

Java: ???(int) vs String(?? ??)? ??? ?? ??

1. Java? ??? ??

Java 虚拟机 运行 在 内存 空间 中 :

1.1 Stack ??

- 存储 变量 的 地址 (primitive type) 的 空间
- 存储 局部 变量 的 地址 , 存储 方法 的 地址
- 存储 垃圾 回收 , GC 的 地址

1.2 Heap ??

- new 创建 的 对象 的 地址 在 堆 内存 中
- ArrayList, String, Integer, 等等 的 地址 在 堆 内存 中
- GC (Garbage Collector) 的 地址 在 堆 内存 中

1.3 String Constant Pool (??? ?)

- Heap 内存 中 的 字符串 常量 池
- 存储 字符串 常量 的 地址 在 堆 内存 中
- "hello" 的 地址 在 堆 内存 中 的 String Constant Pool 中

2. ??? ?? (?: int)

虚拟机 运行 在 内存 空间 中 Stack 内存 空间 . 存储 局部 变量 的 地址 , 存储 方法 的 地址 .

2.1 ?? ??

```
int a = 10;
int b = a;
a = 20;
```

2.2 ??? ??

```

a b:
[Stack]
a → 10
b → 10 (a b)

a = 20; a:
a → 20
b → 10 (a b)

```

int (value) , int
 int

3. String (?? ??)

String ☐ (reference type) ☐ , Stack ☐ ☐ ☐ ☐ **Heap** ☐
Constant Pool ☐ ☐ ☐ .

3.1 ?? ??

```
String s1 = "hello";  
String s2 = s1;  
s1 = s1 + " world";
```

3.2 ??? ??

```

[ ] [ ]
[Stack]      [Heap (String Pool)]
s1 └─┐      "hello"
      └─┬─────────▶
s2 └─┐      (s1, s2 [ ] "hello" [ ])

```

```

[ ] [ ]
[Stack]      [Heap]
s1 → "hello world" ← 0 0 0 0
s2 → "hello"       ← 0 0 0

→ s1 + " world" 0 0 "hello" 0 0 0 0 0 0
   **"hello world" 0 0 0 0 0 0 0 0**

```

String 是不可变 (immutable) 的，一旦创建，其值就不能被修改。因此，String 对象通常存储在堆 (Heap) 中，而不是栈 (Stack) 中。堆内存由垃圾回收 (GC) 管理。

4. 变量与字符串

```

// 变量
int a = 10;
int b = a;
a = 20;      // b 仍然是 10

// 字符串 (String)
String s1 = "hi";
String s2 = s1;
s1 = s1 + "!"; // s2 仍然是 "hi"

```

变量	基本类型 (int 等)	引用类型 (String)
存储位置	Stack	Stack(局部), Heap(全局)
生命周期	随方法结束而结束	随垃圾回收而结束
可变性	可变	不可变 → 不可变字符串
内存管理	自动	手动 (GC)

5. 字符串操作

- 字符串是不可变的，存储在堆 (Heap) 中，而不是栈 (Stack) 中。
- String 是不可变的，一旦创建，其值就不能被修改。
- 字符串操作通常使用 StringBuilder 类，因为它允许修改字符串。

```
// 拼接 (String)
String result = "";
for (int i = 0; i < 1000; i++) {
    result += "a"; // 1000次 拼接 操作
}

// 拼接 (StringBuilder)
StringBuilder sb = new StringBuilder();
for (int i = 0; i < 1000; i++) {
    sb.append("a");
}
String result = sb.toString(); // 只 1次 拼接 操作
```

6. ?? ??

- 基本数据类型 : String, Integer, Boolean, LocalDate
- 集合类型 : StringBuilder, ArrayList, HashMap

在 Java 中 **String** 是不可变的，一旦创建就不能再修改其内容。

1. Java 内存管理

Java 内存管理

- Stack : 栈，用于存放基本数据类型 (int, boolean, etc)
- Heap : new 出来的对象 (String, ArrayList, etc)
- String Constant Pool : 字符串常量池 (Heap 中的一部分)

2. 堆内存管理(int ?)

堆内存 Stack 内存

```
int a = 10;
int b = a;
a = 20;
```

```
[Stack]
a → 10
b → 10    (□□□)

a = 20;   □□
a → 20
b → 10    (□□ □□)
```

- int (value) [] [] [] [] [] [] [] [] [] []
- [] [] [] [] [] [] [] [] [] []

3 . String (?? ??)

Stack, Heap/String Pool

```
String s1 = "hello";
String s2 = s1;
s1 = s1 + " world";
```



```

00 00:
[Stack]           [Heap (String Pool)]
s1 └─┬───────────▶ "hello"
      |
      └───────────▶

s2 └─┬───────────▶

00 00:
[Stack]           [Heap]
s1 ───────────▶ "hello world"   ← 00 00 (0000)
s2 ───────────▶ "hello"         ← 0000 0000 00

```

- `s1 + " world"` → `s1` → `hello` → `world` → `hello world`
- `s1` → `"hello"` → `s2` → `hello` → `world`
- `s1` → `hello` → `s2` → `hello` → `world` (GC → `hello world`)

4. ?? ??

```
// 変数宣言
int a = 10;
int b = a;
a = 20; // bは 10のまま

// String
String s1 = "hi";
String s2 = s1;
s1 = s1 + "!"; // s2は "hi"のまま
```

- $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
- $\text{String} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$

```

1  public String toString() {
2      return "String: " + sb.toString();
3  }
4  }

```