Java 7.21 学习

今日目标：完成多维数组的学习 完成数组的这一章节的练习题

多维数组暂时只学习二维数组（TwoDimensionalArray）

(1)二维数组

//什么是二维数组：  
//1. 从定义形式上看 int[][]  
//2. 可以这样理解，原来的一维数组的 每个元素 是一维数组, 就构成二维数组  
  
//关于二维数组的关键概念  
//(1)二维数组的元素个数  
//System.out.println("二维数组的元素个数=" + arr.length);  
//(2) 二维数组的每个元素是一维数组,  
// 所以如果需要得到每个一维数组的值还需要再次遍历  
//(3) 如果我们要访问第(i+1)个一维数组的第j+1 个值arr[i][j];  
// 举例访问3, =》他是第3 个一维数组的第4 个值arr[2][3]

//请用二维数组输出如下图形  
//0 0 0 0 0 0  
//0 0 1 0 0 0  
//0 2 0 3 0 0  
//0 0 0 0 0 0  
int arr[][] = {{0, 0, 0, 0, 0, 0}, {0, 0, 1, 0, 0, 0},  
 {0, 2, 0, 3, 0, 0}, {0, 0, 0, 0, 0, 0}};  
for (int i = 0; i < arr.length; i++){  
 for(int j = 0; j< arr[i].length; j++){  
 System.*out*.print(arr[i][j] + " ");  
 }  
 System.*out*.println();  
}  
//如果我们要访问第(i+1)个一维数组的第(j+1)个值arr[i][j];  
//比如访问第三个一维数组的第四个值--3  
System.*out*.println(arr[2][3]); //3  
//二维数组的长度 arr.length  
System.*out*.println(arr.length); //4  
//二维数组中第i个一维数组的长度 arr[i].length  
System.*out*.println(arr[0].length); //6

(2)二维数组的使用方式1：动态初始化

//二维数组的使用方式1： 动态初始化  
//1) 语法: 类型[][] 数组名=new 类型[大小][大小]  
//2) 比如: int a[][] = new int[2][3]  
int arr[][] = new int[2][3];  
arr[0][1] = 6; //第一组的一维数组中的第二个值赋值为6  
for(int i = 0; i < arr.length; i++){  
 for (int j = 0; j < arr[i].length;j++){  
 System.*out*.print(arr[i][j] + " ");  
 }  
 System.*out*.println(); // 换行作用  
}

（3）二维数组的内存布局图解：



（4）二维数组的使用方式2 动态初始化

//二维数组的使用方式2： 动态初始化  
//(1)先声明： 类型 数组名[][];  
//(2)再定义(开辟空间): 数组名 = new 类型[大小][大小];  
//(3)赋值(有默认值，比如int 类型的就是0)  
int arr1[][];  
arr1 = new int[2][3];

（5）二维数组的使用方式3 动态初始化---列数不确定



//二维数组的使用方式3： 动态初始化---列数不确定  
//看一个需求：动态创建下面二维数组，并输出  
//i = 0: 1  
//i = 1: 2 2  
//i = 2: 3 3 3  
//创建二维数组 一共有三个一维数组 但是每个一维数组还没有开辟空间  
int arr2[][] = new int[3][];  
for(int i = 0; i < arr2.length;i++){  
 //先给每个一维数组创建空间  
 arr2[i] = new int[i + 1];  
  
 //遍历一遍数组 给一维数组的每个元素赋值  
 for (int j = 0; j < arr2[i].length;j++){  
 arr2[i][j] = i + 1;  
 System.*out*.print(arr2[i][j] + " ");  
 }  
 System.*out*.println();  
}

（6）二维数组的使用方式4 静态初始化

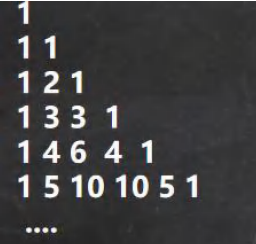
//二维数组的使用方式4 静态初始化  
//定义类型数组名[][] = {{值1,值2..},{值1,值2..},{值1,值2..}}  
//使用即可[ 固定方式访问]  
//比如:  
int[][] arr3 = {{1,1,1}, {8,8,9}, {100}};//哪怕只有一个元素100 也不可以去掉{100}外括号  
//解读  
//1. 定义了一个二维数组arr  
//2. arr 有三个元素(每个元素都是一维数组)  
//3. 第一个一维数组有 3 个元素, 第二个一维数组有 3 个元素, 第三个一维数组有 1 个元素

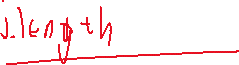
小练习：

//int arr[][] = {{4,6},{1,4,5,7},{-2}};  
//遍历该二维数组，并得到和  
int arr[][] = {{4,6},{1,4,5,7},{-2}};  
int sum = 0;  
for (int i = 0;i < arr.length;i++){  
 for (int j = 0;j < arr[i].length;j++){  
 System.*out*.print(arr[i][j] + " ");  
 sum += arr[i][j];  
 }  
 System.*out*.println();  
}  
System.*out*.println("sum=" + sum);

二维数组的练习（杨辉三角）

//使用二维数组打印一个 10 行杨辉三角





//使用二维数组打印一个 x 行杨辉三角  
//arr[i][j] = arr[i-1][j-1] + arr[i-1][j];  
Scanner myScanner = new Scanner(System.*in*);  
System.*out*.println("请问要打印一个多少行杨辉三角");  
int x = myScanner.nextInt();  
int arr[][] = new int[x][];  
for(int i = 0;i < arr.length;i++){  
 //给每一个一维数组开辟new空间  
 arr[i] = new int[i + 1];  
//每一个数组的打印数值 类似可以看之前写过的空心金字塔和菱形[代码](java7.18金字塔练习.docx)  
 for (int j = 0; j < arr[i].length; j++) {  
 if (j == 0 || j == (arr[i].length - 1)){  
 arr[i][j] = 1;  
 System.*out*.print(1);  
 System.*out*.print(" ");  
 }else {  
 arr[i][j] = (arr[i-1][j-1]) + (arr[i-1][j]);  
 System.*out*.print(arr[i][j]);  
 System.*out*.print(" ");  
 }  
 }  
 System.*out*.println();  
}

（7）二维数组的使用细节和注意事项

(1)一维数组的声明方式有:  
int arr1[]; 或者 int[] arr1;  
  
(2) 二维数组的声明方式有:  
int arr[][]; 或者 int[][] arr; 或 int[] arr[];  
  
(3)二维数组实际上是由多个一维数组组成的，它的各个一维数组的长度可以相同，  
 也可以不相同。比如： map[][] 是一个二维数组  
 int map [][] = {{1,2},{3,4,5}}  
 由map[0]是一个含有两个元素的一维数组，map[1] 是一个含有三个元素的一维数组构成，  
 我们也称为列数不等的二维数组

（8）练习：





