

StringBuilder

Класа `StringBuilder` (`java.lang.StringBuilder`) служи за рад са променљивим стринговима.

Сви методи класе који као повратни тип имају `StringBuilder` враћају референцу на текући објекат, чији су садржај управо променили, тако да је могуће надовезивање већег броја позива метода ове класе у једној наредби, од којих сваки оперише над резултатом претходне операције.

конструктори

`StringBuilder()`

без аргумената, прави „празан“ `StringBuilder` објекат

```
StringBuilder sb = new StringBuilder();
```

`StringBuilder(CharSequence)`

прави `StringBuilder` објекат у коме пише исто што и у аргументу (који је типа `String` или `StringBuilder`)

```
StringBuilder sb1 = new StringBuilder("Promenljivi string");  
// sb1 = new StringBuilder(sb);
```

`length()`

враћа дужину секвенце карактера тренутно представљене овим објектом, типа `int`

```
sb.length()           // 0  
sb1.length()          // 18
```

`append()`

надовезује на крај текућег објекта `String`-репрезентацију

- (јединор) аргумента примитивног типа, типа `char[]`, `String`, `StringBuilder` или било које класе,
- односно дела првог аргумента типа `char[]`, `String` или `StringBuilder` (одређеног другим и трећим аргументом)

```
sb.append(2).append(' ').append(3.14).append(" ");  
// u sb pise "2 3.14 "  
sb.append("abcdOVO SE DOPISUJE1234", 4, 19);  
// dopisuju se karakteri sa pozicija [4, 18]  
// u sb pise "2 3.14 OVO SE DOPISUJE" char[]  
karakter = {'0', '1', '2', '3', '4'};  
sb.append(karakter); // u sb pise "2 3.14 OVO SE DOPISUJE01234"  
char [] samoDeo = {'a', 'b', 'c', 'd', 'e'};  
sb.append(samoDeo, 1, 3);  
// dopisuje se 3 karaktera iz niza samoDeo, pocev od pozicije 1  
// u sb pise "2 3.14 OVO SE DOPISUJE01234bcd"
```

`charAt()`

враћа карактер са задате позиције, позиције се броје од 0,

избацује `StringIndexOutOfBoundsException` уколико позиција није валидна

```
sb.charAt(0)          // '2', prvi karakter
```

`deleteCharAt()`

брише карактер са задате позиције

```
sb.deleteCharAt(5);    // u sb pise "2 3.1 OVO SE DOPISUJE01234bcd"
```

`setCharAt()`

мења карактер на задатој позицији датим карактером

```
sb.setCharAt(21, ' '); // u sb pise "2 3.1 OVO SE DOPISUJE 1234bcd"
```

`equals()`

враћа исто што би за текући и објекат аргумент вратио оператор `==` (да ли се ради о истом објекту у меморији)

```
StringBuilder sb1 = new StringBuilder("abcde");  
StringBuilder sb2 = new StringBuilder("abcde");  
sb1.equals(sb2) // false
```

substring()

издваја подстринг са једним аргументом – од задате

позиције до краја

са два – између две задате позиције (укључујући прву и НЕ укључујући другу)

```
// StringBuilder sb1 = new StringBuilder("abcde");
String od3 = sb1.substring(2); // od3 = "cde"
String od2do4 = sb1.substring(1, 4); // od2do4 = "bcd"
```

delete()

уклања део између две задате позиције (укључујући прву и НЕ укључујући другу)

```
sb1.delete(0, 3); // u sb pise "de"
```

getChars()

издваја део садржаја → низ карактера

Као за String, издваја карактере између две задате позиције у задати низ карактера почев од задате позиције. Низ мора постојати и бити довољно велике дужине

Аргументи су:

индекс првог карактера који се издваја, индекс непосредно
иза последњег карактера који се издваја,
низ карактера у који ће бити смештени издвојени карактери,
позиција у низу од које ће почети смештање издвојених карактера

```
sb1 = new StringBuilder("de123456CDE");
char[] nizKaraktera = new char[] {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
sb1.getChars(2, 6, nizKaraktera, 3); // nizKaraktera = [A, B, C, 1, 2, 3, 4, H]
```

indexOf()

враћа индекс прве појаве

са једним аргументом датог стринга у
текућем објекту

са два

али не од почетка текућег објекта, већ од позиције задате другим
аргументом -1 ако не постоји таква појава

lastIndexOf()

враћа индекс последње појаве

са једним аргументом датог стринга у
текућем објекту

са два

али не од краја текућег објекта, већ почев од позиције задате другим
аргументом -1 ако не постоји таква појава

insert()

умеће, почев од позиције задате првим аргументом, String-репрезентацију

- другог аргумента примитивног типа, типа char[], String, StringBuilder или било које класе,

- односно дела другог аргумента типа char[], String или StringBuilder (одређеног трећим и четвртим аргументом)

append() је специјални случај, прима један аргумент мање - не задаје се позиција,
јер се увек врши надовезивање на крај

replace()

мења део између две задате позиције (укључујући прву и НЕ укључујући другу) задатим стрингом

```
StringBuilder sb3 = new StringBuilder("prepelica");
sb3.replace(2, 8, "omaj"); // u sb3 pise "promaja"
```

reverse()

обрће садржај текућег објекта

```
StringBuilder sb4 = new StringBuilder("abcde");  
sb4.reverse(); // u sb4 pise "edcba"
```

toString()

враћа String-репрезентацију текућег објекта

Напомена: CharSequence је интерфејс.

Између осталих, имплементирају га класе String и StringBuilder.

Последица је да се методима који су дефинисани са параметом типа CharSequence може приликом позива проследити аргумент типа String, односно StringBuilder.

Ово није потпун списак метода класе StringBuilder. Постоје и методи capacity(), ensureCapacity(), setLength(), trimToSize(), ...

Не очекује се да се било шта од овога учи напамет, већ да се користи помоћ коју Eclipse свесрдно пружа након што се уради неко од подешавања из упутства **Attach Source... Source Attachment... eclipse.ini.pdf**