new** StringBuilder();//字符数组长度初始为16

StringBuilder b = **new** StringBuilder(32);//初始长度为32

StringBuilder c = **new** StringBuilder("abcd");

//字符串初始长度为16+本来的长度4 = 20，value = {'a','b'……}

//String f[] = new String[4];

c.append("woshi").append("张宇晋") this用法

System.*out*.println(c);

5. StringBuilder 的常用方法

1\* delete

StringBuilder sb = **new** StringBuilder("abcdefghijklmnopqrstuvwxyz");

sb.delete(3,5).delete(3,5); //包头不包尾

System.*out*.println(sb);

结果为：abcfghijklmnopqrstuvwxyz

2\* reverse

sb.reverse(); //将它从后面向前面排列

System.*out*.println(sb);

输出结果是 zyxwvutsrqponmlkjihgfcba

总结下 substring replace charAt index

4容器

**public** **class** Rongqi{

//public static void mian(String args[]){

StringBuilder s;

Object[] value;

**int** size; //这里要用到它的初始化，非常重要

//}

**public** Rongqi(){ //初始化数组

value = **new** Object[16];

}

**public** Rongqi(**int** size){

value = **new** Object[size];

}

//给容器中放东西

**public** **void** add(Object obj){

value[size] = obj;

size++;

**if**(size >= value.length){

//装不下了扩容吧！

**int** xinrongliang = value.length \* 2;

Object xinmingzi[] = **new** Object[xinrongliang];

**for**(**int** i = 0;i < value.length;i++){

xinmingzi[i] = value[i]; //将旧的东西放到新的东西里

}

value = xinmingzi; //这样就对容量的大小进行了扩容

}

}

**public** Object get(**int** index){ //这个类型需要返回类的名字

**if**(index < 0||index > size-1){ //这里的index必须是在已经有的数组之间之间 【0,size-1】

**try** {

**throw** **new** Exception(); //手动抛出一份异常，之后才会学习

} **catch** (Exception e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} //这里的程序的任务是输出一份异常，

}

**return** value[index];

}

**public** **static** **void** main(String args[]){

Rongqi a = **new** Rongqi(2); 构造器的使用

a.add("张宇晋");

a.add(**new** Shiyan("张宇晋")); 这里用到一个构造器

a.add("刘利");

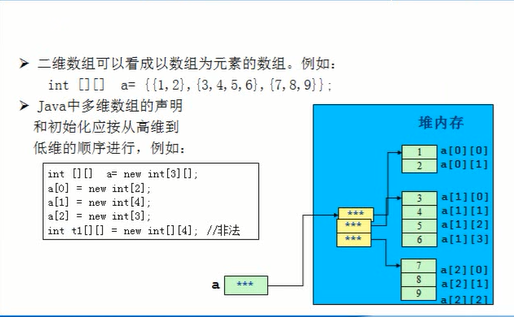
System.*out*.println(a.get(2)); 返回类的名字

}

}

5.多维数组

1.声明和初始化 时候按照从高维到低维（也就是说从一维到多维）



**int**[][] a ={{1,4,5},{1},{5}};

**int**[][] b = **new** **int**[3][];

b[0] = **new** **int**[2];

b[1] = **new** **int**[3];

b[2] = **new** **int**[4];

b[0][0] = 1;

b[0][1] = 2;

b[1][0] = 3;

b[1][1] = 4;

b[1][2] = 5;

2.数组进行运算

1》定义一个数组类型的方法

**public** **static** **int**[][] add(**int**[][] a,**int**[][] b){

**将来返回的是一个数组 （地址就可以了）**

**int** c[][] = **new** **int**[a.length][a.length];

**for**(**int** i = 0;i<c.length;i++){

**for**(**int** j = 0;j<c.length;j++){

c[i][j] = a[i][j] + a[i][j];

}

}

**return** c;

}

2》定义一个类名字类型的方法

**public** Object get(**int** index){

**if**(index < 0||index > size-1){

**try** {

**throw** **new** Exception(); //手动抛出一份异常，之后才会学习

} **catch** (Exception e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} //这里的程序的任务是输出一份异常，

}

**return** value[index]; /返回类的名字 （这里定义的是一个类的数组）

}

3》接1》如何调用数组类型的方法

**public** **static** **void** main(String args[]){

**int**[][] a = {{1,2},{3,4}};

**int**[][] b = {{1,2},{3,4}};

**int** c[][] = *add*(a,b); //这里就是在调用多维数组的方法（字面意思理解可以是 只要c[][]中有东西就是可以成立的）

3.数组的拷贝



1》数组的拷贝

**public** **static** **void** main(String args[]){

String[] a = {"张宇晋","刘利","张葛天"};

String[] b =**new** String[60];

System.*arraycopy*(a,1,b,2,5); //等待拷贝的数组，数组开始的索引号，未来的数组，引进的位置，a中需要拷贝的个数

数组a拷贝到数组b中，从数组a的第1位开始，拷贝到数组b的第2位，拷贝的长度是a中的5位

2》数组的排序和输出

**public** **class** Shuzupaixu {

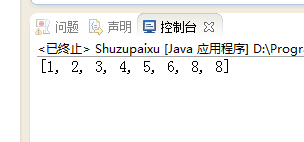
**public** **static** **void** main(String args[]){

**int** a[] = {4,5,6,8,1,2,3,8};

Arrays.*sort*(a); //对数组进行排序

System.*out*.println(Arrays.*toString*(a));

} 用来打印数组

}

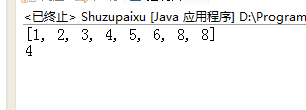
3》 如果我们要查找在排序完成之后的 某个数字的索引号 我们就必须用二分法进行排序

**int** a[] = {4,5,6,8,1,2,3,8};

Arrays.*sort*(a); //使用二分法进行 排序

System.*out*.println(Arrays.*toString*(a));

System.*out*.println(Arrays.*binarySearch*(a, 5));



4》填充（将一段数组中的元素变了）

**int** s[] = {5,4,18,9,2,8,11115};

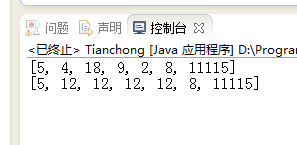
System.*out*.println(Arrays.*toString*(s));

//下面将索引号为1到5之间的数变为12 （包头布包尾） 意思就是 1,2,3,4

Arrays.*fill*(s,1,5,12);

System.*out*.println(Arrays.*toString*(s));

}

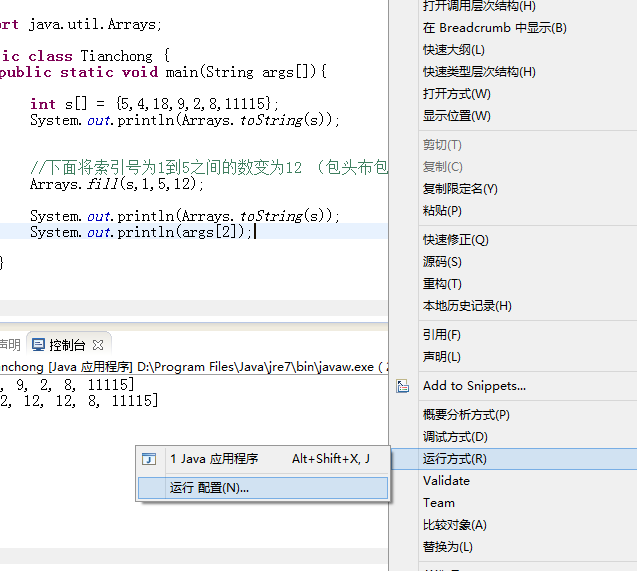


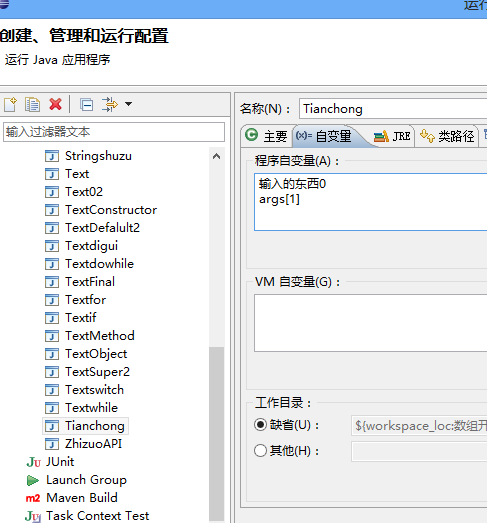
6.String args[] 数组的用法（了解即可）

**public** **static** **void** main(String args[]){

System.*out*.println(args[2])

}





7.JDK 5.0新增内容

《《《《《《《使用for循环快速输出数组》》》》》》》》

相比我们之前输出更加方便

我们之前输出

**for**(**int** i = 0; i < a.length;i++){

System.*out*.println(a[i]);

}

//相当于下面的程序，是一摸一样的

**for**(**int** i : a){

System.*out*.println(i);

}

}