



# Java 核心技术

## 第十章 Java数据结构

### 第一节 数组

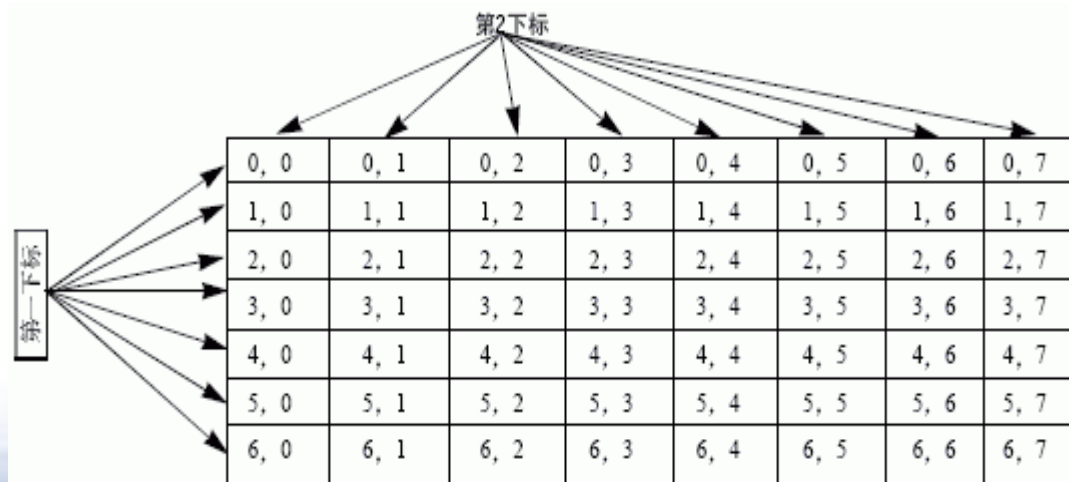
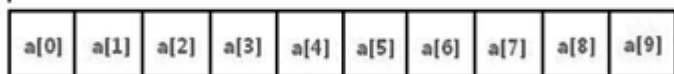
华东师范大学 陈良育

# 数组(1)



- 数组是一个存放 **多个数据** 的容器
  - 数据是同一种类型
  - 所有的数据是线性规则排列
  - 可通过位置索引来快速定位访问数据
  - 需明确容器的长度

数组a, int型



# 数组(2)



- Java数组定义和初始化

```
int a[]; //a 还没有new操作 实际上是null, 也不知道内存位置  
int[] b; //b 还没有new操作 实际上是null, 也不知道内存位置  
int[] c = new int[2]; //c有2个元素, 都是0  
c[0] = 10; c[1] = 20; //逐个初始化
```

```
int d[] = new int[]{0,2,4}; //d有3个元素, 0,2,4, 同时定义和初始化  
int d1[] = {1,3,5}; //d1有3个元素, 1,3,5 同时定义和初始化
```

```
//注意声明变量时候没有分配内存, 不需要指定大小, 以下是错误示例  
//int e[5];  
//int[5] f;  
//int[5] g = new int[5];  
//int h[5] = new int[5];
```

# 数组(3)



- 数组索引

- 数组的length属性标识数组的长度
- 从0开始，到length - 1
- `int[] a = new int[5];` //a[0]~a[4], not a[5]; a.length是5
- 数组不能越界访问，否则会报

`ArrayIndexOutOfBoundsException` 异常



# 数组(4)



- 数组遍历：两种方法

```
//需要自己控制索引位置  
for(int i=0;i<d.length;i++)  
{  
    System.out.println(d[i]);  
}
```

```
//无需控制索引位置  
for(int e : d)  
{  
    System.out.println(e);  
}
```

# 数组(5)



- 多维数组

- 数组的数组

- 存储是按照行存储原则

//规则数组

```
int a[][] = new int[2][3];
```

//不规则数组

```
int b[][];
```

```
b = new int[3][];
```

```
b[0]=new int[3];
```

```
b[1]=new int[4];
```

```
b[2]=new int[5];
```

```
int k = 0;
for(int i=0;i<a.length;i++)
{
    for(int j=0;j<a[i].length;j++)
    {
        a[i][j] = ++k;
    }
}

for(int[] items : a)
{
    for(int item : items)
    {
        System.out.print(item + ", ");
    }
    System.out.println();
}
```

# 数组(6)



- 总结

- 数组是一种确定大小的存储同种数据的容器
- 初始化和遍历方法，特别是两种遍历方式

# 代码(1) ArrayTest.java



```
public class ArrayTest {  
    public static void main(String[] args) {  
  
        int a[]; //a 还没有new操作 实际上是null, 也不知道内存位置  
        int[] b; //b 还没有new操作 实际上是null, 也不知道内存位置  
        int[] c = new int[2]; //c有2个元素, 都是0  
        c[0] = 10; c[1] = 20; //逐个初始化  
  
        int d[] = new int[]{0,2,4}; //d有3个元素, 0,2,4, 同时定义和初始化  
        int d1[] = {1,3,5}; //d1有3个元素, 1,3,5 同时定义和初始化  
  
        //注意声明变量时候没有分配内存, 不需要指定大小, 以下是错误示例  
        //int e[5];  
        //int[5] f;  
        //int[5] g = new int[5];  
        //int h[5] = new int[5];  
  
        //需要自己控制索引位置  
        for(int i=0;i<d.length;i++) {  
            System.out.println(d[i]);  
        }  
  
        //无需控制索引位置  
        for(int e : d) {  
            System.out.println(e);  
        }  
    }  
}
```



# 代码(2) MultiDimArrayTest.java



```
public class MultiDimArrayTest {  
  
    public static void main(String[] args)  
        //Java多维数组是按照行存储  
  
        //规则数组  
        int a[][] = new int[2][3];  
        //不规则数组  
        int b[][];  
        b = new int[3][];  
        b[0]=new int[3];  
        b[1]=new int[4];  
        b[2]=new int[5];  
}
```

```
    int k = 0;  
    for(int i=0;i<a.length;i++)  
    {  
        for(int j=0;j<a[i].length;j++)  
        {  
            a[i][j] = ++k;  
        }  
    }  
  
    for(int[] items : a)  
    {  
        for(int item : items)  
        {  
            System.out.print(item + ", ");  
        }  
        System.out.println();  
    }  
}
```



谢谢!