

# Programiranje 1 — deseta domača naloga

Rok za oddajo: nedelja, 20. januarja 2019, ob 23:55

## Aritmetični izrazi

### Uvod

Ločimo dva tipa aritmetičnih izrazov:

**Števila.** V tej nalogi se bomo omejili na pozitivna enomestna cela števila (0, 1, ..., 9).

**Sestavljeni izrazi.** Tovrstni izrazi so sestavljeni iz *levega izraza*, *operatorja* in *desnega izraza*. *Levi izraz* in *desni izraz* sta poljubna izraza, *operator* pa je eden od znakov +, -, \* in /. Kot smo vajeni, navedeni operatorji po vrsti izražajo seštevanje, odštevanje, množenje in celoštevilsko deljenje.

Na primer, sestavljeni izraz  $(3+4)*5/6$  je sestavljen iz izraza  $(3+4)*5$ , operatorja / in izraza 6. Levi izraz je tudi sam sestavljeni izraz (tvorijo ga izraz 3+4, operator \* in izraz 5), desni izraz pa je seveda število 6. Tudi izraz 3+4 je sestavljeni izraz. Njegovi sestavni deli so izraz (število) 3, operator + in izraz (število) 4.

Denimo, da veljajo sledeče deklaracije:

```
abstract class Izraz {
    ...
}

class Stevilo extends Izraz {
    ...
    public Stevilo(int n) {
        ...
    }
}

class SestavljeniIzraz extends Izraz {
    ...
    public SestavljeniIzraz(Izraz levi, char operator, Izraz desni) {
        ...
    }
}
```

Izraz  $(3+4)*5/6$  bi potemtakem lahko predstavili s sledečim objektom:

```
new SestavljeniIzraz(
    new SestavljeniIzraz(
        new SestavljeniIzraz(new Stevilo(3), '+', new Stevilo(4)),
        '*',
        new Stevilo(5)
    ),
    '/',
    new Stevilo(6)
)
```

## Naloga

Napišite razred `Izraz` s sledečimi metodami:

- `public static Izraz zgradi(String niz)`

Izdela in vrne objekt, ki predstavlja izraz, zapisan v nizu `niz`. Ta niz predstavlja veljaven aritmetični izraz brez presledkov, sestavljen iz pozitivnih enomestnih celih števil, operatorjev `+`, `-`, `*` in `/` ter oklepajev (znakov `(` in `)`). Postopek gradnje izraza na podlagi splošnega niza je opisan v razdelku *Gradnja izraza na podlagi niza*.

- `public int steviloOperatorjev()`

Vrne število operatorjev izraza `this`. Če je izraz število, je število njegovih operatorjev seveda enako 0, število operatorjev sestavljenega izraza  $l \circ d$  (levi izraz  $l$ , operator  $\circ$ , desni izraz  $d$ ) pa izračunamo kot  $o(l) + 1 + o(d)$ , kjer je  $o(l)$  oz.  $o(d)$  število operatorjev v izrazu  $l$  oz.  $d$ .

- `public String postfiksno()`

Vrne postfiksni zapis izraza `this`. Postfiksni zapis števila  $n$  je kar niz  $n$ , postfiksni zapis sestavljenega izraza  $l \circ d$  pa je niz  $LD\circ$ , kjer sta  $L$  in  $D$  postfiksna zapisa izrazov  $l$  oziroma  $d$ . Na primer, postfiksni zapis izraza  $3+4$  je  $34+$ , postfiksni zapis izraza  $(3+4)*5$  je  $34+5*$ , postfiksni zapis izraza  $(3+4)*5/6$  pa je  $34+5*6/$ .

- `public int vrednost()`

Vrne vrednost izraza `this`. Vrednost števila je kar število samo, vrednost sestavljenega izraza  $l \circ d$  pa izračunamo kot  $v(l) \circ v(d)$ , kjer sta  $v(l)$  in  $v(d)$  vrednosti izrazov  $l$  in  $d$ . Na primer, vrednost izraza  $(3+4)*5/6$  izračunamo tako, da vrednost izraza  $(3+4)*5$  delimo z vrednostjo izraza  $6$ .

V nobenem testnem primeru ne bo pri računanju vrednosti prišlo do deljenja z ničlo. Vrednost izraza in vseh njegovih podizrazov bo vedno v intervalu  $[-10^9, 10^9]$ .

Vaša implementacija razreda `Izraz` mora vsebovati gornje štiri metode, glede ostalih reči pa ste povsem svobodni. Po zgledu uvodnega razdelka lahko definirate podrazreda `Stevilo` in `SestavljeniIzraz` (v tem primeru boste morda razred `Izraz` in njegove metode `steviloOperatorjev`, `postfiksno` in `vrednost` deklarirali kot abstraktne), možna pa je tudi rešitev, ki obsega samo razred `Izraz`.

## Testni razredi

V vseh javnih in skritih testnih razredih ima metoda `main` sledečo vsebino:

```
Izraz izraz = Izraz.zgradi(niz);
System.out.println(izraz.steviloOperatorjev());
System.out.println(izraz.postfiksno());
System.out.println(izraz.vrednost());
```

Testni razredi se razlikujejo le po izrazih, zapisanih v nizih `niz`:

- J1–J2, S1–S10: vsak niz je sestavljen iz enega samega števila. V teh testnih primerih se metoda `zgradi` torej pokliče z enim od nizov  $0, 1, \dots, 9$ .
- J3–J4, S11–S20: nizi so lahko tudi oblike  $a \circ b$ , kjer sta  $a$  in  $b$  števili. Na primer:  $3+4$ ,  $2-7$ ,  $9/5$ , ...

- J5–J6, S21–S30: nizi so lahko tudi oblike  $a_1 \circ a_2 \circ \dots \circ a_k$ , kjer so  $a_1, \dots, a_k$  števila,  $\circ$  pa je operator  $+$  ali  $*$ . Na primer:  $7+9+6$ ,  $3*2*7*1*5*6$ , ...
- J7–J8, S31–S40: niz je lahko poljuben aritmetični izraz brez oklepajev.
- J9–J10, S41–S50: niz je lahko poljuben aritmetični izraz.

Dolžina niza, ki je podan kot parameter metode `zgradi`, nikjer ne presega 100.

## Gradnja izraza na podlagi niza

V tem razdelku opisujemo postopek za gradnjo izraza na podlagi niza, ki predstavlja poljuben veljaven izraz. Za nekatere posebne primere (npr. za testne razrede J1–J6 in S1–S30) obstajajo tudi enostavnejši postopki.

Niz pretvorimo v izraz na sledeči način:

1. Zunanje pare oklepajev odstranjujemo tako dolgo, dokler ne dobimo golega števila ali pa niza, v katerem se vsaj en operator nahaja izven oklepajev. Na primer, ko v nizu  $((6-9)*(5/2))$  odstranimo dva para zunanjih oklepajev, dobimo niz  $(6-9)*(5/2)$ , v katerem se vsaj en operator (tj. operator  $*$ ) nahaja izven oklepajev. Postopek odstranjevanja oklepajev zato zaključimo.
2. Če niz vsebuje zgolj število, je to število kar samo svoj izraz (npr. niz 3 predstavlja izraz 3), sicer pa med vsemi operatorji, ki se nahajajo izven oklepajev, izberemo najbolj desni operator  $+$  ali  $-$ . Če takih operatorjev ni, izberemo najbolj desni operator  $*$  ali  $/$  izven oklepajev. Označimo izbrani operator z  $\circ$ , podniz levo od njega z  $L$ , podniz desno od njega pa z  $D$ .
3. Na enak način kot prvotni niz pretvorimo podniza  $L$  in  $D$  v izraza  $l$  in  $d$ , celoten izraz pa nato zgradimo kot sestavljeni izraz, ki ga tvorijo levi izraz  $l$ , operator  $\circ$  in desni izraz  $d$ .

Na primer, v nizu  $(3+4)*5/6$  imamo  $L = (3+4)*5$ ,  $\circ = /$  in  $D = 6$ . Niz  $L$  nadalje razbijemo na  $L_1 = (3+4)$ ,  $\circ_1 = *$  in  $D_1 = 5$ . V nizu  $L_1$  najprej odstranimo zunanji par oklepajev, dobljeni niz pa razbijemo na  $L_2 = 3$ ,  $\circ_2 = +$  in  $D_2 = 4$ . Niza  $L_2$  in  $D_2$  pretvorimo v izraza `new Stevilo(3)` in `new Stevilo(4)`, niz  $L_1$  pa v izraz `new SestavljeniIzraz(new Stevilo(3), '+', new Stevilo(4))`. Za niz  $L$  nato dobimo izraz `new SestavljeniIzraz(new SestavljeniIzraz(new Stevilo(3), '+', new Stevilo(4)), '*', new Stevilo(5))`, za celoten niz pa izraz `new SestavljeniIzraz(new SestavljeniIzraz(new SestavljeniIzraz(new Stevilo(3), '+', new Stevilo(4)), '*', new Stevilo(5)), '/', new Stevilo(6))`.

## Testni primer J9

Testni razred:

```
public class Test09 {

    public static void main(String[] args) {
        Izraz izraz = Izraz.zgradi("(3+4)*5/6");
        System.out.println(izraz.steviloOperatorjev());
        System.out.println(izraz.postfiksno());
        System.out.println(izraz.vrednost());
    }
}
```

```
}  
}
```

Izhod:

```
3  
34+5*6/  
5
```

## Napotek

Za delo z nizi vam bodo morda koristile nekatere metode iz razredov `String` in `Integer`. Če je spremenljivka `niz` tipa `String`, potem

- `niz.length()` vrne dolžino niza;
- `niz.charAt(i)` vrne znak na indeksu  $i$  (prvi znak ima indeks 0);
- `niz.substring(i, j)` vrne podniz od znaka na indeksu  $i$  do vključno znaka na indeksu  $(j - 1)$ ;
- `Integer.parseInt(niz)` vrne število, zapisano v nizu.

## Oddaja naloge

Oddajte datoteko `Izraz.java` in še vse morebitne ostale datoteke, ki vsebujejo vaše razrede. Datotek ne »zipajte«; oddajajte jih posamično!

V prvi vrstici vsake datoteke naj bo v komentarju zapisana vaša vpisna številka. Na primer:

```
// 63180999  
  
public class Izraz {  
    ...  
}
```