

Java: String???

1. ??

String 是 Java 中最重要的类之一，它位于 `java.lang` 包中，因此不需要导入。它用于表示文本数据。

- String 是 `char` 数组，但它不是 `String` 数组。
- String 是不可变的，而 `char[]` 是可变的。
- Java 中的 String 是 **immutable** 的，这意味着一旦创建，就不能更改。

2. ?? ??

2.1 ????? ???

```
String str = "hello"; // str 是 String 类型的变量
```

- String 是 `final` 的，因此不能被重写。
- String 是 `char[]` 数组，但它不是 `char[]` 数组。它提供了许多方法，如 `length()`, `substring()`, `toUpperCase()` 等。

2.2 ??? (Immutable)

- String 是不可变的，这意味着一旦创建，就不能更改。
- 每次对 String 进行修改时，都会创建一个新的 String 对象。

```
String a = "hi";
a += "!";
System.out.println(a); // hi!
```

→ "hi" 和 "hi!" 是两个不同的对象，存储在堆内存中。

2.3 ??? ????? ?? ?(String Constant Pool)? ??

```
String s1 = "java";
String s2 = "java";
System.out.println(s1 == s2); // true → [] [] []
```

→ [] [] [] JVM [] **String Constant Pool**[] []

3. ??? ?? ??

3.1 ??? ?? (??)

```
String s = "Hello";
```

- JVM[] [] [] (Constant Pool)[] [] → [] []
- [] [] [] [] [] []

3.2 ??? ?? (???)

```
String s2 = new String("Hello");
```

- [] [] [] Heap[] []
- [] [] [] [] [] → [] [] []

[] [] []

```
String a = "abc";
String b = new String("abc");

System.out.println(a == b); // false ([] [])
System.out.println(a.equals(b)); // true ([] [])
```

4. ??? ??

4.1 == vs .equals()

```
String a = "test";
String b = "test";
String c = new String("test");

System.out.println(a == b);           // true (== 比较 地址)
System.out.println(a == c);           // false (Heap 上 两个 对象)
System.out.println(a.equals(c));      // true (equals 比较 内容)
```

==	地址
==	比较 两个 对象的 地址 (是否 指向 同一 堆内存)
.equals()	比较 两个 对象的 内容 (是否 相等)

5. ??? ??? ???

```
String s = "hello world";
System.out.println(s.length());       // 11
System.out.println(s.charAt(0));       // h
System.out.println(s.substring(0, 5)); // hello
System.out.println(s.toUpperCase());   // HELLO WORLD
System.out.println(s.replace(" ", "_")); // hello_world
```

6. ?? ?? ? ?? ??

6.1 String ?? ??

- Java 9 之前 : `char[]` 数组 存储
- Java 9 之后 : `byte[]` + `coder` (UTF-8 或 UTF-16) 存储 Compact Strings (节省 空间)

```
public final class String implements java.io.Serializable, Comparable<String>, CharSequence {
    private final char value[];
    private final int offset;
    private final int count;
    ...
}
```

6.2 ??? ?? (Concatenation)

```
String a = "Hello";
String b = "World";
String c = a + " " + b; // Hello World
```

- + 연산자 사용 시 StringBuilders를 생성하여 문자열을 연결 (StringBuilder를 생성하여 문자열을 연결)
- StringBuilder를 생성하여 문자열을 연결

```
StringBuilder sb = new StringBuilder();
sb.append("Hello");
sb.append(" ");
sb.append("World");
System.out.println(sb.toString()); // Hello World
```

7. ??

연산자	String
문자열	문자열을 연결하여 새로운 문자열을 생성
문자열 + 문자열	문자열 (문자열), 문자열 (문자열)
문자열 == 문자열	문자열 == 문자열 / .equals()를 사용하여 문자열을 비교
문자열 문자열	Java 9 문자열 : char[]
문자열 문자열	+, concat(), StringBuilder를 사용하여 문자열을 연결
문자열 문자열	문자열 문자열, 문자열, 문자열