Objektno-orijentisano programiranje

Stringovi

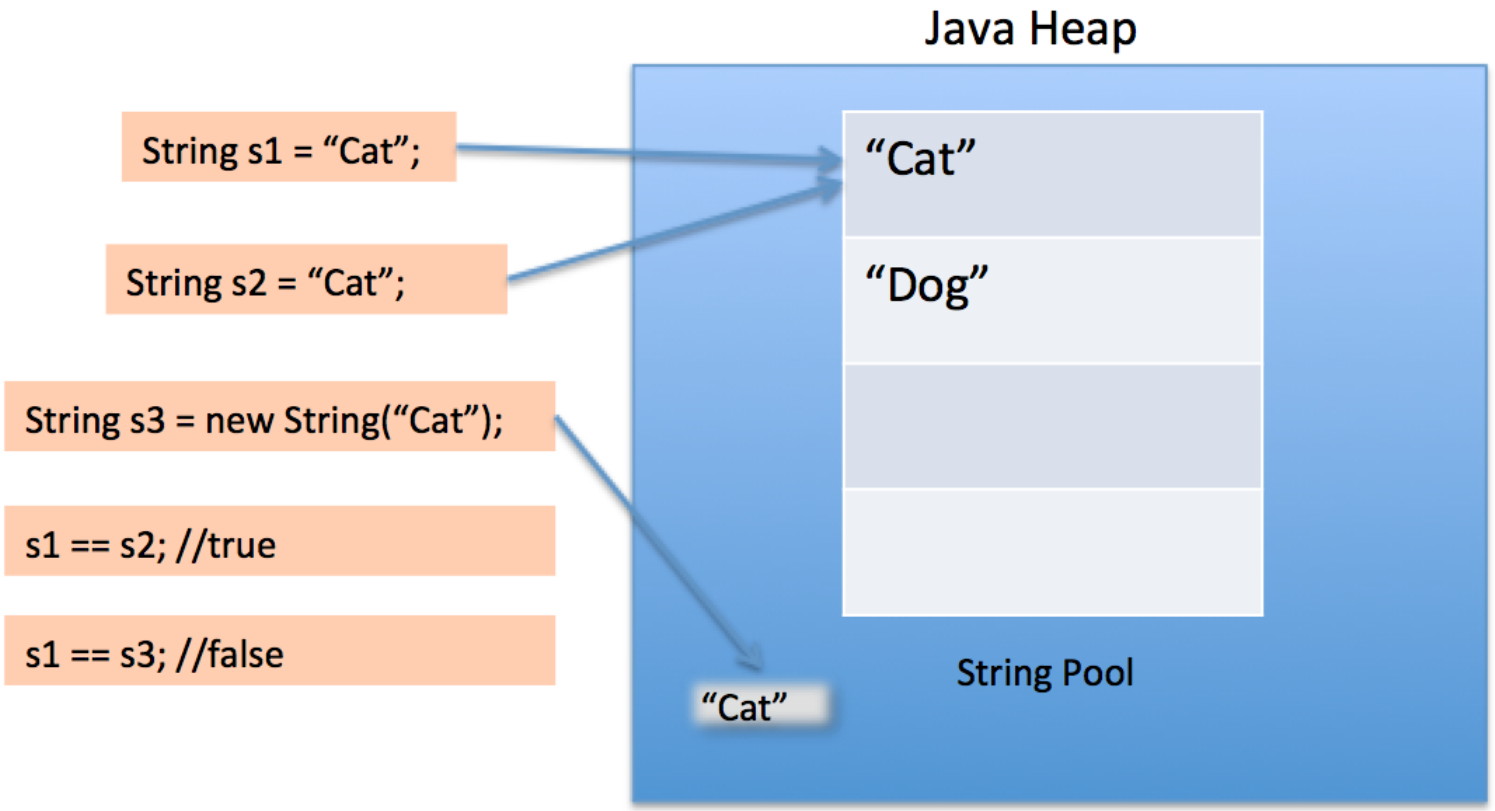
Nevena Ćirić
www.matf.bg.ac.rs/~nevena_ciric

Klasa String

- U Javi, stringovi su objekti klase String, koja se nalazi u paketu java.lang.
- Stringovska konstanta je niz karaktera pod dvostrukim navodnicima.
- Dozvoljena su dva načina za kreiranje objekata klase String:
  - Sintaksa dodele kao kod primitivnih tipova
    String str = “Hello world”;
  - Sintaksa za kreiranje klasnih tipova
    String str = new String(“Hello world”); *
- U oba slučaja kreira se objekat klase String i referenca na objekat (ručka) se smešta u promenljivu str.

* Postoji više različitih konstruktora klase String, kao što je na primer konstruktor koji kao argument uzima niz char-ova.

String pool


Java Heap
String s1 = “Cat”;
String s2 = “Cat”;
String s3 = new String(“Cat”);
s1 == s2; //true
s1 == s3; //false
“Cat”
“Dog”
“Cat”
String Pool

String pool

- String pool je prostor u kome se čuvaju String objekati kreiranih dodelom stringovskih konstanti (sintaksa kao kod primitivnih tipova). Deo je Java hip memorije.
- Svaki put kada se kreira novi String objekat dodelom stringovske konstante JVM prvo proverava da li u String pool-u postoji objekat sa istim sadržajem. U slučaju da takav objekat već postoji, ne kreira se novi već se vraća referenca na postojeći String objekat. U suprotnom, kreira se novi objekat i smešta u String pool.
- Ovaj mehanizam je ugrađen u JVM kako bi se smanjio broj String objekata koji čuvaju isti sadržaj (stringovsku konstantu).

Imutabilnost stringova

- String objekti imaju svojstvo da ne mogu biti promenjeni (immutable). To znači da se ne može promeniti stringovska konstanta koju čuva objekat klase String.
- Kada se izvršava metod nad postojećim String objektom koji kao rezultat treba da izvrši nekakvu transformaciju, uvek se kreira novi objekat klase String.
- Podsećanje: Sam objekat razlikuje se od promenljive kojom se na njega referiše. Jedna ista promenljiva tipa String u različitim delovima programa može referisati na različite objekte klase String. Kada se promeni objekat na koji referiše stringovna promenljiva, odbacuje se referenca na stari objekat i zamenjuje referencom na novi.

Poređenje stringova

- Izraz str1 == str2 proverava da li obe stringovne promenljive referišu na isti objekat. Ako referišu na nezavisne objekte, rezultat izraza je false, bez obzira da li možda ta dva objekta sadrže istu stringovnu konstantu. Dakle, ovakav izraz ne poredi same stringove, već poredi reference na stringove!
- Da bismo poredili dva stringa na jednakost koristimo metod equals() ili metod compareTo() koji vrši leksikografsko poređenje stringova.
- Za poređenje dva stringa na jednakost, ali ignorišući veličinu slova koristi se metod equalsIgnoreCase().

Metod valueOf()

- String klasa sadrži statički metod valueOf() koji kreira String objekat od vrednosti proizvoljnog primitivnog tipa, niske karaktera ili bilo kog objekta čija klasa ima definisan metod toString().
  - public static String valueOf(boolean b)
  - public static String valueOf(char c)
  - public static String valueOf(int i)
  - public static String valueOf(long l)
  - public static String valueOf(float f)
  - public static String valueOf(double d)
  - public static String valueOf(char[] c)
  - public static String valueOf(Object o)

Implementacija metoda valueOf()

```
public static String valueOf (int i) {
    return Integer.toString(i);
}
public static String valueOf (char data[]) {
        return new String(data);
}
public static String valueOf (Object obj) {
    return (obj == null) ? "null" : obj.toString();
}
```

Još neki metodi klase String

- length() – vraća dužine stringa (podsećanje: kod nizova se dužina dobija pristupom atributu length)
- charAt() – vraća karakter na određenom indeksu
- toCharArray() – vraća niza karaktera od od kojih se sastoji string
- substring() - izdvaja podstring iz stringa
- contains() – proverava da li string sadrži dati podstring
- startsWith() - proverava da li string počinje drugim stringom
- endsWith() - proverava da li se string završava drugim stringom
- indexOf() – traži indeks početka datog podstringa u okviru stringa
- split() – razdvaja string na niz podstringova uzimajući za delimiter zadati string
- replace() – vraća string koji se dobija zamenom svakog datog podstringa drugim datim stringom
- trim() – vraća string koji se dobija uklanjanjem belina sa početka i kraja stringa

Klasa StringBuilder

- Za rad sa stringovima koji mogu direktno da se modifikuju može se koristiti klasa StringBuilder.
- Glavne operacije koje klasa StringBuilder pruža a nisu dostupne u klasi String implementirane su metodima append(), insert() i delete(), koji omogćavaju dodavanje sadržaja na kraj stringa, ubacivanje i izbacivanje sadržaja unutar stringa.
- Još neki metodi klase StringBuilder koji nisu dostupni u klasi String:
  - setCharAt()
  - deleteCharAt()
  - reverse()
  - setLength()
  - toString()

Klasa StringBuilder

- StringBuilder objekte treba koristiti kada se stringovi često transformišu ili kada ih izgrađujemo višestrukim nadovezivanjem stringova. Ako uglavnom imamo statičke stringove koje tu i tamo treba da nadovežemo, String objekti su najbolji izbor.
- Metod toString() klase StringBuilder koristi se često od strane kompajlera zajedno sa metodom append() za implementaciju konkatenacije String objekata. Na primer, sledeću naredbu

  ```
  String str = “Hello “ + “world, ” + “how “ + “are “ + “you ” + “today?”;
  ```

  kompajler će interpretirati kao

  ```
  String str = new StringBuilder().append(“Hello ”).append(“world, “)
                                  .append(“how “).append(“are “)
                                  .append(“you “).append(“today?”);
  ```

- NAPOMENA: Za razliku od klase String, kreiranje objekata klase StringBuilder nije moguće dodelom stringovske konstante, već isključivo sintaksom za kreiranje klasnih tipova.