Detailed Comparison: String vs StringBuilder vs StringBuffer

1. Mutability:
- **String**: Immutable. Every modification creates a new object in memory.
- **StringBuilder**: Mutable. Supports direct modification without creating new objects.
- **StringBuffer**: Mutable. Similar to StringBuilder but synchronized.

2. Thread Safety:
- **String**: Thread-safe because it cannot be changed.
- **StringBuilder**: Not thread-safe. Best for single-threaded environments.
- **StringBuffer**: Thread-safe (synchronized). Suitable for multi-threaded applications.

3. Performance:
- **String**: Slow when performing repeated modifications because new objects are created.
- **StringBuilder**: Fastest for text modifications.
- **StringBuffer**: Slightly slower due to synchronization overhead.

4. When to Use:
- **String**: Use when the text does not need to change.
- **StringBuilder**: Use when modifying text frequently in a single thread.
- **StringBuffer**: Use in multi-threaded environments where thread safety is required.

Summary:
String → Immutable
StringBuilder → Best performance for heavy text modification
StringBuffer → Safe for multi-threading