

Varianta A

1 Enunț

Se consideră un fișier care conține cod scris în limbajul GO care trebuie să fie analizat sintactic. Cu alte cuvinte, ne dorim să verificăm dacă apar sau nu erori de sintaxă în cod. Deoarece o astfel de analiză ar fi mult mai complexă și ar depăși scopul temei, vom restrânge elementele de sintaxă ale limbajului.

2 Sintaxă

O să considerăm doar următoarele elemente de sintaxă ale limbajului GO:

2.1 Comentarii

În GO există două tipuri de comentarii:

- comentariu care începe cu `//` și continuă până la sfârșitul liniei;
- comentariu care începe cu `/*` și se termină cu `*/` (poate conține mai multe linii).

Conținutul comentariilor o să fie ignorat în timpul analizei.

Exemplu:

```
// First line of a block comment
// Second line of a block comment

/*
Everything here
will be considered
a block comment
*/
```

2.2 Variabile

Variabilele se pot declara în limbajul GO folosind cuvântul cheie **var**.

```
var a = "initial"
var e int
```

Acest limbaj permite declararea mai multor variabile simultan.

```
var b, c int = 1, 2
var a, b, c = 3, 4, "foo"
```

Limbajul GO are posibilitatea de a infera tipul variabilelor inițializate.

```
var a = true // va infera tipul bool pentru variabila a
var b = "sir" // va infera tipul string pentru variabila b
```

Elementul de sintaxă `:=` permite prescurtarea declarării și inițializării unei variabile. Astfel, declarația `var f string = "apple"` se poate rescrie ca:

```
f := "apple"
```

De asemenea, operatorul `:=` oferă posibilitatea declarării mai multor variabile simultan.

```
x, y, z := 1, 2.5, false
```

Această prescurtare poate fi folosită doar pentru variabile ce nu au mai fost declarate anterior.

```
var x int
x := 7 // va genera eroare la compilare
```

Eroarea afișată va trebui să fie `No new variables on the left side of :=`

Aceeași eroare va fi afișată și pentru următoarea secvență:

```
x := 7
x := 8
```

Variabilele nu pot fi redeclarate în interiorul aceluiași bloc. Dacă se descoperă o redeclararea de variabilă se va afișa mesajul `'<variable_name>' redeclared in this block.`

2.3 Tipuri

Vom presupune că există doar următoarele 4 tipuri în limbajul GO:

1. tipul `int` – pentru numere întregi;

```
var x int
x = 1
```
2. tipul `float32` – pentru numere reale;

```
var x float32
x = 1.5
```
3. tipul `string` – pentru șiruri de caractere;

```
var x string
x = "LFA"
```
4. tipul `bool` – pentru valori binare (`true` sau `false`).

```
var x bool
x = true
```

În cadrul analizatorului nostru vom verifica dacă valorile atribuite unor variabile sunt în concordanță cu tipurile acestora.

În caz de eroare se va afișa o eroare de genul: `(type1) cannot be represented by the (type2).`

Pentru următorul exemplu se va afișa mesajul `(string) cannot be represented by the (float32).`

```
var x float32
x = "ana"
```

2.4 Funcții

Funcțiile se declară în limbajul GO folosind cuvântul cheie `func`.

Argumentele funcției sunt specificate între paranteze, iar pentru fiecare argument o să fie specificată o pereche de forma: identificator și tip.

Funcțiile care vor întoarce un rezultat vor avea specificat tipul rezultatului după specificarea parametrilor.

Corpul funcției este oferit între acolade.

```
func identity(f float32) float32 { return f }

func second(a, b int) int
{
    return b
}
```

```
func first(a int, b int) int {
    return a
}

func main() {
    var bVal bool
    bVal = true
}
```

După cum am observat în exemplul anterior, putem avea și funcții care să nu întoarcă rezultat. Dacă o funcție nu are specificat un rezultat în antet, dar în cadrul implementării acesteia se va returna un rezultat, atunci se va afișa mesajul de eroare **Too many arguments to return**.

Analizatorul nostru va verifica dacă rezultatul întors de funcție are sau nu tipul potrivit. În cazul în care tipurile nu coincid, se va afișa mesajul **Invalid return type for function '<function_name>'**.

În cadrul unui fișier nu vom putea defini două funcții cu același nume. În cazul în care analizatorul întâlnește funcții cu același nume, se va afișa mesajul de eroare: **'<function_name>' redeclared in this file**

Important

Analizatorul nostru va contoriza liniile din fișier, iar pentru fiecare eroare întâlnită se va specifica inițial indicele liniei.

Atunci când se întâlnește o eroare analizatorul nu își încheie execuția, ci va continua!

Dacă doriți, puteți presupune că orice variabilă va avea identificatorul format dintr-o singură literă.

Orice concept de sintaxă care nu a fost precizat anterior vom considera că nu există!

3 Exemple

3.1 Exemplul 1

Intrare

```
1 // Exemplul 1
2 func first(a, b, c int) int {
3     return a
4 }
5
6 func second(a, b int) int { return b }
7
8 func first(a int, b int) {
9     return a
10 }
11
12 func main() {
13     var a, b, c := 2, 3, 4
14     var d int = a
15     var e, f string = b, "ana"
16 }
```

Ieșire

line 8: 'first' redeclared in this file

line 9: Too many arguments to return

line 15: (int) cannot be represented by the (string)

3.2 Exemplul 2

Intrare

```
1  /*  
2      Exemplul 2  
3  */  
4  func second(a, b, c int) bool {  
5      return b  
6  }  
7  
8  func main() {  
9      var a int  
10     var b int  
11     a := 7  
12     c := true  
13     var d string = c  
14     var a bool  
15 }
```

Ieşire

line 5: Invalid return type for function 'second'
line 11: No new variables on the left side of :=
line 13: (bool) cannot be represented by the (string)
line 14: 'a' redeclared in this block

