Java Method Overloading

Method Overloading

In Java, two or more methods may have the same name if they differ in parameters (different number of parameters, different types of parameters, or both). These methods are called overloaded methods and this feature is called method overloading.

Suppose, you have to perform the addition of given numbers but there can be any number of arguments (let's say either 2 or 3 arguments for simplicity).

In order to accomplish the task, you can create two methods sum2num(int, int) and sum3num(int, int, int) for two and three parameters respectively. However, other programmers, as well as you in the future may get confused as the behavior of both methods are the same but they differ by name.

The better way to accomplish this task is by overloading methods. And, depending upon the argument passed, one of the overloaded methods is called. This helps to increase the readability of the program.

With method overloading, multiple methods can have the same name with different parameters:

Example

```
int myMethod(int x)

float myMethod(float x)

double myMethod(double x, double y)
```

Consider the following example, which has two methods that add numbers of different type:

Example

```
static int plusMethodInt(int x, int y) {

  return x + y;

}
```

```
static double plusMethodDouble(double x, double y) {
  return x + y;
}

public static void main(String[] args) {
  int myNum1 = plusMethodInt(8, 5);
  double myNum2 = plusMethodDouble(4.3, 6.26);
  System.out.println("int: " + myNum1);
  System.out.println("double: " + myNum2);
}
```

Instead of defining two methods that should do the same thing, it is better to overload one.

In the example below, we overload the plusMethod method to work for both int and double:

Example

```
static int plusMethod(int x, int y) {
  return x + y;
}

static double plusMethod(double x, double y) {
  return x + y;
}

public static void main(String[] args) {
  int myNum1 = plusMethod(8, 5);
```

```
  double myNum2 = plusMethod(4.3, 6.26);

  System.out.println("int: " + myNum1);

  System.out.println("double: " + myNum2);

}
```

Note:

- Multiple methods can have the same name as long as the number and/or type of parameters are different.
- Two or more methods can have the same name inside the same class if they accept different arguments. This feature is known as method overloading.

Method overloading is achieved by either:

- changing the number of arguments.
- or changing the data type of arguments.
- It is not method overloading if we only change the return type of methods. There must be differences in the number of parameters.

References

- https://www.javatpoint.com/method-overloading-in-java
- https://www.programiz.com/java-programming/method-overloading
- https://www.w3schools.com/java/java_methods_overloading.asp