

Introduction to Programming

Rania Baghernejad

Java – part 07.1

Varargs

مباحث

- ٢ تعریف آرگومان (Argument)
- ٢ تعریف پارامتر (Parameter)
- ٢ مبحث Varargs (...args)

تعریف آرگومان (Argument)

آرگومان مقداری است که هنگام فراخوانی یک متد یا تابع، به آن ارسال می‌شود.

یعنی وقتی شما یک متد را صدا می‌زنید (call)، مقدارهایی که داخل پرانتز قرار می‌گیرند، آرگومان نام دارند.

تعریف پارامتر (Parameter)

پارامتر متغیری است که در تعریف متد به کار می‌رود، و انتظار دریافت مقدار را دارد.

در واقع، پارامترها هنگام تعریف متد نوشته می‌شوند، ولی آرگومان‌ها هنگام استفاده از آن متد وارد می‌شوند.

مثال ساده

```
public static void greet(String name) { // ← "name" is the parameter
    System.out.println("Hello, " + name);
}

greet("Sara"); // ← "Sara" is the argument
```

تحلیل:

- در تعریف متد greet، متغیر name یک پارامتر است.
 - هنگام فراخوانی متد با greet("Sara")، مقدار "Sara" یک آرگومان است که به پارامتر name منتقل می‌شود.
-

مبحث Varargs (...args)

تعریف

در جاوا، اگر بخواهیم متدی تعریف کنیم که بتواند تعداد متغیری از آرگومان‌ها را به عنوان ورودی بگیرد، از varargs استفاده می‌کنیم. این قابلیت به ما اجازه می‌دهد تا بدون تعریف چندین نسخه‌ی مختلف از یک متد (overloading)، یک متد انعطاف‌پذیرتر داشته باشیم.

نحوه تعریف

از ... پیش از نام آرایه استفاده می‌شود، و باید آخرین پارامتر متد باشد.

```
public static void printNumbers(int... numbers) {  
    for (int num : numbers) {  
        System.out.println(num);  
    }  
}
```

در اینجا numbers یک آرایه‌ی از نوع int است که می‌تواند صفر یا چند مقدار را دریافت کند.

نحوه استفاده

```
printNumbers();           // No argument  
printNumbers(5);          // One argument  
printNumbers(1, 2, 3, 4); // Multiple arguments
```

در هر بار فراخوانی، مقادیر به صورت یک آرایه به متد ارسال می‌شوند.

محدودیت‌ها

۱. تنها یک varargs در تعریف متد مجاز است.

۲. varargs باید در آخرین موقعیت پارامترها باشد.

○ مجاز: method(String name, int... scores)

○ غیرمجاز: method(int... scores, String name)

کاربرد عملی

فرض کنید متدی بنویسیم که میانگین نمرات یک دانشجو را محاسبه کند:

```
public static double calculateAverage(double... scores) {  
    if (scores.length == 0) return 0;  
  
    double sum = 0;  
    for (double score : scores) {
```

```
        sum += score;
    }
    return sum / scores.length;
}
```

فراخوانی:

```
double avg1 = calculateAverage(18.5, 17, 19.75);
double avg2 = calculateAverage(20);
double avg3 = calculateAverage(); // returns 0
```

تفاوت با آرایه معمولی

ممکن است بپرسید: «آیا varargs همان آرایه نیست؟»

در پاسخ: از نظر پیاده‌سازی در زمان اجرا بله، varargs در واقع یک آرایه است؛ اما مزیت آن این است که استفاده از آن در فراخوانی متد ساده‌تر و قابل‌انعطاف‌تر است. یعنی نیازی نیست ابتدا آرایه‌ای بسازید.
