METHODS AND ARGUMENTS: HANDLING PARAMETERS IN JAVA

Mahkamov Shohruh Sarvar o‘g‘li
Assistant professor, Sambhram university, Jizzax
Dandu Jayabharath Reddy
Assistant professor, Sambhram university, Jizzax
Aynakulov Toxir Turg‘un o‘g‘li
Assistant professor, Jizzakh branch of the National University of Uzbekistan
Saitqulov Shavkat Abdunabi o‘g‘li,
Student, Sambhram university, Jizzax

Annotatsiya: Ushbu maqola Java dasturlash tilida metodlar va ularning argumentlari bilan ishlashni o‘rganishga bag‘ishlangan. Metodlarning parametrlari qanday uzatilishi, metodga o‘zgartiriladigan va o‘zgartirilmaydigan qiymatlar yuborilishi, shuningdek, varargs (o‘zgaruvchan argumentlar) va parametrlarni to‘g‘ri boshqarish usullari haqida tushunchalar beriladi. Maqola, shuningdek, pass-by-value va pass-by-reference tushunchalari va ularning Java’dagi amaliy qo‘llanishini ham tahlil qiladi.

Kalit so‘zlar: Java metodlari, argumentlar, pass-by-value, pass-by-reference, varargs, parametrlar, metodlarning ishlatilishi, metod overloading.

Аннотация: Эта статья посвящена изучению работы с методами и их аргументами в языке программирования Java. Рассматривается передача параметров в методы, отправка изменяемых и неизменяемых значений в методы, а также использование varargs (переменное количество аргументов) и управление параметрами. Также анализируются концепции pass-by-value и pass-by-reference и их практическое применение в Java.

Ключевые слова: Методы Java, аргументы, pass-by-value, pass-by-reference, varargs, параметры, использование методов, перегрузка методов.

Abstract: This article focuses on the handling of methods and their arguments in the Java programming language. It explores how method parameters are passed, the difference between passing mutable and immutable values, and the proper usage of varargs (variable-length arguments) and parameter management. Additionally, the concepts of pass-by-value and pass-by-reference are analyzed, along with their practical application in Java.

Keywords: Java methods, arguments, pass-by-value, pass-by-reference, varargs, parameters, method usage, method overloading.

In the Java programming language, methods are one of the core constructs that enable developers to perform actions without repeating code. By utilizing methods, repetitive tasks can be encapsulated into distinct functions that can be invoked whenever needed. A key aspect of using methods effectively lies in how variables, known as parameters, are passed to them.

Java follows the pass-by-value principle when passing arguments to methods, meaning values are passed as copies. This requires developers to have a solid understanding of how this impacts mutable and immutable data types. Additionally, Java allows the use of varargs, which enables methods to accept a variable number of arguments.

Pass-by-value - In Java, all arguments are passed as copies. This means that if a parameter is modified within a method, the original value remains unchanged outside the method.

```
    public class Article {
public static void changeValue(int number) {
  number = 10;
```

```
    }
    public static void main(String[] args) {
        int originalValue = 5;
        changeValue(originalValue);
        System.out.println("Original value: " + originalValue); // 5
    }
}
```

In this example, even though originalValue is changed inside the method, its original value remains intact because Java passes the value as a copy, following the pass-by-value principle.

Working with mutable objects - Although objects are also passed by value in Java, if the internal state of the object is modified, the changes will reflect in the original object.

```
        public class Article {
    public static void changeArray(int[] arr) {
        arr[0] = 10;
    }
    public static void main(String[] args) {
        int[] originalArray = {1, 2, 3};
        changeArray(originalArray);
        System.out.println("Original array: " + originalArray[0]); // 10
    }
}
```

In this case, the first element of the array is modified inside the method, and this change is reflected in the original array, as the array reference is passed by value but points to the same object.

Varargs (variable-length arguments) - Java provides the varargs technique to allow methods to accept a variable number of arguments.

```
        public class Article {
    public static void printNumbers(int... numbers) {
        for (int number : numbers) {
            System.out.println(number);
        }
    }
    public static void main(String[] args) {
        printNumbers(1, 2, 3, 4, 5); // 1 2 3 4 5
    }
}
```

In this example, the printNumbers method uses varargs, allowing it to accept any number of arguments, making it more flexible for various inputs.

The results of the study reveal that Java follows the pass-by-value principle when passing arguments to methods. This principle means that values are passed as copies, and the original value remains unchanged. This behavior is evident when working with primitive data types—although a value may be modified inside the method, the original variable retains its initial value.

However, working with objects introduces certain complexities in understanding this principle. While Java still passes objects by value, what is actually passed is a copy of the object reference. Therefore, when an object's internal state is modified within a method, the changes are reflected outside the method as well. This distinction between primitive types and object references is crucial for developers to grasp, as it directly impacts how data is manipulated and how side effects occur in the code.

Additionally, the use of the varargs technique allows methods to accept a variable number of arguments, making them more flexible and adaptable to different use cases. However, care must be taken when using varargs, as improper usage can lead to inefficiencies and reduce the

readability of the code. When not handled correctly, methods utilizing varargs may become less efficient or harder to debug.

The discussion highlights the importance of understanding the differences between pass-by-value and pass-by-reference for Java developers. Correctly applying these concepts can significantly optimize code and help prevent errors when working with methods and arguments. The careful use of varargs and proper understanding of reference manipulation will lead to more robust and maintainable codebases.

Working with methods and their arguments in Java requires developers to have a solid understanding of key technical concepts. This study examined how Java passes values to methods based on the pass-by-value principle. The results show that when primitive types are passed to methods, their values are copied, and any changes made within the method do not affect the original value.

However, when dealing with objects, references are passed, leading to changes in the internal state of the object within the method. This distinction is critical for developers to manage when working with both objects and primitive data types.

Additionally, the varargs technique enhances the flexibility of Java methods by allowing them to accept a variable number of arguments. Nonetheless, careful handling is required, as improper use of varargs can lead to complications in method readability and performance.

In conclusion, Java developers must fully understand concepts like pass-by-value and object references. Mastering these principles is essential for improving code efficiency, clarity, and maintainability.

1. Foydalanilgan adabiyotlar:

2. Махкамов, Ш. (2023). Теоретические основы базы данных (мб) и системы управления базами данных (мббт). Информатика и инженерные технологии, 1(1), 90-94.

3. Аликулов, С. Т., & Махкамов, Ш. С. (2018). Дидактическая компетентность педагога. In Развитие интеллектуально-творческого потенциала молодежи: из прошлого в современность (pp. 150-152).

4. Махкамов, Ш. (2024). Davlat boshqaruv tizimining vujudga kelishi hamda davlat va uning moxiyati. Новый Узбекистан: наука, образование и инновации, 1(1), 379-382.

5. Norqo‘ziyev, Q., Aynakulov, T., Mahkamov, S., & Sodiqov, D. (2024). IMPLEMENTING MOBILE ROBOT WALKING PATH CONTROL AND TRACKING IN C# WINDOWS FORM APPLICATION. International Journal of scientific and Applied Research, 1(3), 95-99.

6. Mahkamov, S., Aynakulov, T., & Baratov, J. (2024). ANDROIDDA MODEL VIEW PRESENTER (MVP) ARXITEKTURASI. International Journal of scientific and Applied Research, 1(3), 199-202.

7. Sarvar o’g’li, M. S. (2024). MODEL VIEW PRESENTER (MVP) ARXITEKTURASI. Ta'limning zamonaviy transformatsiyasi, 8(4), 186-191.

8. qizi Akramova, M. A. (2023). 3D MODELLASHTIRISHNING HOZIRGI ZAMONDAGI O’RNI VA AHAMIYATI.

9. Toxir Turg‘un o‘g, A., & Komiljon o‘g, N. Z. Q. (2024). CALCULATION OF SPECIAL FUNCTIONS WITH ARBITRARY PRECISION IN THE. NET FRAMEWORK ENVIRONMENT. Лучшие интеллектуальные исследования, 19(3), 142-149.

10. Aynakulov, T. (2024). MOBIL HISOBLASH PLATFORMASIGA ASOSLANGAN AQLLI BINAURAL ESHITISH APPARATI ARXITEKTURASI. International Journal of scientific and Applied Research, 1(3), 119-122.

11. Rakhmonkulov, F. P. (2022). THE IMPORTANCE OF SOFTWARE IN WEBSITE DESIGN. In СОВРЕМЕННАЯ НАУКА, ОБЩЕСТВО И ОБРАЗОВАНИЕ: АКТУАЛЬНЫЕ ВОПРОСЫ, ДОСТИЖЕНИЯ И ИННОВАЦИИ (pp. 129-131).