

# Методите reverse, copy и fill при работа с колекции в Java

В този урок ще покажем 3 метода, които са от класа **Collections** \* reverse - разменя елементите на списък, като първият става последен, последният първи и т.н. \* copy - копира елементите от един списък в друг \* fill - запълва всички елементи от даден списък с дадена променлива

## Подготовка

Като за начало ще създадем един масив от символи, след което ще от него ще създадем списък.

```
Character[] array = {'p', 'w', 'n'};
```

Този път за копирането използваме

```
List<Character> list = Arrays.asList(array);
```

*Този начин за преобразуване на масив в списък е по-кратък, но не е за предпочитане да се използва в истински програми, защото често се случва да се получи грешка.*

Нека след това да принтираме нашият масив в конзолата.

```
System.out.println("List is : ");  
output(list);
```

Методът **output()**; все още не съществува. Него ще напишем в края на урока.

## reverse()

Методът **reverse()** е от класа **Collections** и приема като единствен параметър колекция, а след това разменя елементите в нея.

```
// reverse and print out the list  
Collections.reverse(list);  
System.out.println("After reverse: ");  
output(list);
```

## copy()

Методът **copy()** е от класа **Collections** и приема два параметъра. Първият е списъкът, в който да се копират елементите, а вторият е списъкът, източник на информацията.

Преди да копираме обаче ще трябва да направим нов списък. За целта ще направим втори масив с 3 празни клетки и от него ще направим новия списък.

```
// create new array and new list
Character[] newArray = new Character[3];
List<Character> listCopy = Arrays.asList(newArray);
```

След това, използвайки методът **copy()**, ще копираме информацията от стария списък в новия.

```
// copy contents of list into listCopy
Collections.copy(listCopy, list);
System.out.println("Copy of list: ");
output(listCopy);
```

## fill()

Методът **fill()** е от класа **Collections** и приема два параметъра. Първият е колекция, а вторият е променлива, с която да бъдат заменени всички елементи в колекцията.

```
// fill collection with crap
Collections.fill(list, 'X');
System.out.println("After filling the list: ");
output(list);
```

## методът за принтиране в конзолата output()

За да работи нашата програма, трябва да напишем и методът за принтиране на колекции в конзолата. Това ще направим с цикъл, който минава през всички елементи на колекцията и принтира техните стойности, след което принтира празен ред.

```
//output method
public static void output(List<Character> thelist){
    for(Character thing : thelist){
        System.out.printf("%s ", thing);
    }
    System.out.println();
}
```

## Краен резултат

Цялата програма изглежда така

```
import java.util.*;
class Demo{
    public static void main(String[] args){
```

```

// create an array and convert to list
Character[] array = {'p', 'w', 'n'};
List<Character> list = Arrays.asList(array);
System.out.println("List is : ");
output(list);

// reverse and print out the list
Collections.reverse(list);
System.out.println("After reverse: ");
output(list);

// create new array and new list
Character[] newArray = new Character[3];
List<Character> listCopy = Arrays.asList(newArray);

// copy contents of list into listCopy
Collections.copy(listCopy, list);
System.out.println("Copy of list: ");
output(listCopy);

// fill collection with crap
Collections.fill(list, 'X');
System.out.println("After filling the list: ");
output(list);
}

//output method
public static void output(List<Character> thelist){
    for(Character thing : thelist){
        System.out.printf("%s ", thing);
    }
    System.out.println();
}
}

```

это го и изхода в конзолата

```

>>> Demo
>>> List is :
>>> p w n
>>> After reverse:
>>> n w p
>>> Copy of list:
>>> n w p
>>> After filling the list:
>>> X X X

```