

04-数组-作业

1.数组初始化考核

需求描述：

使用三种不同方式创建数组并初始化，数组包含a、b、c三个元素

答案

```
public class Test01_init {  
    public static void main(String[] args) {  
        //第一种  
        String[] arr1 = new String[] {"a", "b", "c"};  
        //第二种  
        String[] arr2 = {"a", "b", "c"};  
        //第三种  
        String[] arr3 = new String[3];  
        arr3 = new String[] {"a", "b", "c"};  
    }  
}
```

2.数组遍历考核

需求描述：

有数组 `String[] arr = new String[]{"hello","world","!"};` 请使用2种不同的方式进行遍历

答案

```

public class Test02_traversal {
    public static void main(String[] args) {
        String[] arr = new String[]{"hello", "world", "!"};
        //第一种方式：普通for循环
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }

        //第二种方式：增强for循环
        for (String str : arr) {
            System.out.println(str);
        }
    }
}

```

3.获取次大值

需求描述：

补全方法，实现求一个int[]数组的次大值

思路：根据数组的第一个和第二个元素的大小关系确定最大值和次大值从第三个元素开始依次比较和最大值、次大值的大小关系，确定是否需要改变最大值和次大值的值：

```

public static int getSecondValue(int[] arr){
    return 0;
}

```

```

public class Test03_secondValue {
    public static int getSecondValue(int[] arr) {
        // 默认使用第一个值作为最大值
    }
}

```

```

int max = (arr[0] > arr[1]) ? arr[0] : arr[1];
// 默认使用第二个值作为第二大值
int sec = (arr[0] > arr[1]) ? arr[1] : arr[0];
// 补全代码
for (int i = 2; i < arr.length; i++) {
    if (arr[i] > max) {
        // 第3个值比最大值大
        max = arr[i];
        sec = max;
    }else if (arr[i] > sec) {
        // 第3个值比最大值小，比次大值大
        sec = arr[i];
    }
    // else {
    //     // 第3个值比次大值小，无需改变
    // }
}
return sec;
}

public static void main(String[] args) {
    int[] arr = {12,4,56,45,67,34,52,28,65};
    int num = getSecondValue(arr);
    System.out.println("次大值: " + num);
}
}

```

4.增强for循环考核

需求描述：

分析以下代码运行结果

```
public class Test04_foreach {  
    public static void main(String[] args) {  
        int[] arr1 = {1,2,3};  
        for (int num : arr1){  
            num = 4;  
        }  
        System.out.println("输出: " + arr1[0]);  
    }  
}
```

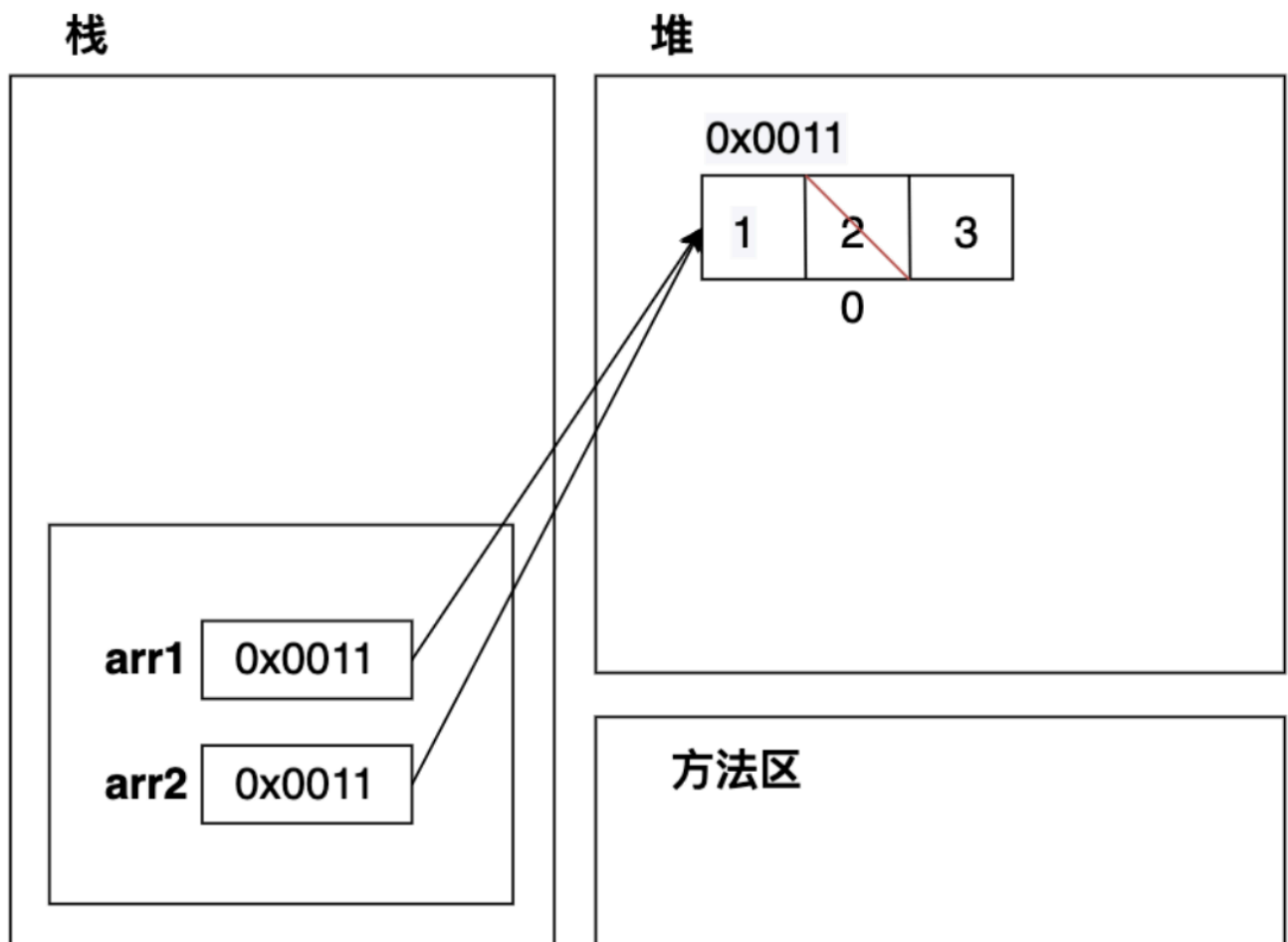
1

5.数组内存考核

需求描述:

分析以下代码的运行结果，并画出代码对应的内存图

```
public class Test05_memory {  
    public static void main(String[] args) {  
        int[] arr1 = {1,2,3};  
        int[] arr2 = arr1;  
        arr2[1] = 0;  
        System.out.println("arr1: " + Arrays.toString(arr1));  
        System.out.println("arr2: " + Arrays.toString(arr2));  
    }  
}
```



```
arr1: [1, 0, 3]
arr2: [1, 0, 3]
```

6.生成随机数

需求描述：

产生10个随机数，范围在[1, 100]，使用java.lang.Math类实现

答案

```
public class Test06_random {
    public static void main(String[] args) {
        // 定义存放生成的随机数的数组
    }
}
```

```

int[] arr = new int[10];

double random = 0;
// 循环遍历数组，为每个索引位置生成随机数，并输出生成的随机数
for (int i = 0; i < arr.length; i++) {
    random = Math.random();
    arr[i] = (int) (random * 100 + 1);
}

//打印数组
System.out.println("数组元素: "+Arrays.toString(arr));
}
}

```

7.生成验证码

需求描述：

随机产生一个长度为4位的验证码，包含大小写字母以及数字

答案

```

public class Test07_validateCode {
    public static void main(String[] args) {
        //创建存放验证码的数组
        char[] validateCode = new char[4];

        //创建存放大小写字母以及数字数组
        char[] data = new char[62];

        //给data输出添加元素值，提示：通过循环实现
        // 添加大写字母

```

```

char c = 'A';
for (int i = 0; i < 26; i++) {
    data[i] = c++;
}

// 添加小写字母
c = 'a';
for (int i = 26; i < 52; i++) {
    data[i] = c++;
}

// 添加数字
c = '0';
for (int i = 52; i < data.length; i++) {
    data[i] = c++;
}

//循环遍历，为存放验证码的数组每个位置赋值，并输出验证码
double random = 0;
int index = 0;
for (int i = 0; i < validateCode.length; i++) {
    random = Math.random(); // [0,1)
    index = (int) (random * data.length); // [0, 62)
    validateCode[i] = data[index];
}

//输出验证码
System.out.println("验证码内容: "+
Arrays.toString(validateCode));
}
}

```

8.统计次数

需求描述：

1. 产生100个[1,6]之间的随机数，并统计每个数字出现的概率
2. 可以创建2个数组分别保存随机数信息和统计产生次数信息
3. 例如：随机数生成了10个1，20个2，30个3，5个4，10个5，25个6，对应的统计数组内容[10,20,30,5,10,25]，即统计数组的每一个位置上存放的是对应下标出现的次数

答案

```
public class Test08_Count {  
    public static void main(String[] args) {  
        //定义一个长度为100的数组存放随机生成的数据  
        int[] randomArr = new int[100];  
        //定义统计数组  
        int[] countArr = new int[6];  
        //初始化数组randomArr  
        Random random = new Random();  
        for (int i = 0; i < randomArr.length; i++) {  
            int randomNum = random.nextInt(6) + 1;  
            randomArr[i] = randomNum;  
            countArr[randomNum - 1]++;  
        }  
        //统计每个数字出现的概率，输出每个数字出现的概率  
        for (int i = 0; i < countArr.length; i++) {  
            System.out.println("数字" + (i + 1) + "出现的频率  
为" + countArr[i] + "%");  
        }  
    }  
}
```