# Strings

Строки в Java представлены 3 final классами - от них нельзя наследоваться:

**String** — при создании обьекта содержит в себе массив символов и представляет собой строку. И обьект данного класса не изменяемый (immutable), при изменении символа — создаётся новый обьект.

Неизменяемость строк предоставляет ряд возможностей: использование строк в многопоточных средах (String является потокобезопасным (thread-safe)).

Использование String Pool (это коллекция ссылок на String объекты, используется для оптимизации памяти)

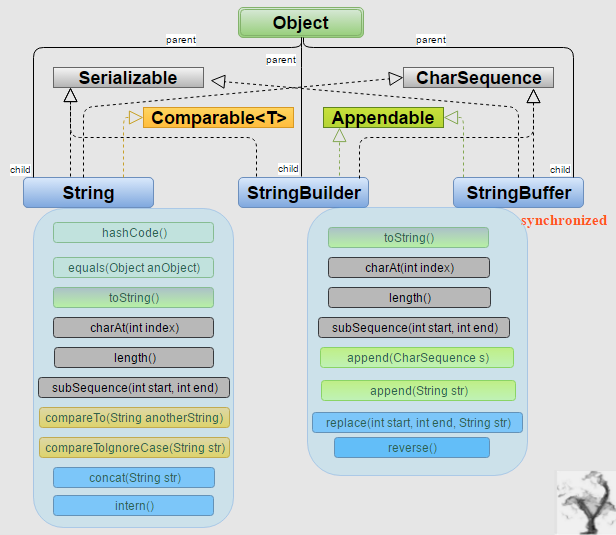
Использование строк в качестве ключей в HashMap (ключ рекомендуется делать неизменяемым)

**String** и **классы обёртки** для приметивных типов – являются не изменяемыми, при их модификации создаётся новый обьект.

**StringBuffer** — строковый класс который является изменяемым (mutable) и используется в многопоточных средах, все методы ориентированны на многопоточность (являются синхронизированными). Потокобезопасный!

**StringBuilder** — класс, что представляет изменяемую последовательность символов. Имеет полностью идентичный API с StringBuffer. Единственное отличие — StringBuilder не синхронизирован. Это означает, что его использование в многопоточных средах есть нежелательным.

Следовательно, если вы работаете с многопоточностью, Вам идеально подходит StringBuffer, иначе используйте StringBuilder, который работает намного быстрее в большинстве реализаций.

Первый, который был в Java изначально — StringBuffer и более новый StringBuilder (появился начиная с версии 1.5).

**Класс String**

**concat()**: объединяет строки

**valueOf (int,double,и т.д.)**: преобразует объект в строковый вид

**join()**: соединяет строки с учетом разделителя

**сompareTo()**: сравнивает две строки

**charAt()**: возвращает символ строки по индексу

**getChars()**: возвращает группу символов

**equals()**: сравнивает строки с учетом регистра

**equalsIgnoreCase()**: сравнивает строки без учета регистра

**regionMatches()**: сравнивает подстроки в строках

**indexOf()**: находит индекс первого вхождения подстроки в строку

**lastIndexOf()**: находит индекс последнего вхождения подстроки в строку

**startsWith()**: определяет, начинается ли строка с подстроки

**endsWith()**: определяет, заканчивается ли строка на определенную подстроку

**replace()**: заменяет в строке одну подстроку на другую

**trim()**: удаляет начальные и конечные пробелы

**substring()**: возвращает подстроку, начиная с определенного индекса до конца или до определенного индекса

**toLowerCase()**: переводит все символы строки в нижний регистр

**toUpperCase()**: переводит все символы строки в верхний регистр

**Классы StringBuilder and StringBuffer**

**Список методов, поддерживаемых классом StringBuffer:**

|  |  |
| --- | --- |
| № | Описание |
| 1 | [**public StringBuffer append (String s)**](http://proglang.su/java/strings-stringbuilder-stringbuffer-append) Обновляет значение объекта, который вызывает метод. Этот метод принимает boolean, char, int, long, Strings и т.д. |
| 2 | [**public StringBuffer reverse()**](http://proglang.su/java/strings-stringbuilder-stringbuffer-reverse) Этот метод изменяет значение объекта StringBuffer, который вызывает метод. |
| 3 | [**public delete (int start, int end)**](http://proglang.su/java/strings-stringbuilder-stringbuffer-delete) Удаляет строку, начиная с начального индекса до конечного индекса. |
| 4 | [**public insert (int offset, int i)**](http://proglang.su/java/strings-stringbuilder-stringbuffer-insert) Этот метод вставляет строку s в позицию упомянутую по смещению. |
| 5 | [**replace (int start, int end, String str)**](http://proglang.su/java/strings-stringbuilder-stringbuffer-replace) Этот метод заменяет символы в подстроке данного StringBuffer символами в указанной строке. |

**Список других методов**

**(за исключением методов set), которые очень похожи на**[**методы класса строк**](http://proglang.su/java/strings)**:**

|  |  |
| --- | --- |
| **№** | **Описание** |
| 1 | **int capacity ()** Возвращает текущую вместимость буфера String. |
| 2 | **char charAt (int index)** Возвращается указанный символ последовательности, в настоящее время представленный буфером строки, указанный индексом аргумент. |
| 3 | **void ensureCapacity (int minimumCapacity)** Гарантирует вместимость буфера, по крайней мере равным указанному минимуму. |
| 4 | **void getChars (int srcBegin, int srcEnd, char[] dst, int dstBegin)** Символы копируются из этого буфера строки в символьный массив назначения dst. |
| 5 | **int indexOf (String str)** Возвращает индекс в данной строке первого вхождения указанной подстроки. |
| 6 | **int indexOf (String str, int fromIndex)** Возвращает индекс в данной строке первого вхождения указанной подстроки, начиная с указанного индекса. |
| 7 | **int lastIndexOf (String str)** Возвращает индекс в данной строке последнего вхождения указанной подстроки. |
| 8 | **int lastIndexOf (String str, int fromIndex)** Возвращает индекс в данной строке последнего вхождения указанной подстроки, начиная с указанного индекса. |
| 9 | **int length ()** Возвращает длину строкового буфера (количество символов). |
| 10 | **void setCharAt (int index, char ch)** Символ с указанным индексом этого буфера строки имеет значение ch. |
| 11 | **void setLength (int newLength)** Устанавливает длину буфера строки (Stringbuffer). |
| 12 | **CharSequence subSequence (int start, int end)** Возвращает новую последовательность символов, которая является подпоследовательностью этой последовательности. |
| 13 | **String substring (int start)** Возвращает новую строку, которая содержит подпоследовательность символов в данный момент содержащихся в StringBuffer. Подстрока начинается с указанного индекса и продолжается до конца StringBuffer. |
| 14 | **String substring (int start, int end)** Возвращает новую строку, которая содержит подпоследовательность символов в данный момент содержащихся в этом StringBuffer. |
| 15 | **String toString ()** Преобразование в строку, представляющую данные в этой строке буфера. |