

문자열 클래스

Assembled by GimunLee (2019-10-28)

Goal

- JAVA의 3가지 문자열 클래스에 대해 간략하게 설명할 수 있다.
- 각 문자열 클래스의 차이점에 대해 설명할 수 있다.
- 상황에 맞게 문자열 클래스를 사용할 수 있다.

Introduction

JAVA에는 문자열 클래스로 String, StringBuffer, StringBuilder 3가지가 있습니다. 사소해보이지만 상황에라 어떤 클래스를 쓰냐에 따라, 성능차이가 발생하는데요. 어떤 차이점이 있는지 알아보도록 하겠습니다.

String vs StringBuffer vs StringBuilder

Index	String	StringBuffer	StringBuilder
Storage Area	Heap or Constant String Pool	Heap	Heap
Modifiable	No(immutable)	Yes(mutable)	Yes(mutable)
Thread-Safe	YES	YES	NO

String과 다른 클래스(StringBuffer, StringBuilder)의 기본적인 차이는 String은 Immutable(불변), StringBuffer, StringBuilder는 Mutable(가변)에 있습니다.

String Class

String 객체는 한번 생성되면 할당된 메모리 공간이 변하지 않습니다. 즉, '+' 연산 또는 concat 메서드를 통해 기존에 생성된 String 객체에 다른 문자열을 붙여도 기존 문자열에 새로운 문자열을 붙이는 것이 아닙니다. 새로운 String 객체를 만든 후, 이 객체에 연결된 문자열을 저장하고, 그 객체를 참조하도록합니다.

- 장점
 - String Class의 객체는 Immutable(불변)하기 때문에 단순히 읽어가는 조회 연산에서는 타 클래스보다 빠르게 읽을 수 있습니다.
 - Immutable(불변)하기 때문에 멀티쓰레드 환경에서 동기화를 신경 쓸 필요가 없습니다. (Thread-Safe)
- 단점
 - 문자열 연산('+', concat 등)을 많이 일어나는 경우, 더이상 참조되지 않는 기존 객체는 Garbage Collection(이하, GC)에 의해 제거되어야하기 때문에 성능이 좋지 않습니다.

- 또한, 문자열 연산이 많아질 때 연산 내부적으로 char 배열을 사용하고, 계속해서 객체를 만드는 오버헤드가 발생하므로 성능이 떨어질 수 밖에 없습니다.

StringBuffer와 StringBuilder Class

StringBuffer와 StringBuilder 클래스는 String과 다르게 mutable(변경가능)합니다. 즉, 문자열 연산에 있어서 클래스를 한번만 만들고(new), 연산이 필요할 때 크기를 변경시켜서 문자열을 변경합니다. 그러므로 문자열 연산이 자주 있을 때 사용하면 성능이 좋습니다.

StringBuffer와 StringBuilder 클래스가 제공하는 메서드는 서로 동일합니다. 그렇다면 두 클래스의 차이점을 무엇일까요? 바로 **동기화 여부**입니다.

StringBuffer는 각 메서드별로 Synchronized Keyword가 존재하여, Multi-Thread 환경에서도 동기화를 지원하여 Thread-Safe합니다.

반면, StringBuilder는 동기화를 보장하지 않습니다. 하지만 StringBuilder는 Single-Thread 환경에서 동기화를 고려하지 않기 때문에 StringBuffer에 비해 연산처리가 빠릅니다.

그렇기 때문에 Multi-Thread 환경이라면 값 동기화 보장을 위해 StringBuffer를 사용하고, Single-Thread 환경이라면 StringBuilder를 사용하는 것이 좋습니다.

Conclusion

String Class는 JDK 1.5버전 이전에 문자열연산('+', concat)을 할 때에는 조합된 문자열을 새로운 메모리에 할당하여 참조함으로 인해서 성능상의 이슈가 있었습니다. 그러나 JDK1.5 버전 이후에는 컴파일 단계에서 String 객체를 사용하더라도 StringBuilder로 컴파일 되도록 변경되었습니다. 그리하여 JDK 1.5 이후 버전에서는 String 클래스를 활용해도 StringBuilder와 성능상으로 차이가 없어졌습니다. 하지만 반복 루프를 사용해서 문자열을 더할 때에는 객체를 계속 추가한다는 사실에는 변함이 없습니다.

String Class를 쓰는 대신, Thread와 관련이 있으면 StringBuffer를, Thread 안전 여부와 상관없이 없으면 StringBuilder를 사용하는 것을 권장합니다.

단순히 성능만 놓고 본다면 연산이 많은 경우, `StringBuilder > StringBuffer >>> String` 라고 합니다.

Reference & Additional Resources

- <https://cjh5414.github.io/why-StringBuffer-and-StringBuilder-are-better-than-String/>
- https://novemberde.github.io/2017/04/15/String_0.html
- <https://jeong-pro.tistory.com/85>
- <https://12bme.tistory.com/42>
- <https://itblackbelt.wordpress.com/2015/01/31/difference-between-string-stringbuilder-and-stringbuffer-classes-with-example-java/>