

Pismeni deo ispita iz Programskih prevodilaca

1. Deklaracija tipa u jednom programskom jeziku definisana je sledećom gramatikom:

$TypeDeclaration \rightarrow \text{type } TypeSpec \mid \text{type } (TypeSpecList)$

$TypeSpecList \rightarrow TypeSpecList TypeSpec \mid TypeSpec$

$TypeSpec \rightarrow \text{ID } TypeName$

$TypeName \rightarrow \text{ID} \mid \text{int} \mid \text{float}$

- Transformisati datu gramatiku u LL(1) gramatiku i dokazati da tako transformisana gramatika jeste LL(1) gramatika.
- Kreirati LR sintaksnu tabelu date gramatike.

2. Kreirati *cup* specifikaciju za generisanje sintaksnog (i semantičkog) analizatora deklaracije tipa koja je opisana gramatikom iz prethodnog zadatka. Za generisanje potrebnog leksičkog analizatora kreirati odgovarajuću *flex* specifikaciju. Terminalni simbol NAME predstavlja niz brojki i slova u kojem je prvi znak obavezno slovo.

Semantička pravila o kojima treba voditi računa su:

- ne može definisati više tipova sa istim imenom,
- ne može se koristiti ime tipa koji prethodno nije definisan.

Primer:

Ispravna deklaracija	Višestruko definisanje istog tipa	Korišćenje imena tipa koji nije deklarisan
<pre> type (dan int mesec int godina int radniDan dan) </pre>	<pre> type (xKoordinata float yKoordinata float xKoordinata int yKoordinata int) </pre>	<pre> type (dn int mesec int godina int radniDan dan) </pre>

3. Definirati klasu za predstavljanje „while petlje“ u apstraktnom sintaksnom stablu. „While petlja“ je definisana sledećim smenama:

$while\text{-}statement \rightarrow \text{while } condition\text{-}list \text{ code_block}$

$condition\text{-}list \rightarrow expression \mid condition\text{-}list, expression$

Značenje petlje: Petlja se ponavlja dok su svi uslovi u listi zadovoljeni.

Definisati medjukod niskog nivoa koji odgovara „while petlji“ u ovom programskom jeziku i u klasi koja „while petlju“ predstavlja u apstraktnom sintaksnom stablu implementirati funkciju za generisanje takvog medjukoda.

4. Definirati 8086 kod i izgled aktivacionog sloga za funkcije *stepen*. Smatrati da se rezultat funkcije prenosi kroz listu parametara.

```

int stepen(int k, int n) {
    int rez;
    if ( n>1 )
        rez = k*stepen(k,n-1);
    else
        rez = k;
    return rez;
}
                    
```