

Java: String? ?? ???

1. String? ????? ???

1.1 ??

String 类型 是 不可变 的 字符串 类 。 例如 , "hello".length() 返回 5 , 表示 字符串 的长度 。

1.2 ?? ????? ??? + ??? ??

Java 中 String 类型 是 不可变 的 字符串 类 。 字符串 一旦 创建 就 存储在 内存 中 , 且 不会 被 修改 。 String 类型 是 不可变 的 , 因此 在 堆 内存 中 创建 的 字符串 对象 是 不可变 的 (*不可变 (immutability)*)。

- length()

返回 字符串 的 长度 。

例 : "hello".length() → 5

时间复杂度 : 是 `char[]` 类型 的 字符串 , 时间复杂度 是 $O(1)$ 。

- charAt(int index)

返回 字符串 中 指定 索引 的 字符 。

例 : "java".charAt(1) → 'a'

时间复杂度 : 是 常数 时间 复杂度 , 返回 指定 索引 的 字符 。 `IndexOutOfBoundsException` 异常 。

- substring(int begin, int end)

返回 字符串 中 指定 索引 范围 的 子字符串 。

例 : "hello".substring(1, 3) → "el"

时间复杂度 : 是 常数 时间 复杂度 , 返回 指定 索引 范围 的 子字符串 。

- indexOf(String s)

返回 字符串 中 指定 子字符串 的 索引 。

例 : "hello".indexOf("l") → 2

时间复杂度 : 是 线性 时间 复杂度 , 返回 指定 子字符串 的 索引 。 $O(n)$ 。

- `lastIndexOf(String s)`

문자열에서 문자열이 마지막으로 나타나는 인덱스를 반환합니다.

예: `"hello".lastIndexOf("l")` → 3

참고: 문자열이 여러 번 나타나면, 가장 오른쪽에 있는 인덱스를 반환합니다.

- `contains(String s)`

문자열이 주어진 문자열을 포함하는지 여부를 boolean으로 반환합니다.

예: `"hello".contains("he")` → true

참고: `indexOf(...)` >= 0이면 포함합니다.

- `startsWith(String prefix)`

문자열이 주어진 접두어로 시작하는지 여부를 반환합니다.

예: `"hello".startsWith("he")` → true

참고: 문자열이 접두어와 일치하는지 확인합니다.

- `endsWith(String suffix)`

문자열이 주어진 접미어로 끝나는지 여부를 반환합니다.

예: `"hello".endsWith("lo")` → true

참고: 문자열이 접미어와 일치하는지 확인합니다.

- `equals(String s)`

문자열이 주어진 문자열과 동일한지 여부를 반환합니다.

예: `"hi".equals("hi")` → true

참고: 문자열이 null이면 false를 반환합니다. true를 반환하는 문자열도 있습니다.

- `equalsIgnoreCase(String s)`

문자열이 주어진 문자열과 대소문자를 무시하여 동일한지 여부를 반환합니다.

예: `"Hi".equalsIgnoreCase("hi")` → true

참고: `toLowerCase()` 또는 `equals()`를 사용하여 대소문자를 무시합니다.

- `toUpperCase()`

문자열을 대문자로 변환하여 새로운 문자열을 반환합니다.

예: `"java".toUpperCase()` → "JAVA"

참고: 문자열이 이미 대문자이면, 문자열이 변경되지 않습니다.

- `toLowerCase()`

문자열을 소문자로 변환하여 새로운 문자열을 반환합니다.

예: `"JAVA".toLowerCase()` → "java"

참고: `toUpperCase()`와 함께 사용하여 대소문자를 변환합니다.

- `trim()`

去除字符串首尾的空格。
例：`" hello ".trim()` → `"hello"`

注意：该方法在 JDK 5 之前是不存在的。

- `replace(CharSequence old, CharSequence new)`

将字符串中的指定子字符串替换为新的子字符串。

例：`"apple".replace("p", "b")` → `"abble"`

注意：该方法在 JDK 5 之前是不存在的。

- `split(String regex)`

根据指定的正则表达式将字符串分割成数组。

例：`"a,b,c".split(",")` → `["a", "b", "c"]`

注意：返回的数组是 `String[]` 类型。

1.3 字符串常量池

- 字符串常量池

String 类是不可变的（immutable），因此 JVM 为 String 类维护了一个常量池。

2. String 的创建

2.1 通过字面量创建

String 类提供了 `valueOf()` 静态方法，用于通过字面量创建 String 对象。

例：`String s = "hello";`

2.2 通过 new 关键字创建

Java 中 String 类提供了 `new` 关键字来创建 String 对象（static method）。
例：`String s = new String("hello");`

- `String.valueOf(...)`

该方法用于将各种数据类型转换为 String 类型。

例：`String.valueOf(123)` → `"123"`

注意：该方法在 JDK 5 之前是不存在的。

对于 `null` 值，`String.valueOf(null)` 返回 `"null"`。

- `String.valueOf(char[])`

char[] 参数 返回 String 对象。
例：`String.valueOf(new char[]{'h','i'})` → "hi"
例：`String.valueOf(char[])` 返回 String 对象。

- `String.format(String format, Object... args)`

格式字符串 返回 String 对象。
例：`String.format("%04d", 5)` → "0005"
例：`printf` 方法 返回 String 对象。使用 `Formatter` 类。
例：`String.format("Heap %d", 10)` → "Heap 10"。

- `String.join(String delimiter, CharSequence...)`

分隔符 返回 String 对象。
例：`String.join("-", "a", "b", "c")` → "a-b-c"
例：`String.join(StringBuilder)` 返回 String 对象。
例：`String.join(String)` 返回 String 对象。

- `String.valueOf(char[])`

char[] 参数 返回 String 对象。
例：`String.valueOf(new char[]{'j','a','v','a'})` → "java"
例：`String.valueOf(char[], offset, count)` 返回 String 对象。

2.3 ??

- 使用 `String.format(...)` 方法。
 - 使用 `String.format` 方法，使用 `Formatter` 类。
 - 使用 `String.join` 方法。
 - `static` 方法，使用 `String` 类。
 - 使用 `String` 类，使用 `Heap` 类。
-