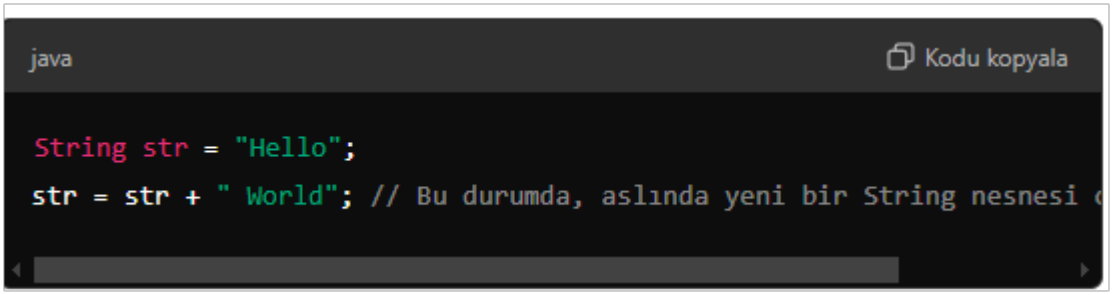
💻👨‍💻 **Java'da `StringBuilder` ve `StringBuffer`** 💻👾

Java'da iki önemli sınıf olan **`StringBuilder`** ve **`StringBuffer`**, değiştirilebilir (mutable) stringler oluşturmak için kullanılır. İkisi de benzer amaçlar için kullanılsa da aralarında bazı önemli farklar vardır. Şimdi adım adım her iki sınıfı inceleyelim.

🧩 **1. `String` Nedir?**

- Java'da **`String`** sınıfı, immutable'dir yani değiştirilemez. Bir **`String`** nesnesi oluşturulduğunda, bu nesne içindeki karakterler değiştirilemez.

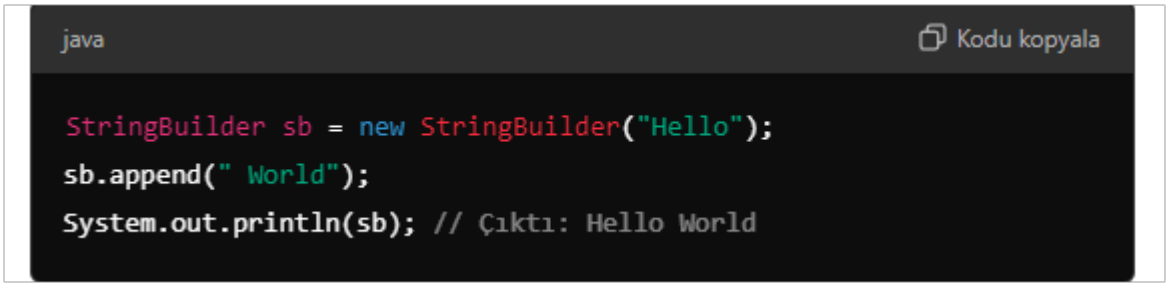
- Örneğin:



🧩 **2. `StringBuilder`**

- **`StringBuilder`**, Java'da değiştirilebilir stringler üzerinde işlem yapmak için kullanılır. Daha hızlıdır çünkü **thread-safe** değildir (yani aynı anda birden fazla thread kullanımı için güvenli değil).

📌 **Temel Kullanım**



📌 **Özellikleri**

- Değiştirilebilir stringler sağlar.

- Hızlıdır çünkü senkronize edilmez.

- Hafızada yer tasarrufu sağlar çünkü her değişiklik yeni bir nesne oluşturmaz.

- **Kullanım yerleri**: Tek thread'li işlemlerde.

📌 **Bazı Yöntemler**

- **`append(String str)`**: String ekler.

- **`insert(int offset, String str)`**: Belirli pozisyona string ekler.

- **`delete(int start, int end)`**: Belirli aralıktaki karakterleri siler.

- **`reverse()`**: String'i tersine çevirir.

🧩 **3. `StringBuffer`**

- **`StringBuffer`**, `StringBuilder` gibi değiştirilebilir stringler üzerinde işlem yapar. Ancak **thread-safe** (çoklu thread kullanımına uygun) olacak şekilde tasarlanmıştır, bu yüzden biraz daha yavaş olabilir.

📌 **Temel Kullanım**

📌 **Özellikleri**

- Değiştirilebilir stringler sağlar.

- Senkronizedir, yani thread-safe'dir.

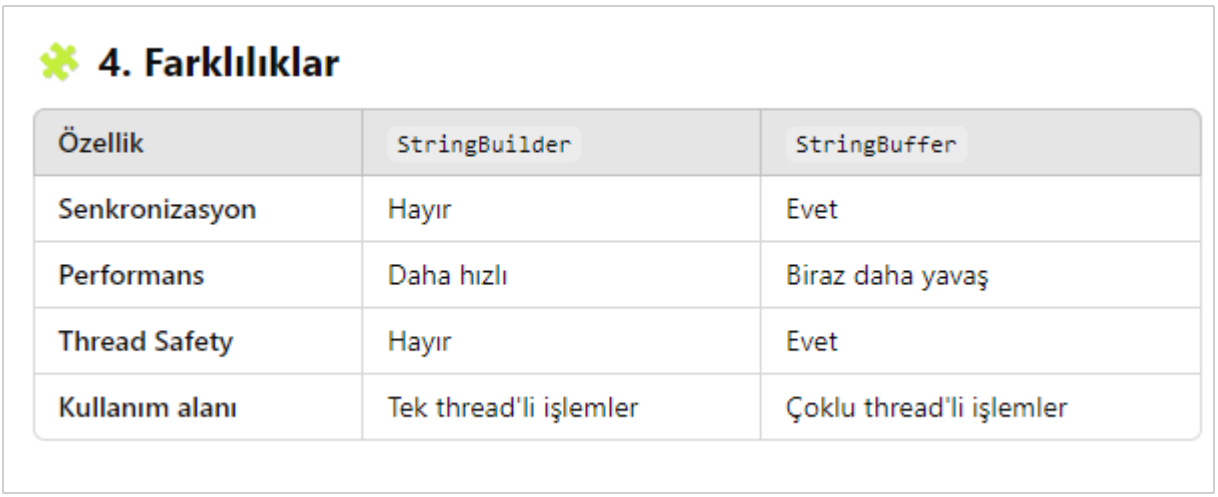
- Birden fazla thread'in aynı string üzerinde çalıştığı durumlarda güvenlidir.

- **Kullanım yerleri**: Çoklu thread kullanımı olan durumlar.

📌 **Bazı Yöntemler**

- **`append(String str)`**: String ekler.
- **`insert(int offset, String str)`**: Belirli pozisyona string ekler.
- **`delete(int start, int end)`**: Belirli aralıktaki karakterleri siler.
- **`reverse()`**: String'i tersine çevirir.

🧩 **4. Farklılıklar**



🧩 4. Farklılıklar

Özellik	`StringBuilder`	`StringBuffer`
Senkronizasyon	Hayır	Evet
Performans	Daha hızlı	Biraz daha yavaş
Thread Safety	Hayır	Evet
Kullanım alanı	Tek thread'li işlemler	Çoklu thread'li işlemler

🎯 **StringBuilder ve StringBuffer Kullanımına Beyin Haritası** 🎯

🧠 Başlık: Java'da String Sınıfları

- **String (Immutable)**
  - Değiştirilemez
  - Her değişiklikte yeni nesne oluşturur
- **StringBuilder (Mutable)**
  - Thread-safe değil (senkronize değil)
  - Daha hızlı
  - Değiştirilebilir
  - Tek thread'li işlemler için ideal

  - Yöntemler:
    - `append()`
    - `insert()`
    - `delete()`
    - `reverse()`
- **StringBuffer (Mutable)**
  - Thread-safe (senkronize)
  - Biraz daha yavaş
  - Değiştirilebilir
  - Çoklu thread'li işlemler için uygun
  - Yöntemler:
    - `append()`
    - `insert()`
    - `delete()`
    - `reverse()`