

Java基础

ZCX

一、一维数组

1. 数组的定义之一

- 定义：数组是一个数据容器，可用来存储一批**同类型的数据**
- **静态**初始化数组：

```
int[] arr = {1,2,3,4,5};  
String[] arr = {"zcx1","zcx2","zcx3"};
```

- **动态**初始化数组：

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

- 数组的**访问**：

```
int a = array[0]
```

- **获取数组长度**：

```
int a = array.length;
```

2. 数组的定义之二

- 定义数组可以先不存储具体的值：

```
int[] array = new int[8]; 只存储数据类型和长度，即动态存储
```

- 动态初始化数组元素默认值规则：

数据类型	明细	默认值
基本类型	byte、short、char、int、long	0
基本类型	float、double	0.0
基本类型	boolean	false
引用类型	类、接口、数组、String	null

3. 数组的遍历

- 一个一个数据的访问，可以通过for循环遍历
- 实例：请输入8为学生的成绩，并求其平均成绩，最小值，最大值

- ```

 package com.cxz;
 import java.util.Scanner;
 public class arraydemo_02 {
 public static void main(String[] args) {
 array_demo_02();
 }
 public static void array_demo_02(){
 double[] grade = new double[8]; 动态初始化数组
 Scanner sc = new Scanner(System.in);
 for(int i=0;i<8;i++){
 grade[i]=sc.nextDouble();
 } 通过for循环输入数组元素
 double average=0;
 double sum=0;
 for(int i=0;i<8;i++){
 sum+=grade[i];
 } 遍历数组求和
 average=sum/8.0;
 double min=grade[0];
 double max=grade[0];
 for(int i=1;i<8;i++){
 if(min>=grade[i]){
 min=grade[i];
 }
 else if(max<=grade[i]){
 max=grade[i];
 }
 } 遍历数组求最大值和最小值
 System.out.println("average:"+ average);
 System.out.println("max:"+ max);
 System.out.println("min:"+ min);
 }
 }

```

#### 4. 一维数组综合案例

- 目标：存储扑克牌54张(做牌)，并洗牌

- ```

        package com.cxz.array; 做牌操作
        public class array_test {
            public static void main(String[] args) {
                test();
            }
        }
    
```

```

public static void test(){
    //做牌，创建一个动态初始化数组存储54张牌。
    String[] cards=new String[54];
    //card=null...
    //准备四种花色和点数，再做牌存入数组中
    String[] colors={"♦","♠","♥","♣"};
    String[] points={"3","4","5","6","7","8","9","10","J","Q","K","A","2"};
    //拼装花色和点数
    for(int i=0;i<points.length;i++){
        for(int j=0;j<colors.length;j++){
            cards[i*4+j]=colors[j]+points[i];
        }
    } 通过循环嵌套实现，外层遍历点数，内层遍历花色
    cards[52]="小王";
    cards[53]="大王"; 牌组最后两位是王
    for(int i=0;i<points.length;i++){
        for(int j=0;j<colors.length;j++){
            System.out.print(cards[i*4+j]+"\t");
        }
        System.out.println();
    }
    System.out.println(cards[52]);
    System.out.println(cards[53]);
} 打印所有牌
}

```

• 洗牌思路

- a. 遍历数组，遍历到的元素与随机的一个索引对应的元素交换

- o for(int i=0;i<54;i++){ 遍历数组

int num=(int)(Math.random()*cards.length); 生成随机数索引

String a = cards[i]; 交换

cards[i]=cards[num];

cards[num]=a;

}

for(int i=0;i<cards.length;i++){ 打印

System.out.print(cards[i]);

}

- b. 遍历数组，每次随机的两个索引对应的元素交换

- o for(int i=0;i<54;i++){ 遍历数组

int num1=(int)(Math.random()*cards.length);

int num2=(int)(Math.random()*cards.length); 生成随机数索引两个

```

String a = cards[num1]; 交换
cards[num1]=cards[num2];
cards[num2]=cards[num1];
}
for(int i=0;i<cards.length;i++){ 打印
    System.out.print(cards[i]);
}

```

二、二维数组

1. 二维数组的定义

- 数组中的每个元素都是一维数组，这个数组就是二维数组
- 静态初始化：
- `int[][] arr = new int[][]{{1,2,3},{4,5,6},{7,8}}`
- 动态初始化：
- `int[][] arr = new int[3][5];`

2. 二维数组的访问

- 二维数组的访问，若只访问行索引就是访问一个一维数组：
- `int[] a = array[1];` **访问行，得到一个一维数组**
- 二维数组的访问，若同时访问行索引和列索引，就是访问一个元素：
- `int a = array[1][2];` **访问行和列，得到1行2列的那个元素**
- 获得二维数组的长度，若直接`arr.length`，就是获取二维数组的行数：
- `int a = array.length;` **获取二维数组的行数**
- 获得二维数组的长度，若`arr[1].length`，就是获取`arr[1]`这个一维数组的长度
- `int a = arr[1].length;` **获取arr[1]这个一维数组的长度**

3. 二维数组的遍历

- for循环嵌套遍历：
- `for(int i=0;i<classroom.length;i++){
 for(int j=0;j<classroom[i].length;j++){
 System.out.print(classroom[i][j]);
 }
 System.out.println();
 }`

第一个for循环遍历行数，第二个for循环遍历这一行的每一列

4. 二维数组综合案例

- 设计一个石头迷阵(数字华容道)，做到顺序插入数据后，再进行打乱

```
• package com.cxz.array;
public class arraydem {
public static void main(String[] args) {
start(5);
}
public static void start(int n) {
int[][] num=new int[n][n]; 声明一个二维数组
int a=1;
for(int i=0;i<n;i++){ 二维数组的初始化
for(int j=0;j<n;j++){
num[i][j]=a;
a++;
}
}
for(int i=0;i<n;i++){ 二维数组的输出
for(int j=0;j<n;j++){
System.out.print(num[i][j]+" ");
}
System.out.println();
}
System.out.println();
for(int i=0;i<n;i++){ 二维数组的打乱
for(int j=0;j<n;j++){
int u =(int)(Math.random()*n); 生成两个随机数索引，表示随机到的二维数组的行列
int v=(int)(Math.random()*n);
int swap=num[i][j]; 把遍历到的位置与随机到的位置交换
num[i][j]=num[u][v];
num[u][v]=swap;
}
}
for(int i=0;i<n;i++){ 输出打乱后的二维数组
for(int j=0;j<n;j++){
System.out.print(num[i][j]+" ");
}
System.out.println();
}
}
}
```