**StringBuffer and StringBuilder**

String is Immutable, StringBuffer and StringBuilder are Mutable.

Both StringBuilder and StringBuffer are mutable. That means you can change the content of them, with in the same location.

**Differences**: StringBuffer is mutable and synchronized as well. Where as StringBuilder is mutable but not synchronized by default.

**Meaning of synchronized (synchronization)**: When some thing is synchronized, then multiple threads can access, and modify it with out any problem or side effect. StringBuffer is synchronized, so you can use it with multiple threads with out any problem.

**Which one to use when?** StringBuilder : When you need a string, which can be modifiable, and only one thread is accessing and modifying it. StringBuffer : When you need a string, which can be modifiable, and multiple threads are accessing and modifying it.

String and StringBuffer

|  |  |  |
| --- | --- | --- |
| **No.** | **String** | **StringBuffer** |
| 1) | String class is immutable. | StringBuffer class is mutable. |
| 2) | String is slow and consumes more memory when you concat too many strings because every time it creates new instance. | StringBuffer is fast and consumes less memory when you cancat strings. |
| 3) | String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method. | StringBuffer class doesn't override the equals() method of Object class. |

StringBuffer and StringBuilder

|  |  |  |
| --- | --- | --- |
| **No.** | **StringBuffer** | **StringBuilder** |
| 1) | StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. |
| 2) | StringBuffer is *less efficient* than StringBuilder. | StringBuilder is *more efficient* than StringBuffer. |

Example:

public static void main(String[] args) {

String strObj ="";

long t0 = System.currentTimeMillis();

for (int i = 0 ; i < 100000; i++){

strObj +="some string";

}

System.out.println("Strings:" + (System.currentTimeMillis() - t0));

t0 = System.currentTimeMillis();

StringBuffer bufferObj = new StringBuffer();

for (int i = 0 ; i < 100000; i++){

bufferObj.append("some string");

}

System.out.println("Buffers : "+(System.currentTimeMillis() - t0));

t0 = System.currentTimeMillis();

StringBuilder buildingObj = new StringBuilder();

for (int i = 0 ; i < 100000; i++){

buildingObj.append("some string");

}

System.out.println("Builder : "+(System.currentTimeMillis() - t0));

}