Java7.19 数组的使用2

1. 数组的赋值机制及图解

//数组的赋值机制  
//1) 基本数据类型赋值，这个值就是具体的数据，而且相互不影响。  
int n1 = 2;  
int n2 = n1;  
n2 = 10;//n1 和 n2 互不影响  
//2) 数组在默认情况下是引用传递，赋的值是地址。  
//数组的赋值方式是引用赋值 是一个地址 arr2变化会影响到arr1  
int arr1[] = {1,2,3};  
int arr2[] = arr1;  
arr2[0] = 99;  
System.*out*.println(arr1[0]);



图解 ：



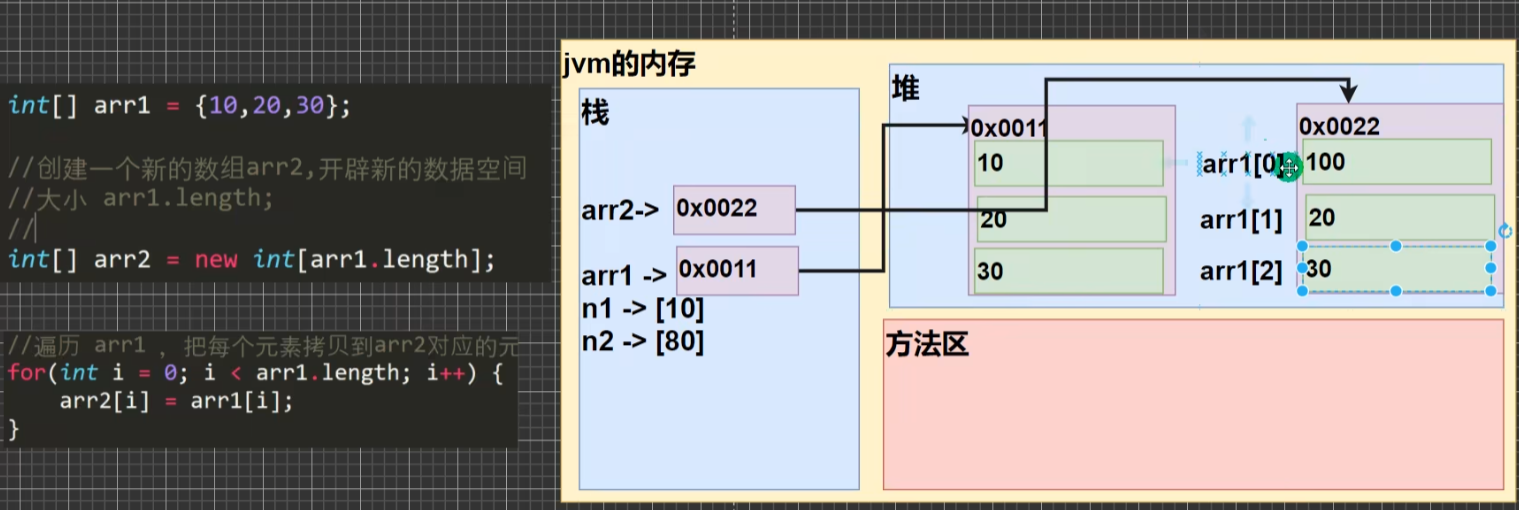


1. 数组拷贝

要求：实现数组拷贝内容赋值

//将int[] arr1 = {10,20,30}; 拷贝到arr2 数组, 要求数据空间是独立的.  
int arr1[] = {1,2,3};  
int arr2[] = new int[arr1.length];  
//遍历arr1 ，把每个元素拷贝到arr2 对应的元素位置  
for (int i = 0; i < arr1.length; i++){  
 arr2[i] = arr1[i];  
}  
//此时修改arr2，不会对arr1有影响.  
arr2[0] = 10;  
System.*out*.println(arr2[0]);//10  
System.*out*.println(arr1[0]);//1

图解：



1. 数组翻转

方法1：

找规律

//要求：把数组的元素内容反转。  
//arr {11,22,33,44,55,66} -> {66, 55,44,33,22,11}  
//思路分析：  
//1. 把arr[0] 和arr[5] 进行交换{66,22,33,44,55,11}  
//2. 把arr[1] 和arr[4] 进行交换{66,55,33,44,22,11}  
//3. 把arr[2] 和arr[3] 进行交换{66,55,44,33,22,11}  
//4. 一共要交换3次 = arr.length / 2  
//5. 每次交换时，对应的下标是arr[i] 和 arr[arr.length - 1 -i]  
int arr1[] = {11,22,33,44,55,66};  
for (int i = 0; i< (arr1.length / 2); i++){  
 //先定义一个临时变量  
 int temp = arr1[arr1.length - 1 - i];  
 arr1[arr1.length - 1 - i] = arr1[i];  
 arr1[i] = temp;  
}  
for (int i = 0; i < 6; i++){  
 System.*out*.print(arr1[i] + " ");  
}

方法2：

逆序赋值的方法

//数组的反转(方法2)  
//逆序赋值的方法  
//1.创建一个新的数组 arr2  
//2.逆序遍历arr1 每个元素拷贝到arr2  
//3.将arr2 赋值给 arr1  
int arr1[] = {11,22,33,44,55,66};  
int arr2[] = new int[arr1.length];  
for (int i = 6; i > 0; i--){  
 arr2[6-i] = arr1[i-1];  
}  
for (int i = 0; i < 6; i++){  
 System.*out*.print(arr2[i]+ " ");  
}  
System.*out*.println();  
arr1 = arr2;  
for (int i =0; i < 6;i++){  
 System.*out*.print(arr1[i]+ " ");  
}

另一种写法：

int arr1[] = {11,22,33,44,55,66};  
int arr2[] = new int[arr1.length];  
for (int i = arr1.length-1,j = 0; i >= 0 ; i--,j++) {  
 arr2[j] = arr1[i];  
}  
arr1 = arr2;  
for (int i = 0; i < arr1.length;i++){  
 System.*out*.print(arr1[i]+" ");  
}

1. 数组的添加(扩容)

//要求：实现动态的给数组添加元素效果，实现对数组扩容。  
//1) 原始数组使用静态分配int[] arr = {1,2,3}  
//2) 增加的元素4，直接放在数组的最后arr = {1,2,3,4}  
//3) 用户可以通过如下方法来决定是否继续添加，添加成功，是否继续？y/n  
int arr[] = {1,2,3};  
int arrNew[] = new int[arr.length + 1];  
for (int i = 0; i < arr.length;i++){  
 arrNew[i] = arr[i];  
}  
arrNew[arr.length] = 4;  
arr = arrNew;  
for(int i = 0;i < arr.length;i++){  
 System.*out*.print(arr[i]+"\t");  
}  
System.*out*.println();  
//自己练习所写：(关于数组添加动态的)  
Scanner myScanner = new Scanner(System.*in*);  
int arr1[] = {1,2,3};  
while(true) {  
 System.*out*.println("是否给数组arr1添加值");  
 char answer = myScanner.next().charAt(0);  
 if (answer == 'y'){  
 int arr2[] = new int[arr1.length + 1];  
 for (int i = 0; i < arr1.length; i++) {  
 arr2[i] = arr1[i];  
 }  
 System.*out*.println("请输入要添加的值");  
 int num = myScanner.nextInt();  
 arr2[arr1.length] = num;  
 arr1 = arr2;  
 for (int i = 0;i < arr1.length;i++){  
 System.*out*.print(arr1[i] + " ");  
 }  
 System.*out*.println();  
 }else if (answer == 'n'){  
 System.*out*.println("不添加就结束");  
 break;  
 }  
}

1. 数组缩减

//有一个数组{1， 2， 3， 4， 5}， 可以将该数组进行缩减，  
//提示用户是否继续缩减，每次缩减最后那个元素。当只剩下最后一个元素，提示，不能再缩减。  
int arr1[] = {1, 2, 3, 4, 5};  
Scanner myScanner = new Scanner(System.*in*);  
do {  
 System.*out*.println();  
 System.*out*.println("是否继续缩减");  
 char answer = myScanner.next().charAt(0);  
 if (answer == 'y') {  
 if (arr1.length > 1) {  
 int arr2[] = new int[arr1.length - 1];  
 for (int i = 0; i < arr2.length; i++) {  
 arr2[i] = arr1[i];  
 }  
 arr1 = arr2;  
 for (int i = 0; i < arr1.length; i++) {  
 System.*out*.print(arr1[i] + " ");  
 }  
 }else System.*out*.println("数组只剩下一个元素了，不能缩减了");  
 }else break;  
}while (true);  
System.*out*.println("结束");