Tedu.cn 达内教育

Java语言基础

JSD BASICS DAY05

未来已来 | 教与学的革命

Tedu.cn 达内教育

技能回顾

- 分支结构switch…case：
  1)优点：效率高、结构清晰
  2)缺点：只能对整数判断相等----也支持String和枚举
- 循环：
  1)反复多次执行一段相同或相似的代码
  2)循环三要素：循环变量初始化、循环条件、循环变量改变
  3)循环结构：while、do…while、for
- break和continue
  1)break:跳出循环
  2)continue:跳过循环体中剩余语句而进入下一次循环

今日目标

01 掌握嵌套循环的使用规则

02 了解什么是数组？数组解决什么问题？

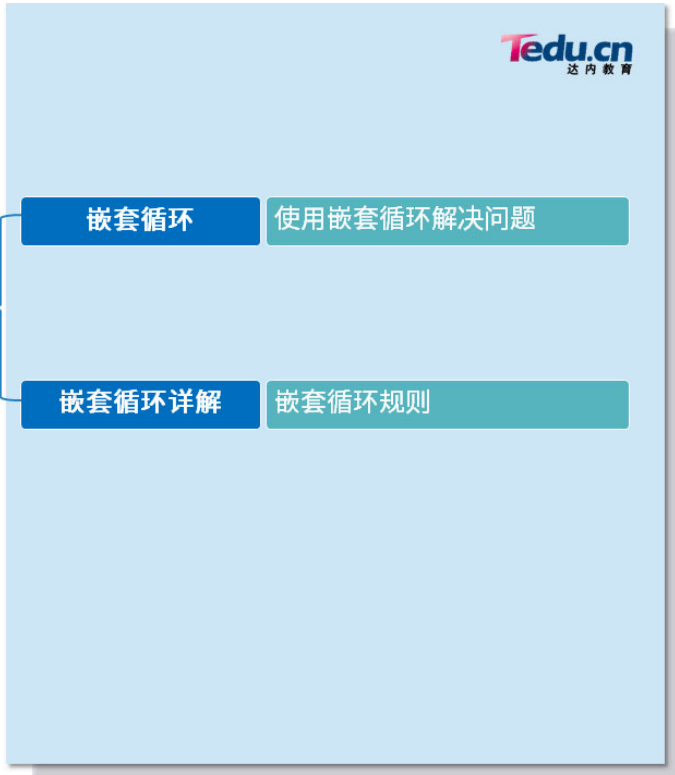
03 掌握数组的定义、初始化

04 掌握数组的访问、遍历

05 掌握数组的复制、排序

06 预计代码量：课上130行、作业260行

01 嵌套循环



嵌套循环

使用嵌套循环解决问题

- 打印九九乘法表：

课堂练习

```
1*1=1
1*2=2  2*2=4
1*3=3  2*3=6  3*3=9
1*4=4  2*4=8  3*4=12 4*4=16
1*5=5  2*5=10 3*5=15 4*5=20 5*5=25
1*6=6  2*6=12 3*6=18 4*6=24 5*6=30 6*6=36
1*7=7  2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49
1*8=8  2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64
1*9=9  2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81
```

使用嵌套循环解决问题

知识讲解

```
for(int num=1;num<=9;num++){
    for(int i=1;i<=num;i++){
        System.out.println(i+"*"+num+"="+i*num+"\t");
    }
    System.out.println(); //换行
}
```

num=1
- i=1 1*1=1
- i=2 false
- 换行

num=2
- i=1 1*2=2
- i=2 2*2=4
- i=3 false
- 换行

num=3
- i=1 1*3=3
- i=2 2*3=6
- i=3 3*3=9
- i=4 false
- 换行

……

嵌套循环详解

嵌套循环规则

知识讲解

- 循环中套循环，常常多行多列时使用，外层循环控制行，内层循环控制列

```
for(int num=1;num<=9;num++){      //控制行
    for(int i=1;i<=num;i++){      //控制列
        System.out.print(i+"*"+num+"="+i*num+"\t");
    }
    System.out.println();//换行
}
```

```
1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
1*6=6   2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7   2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8   2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9   2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81
```

嵌套循环规则(续)

知识讲解

- 执行规则：外层循环走一次，内层循环走所有次

```
for(int i=1;i<=100;i++){
    for(int j=1;j<=200;j++){
        for(int k=1;k<=300;k++){
            System.out.println("我爱Java");
        }
    }
}
```

我爱Java，会输出多少次？ 6000000次

嵌套循环规则(续)

知识讲解

- 建议：嵌套层数越少越好，能用一层就不用两层，能用两层就不用三层…
- break默认只能跳出当前一层循环

```
for(int i=1;i<=100;i++){
    for(int j=1;j<=200;j++){
        for(int k=1;k<=300;k++){
            if(k>=100){
                break;//跳出k层循环
            }
            System.out.println("我爱Java");
        }
    }
}
```

02 数组

什么是数组——数据的组合

为什么用数组

知识讲解

- 实现：存储100名学生的成绩

```
double score0,score1,score2,score3,score4,score5,……,score100;
```

- 实现：找到这100名学生成绩当中的最高分

```
int max = score0;
if(score1>max){ … }
if(score2>max){ … }
if(score3>max){ … }
…
if(score100>max){ … }
```

问题：重复、麻烦

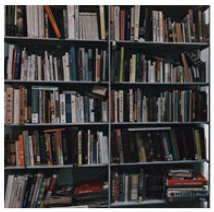
用数组解决

数组的概念

知识讲解

- 相同数据类型元素的集合

数据

- 是一种引用数据类型
- 将一组类型相同、逻辑相关的数据，存到数组中

数组的定义

定义数组

知识讲解

```
int   a;  //声明一个整型变量，名为a
int[ ] b; //声明一个整型数组变量，名为b

int   a = 5;
int[ ] b = 5;  //编译错误，数据类型不匹配
int[ ] b =  new int[4];
```

定义数组(续)

知识讲解

- 定义数组的语法：

  数据类型[ ] 数组名 = new 数据类型[大小];

  int[ ] arr = new int[5];

- 定义数组：

```
//声明整型数组a，包含3个元素，每个元素都是int类型，默认值为0
1) int[ ] a =  new int[3];

//声明浮点型数组，包含10个元素，每个元素都是double类型，默认值为0.0
2) double[ ] d =  new double[10];

//声明布尔型数组，包含26个元素，每个元素都是boolean类型，默认值为false
3) boolean[ ] b =  new boolean[26];
```

定义数组(续)

知识讲解

- 数组元素默认值规则：
  1) byte,short,int,long,char----------默认值为0
  2) float,double----------默认值为0.0
  3) boolean----------默认值为false

数组的初始化

初始化数组

知识讲解

- 给元素初始化：

1) int[ ] arr = new int[3];　//0,0,0

2) int[ ] arr = {2,5,8};　//2,5,8

3) int[ ] arr = new int[ ]{2,5,8};　//2,5,8

[ ]中不能写元素个数了

4) int[ ] arr;

arr = {2,5,8};　//编译错误，此方式只能声明同时初始化

arr = new int[ ]{2,5,8};　//正确

数组的访问

获取数组的长度

- 数组的长度即元素的个数
- 通过数组名.length可以获取数组的长度

知识讲解

```
int[ ] arr = new int[3];
System.out.println(“数组的长度为”+arr.length);  //数组的长度为: 3
```

通过下标访问数组的元素

```
int[ ] arr = new int[3];   //0,0,0
//给第2个元素赋值为100⟶给每个元素标个号
```

知识讲解

```
int a = 0;
int b = 0;
int c = 0;
//给第2个数赋值为100
b = 100;
```

通过下标访问数组的元素(续)

知识讲解

- 通过下标/索引来访问元素，下标从0开始，最大到(数组长度-1)

```
int[ ] arr = new int[3];
arr[0] ----------代表arr中的第1个元素（int)
arr[1] ----------代表arr中的第2个元素(int)
arr[2] ----------代表arr中的第3个元素(int)

//给第2个元素赋值为100
arr[1] = 100;
```

数组的遍历

遍历数组

- 遍历/迭代: 从头到尾挨个走一遍

知识讲解

```
int[ ] arr = new int[10];
arr[0] = 100;
arr[1] = 100;
arr[2] = 100;
arr[3] = 100;
arr[4] = 100;
arr[5] = 100;
arr[6] = 100;
arr[7] = 100;
arr[8] = 100;
arr[9] = 100;
```

需求:给每个元素都赋值为100

→ 大量重复

遍历数组(续)

知识讲解

```
int[ ] arr = new int[10];
for( int i=0; i<arr.length; i++ )      //遍历arr数组
   arr[i] = 100;                       //给每个元素赋值为100
   System.out.println( arr[i] );       //输出每个元素的值
}
```

i<10 代表arr.length

求数组元素的最大值

- 定义一个整型数组，包含10个元素，每个元素为0到99的随机数，并输出
- 找到数组元素中的最大值，并输出

课堂练习

Tedu.cn 达内教育

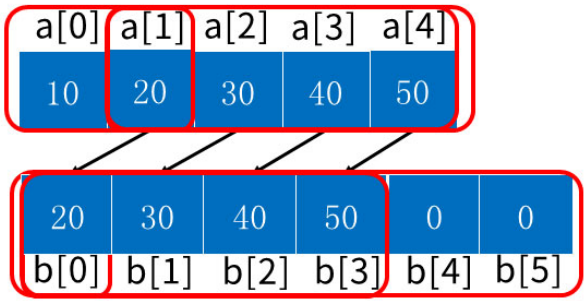
数组的复制

System.arraycopy()数组的复制

知识讲解

- 使用System.arraycopy( ) 可以实现数组的复制

```
int[ ] a   = { 10 ,20 ,30 ,40 ,50 };
int[ ] b = new int[ 6 ] ;     //0,0,0,0,0,0
//a: 源数组
//1: 源数组的起始下标
//b: 目标数组
//0: 目标数组的起始下标
//4: 要复制的元素个数
System.arraycopy( a , 1 , b , 0 , 4 );
```



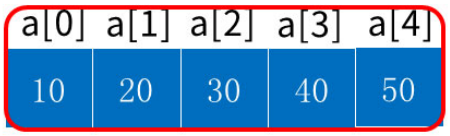
Arrays.copyOf()数组的复制

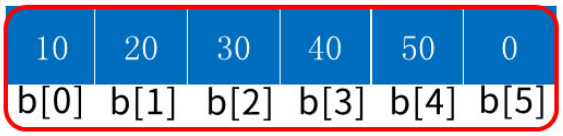
知识讲解

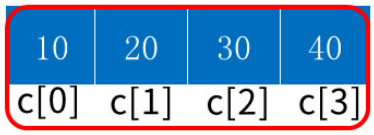
- 使用Arrays.copyOf( ) 可以实现数组的复制

```
int[ ] a   = { 10 ,20 ,30 ,40 ,50 };
//a: 源数组
//b: 目标数组
//6: 目标数组的长度
int[ ] b = Arrays.copyOf(a, 6);

int[ ] c = Arrays.copyOf(a, 4);
```





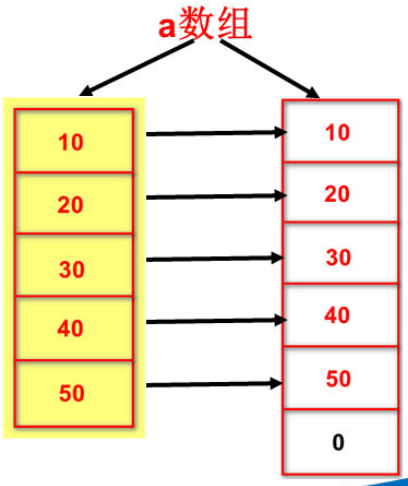


数组的扩容

知识讲解

- 数组的长度在创建数组后是不可改变的，数组元素的内存空间是连续的
- 所谓扩容是指创建一个更大的新数组并将源数组的内容复制进去

```
int[ ] a  = { 10 ,20 ,30 ,40 ,50 };
a = Arrays.copyOf(a, a.length+1);    //扩容
```



求数组元素的最大值并将其放在数组最后一个元素的下一位置

课堂练习

- 定义一个整型数组，包含10个元素，每个元素为0到99的随机数，并输出
- 找到数组元素中的最大值，并输出
- 将最大值放在数组最后一个元素的下一个位置

数组的排序

数组排序

知识讲解

- 排序是对数组施加的最常用的算法；
- 所谓排序，指将数组元素按照从小到大(升序)或从大到小(降序)的顺序重新排列；
- 常见排序算法有：冒泡排序、插入排序、快速排序、选择排序、希尔排序等。

使用Arrays.sort()对数组排序

知识讲解

- Java提供的Arrays.sort() 封装了数组的排序算法：

```
int[ ] arr = { 49, 81, 1, 64, 77, 50, 0, 54, 77, 18 };
Arrays.sort ( arr ) ;//对arr数组做升序排列
for( int i=0; i<arr.length; i++) {
    System.out.println( arr[i] );
}
```

输出结果：0 1 18 49 50 54 64 77 77 81

总结 | SUMMARY

- **嵌套循环执行规则：外层循环走一次，内层循环走所有次**
- **数组：相同数据类型元素的集合，是一种引用数据类型**
- **数组的操作：**
  1) **定义**
  2) **初始化**
  3) **访问**
  4) **遍历**
  5) **复制**
  6) **排序**

小T提示

- 嵌套循环的应用场景有哪些
- 数组的应用场景有哪些
- 数组的初始化方式有哪些
- 冒泡排序算法、选择排序算法、快速排序算法的实现

未来已来

AI助力IT技术的全新革命