

Prevođenje programskih jezika - Septembar 1 27.08.2018

Uputstvo za rad:

Na Desktopu napraviti direktorijum oblika PPJ.Sep2018. Ime_Prezime.Indeks.InicijaliAsistenta i u njemu sačuvati svoj rad. Rešenja zadataka sačuvati u posebnim direktorijumima u okviru svog PPJ.Sep... direktorijuma sa nazivima koji odgovaraju rednim brojevima zadataka. Ispit traje 3h.

1. Potrebno je napraviti interpreter za rad sa kvaternionima. U matematici kvaternion se može posmatrati kao proširenje skupa kompleksnih brojeva i definisan je na sledeći način:

$$q = a + bi + cj + dk$$

pri čemu su $a, b, c, d \in \mathbb{R}$, a i, j, k su kvaternionijske jedinice, tj. $i^2 = j^2 = k^2 = -1$.

- a) Komandom `def` se definiše novi kvaternion na sledeći način `def(ime, s:broj, i:broj, j:broj, k:broj)`. Komanda `def` podržava skraćeni oblik definisanja koji podrazumeva da se može izostaviti bilo koja komponenta kvaterniona sem njegovog imena. Redosled navođenja baznih elemenata može biti proizvoljan. Komandom `print` se ispisuje kvaternion na standardni izlaz u skladu sa zadatim formatom (%q – štampa se ceo kvaternion, %s – štampa se slobodni član, %i – štampa se koeficijent uz i , %j – uz j , %k – uz k).

```
def(a); // definiše novi kvaternion a = 0 + 0i + 0j + 0k
def(b, s:1, i:-2, j:3, k:-4); // definiše novi kvaternion b = 1 - 2i + 3j - 4k
def(c, j:5, k:-10.2); // definiše novi kvaternion c = 0 + 0i + 5j - 10.2k
def(d, j:1, k:-2, s:3, i:-4); // definiše novi kvaternion d = 3 - 4i + 1j - 2k
def(c, j:5, j:-10.2); // greška, bazni element ne može da se definiše dva puta
def(c, m:5, n:-10.2) // greška, bazni element su s, i, j i k
print("%q", a); // štampa ceo kvaternion na stdout: $: a = 0 + 0i + 0j + 0k
print("%s", b); // štampa samo slobodni član na stdout: $: 1
print("%s", k); // greška, nije definisan kvaternion k
```

- b) Omogućiti korisniku da izračunava zbir, razliku, množenje skalarom i normu već definisanih kvaterniona. Uvesti operator dodele `:=` koji omogućava definisanje kvaterniona sabiranjem i oduzimanjem postojećih.

```
def(a); // definiše novi kvaternion a = 0 + 0i + 0j + 0k
def(b, s:1, i:-2, j:3, k:-4); // definiše novi kvaternion b = 1 - 2i + 3j - 4k
c := a + b; // definiše novi kvaternion c = 1 - 2i + 3j - 4k
d := a - b; // definiše novi kvaternion d = -1 + 2i - 3j + 4k
a := 10*(d - a); // definiše novi kvaternion a = -10 + 20i - 30j + 40k
e := -a; // definiše novi kvaternion e = 10 - 20i + 30j - 40k
norm(a); // štampa ceo kvaternion na stdout: $: |a| = 54.77
norm(b); // štampa ceo kvaternion na stdout: $: |b| = 5.48
```

- c) Omogućiti korisniku da izračunava proizvod i količnik kvaterniona. Dodefinisati operator dodele `:=` tako da podržava definisanje kvaterniona množenjem i deljenjem postojećih.

```
def(a, s:1, i:2, j:3, k:4); // definiše novi kvaternion a = 1 + 2i + 3j + 4k
def(b, s:1, i:-2, j:3, k:-4); // definiše novi kvaternion b = 1 - 2i + 3j - 4k
c := a * b; // definiše novi kvaternion c = 12 - 24i + 6j + 12k
d := a / b; // definiše novi kvaternion d = -0.33 - 0.67i + 0j + 0.67k
e := a + b/c; // definiše novi kvaternion e = 1.03 + 2.07i + 3.17j + 4.00k
f := 10*e - 3*(a-b*c); // definiše novi kvaternion f = -10.67 + 50.67i + 508.67j + 100.00k
```

Napomena: Svaka greška na ulazu izaziva prekidanje programa. Voditi računa o prioritetima operacija. Imena kvaterniona su stringovi proizvoljne dužine. Broj promenljivih je ograničen samo dostupnom memorijom računara.

Algebra kvaterniona:

Neka su dati kvaternioni q i r :

$$q = q_0 + q_1i + q_2j + q_3k$$

$$r = r_0 + r_1i + r_2j + r_3k$$

Algebarske operacije u skupu kvaterniona definisane su na sledeći način:

a) Sabiranje i oduzimanje kvaterniona:

$$n = q + r = (q_0 + r_0) + (q_1 + r_1)i + (q_2 + r_2)j + (q_3 + r_3)k$$

$$m = q - r = (q_0 - r_0) + (q_1 - r_1)i + (q_2 - r_2)j + (q_3 - r_3)k$$

b) Suprotni element:

$$n = -q = -q_0 - q_1i - q_2j - q_3k$$

c) Množenje skalarom i norma:

$$\lambda q = \lambda q_0 + \lambda q_1i + \lambda q_2j + \lambda q_3k$$

$$||q|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$$

d) Množenje i deljenje kvaterniona:

$$n = q * r = n_0 + n_1i + n_2j + n_3k$$

$$t = \frac{q}{r} = t_0 + t_1i + t_2j + t_3k$$

Koeficijenti rezultujućeg kvaterniona se dobijaju na sledeći način:

$$n_0 = (r_0q_0 - r_1q_1 - r_2q_2 - r_3q_3)$$

$$n_1 = (r_0q_1 + r_1q_0 - r_2q_3 + r_3q_2)$$

$$n_2 = (r_0q_2 + r_1q_3 + r_2q_0 - r_3q_1)$$

$$n_3 = (r_0q_3 - r_1q_2 + r_2q_1 + r_3q_0)$$

$$t_0 = \frac{r_0q_0 + r_1q_1 + r_2q_2 + r_3q_3}{r_0^2 + r_1^2 + r_2^2 + r_3^2}$$

$$t_1 = \frac{r_0q_1 - r_1q_0 - r_2q_3 + r_3q_2}{r_0^2 + r_1^2 + r_2^2 + r_3^2}$$

$$t_2 = \frac{r_0q_2 + r_1q_3 - r_2q_0 - r_3q_1}{r_0^2 + r_1^2 + r_2^2 + r_3^2}$$

$$t_3 = \frac{r_0q_3 - r_1q_2 + r_2q_1 - r_3q_0}{r_0^2 + r_1^2 + r_2^2 + r_3^2}$$

2. Tehnikom rekurzivnog spusta napisati parser za uprošćenu gramatiku kvaterniona iz prvog zadatka. Gramatika treba da podržava definisanje kvaterniona uprošćenom komandom **def**, koja ne podržava skraćeni oblik definisanja. Pored toga, gramatika treba da podržava štampanje kvaterniona uprošćenom komandom **print** i definisanje novih kvaterniona izračunavanjem zbira, razlike, proizvoda i količnika postojećih. Voditi računa o prioritetima operacija. Podrazumevati da će korisnik uvek pozivati komandu **def** na ispravan način.

```
def(a, s:1, i:2.5, j:-3.7, k:-4);
```

```
def(b, s:1, i:-2, j:3, k:-4);
```

```
print(aj;
```

```
print(b);
```

```
c := a+b;
```

```
d := a-b;
```

```
e := a*b + c;
```

```
f := a - b/c + c*(d+a/c);
```