Grundkurs

Die Hilfs-Klasse Arrays

André Wiechens
12.04.2018

Die Hilfsklasse Arrays

Class Arrays
java.lang.Object
    java.util.Arrays

Die Hilfsklasse java.util.Arrays deklariert nützliche statische Methoden im Umgang mit Arrays. So bietet sie Möglichkeiten zum Vergleichen, Sortieren und Füllen von Feldern sowie zur binären Suche.

Methoden der Klasse Arrays

Class Arrays
java.lang.Object
java.util.Arrays

Methodenname	Beschreibung
binarySearch(**array, value**)	Gibt den Index von ‚value' in dem (sortierten) array (oder < 0 bei Fehlen)
binarySearch(**array, minIndex, maxIndex, value**)	Wie oben nur im Bereich zwischen den Indexes min|max
copyOf(**array, length**)	Gibt die Kopie eines neuen Arrays in der vorgegebenen Länge zurück
equals(**array1, array2**)	Gibt ‚true' zurück, wenn beide Arrays gleiche Inhalte in gleicher Reihenfolge haben

Methoden der Klasse Arrays

Class Arrays
java.lang.Object
java.util.Arrays

Methodenname	Beschreibung
fill(**array, value**)	Setzt jedes Element auf den Wert von ‚value‘
sort (**array**)	Ordnet die Elemente in sortierter Reihenfolge an
toString(**array**)	Gibt eine Zeichenfolge zurück, die das Array darstellt, z. B. „[30, 29, -76, 0]“
asList(**T... a**)	Gibt eine Liste fester Größe zurück, die vom angegebenen Array unterstützt wird.

Beispiel 1

```
1   package themen.arrays;
2   import java.util.Arrays;
3   public class Arrays1 {
4       public static void main(String[] args) {
5           String[] namen = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
            System.out.println(namen);
            System.out.println(namen.toString());
8           System.out.println(Arrays.toString(namen));
9           Arrays.sort(namen);
10          System.out.println(Arrays.toString(namen));
11      }
12  }
```

```
run:
[Ljava.lang.String;@6d06d69c
[Ljava.lang.String;@6d06d69c
[Zora, Gabi, Tamara, Michaela, Bärbel, Katja]
[Bärbel, Gabi, Katja, Michaela, Tamara, Zora]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 2

```
1   package themen.arrays;
2   import java.util.Arrays;
3   public class Arrays2 {
4       public static void main(String[] args) {
5           String[] namen1 = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
6           String[] namen2 = Arrays.copyOf(namen1, 6);
7           System.out.println("Inhalt von namen1: " + Arrays.toString(namen1));
8           System.out.println("Inhalt von namen2: " + Arrays.toString(namen2));
9           System.out.println("Beide Arrays sind gleich?! " + Arrays.equals(namen1, namen2));
10      }
11  }
```

```
run:
Inhalt von namen1: [Zora, Gabi, Tamara, Michaela, Bärbel, Katja]
Inhalt von namen2: [Zora, Gabi, Tamara, Michaela, Bärbel, Katja]
Beide Arrays sind gleich?!    true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 3

```java
package themen.arrays;
import java.util.Arrays;
public class Arrays3 {
    public static void main(String[] args) {
        String[] namen1 = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
        String[] namen2 = Arrays.copyOf(namen1, 6);
        System.out.println("Inhalt von namen1: " + Arrays.toString(namen1));
        Arrays.sort(namen2);
        System.out.println("Inhalt von namen2: " + Arrays.toString(namen2));
        System.out.println("Beide Arrays sind gleich?! " + Arrays.equals(namen1, namen2));
    }
}
```

```
run:
Inhalt von namen1: [Zora, Gabi, Tamara, Michaela, Bärbel, Katja]
Inhalt von namen2: [Bärbel, Gabi, Katja, Michaela, Tamara, Zora]
Beide Arrays sind gleich?! false
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 4

```
1   package themen.arrays;
2   import java.util.Arrays;
3   public class Arrays4 {
4       public static void main(String[] args) {
5           String[] namen = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
6           String[] namen1 = Arrays.copyOf(namen, 7);
7           String[] namen2 = Arrays.copyOf(namen, 5);
8           System.out.println("Inhalt von namen1: " + Arrays.toString(namen1));
9           System.out.println("Inhalt von namen2: " + Arrays.toString(namen2));
10      }
11  }
```

```
run:
Inhalt von namen1: [Zora, Gabi, Tamara, Michaela, Bärbel, Katja, null]
Inhalt von namen2: [Zora, Gabi, Tamara, Michaela, Bärbel]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 5

```
1  package themen.arrays;
2  import java.util.Arrays;
3  public class Arrays5 {
4      public static void main(String[] args) {
5          String[] namen = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
6          String[] namen1 = Arrays.copyOf(namen, 7);
7          String[] namen2 = Arrays.copyOf(namen, 5);
8          Arrays.sort(namen);
9          String[] namen3 = namen;
10         System.out.println("Katja ist im Array namen1 an der Stelle " + Arrays.binarySearch(namen1, "Katja"));
11         System.out.println("Katja ist im Array namen2 an der Stelle " + Arrays.binarySearch(namen2, "Katja"));
12         System.out.println("Katja ist im Array namen3 an der Stelle " + Arrays.binarySearch(namen3, "Katja"));
13     }
14 }
```

```
run:
Katja ist im Array namen1 an der Stelle -3
Katja ist im Array namen2 an der Stelle -1
Katja ist im Array namen3 an der Stelle 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 6

```java
package themen.arrays;
import java.util.Arrays;
public class Arrays6 {
    public static void main(String[] args) {
        String[] namen = {"Zora", "Gabi", "Tamara", "Michaela", "Bärbel", "Katja"};
        String[] namen1 = Arrays.copyOf(namen, 7);
        Arrays.fill(namen, "John Doe");
        String[] namen2 = namen;
        System.out.println("Vorher: " + Arrays.toString(namen1));
        System.out.println("Nachher: " + Arrays.toString(namen2));

    }
}
```

```
run:
Vorher: [Zora, Gabi, Tamara, Michaela, Bärbel, Katja, null]
Nachher: [John Doe, John Doe, John Doe, John Doe, John Doe, John Doe]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Beispiel 7

```java
package themen.arrays;
import java.util.Arrays;
import java.util.List;
public class Arrays7 {
    public static void main(String[] args) {
        List<String> stooges = Arrays.asList("Larry", "Moe", "Curly");
        System.out.println(stooges);
    }
}
```

```
run:
[Larry, Moe, Curly]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Aufgabe 1

Given the following two methods, which method call will not compile?

```
public void printStormName(String... names) {
        System.out.println(Arrays.toString(names));
}public void printStormNames(String[] names) {
        System.out.println(Arrays.toString(names));
}
```

- A. printStormName("Arlene");
- B. printStormName(new String[] { "Bret" });
- C. printStormNames("Cindy");
- D. printStormNames(new String[] { "Don" });

Aufgabe 2

What are the names of the methods to do searching and sorting respectively on arrays?

- A. Arrays.binarySearch() and Arrays.linearSort()
- B. Arrays.binarySearch() and Arrays.sort()
- C. Arrays.search() and Arrays.linearSort()
- D. Arrays.search() and Arrays.sort()

Aufgabe 3

What does this code output?

```
String[] nums = new String[] { "1", "9", "10" };
Arrays.sort(nums);
System.out.println(Arrays.toString(nums));
```

- A. [1, 9, 10]
- B. [1, 10, 9]
- C. [10, 1, 9]
- D. None of the above

Aufgabe 4

Which statement most accurately represents the relationship between searching and sorting with respect to the Arrays class?

- A. If the array is not sorted, calling Arrays.binarySearch() will be accurate, but slower than if it were sorted.
- B. The array does not need to be sorted before calling Arrays.binarySearch() to get an accurate result.
- C. The array must be sorted before calling Arrays.binarySearch() to get an accurate result.
- D. None of the above

Aufgabe 5

What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "Mac"));
```

- A. 0
- B. 1
- C. 2
- D. The output is not defined.

Aufgabe 6

What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "RedHat"));
```

- A. -1
- B. -2
- C. -3
- D. The output is not defined.

Aufgabe 7

What is the output of the following when run as java unix.EchoFirst seed flower?

```
package unix;
import java.util.*;
public class EchoFirst {
        public static void main(String[] args) {
                String one = args[0];
                Arrays.sort(args);
                int result = Arrays.binarySearch(args, one);
        System.out.println(result);
        }
}
```

- A. 0
- B. 1
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

Aufgabe 8

What is the result of the following when called as java counting.Binary?

```
package counting;
import java.util.*;
public class Binary {
        public static void main(String... args) {
                Arrays.sort(args);
                System.out.println(Arrays.toString(args));
        }
}
```

- A. null
- B. []
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

Aufgabe 9

What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
System.out.println(Arrays.binarySearch(os, "Linux"));
```

- A. 0
- B. 1
- C. 2
- D. The output is not defined.

Aufgabe 10

What does the following output?

```
String[] os = new String[] { "Linux", "Mac", "Windows" };
System.out.println(Arrays.binarySearch(os, "Linux"));
```

- A. 0
- B. 1
- C. 2
- D. The output is not defined.

Aufgabe 11

What is the output of the following when run as java unix.EchoFirst seed flower?

```
package unix;
import java.util.*;
public class EchoFirst {
	public static void main(String[] args) {
		Arrays.sort(args);
		String result = Arrays.binarySearch(args, args[0]);
		System.out.println(result);
	}
}
```

- A. 0
- B. 1
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

Aufgabe 12

What does the following code output?

```
List<String> drinks = Arrays.asList("can", "cup");
for (int container = drinks.size() - 1; container >= 0; container )
System.out.print(drinks.get(container) + ",");
```

- A. can,cup,
- B. cup,can,
- C. The code does not compile.
- D. None of the above

Aufgabe 13

What does the following code output?

```
public static void main(String[] args) {
        List<String> bottles = Arrays.asList("glass", "plastic");
        for (int type = 0; type < bottles.size();) {
                System.out.print(bottles.get(type) + ",");
                break;
        }
        System.out.print("end");
}
```

- A. glass,end
- B. glass,plastic,end
- C. The code does not compile.
- D. None of the above