Difference between concat() and + operator in Java

concat() Method

The Java String concat() method concatenates one string to the end of another string. This method returns a string with the value of the string passed into the method, appended to the end of the string.

Example:

```
// Java program to demonstrate
// working of concat() method

class Gfg {
    public static void main(String args[])
    {
        String s = "Gfg";
        s = s.concat("! is the best.");
        System.out.println(s);
    }
}
```

Output:

```
Gfg! is the best.
```

+ operator

+ operator is used to concatenate strings on either side.

Example:

```
// Java program to demonstrate
// working of concat() method

class Gfg {
    public static void main(String args[])
    {
        String s1 = "Gfg";
        String s2 = "! is the best";

        String s3 = s1 + s2;

        System.out.println(s3);
    }
}
```

Output:

```
Gfg! is the best.
```

Although concat() and + operator are both used for concatenation of strings, but there are some differences between them:

1. **Number of arguments the concat() method and + operator takes:**
   - **concat()** method takes only one argument of string and concat it with other string.
   - **+ operator** takes any number of arguments and concatenates all the strings.

```
public class GFG {
    public static void main(String[] args)
    {
        String s = "Geeks", t = "for", g = "geeks";

        System.out.println(s + t + g);
        System.out.println(s.concat(t));
    }
}
```

Output:

```
Geeksforgeeks
Geeksfor
```

2. **Type of arguments :**
   - strong>concat() method takes only string arguments, if there is any other type is given in arguments then it will raise an error.
   - **+ operator** takes any type and converts to string type and then concatenates the strings.

3. **concat() method raises java.lang.NullPointer Exception**
   - **concat() method** throws NullPointer Exception when string is concatenated with null
   - **+ operator** did not raise any Exception when the string is concatenated with null.

```
public class GFG {
    public static void main(String[] args)
    {
        String s = "Geeks";
        String r = null;
        System.out.println(s + r);

        // It raises an NullPointer Exception
        System.out.println(s.concat(r));
    }
}
```

Output:

```
Geeksnull
Exception in thread "main" java.lang.NullPointerException
    at java.lang.String.concat(String.java:2027)
    at GFG.main(GFG.java:7)
```

4. **Creates a new String object.**
   - **concat() method** takes concatenates two strings and return new string object only string length is greater than 0, otherwise it returns same object.
   - **+ operator** creates a new string object every time irrespective of length of string.

```
public class GFG {
    public static void main(String[] args)
    {

        String s = "Geeks", g = "";
        String f = s.concat(g);
        if (f == s)
            System.out.println("Both are same");
        else
            System.out.println("not same");
        String e = s + g;
        if (e == s)
            System.out.println("Both are same");
        else
            System.out.println("not same");
    }
}
```

Output:

```
Both are same
not same
```

5. **Performance:**
   concat() method is better than **+ operator** because it creates a new object only when the string length is greater than zero(0) but + operator always a creates a new string irrespective of length of string.

Difference table:

POINTS	CONCAT() METHOD	+ OPERATOR
Definition	A **concat() method** is method to combine two strings .	**+ operator** used to concatenate any number of strings.
Number of arguments	In **concat()** method, takes only one argument of string and concatenate it with another string.	In **+ operator**takes any number of arguments and combines all strings.
Type of arguments	**concat()** method takes arguments of string type only.	**+ operator** takes any type of argument and converts it to string type and then combine them.
Creates new string	**concat()** takes concatenates two strings and return new string object only string length is greater than 0, otherwise it returns same object..	**+ operator**creates a new string object every time irrespective of length of string.

POINTS	CONCAT() METHOD	+ OPERATOR
NullPointer Exception	In **concat()** method raises NullPointer Exception when string is concatenated with null .	**+ operator** concatenates string with without any error.
Performance	**concat() method** is better than + operator because it creates a new object only when the string length is greater than zero(0), so it uses less amount of memory.	**+ operator** always a creates a new string irrespective of length of string therefore it takes more memory.

Recommended Posts:

IntStream concat() in Java
Stream.concat() in Java
DoubleStream concat() in Java
LongStream concat() in Java
What is the difference between the | and || or operator in php?
Java String concat() with examples
Ints concat() function | Guava | Java
Java Guava | Bytes.concat() method with Examples
Java Guava | Floats.concat() method with Examples
Java Guava | Longs.concat() method with Examples
Java Guava | Doubles.concat() method with Examples
Java Guava | Chars.concat() method with Examples
Java Guava | Shorts.concat() method with Examples