

# Binary Search - Java

---

ПРЕПОДАВАТЕЛ: ИНЖ. ВЕСЕЛИНА МАРИНОВА

## Binary Search in Java

Двоичното търсене се използва за търсене на ключов елемент от множество елементи.

Методът е по-бърз от Линейното търсене.

Двоичното търсене се използва при сортиран масив в нарастващ ред.

### Алгоритъм:

Търсенето се осъществява с повтарящи се разделения на интервала от елементи на половина.

Ако стойността на ключа е по-малка от средният елемент, се взема интервала преди този елемент.

И процесът се повтаря докато се открие търсеният ключ или интервалът е празен.

Идеята на Двоичното търсене в сортиран масив намалява времето за изпълнение:  
Сложност =  $O(\log n)$ .

# Binary Search

Search 23

0	1	2	3	4	5	6	7	8	9
2	5	8	12	16	23	38	56	72	91

23 > 16  
take 2<sup>nd</sup> half

L=0	1	2	3	M=4	5	6	7	8	H=9
2	5	8	12	16	23	38	56	72	91

23 > 56  
take 1<sup>st</sup> half

0	1	2	3	4	L=5	6	M=7	8	H=9
2	5	8	12	16	23	38	56	72	91

Found 23,  
Return 5

0	1	2	3	4	L=5, M=5	H=6	7	8	9
2	5	8	12	16	23	38	56	72	91

```
public class BinarySearch1 {  
    public static void binarySearch(int arr[], int first, int last, int key){  
        int mid = (first + last)/2;  
        while( first <= last ){  
            if ( arr[mid] < key ) first = mid + 1;  
            else  
                if ( arr[mid] == key ){  
                    System.out.println("Element is found at index: " + mid); break; }  
                else last = mid - 1;  
            mid = (first + last)/2;  
        }  
        if ( first > last ) System.out.println("Element is not found!");  
    }  
  
    public static void main(String[] args) {  
        int arr[] = {10,20,30,40,50};  
        int key = 30;  
        int last=arr.length-1;  
        binarySearch(arr,0,last,key);  
    }  
}
```



метод



Массив - int

```
package demo;  
import java.util.*;
```

Метод – ArrayList - String

```
public class Example_ArrayList {  
    public static void binarySearch(ArrayList<String> listx, int first, int last, String key){  
        int mid = (first + last)/2;  
        while( first <= last ){  
            if ( listx.get(mid).compareTo(key)<0) first = mid + 1;  
            else  
                if ( listx.get(mid).equals(key))  
                    { System.out.println("Element is found at index: " + mid);  
                      break;  
                    }  
                else last = mid - 1;  
            mid = (first + last)/2;  
        }  
        if ( first > last ) System.out.println("Element is not found!");  
    }  
}
```

Next

```
public static void main(String[] args) {
```

```
    ArrayList<String> list1=new ArrayList<>();
```

```
    list1.add("Ana");
```

```
    list1.add("Dani");
```

```
    list1.add("Georgi");
```

```
    list1.add("Toni");
```

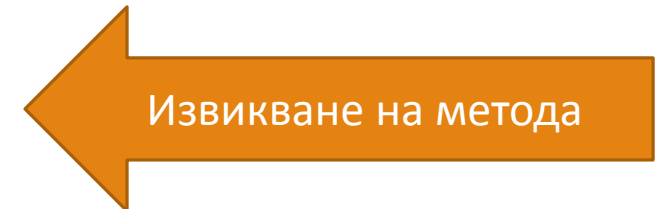
```
    String key1="Georgi";
```

```
    int last=list1.size()-1;
```

```
    binarySearch(list1,0,last,key1);
```

```
}
```

```
}
```



```
package demo;
import java.util.*;
public class Example_double {
    public static double arr[] = new double[100];
    public static void binarySearch(double arr[], int first, int last, double key){
        int mid = (first + last)/2;
        while( first <= last ){
            if ( arr[mid] < key ) first = mid + 1;
            else
                if ( arr[mid] == key ){ System.out.println("Element is found at index: " + mid);
                                     break; }
                else last = mid - 1;
            mid = (first + last)/2;
        }
        if ( first > last ) System.out.println("Element is not found!");
    }
}
```

Метод `binarySearch()`

Next

```
public static void InsertionSort(double arr[],int n){  
    double temp;  
    int i,j;  
    for ( i = 1; i < n; i++)  
    {  
        temp = arr[i];  
        j = i;  
        while ((j > 0)&& (arr[j-1]> temp))  
        {  
            arr[j] = arr[j-1];  
            j--;  
        }  
        arr[j] = temp;  
    }  
}
```

Метод InsertionSort()

Next



```
public static void main(String[] args) {  
    Scanner kb=new Scanner(System.in);  
    int i,n;  
    do{System.out.println("Въведи размерност за масива");  
    n=kb.nextInt();  
    }while(n<0 || n>100);  
    System.out.println("Въведи елементите на масива:");  
    for(i=0;i<n;i++)  
        arr[i]=kb.nextDouble();  
    InsertionSort(arr,n);  
    for(i=0;i<n;i++)  
        System.out.println("sort:"+arr[i]);  
    System.out.println("Въведи елемент за търсене:");  
    double key =kb.nextDouble() ;  
    int last=n-1;  
  
    binarySearch(arr,0,last,key);  
  
    kb.close();  
}
```



Извикване на метода за сортиране

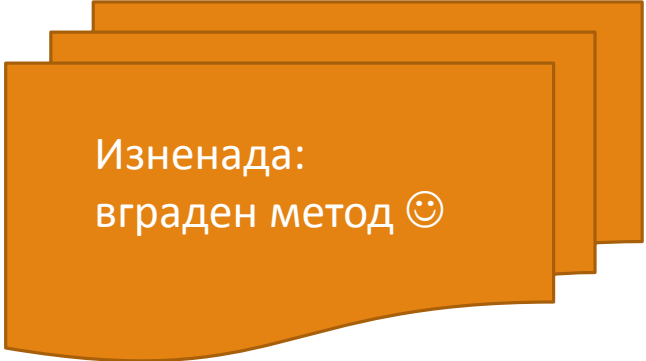


Извикване на метода за търсене

```
package demo;
import java.util.Arrays;
public class Example2 {

    public static void main(String[] args) {
        // използване на вградените методи
        int arr[] = {10,20,30,40,50};
        int key = 30;
        int result = Arrays.binarySearch(arr,key);
        if (result < 0)
            System.out.println("Element is not found!");
        else
            System.out.println("Element is found at index: "+result);
    }

}
```



Изненада:  
вграден метод 😊

**Благодаря за вниманието!**