Objektno orijentisano programiranje
vežbe

Staša Vujičić Stanković

Rad sa promenljivim stringovima

- Objekti klase String ne mogu se promeniti.
  Za rad sa stringovima koji mogu direktno da se modifikuju u Javi postoje dve standardne klase: **StringBuffer i StringBuilder**.

- Što se tiče operacija koje omogućuju,
  ove dve klase se ne razlikuju,
  jedina razlika je što se objekti klase StringBuffer mogu sigurno koristiti od strane više niti.

Rad sa promenljivim stringovima

- Izmenljive stringove treba koristiti kada su česte transformacije stringova – dodavanje, brisanje, menjanje podstringova u stringu.

  StringBuffer strbuf = new StringBuffer("Izmenljivi string");

- Za razliku od klase String, ne može se koristiti sledeća notacija:

  StringBuffer strbuf = "Izmenljivi string";

- Moguće je sledeće:

```
String str = “Izmenljivi string”;
StringBuffer strbuf = new
  StringBuffer(str);
```

- Moguće je i sledeće:

```
StringBuffer strbuf = null;
StringBuffer strbuf = new
  StringBuffer(“Izmenljivi string”);
```

Rad sa promenljivim stringovima

- StringBuffer sadrži blok memorije koji se naziva **buffer**, koji može i ne mora da sadrži string.
- Ako buffer sadrži string, on ne mora da zauzima ceo bafer. Dakle, dužina stringa u objektu klase StringBuffer može biti različita od dužine bafera sadržanog u objektu.
- Veličina bafera naziva se **kapacitetom** (capacity) StringBuffer objekta.

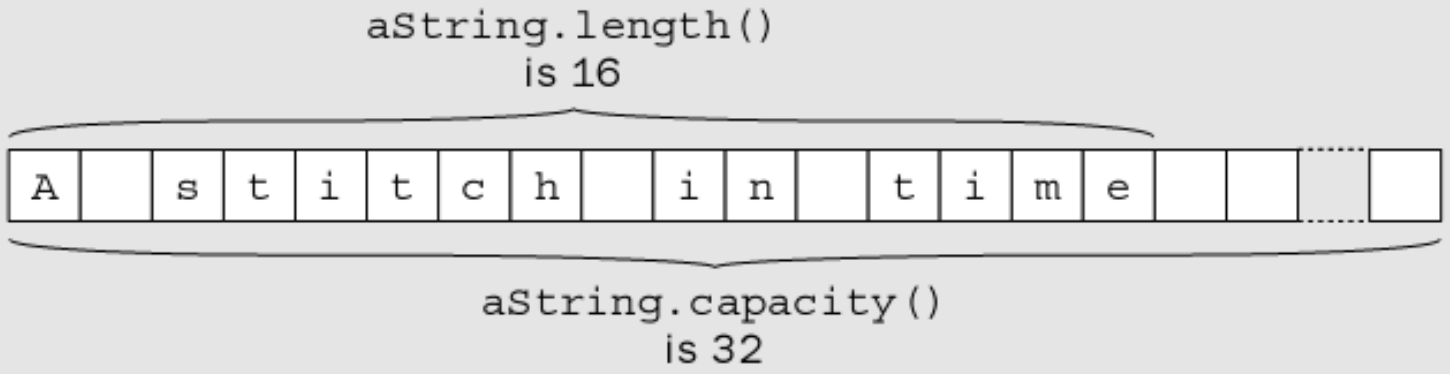
Rad sa promenljivim stringovima

- Kada se jednom kreira StringBuffer objekat, dužina stringa sadržanog u njemu se može dobiti pozivom metoda length().

```
StringBuffer strbuf = new
   StringBuffer("String");
int strlength = strbuf.length();
```

Rad sa promenljivim stringovima

```
StringBuffer aString = new StringBuffer("A stitch in time");
```



Rad sa promenljivim stringovima

- Možemo eksplicitno da zadamo kapacitet na sledeći način:

```
StringBuffer newStr = new
StringBuffer(50);

/* Kapacitet je 50 Unicode karaktera,
   dakle 100 bajtova.

   Ako se ne navede kapacitet,
   podrazumevana vrednost je
   16 Unicode karaktera.

   Bafer je inicijalno prazan,
   tj. ne sadrži nijedan string. */
```

Rad sa promenljivim stringovima

- Metod **capacity()** vraća kapacitet bafera.

- Metod **ensureCapacity()** omogućuje da se promeni podrazumevani kapacitet StringBuffer objekta.
  Specificira se minimalni kapacitet kao argument metoda.

Rad sa promenljivim stringovima

- Na primer,

newStr.ensureCapacity(40);

/* Ako je tekući kapacitet objekta newStr manji od 40, kapacitet će biti uvećan alokacijom novog većeg bafera*/

Rad sa promenljivim stringovima

- Promena dužine stringa u StringBuffer objektu može se izršiti pozivom metoda **setLength().**
- Napomena: **dužina** je karakteristika stringa sadržanog u StringBuffer objektu, dok je **kapacitet** karakteristika StringBuffer objekta.
- Kada se uvećava dužina stringa u StringBuffer objektu, dodaju se karakteri na postojeći string i svi oni sadrže '\u0000'. Češće se ovaj metod koristi za odsecanje stringa.

Rad sa promenljivim stringovima

- Da bi se dodao string na kraj postojećeg stringa u StringBuffer objektu koristi se metod **append()**.

```
StringBuffer aString = new StringBuffer("Primer");
aString.append(" nadovezivanja");

/* sada aString sadrži string "Primer nadovezivanja" */
```

- Kapacitet će se uvek automatski uvećati ako je potrebno prihvatiti veći string.

- Metod append() vraća referencu na prošireni StringBuffer objekat i ona se može dodeliti drugom StringBuffer objektu.

StringBuffer aString = new StringBuffer(“Primer”);

StringBuffer bString = aString.append(“ nadovezivanja”);

Sada i aString i bString referišu na isti StringBuffer objekat

- **append(String)** – dodaje String na kraj
- **append(String, int, int)** – dodavanje podstringa
- **append(prim_type)** – dodavanje primitivnog tipa
- Dodavanje niza karaktera:

```
StringBuffer buf = new
   StringBuffer("Test");
char[] text = {'a', 'b', 'c', 'd'};
buf.append(text, 1, 2);
```

Rad sa promenljivim stringovima

- Može se pretraživati bafer u StringBuffer objektu da bi se pronašao odgovarajući podstring metodama:
  - **lastIndexOf(String)** – za dati podstring vraća poziciju poslednjeg pojavljivanja.
  - **lastIndexOf(String, int)** – pretraga podstringa počev od zadate date pozicije u baferu.

- Zamena podstringa u baferu:

```
StringBuffer phrase = new StringBuffer("one two three four");
String substring = "two";
String replacement = "twenty";
int position = phrase.lastIndexOf(substring);            // Find start of "two"
phrase.replace(position, position+substring.length(), replacement);
```

Rad sa promenljivim stringovima

- Umetanje stringa:**insert(int, String)**
  – prvi argument je indeks pozicije u objektu gde treba umetnuti prvi karakter stringa.

  - Na primer, ako bafer buf sadrži string “Ovo je primer”,
    naredba: **buf.insert(6, “ samo”)**
    će umetnuti string “ samo” počevši od indeksa 6, tako da sada buf sadrži string “Ovo je samo primer”

- **setCharAt(int , char)** - Promena jednog karaktera u StringBuffer objektu.
- **deleteCharAt(int)** - Brisanje jednog karaktera.
- **delete(int, int)** - Brisanje više karaktera. Prvi argument je indeks prvog karaktera za brisanje, a drugi je indeks pozicije posle poslednjeg karaktera za brisanje.
- **reverse()** –obrće sekvencu karaktera u StringBufferu.

Rad sa promenljivim stringovima

- Kreiranje objekta klase String od objekta klase StringBuffer može se izvršiti upotrebom metoda **toString()** klase StringBuffer.

  StringBuffer strbuf = new StringBuffer("Primer stringa");

  String s = strbuf.toString();

  Objekat s sada sadrži string "Primer stringa".

- Metod toString() klase StringBuffer koristi se često od strane kompajlera zajedno sa metodom append() za implementaciju konkatenacije String objekata.

- Kada se napiše naredba:

```
String saying = "Many" + " hands" + " make" + " light" + " work";
```

- Kompajler će je implementirati kao:

```
String saying = new StringBuffer().append("Many").append(" hands").
                                   append(" make").append(" light").
                                   append(" work").toString();
```

ZADATAK

```
public class StringBufferTest {
   /*
    * Rad sa objektima klase StringBuffer,
    * demonstriranje funkcionalnosti nekih funkcija
    * za rad sa objektima ove klase.
    */
   public static void main(String[] args)
   {
      StringBuffer recenica = new StringBuffer(20);
      System.out.println("Kapacitet StringBuffer
objekta je: "+ recenica.capacity()+
            " a duzina stringa je: "+recenica.length());
```

```
// Dodajemo reci niza u StringBuffer objekat
String[] niz_reci = {"Ako" , "danas", "bude", "snega", "bicu", "srecan"};
recenica.append(niz_reci[0]);
for(int i = 1 ; i<niz_reci.length ; i++) {
recenica.append(' ').append(niz_reci[i]);
}
```

```
System.out.println("\nString koji se
nalazi u StringBuffer objektu je:\n" +
recenica.toString());

System.out.println("Kapacitet
StringBuffer objekta je sada: "+
recenica.capacity()+
" a duzina stringa je: "+recenica.length());
```

```
// Vrsimo zamenu neke niske u objektu nekom
// drugom
        int poz=recenica.lastIndexOf("snega");
        recenica.replace(poz, poz+5,"kise");

// Vrsimo umetanje nekih niski u objekat klase
// StringBuffer

recenica.insert(recenica.lastIndexOf("srecan"),
"ne");
```

```
System.out.println("\nString koji se nalazi u StringBuffer objektu je:\n" + recenica);

System.out.println("Kapacitet StringBuffer objekta je sada: "+ recenica.capacity()+
" a duzina stringa je: " + recenica.length());
    }
}
```

Vežbanje - Zadatak 1.

- Zadajemo dimenzije i samu matricu sa standardnog ulaza i ispitujemo da li je relacija zadata ovom matricom refleksivna i simetrična.
- Treba napisati funkcije za ispitivanje da li je matrica refleksivna, da li je simetrična i funkciju za ispis matrice.

Vežbanje - Zadatak 2.

- Napisati program koji sa standardnog ulaza učitava granice a i b intervala [a,b] i korak h i za odabranu polinomijalnu funkciju (npr. f(x)=x^3-6x+2) računa vrednost funkcije u tačkama mreže:
  a, a+h, a+2h,..., a+kh, k∈N,
  gde je a+kh<b<a+(k+1)h.

Vežbanje - Zadatak 3.

- Učitati redni broj dana u nedelji i ispisati koji je to dan.
  - 1 ponedeljak
  - 2 utorak
  - 3 sreda
  - 4 četvrtak ...

Vežbanje - Zadatak 4.

- Učitati jedan string i jedan ceo broj i ispisati da li se na tom mestu u stringu nalazi malo ili veliko slovo, samoglasnik ili suglasnik.