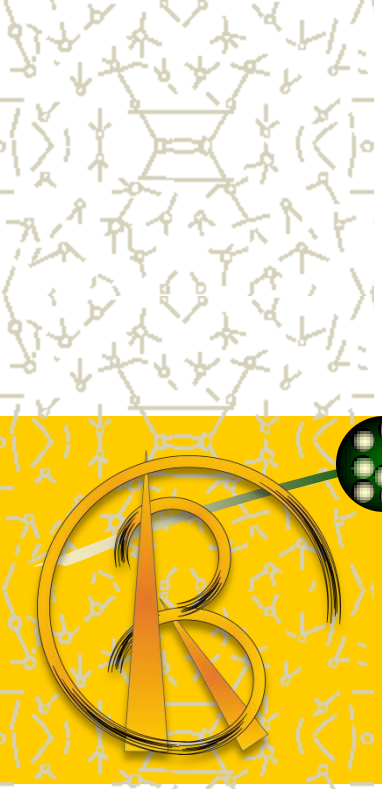


Символни низове

Разработил: инж.А.Анчев

Общи сведения

- В компютърните науки низ е крайна поредица от символи, представляващи краен брой знаци.
- Елементите му евентуално могат да бъдат променяни, както и дължината му.
- Низовете обикновено се разглеждат като тип данни и често са имплементирани като масиви от байтове (или думи), които съхраняват последователност от символите.

- В Java съществува клас String, който репрезентира типа данни символен низ.
- Тъй като String е клас, то за да запишем в дадена променлива низ, то ни трябва обект от тип класа String.
- Символните низове се записват като последователност от символи, оградени от кавички. Кавичките на влизат в състава на символния низ.

- Java Heap Space – той се създава от Java Virtual Machine, когато стартира;
- Паметта се използва, докато приложението работи. Когато даден обект е създаден, той винаги се създава в Heap и има глобален достъп;
- Това означава, че всички обекти могат да бъдат видими и достъпни от всяко място в приложението;

- Java Memory Stack - Това е временната памет, където се съхраняват стойности на променливи, когато се извикват методите им. След като методът приключи, паметта, съдържаща тези стойности, се изчиства, за да се освободи място за нови методи.
- Когато се извиква нов метод, в Java Memory Stack ще бъде създаден нов блок памет.

- String е клас, а не прост тип данни като int, float, boolean и др. Поради това променливите от тип String може да имат стойност null;
- Важна особеност на String е, че са неизменни (immutable). Това означава, че ако един път е създаден обект от тип String с конкретна стойност в Java Heap Space, *то той никога не може да бъде модифициран*.

- Дори и да искаме да реинициализираме променливата с друга стойност, автоматично се създава нов обект, ако такъв (със същата стойност)вече не съществува.
- Тоест *ако в паметта съществува обект със стойност, с която искаме да създаваме новия String, то нов не се създава, а променливата започва да реферира него.*

Създаване

```
String str = new String("Hello eng.Anchev");
```

- ключовата дума new указва, че задължително трябва да се задели нова памет за обекта.
- Именно затова ако стринг се създава по този начин, механизмът за проверка дали не съществува случайно вече такъв обект в паметта не сработва и така винаги ще създаваме нова променлива в паметта .

```
String str = “Hello eng.Anchev”;
```

- Тук е важно да запомним, че ако в паметта съществува обект със стойност, която е еднаква с тази на новия String, *то нов не се създава, а променливата започва да реферира него*. Това означава, че може да имаме 3 променливи от тип String, които сочат едновременно към един и същи обект в динамичната памет.

Методи на класа String

Метод	Действие
char charAt(int index)	Връща символ на определена позиция
boolean equals(Object obj)	Сравнява два стринга и връща true, ако и двата са еднакви
String concat(String str)	Конкатенира (залепва) два стринга
int length()	Връща дължината на текущия стринг
boolean equalsIgnoreCase(String string)	Сравнява два стринга, без да има предвид главни и малки букви
int indexOf(String str)	Връща индекса на мястото, на което за първи път е срещнал даден подниз(str) в текущия низ
int lastindexOf(String str)	Връща индекса на мястото, на което за последен път е срещнал даден подниз (str) в текущия низ

Конвертиране на String към int

```
int secondNumber = Integer.parseInt(StrNUM);
```

- Преобразува символния низ StrNUM в число от тип Integer и го записва в променливата secondNumber.
- За да използваме метод parseInt(), важно е да се каже, че всички символи трябва да са цифри, а само първия може да е символ, различен от цифра – символът минус („-“).

Конвертиране на String към Double

```
double var = Double.parseDouble(StrNUM);
```

- Преобразува символния низ StrNUM в число от тип Double и го записва в променливата var;
- Всички символи трябва да са цифри, а само първия може да е символ, различен от цифра – символът минус („-“).

Конвертиране на String към Float

```
float number = Float.parseFloat(StrNUM);
```

- Преобразува символния низ StrNUM в число от тип Float и го записва в променливата number;
- Всички символи трябва да са цифри, а само първия може да е символ, различен от цифра – символът минус („-“).

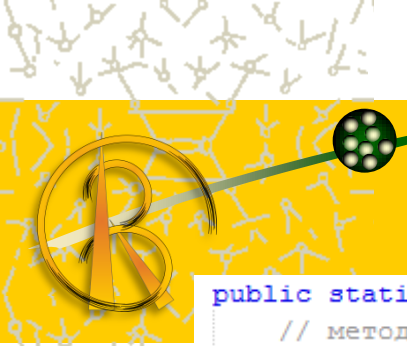
Динамични низове

- Във връзка с проблемът от излишното създаване на обекти при всеки опит за промяна върху стойността им в Java съществуват т.нар. *динамични низове*.
- При тях *низът представлява динамичен масив от символи, който може да бъде изменян по всякакъв начин*.
- За реализиране на такъв динамичен масив ни помага класът StringBuilder, както и неговият аналог StringBuffer.

- Разликата между двата класа е, че ***StringBuffer*** представлява синхронизирана имплементация на **StringBuilder**. Тоест, ако динамичният масив е споделен ресурс и се налага синхронизация, то е редно да използваме синхронизираната имплементация. Използването на **StringBuffer** и изобщо на всички подобни обекти, грижещи се за синхронизация, е изключително ресурсоемко и влияе на производителността. По-често срещаният вариант е да не се налага синхронизиране, затова се използва **StringBuilder**.

Методи

Метод	Действие
public StringBuilder append(String s)	Добавя нов стринг към дадения
public StringBuilder reverse()	Обръща наобратно стринга
public insert(int offset, value)	Вмъква от опредлено отстояние (offset) на определена стойност value. Тя може да бъде boolean, char, char[],float, double, String и тн.
public StringBuilder delete(int start, int end)	Изтрива подниз от дадения, от определена позиция (start) на настоящия низ до определена позиция(end)
replace(int start, int end, String str)	Замества символи от определена позиция (start) на настоящия низ до определена позиция(end) с нов низ (str).



```java
public static void main(String[] args) {
    // метод append
    StringBuilder sb = new StringBuilder("Здравейте");      // основен символен низ
    sb.append(", инж.Анчев");                               // добаване на допълнителния низ
    System.out.println("Добавяне на низ : " + sb);

    //метод insert
    StringBuilder sb1 = new StringBuilder("abcdefghijk");
      sb1.insert(3, 12);                        // вмъкване на цяло число
      sb1.insert(3, 12.05);                     // вмъкване на десетично число
    sb1.insert(3, "123456");                    // вмъкване на текст
    System.out.println("Вмъкване на низ : " + sb1);

    // метод reverse
    StringBuilder buffer = new StringBuilder("ABCDEFGH12345");
    buffer.reverse();
    System.out.println("Обърнат низ : " + buffer);

    // метод delete
    StringBuilder nameA = new StringBuilder("Николай Георгиев Петров");
    nameA.delete(8, 17);                            // изтрива символите от позиция 8 до позиция 17
    System.out.println("Новият низ след изтриването : " + nameA);

    // метод repleace
    StringBuilder nameB = new StringBuilder("Иван Георгиев Ангелов");
    nameB.replace(5, 14, "Иванов ");                // замества символите от позиция 5 до позиция 14
    System.out.println("Новият низ след заместването : " + nameB);
}
```