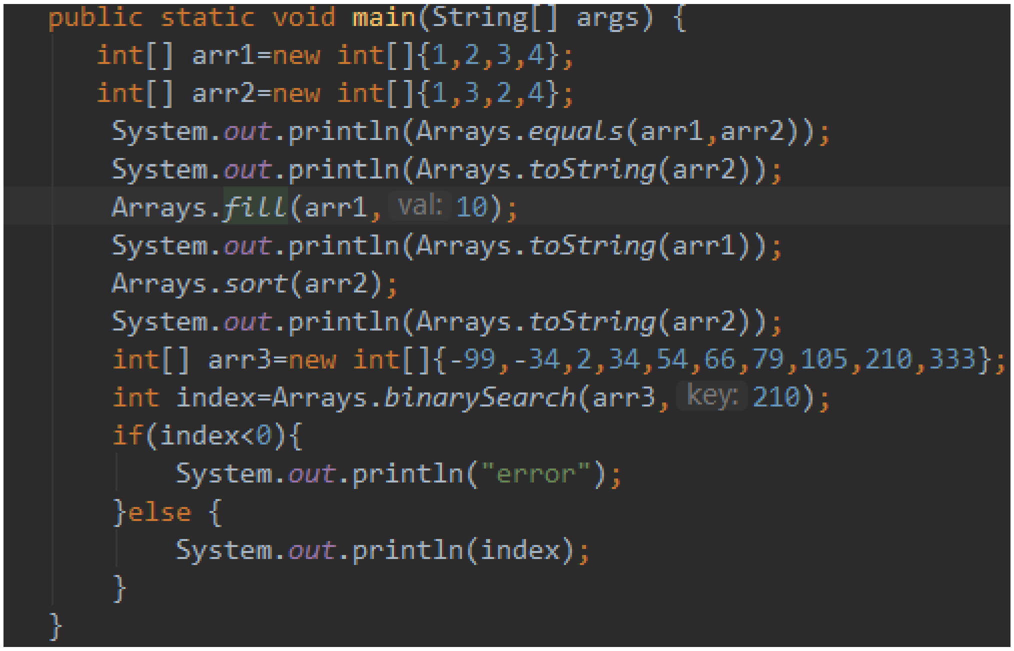
2021.5.16 dmd.java基础 6-15 数组

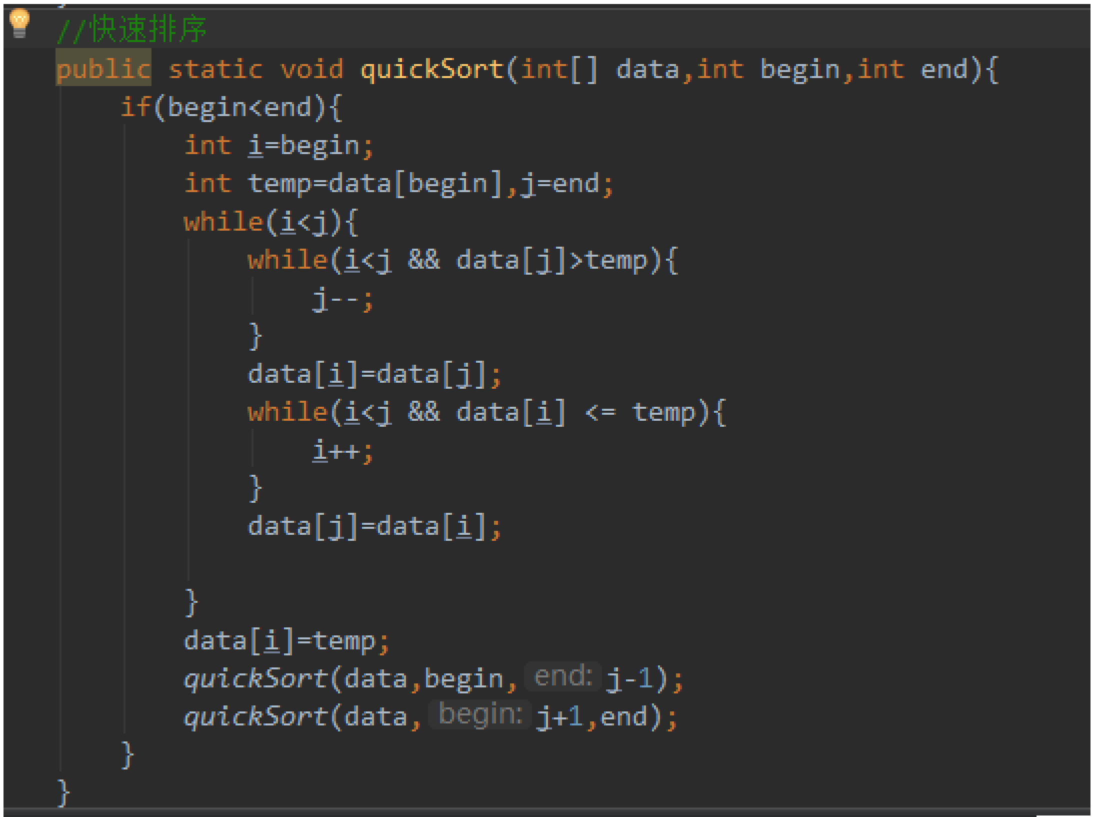
笔记本:	我的第一个笔记本		
创建时间:	2021/5/16 11:16	**更新时间:**	2021/5/16 14:44
作者:	dmd1810411032@163.com		

2021.5.16 dmd.java基础 6-15 数组

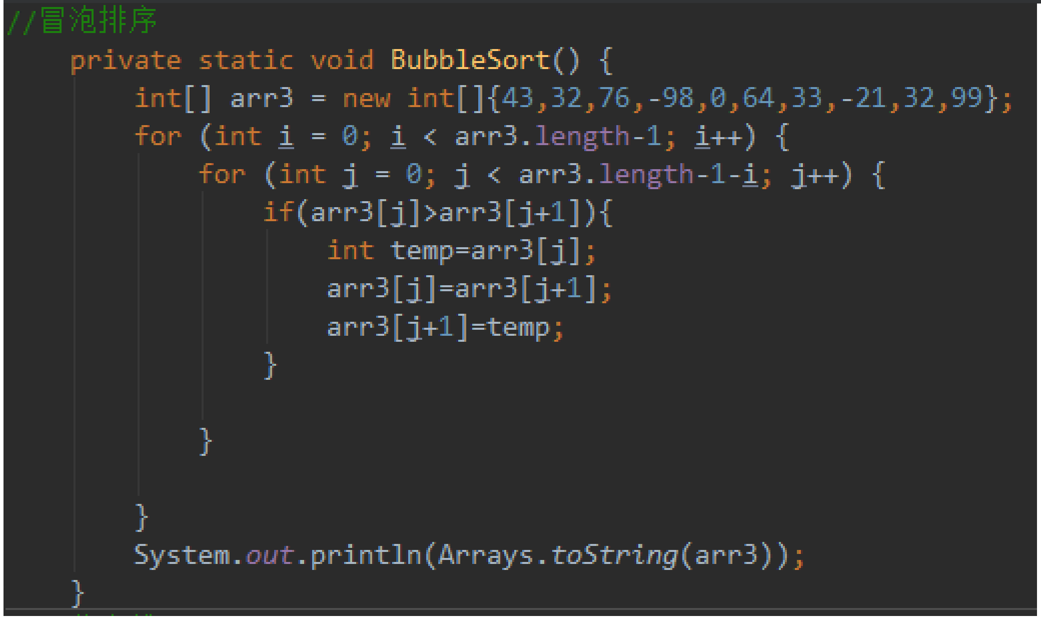
对于字符串来说，比较内容最好用.equals(),不要直接是==。不然可能比较的是地址。



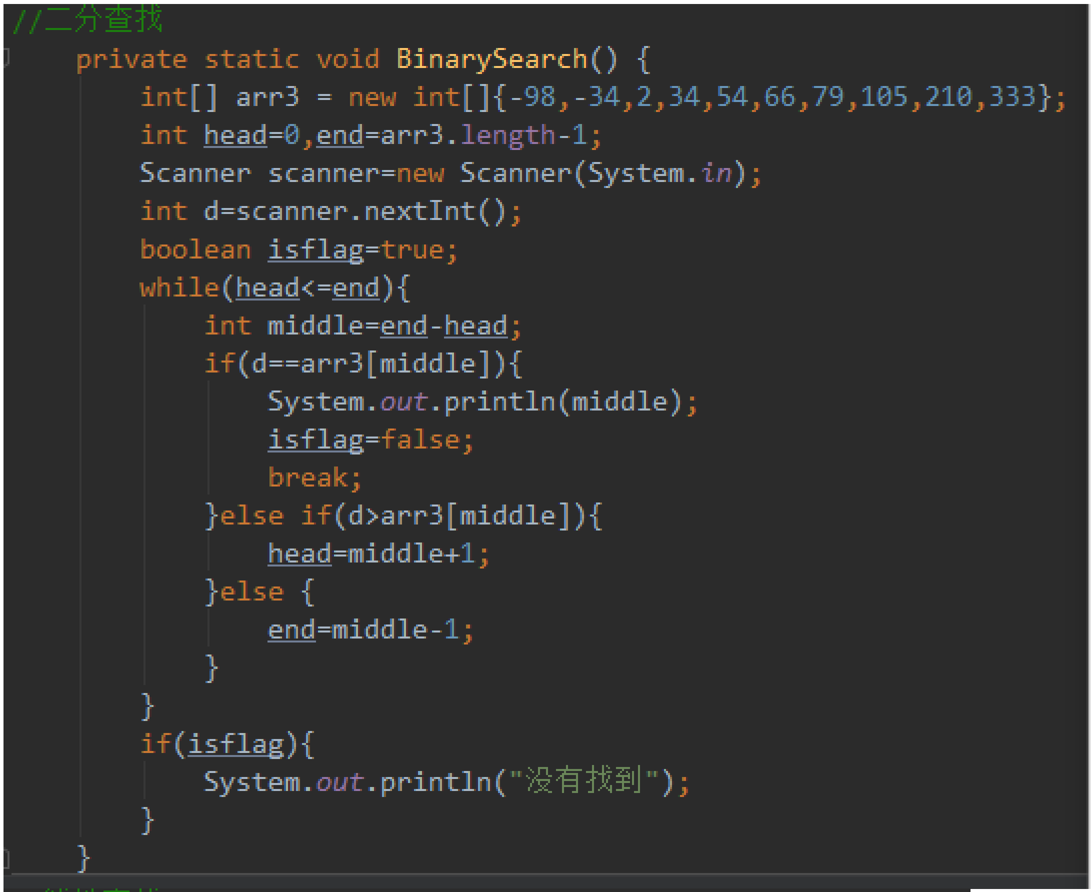
```
public static void main(String[] args) {
    int[] arr1=new int[]{1,2,3,4};
    int[] arr2=new int[]{1,3,2,4};
     System.out.println(Arrays.equals(arr1,arr2));
     System.out.println(Arrays.toString(arr2));
     Arrays.fill(arr1, val: 10);
     System.out.println(Arrays.toString(arr1));
     Arrays.sort(arr2);
     System.out.println(Arrays.toString(arr2));
     int[] arr3=new int[]{-99,-34,2,34,54,66,79,105,210,333};
     int index=Arrays.binarySearch(arr3, key: 210);
     if(index<0){
         System.out.println("error");
     }else {
         System.out.println(index);
     }
}
```



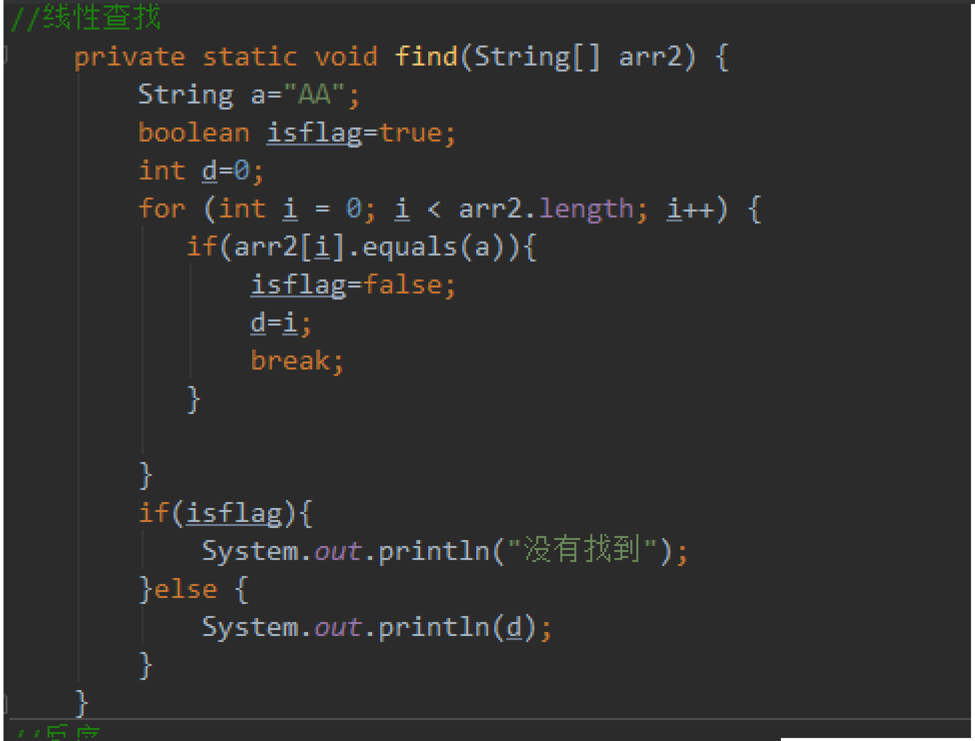
```
//快速排序
public static void quickSort(int[] data,int begin,int end){
    if(begin<end){
        int i=begin;
        int temp=data[begin],j=end;
        while(i<j){
            while(i<j && data[j]>temp){
                j--;
            }
            data[i]=data[j];
            while(i<j && data[i] <= temp){
                i++;
            }
            data[j]=data[i];

        }
        data[i]=temp;
        quickSort(data,begin, end: j-1);
        quickSort(data, begin: j+1,end);
    }
}
```



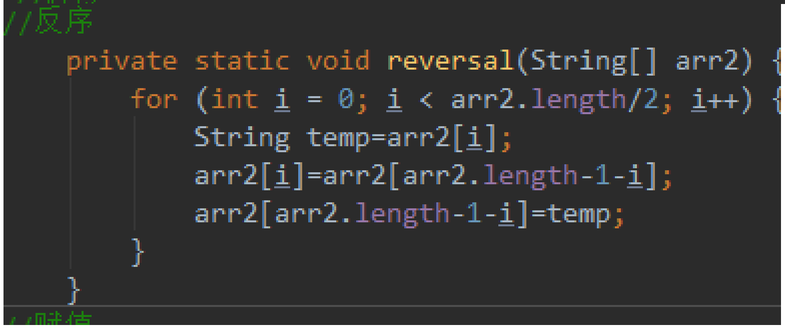
```
//冒泡排序
    private static void BubbleSort() {
        int[] arr3 = new int[]{43,32,76,-98,0,64,33,-21,32,99};
        for (int i = 0; i < arr3.length-1; i++) {
            for (int j = 0; j < arr3.length-1-i; j++) {
                if(arr3[j]>arr3[j+1]){
                    int temp=arr3[j];
                    arr3[j]=arr3[j+1];
                    arr3[j+1]=temp;
                }

            }

        }
        System.out.println(Arrays.toString(arr3));
    }
```



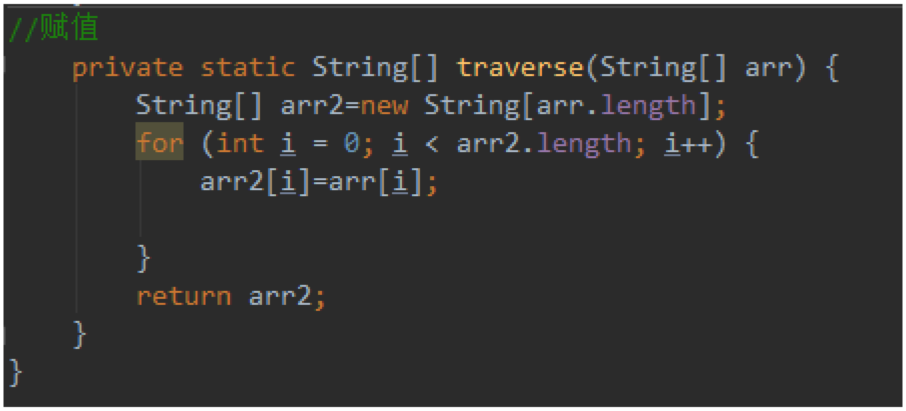
```
//二分查找
    private static void BinarySearch() {
        int[] arr3 = new int[]{-98,-34,2,34,54,66,79,105,210,333};
        int head=0,end=arr3.length-1;
        Scanner scanner=new Scanner(System.in);
        int d=scanner.nextInt();
        boolean isflag=true;
        while(head<=end){
            int middle=end-head;
            if(d==arr3[middle]){
                System.out.println(middle);
                isflag=false;
                break;
            }else if(d>arr3[middle]){
                head=middle+1;
            }else {
                end=middle-1;
            }
        }
        if(isflag){
            System.out.println("没有找到");
        }
    }
```



```
//线性查找
    private static void find(String[] arr2) {
        String a="AA";
        boolean isflag=true;
        int d=0;
        for (int i = 0; i < arr2.length; i++) {
           if(arr2[i].equals(a)){
               isflag=false;
               d=i;
               break;
           }

        }
        if(isflag){
            System.out.println("没有找到");
        }else {
            System.out.println(d);
        }
    }
```



```
//反序
    private static void reversal(String[] arr2) {
        for (int i = 0; i < arr2.length/2; i++) {
            String temp=arr2[i];
            arr2[i]=arr2[arr2.length-1-i];
            arr2[arr2.length-1-i]=temp;
        }
    }
//赋值
```



```
//赋值
    private static String[] traverse(String[] arr) {
        String[] arr2=new String[arr.length];
        for (int i = 0; i < arr2.length; i++) {
            arr2[i]=arr[i];

        }
        return arr2;
    }
}
```