

Programski prevodioci: Projekat

Šifra:PP-J5FT

Sadržaj

1. Napomena	1
2. Zadatak 1	1
3. Zadatak 2	1
4. Zadatak 3	2

1. Napomena

Za svaki zadatak potrebno je uraditi:

1. Sintaksnu analizu
2. Semantičku analizu
3. Generisanje koda

2. Zadatak 1

Omogućiti upotrebu operacija množenja i deljenja.

3. Zadatak 2

Proširiti jezik PARA iskazom koji ima sledeći oblik:

```
"para" <type> <id> "en" "(" <lit1> ".." <lit2> ")"  
    <statement>
```

Gde:

- <type> predstavlja tip podatka
- <id> predstavlja naziv promenljive, iterator
- <lit1> i <lit2> predstavljaju literala, gornju i donju granicu iteratora
- <statement> predstavlja iskaz

Realizovati semantičke provere:

1. <id> treba da bude lokalna promenljiva za para iskaz (sledeći para iskaz može da definiše iterator sa istim imenom).
2. <lit1> i <lit2> moraju biti istog tipa kao i <id>.

Izvršavanje:

- Inicijalizacija iteratora se vrši samo jednom, pre prvog izvršavanja petlje. Iterator se inicijalizuje na vrednost prvog literala (<lit1>).
- Ako je $\text{lit2} \geq \text{lit1}$:
 - Na početku svake iteracije treba proveriti da li je iterator veći od drugog literala (<lit2>), ako nije izvršiti telo petlje.
 - Nakon izvršavanja tela petlje, iterator uvećati za 1.
- Ako je $\text{lit2} < \text{lit1}$:
 - Na početku svake iteracije potrebno je proveriti da li je iterator manji od drugog literala (<lit2>), ako nije izvršiti telo petlje.
 - Nakon izvršavanja tela petlje, iterator se umanjuje za 1.



Omogućiti i ugnježdene para iskaze.

Primer:

```
int zbir = 0;
int razlika = 0;
para int i en(0 .. 13){
    zbir = zbir + i;
    razlika = razlika - i;
}
para int i en ( 10 .. 4)
    razlika = zbir - i;
```

4. Zadatak 3

Proširiti jezik SWITCH iskazom koji ima sledeći oblik:

```
"switch" "[" <switch_expression> "]" "{"
    "case" <constant_expression> "->" <case_body> ["finish" ";"]
    ...
    ["default" "->" <default_statement>]
"}
```

Gde:

- <switch_expression> predstavlja ime promenljive
- <constant_expression> predstavlja konstantu
- <case_body> predstavlja iskaz (statement)
- <default_statement> predstavlja iskaz (statement)



Mora postojati bar jedna case naredba. Finish naredba se opciono može pojaviti samo na kraju case naredbe. Default naredba je opcionalna i može se pojaviti samo posle svih case naredbi

Realizovati sledeće semantičke provere:

1. Promenljiva u <switch_expression> mora biti prethodno deklarirana
2. Konstante u svim case iskazima moraju biti jedinstvene
3. Tip konstante u case naredbi mora biti isti kao tip promenljive u <switch_expression>

Izvršavanje:

- Na početku switch iskaza se izvrši provera vrednosti promenljive switch_expression.
- U zavisnosti od te vrednosti preusmerava se tok izvršavanja na telo odgovarajuće case naredbe.
- Ukoliko se na kraju case naredbe nalazi finish naredba, tok izvršavanja se preusmerava na kraj switch iskaza; a ako je finish naredba izostavljena, "propada" se na izvršavanje sledeće case naredbe.
- Default naredba se izvršava ukoliko se vrednost switch promenljive razlikuje od svih konstanti navedenih u svim case naredbama.

Primer:

```
switch [a] {  
  case 1 ->  
    a = a + 5;  
    finish;  
  case 5 ->  
  {  
    b = 3;  
  }  
  default ->  
    a = a - 1;  
}
```