Day03

java课程-李彦伯

Scanner问题(next和nextLine和nextInt)

> nextInt只会读取整数,后面的回车不会读取,所以如果出现读取完整数以后,不能读取后面的字符串,在两个读取中间加一句
>
> `sc.nextLine();`

```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
sc.nextLine();
String str = sc.nextLine();
System.out.println(a+"------"+str);
```

循环控制语句练习

> 章节的break和continue

```
for (int i = 1; i < 6; i++){
if(i == 5){
break;
}

for(int j = 1; j < 5 + i; j++){
if(j == 3){
    continue;
}

System.out.println("第"+i+"章,第"+j+"节");
}
}
```

乘法表

```
for (int i = 1; i <= 9; i++){
for (int j = 1; j<= i ;j++){
System.out.print(j+"*"+i+"="+j*i+"\t");
}
System.out.println();
}
```

计算1到100的前n项和

```
/*
 * 1. for循环从1开始到100结束
 * 2. 定义整型变量,每次for循环将值加入这个变量
 * 3. for循环结束以后输出这个变量
 *
 *
 */
int sum = 0;
for(int i = 1; i < 101; i++){
    sum += i;
}
System.out.println(sum);
```

计算水仙花数:3位数,各位数字立方和等于这个数本身

```
//建议把变量定义在外部
int bai = 0;
int shi = 0;
int ge = 0;
for(int i = 100; i < 1000;i++){
    bai = i / 100;
    shi = i / 10 % 10;
    ge = i % 10;
    if(bai*bai*bai+shi*shi*shi+ge*ge*g
e==i){
        System.out.println(i);
    }
}
```

输出A-Z,a-z

```java
/*
 * 1. 定义两个整型变量(for的外部)
 * 2. 一个变量是65,一个变量是97
 * 3. 写一个for循环,循环26次,没次让这两个变量
加1
 * 4. 将这两个变量转换为char进行输出
 *
 */

int littleNum = 97;
int bigNum = 65;
for (int i = 0; i < 26; i++){
    char littleLetter = (char)littleNum;
    char bigLetter = (char)bigNum;
    System.out.println(littleLetter+"----
"+bigLetter);
    littleNum++;
    bigNum++;
}

char littleLetter = 'a';
char bigLetter = 'A';
for (int i = 0; i < 26; i++){
    System.out.println(littleLetter+"----
"+bigLetter);
    littleLetter++;
    bigLetter++;
}
```

猜数字游戏

```
/*
* 1.定义随机数1-100
* 2.写一个while死循环
* 2.1 请输入1-100的整数,创建Scanner,读取整数,判断整数和随机数是否相同,如果相同break,如果不同输出大还是小
*/

Random ran = new Random();
int num = ran.nextInt(100)+1;
while(true){
System.out.println("请输入1-100的整数");
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
if (a == num){
    System.out.println("人才");
    break;
}
System.out.println(a>num?"猜大了":"猜小了");
}
```

数组

一维数组

- 数组的声明`类型 [] 数组名;`
- []在前后没有影响,一般是写在名称前方
- 数组的创建`数组名= new 类型[数组长度]`
- 一般建议写成`类型 [] 数组名 = new 类型[数组的长度]`
- 数组的访问是通过索引来进行访问的,注意索引值是从0开始`System.out.println("arr[0]=" + arr[0]); //获取数组的第一个元素`
- 获取数组的长度`System.out.println(arr.length)`
    - 数组的长度和索引的关系

- 定义完数组的时候系统会给数组默认值,叫做动态初始化,不同的类型初始化的值不同

类型	初始化的值
byte,short,int,long	0
float,double	0.0
char	空格
boolean	false
引用类型	null

- 数组的静态初始化两种方式

    - `类型[] 数组名 = new 类型[]{元素，元素，……};`
    - `类型[] 数组名 = {元素，元素，元素，……};`推荐使用这种方式

- 数组的遍历
    - 普通for循环遍历
    - for each遍历:

```
    int[] arr = { 1, 2, 3, 4, 5 };
    for (int i = 0; i < arr.length; i++)
{
        System.out.println(arr[i]);
    }
    for (int a : arr){
        System.out.println(a);
    }
```

- 数组的常见问题
    - 数组越界异常

```
thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
m.zhiyou100.demo.demo01.ArrayDemo.main(ArrayDemo.java:18)
```

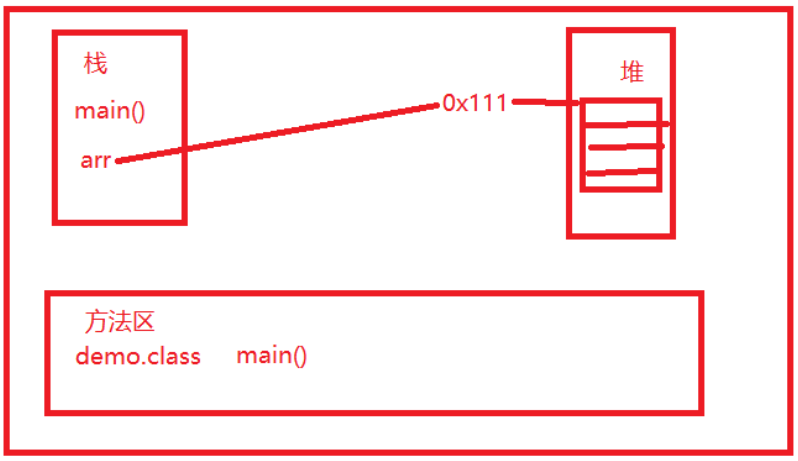
数组的越界异常

JVM的内存划分

- 程序计数器:记录cpu该去执行线程中哪条指令,说白了就是内存和cpu进行通信不需要我们去关心
- 本地方法栈:jvm调用操作系统的方法 的区域,如操作系统支持的复制和粘贴不需要我们去关心
- 方法栈:执行方法,保存局部变量
- 方法区:存储了每个类的信息（包括类的名称、方法信息、字段信息）、静态变量、常量以及编译器编译后的代码等
- 堆:用来存储对象本身的以及数组

一维数组内存原理

demo.class




```
public class ArrayDemo {

    public static void main(String[] arg
s) {
        int [] arr = new int [4];
        System.out.println(arr);

    }

}
```

1. ArrayDemo.class先进入JVM的方法区
2. 代码执行main方法,main方法复制压栈开始执行
3. 遇到创建数组的代码,在堆中创建一块内存,开辟4个空间,给每个变量有个默认值是0
4. 将数组的首地址给栈中的arr变量,所以这个arr变量就有了对堆中对象的引用
5. 代码执行到输出语句,就会输出arr所引用的堆中的内存地址

> 对引用类型,只要new一次,就代表在堆中开辟内存空间

二维数组

- 二维数组声明`类型 [][] 数组名;`
- 二维数组的赋值`数组名= new 类型[长度][长度]`
- 一般建议写成`类型 [][] 数组名 = new 类型[长度][长度]`
- 赋值

```
int [] arr = new int [4];
arr[0] = 100;
arr[1] = 200;
arr[2] = 300;
arr[3] = 400;
```

- 长度是 `二维数组名.length;`
- 二维数组中一维数组的长度:`二维数组名[索引].length`
- 二维不定长度的格式

```
int[][] arr = new int[3][];
arr[0] = new int []{1,2,3};
arr[1] = new int []{4,5};
arr[2] = new int []{6};
```

- 简写形式 `int[][] arr = {{1,2,3},{4,5},{6}};`

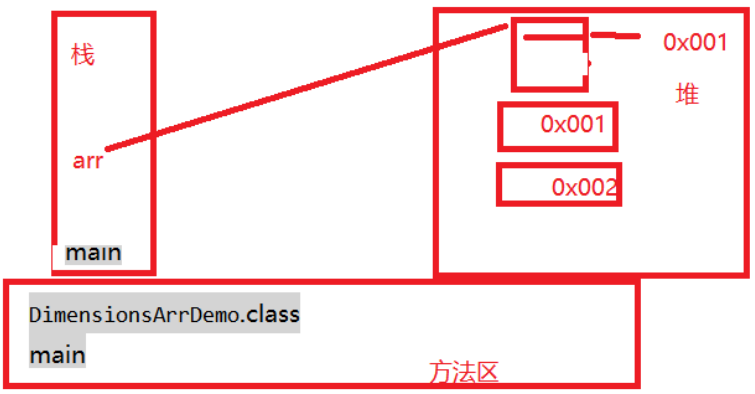
> 作业:
> 二维数组累加求和

```java
//三个小组,小组人数3,4,5
//小组一 : 1020 1505 1208
//小组二 : 4001 5634 3214 6547
//小组三: 7658 8976 7866 8900 9087

int [][] bigArr = {{1020,1505,1208},{400
1,5634,3214,6547},{7658,8976,7866,8900,90
87}};

int sum = 0;
for(int i = 0; i < bigArr.length; i++){
    for (int j = 0; j < bigArr[i].length;
j++){
        sum += bigArr[i][j];
    }
}
System.out.println(sum);

int sum2 = 0;
for(int [] littleArr : bigArr){

    for(int num : littleArr){
        sum2 += num;
    }
}

System.out.println(sum2);
```

一维数组和二维数组的内存关系

1. class文件先进入方法区
2. main压栈执行
3. 在堆中创建两个数组,并初始化内容为0,各自的地址为数组中首元素的地址
4. 在堆中创建一个数组,数组的长度为2,数组的中有两个对之前的数组的引用
5. 让arr引用二维数组的地址

```
int [][] arr =
new int[2][3];
arr[0][0] = 1;
arr[0][1] = 2;
arr[0][2] = 3;
```



作业:

1. 找出数组的最大值
2. 一维数组累加求和
3. 数组的元素倒序
4. 数组的元素排序
   4.1 选择排序
   4.2 冒泡排序