

Spisak konvencija u korišćenom pseudo jeziku:

1. Osnovni tipovi podataka su:

- a. numerički (celi i realni brojevi sa pokretnim zarezom),
- b. logički (sa vrednostima true (tačno) i false (netačno)),
- c. znakovni i
- d. pokazivači.

Ostali tipovi podataka se uvode po potrebi, stringovi, nizovi, structure, itd...

2. Promenljive se **ne deklariraju formalno**, već se tip određuje iz konteksta upotrebe.
3. **Komentari** se navode iza simbola dve kose crte (//).
4. **Naredbe petlje i grananja** imaju uobičajnu interpretaciju. Na raspolaganju su naredbe **for-do**, **while-do**, **repeat-until**, kao naredbe petlje. U for petlji iterativna promenljiva zadržava vrednost nakon izlaska iz petlje. Kao **naredbe grananja**, na raspolaganju su **if-then-else** i **case** naredbe. else grana nije obavezna u naredbi.
5. Svaka naredba se završava tačkom-zarezom (;). U jednom redu algoritma može biti zapisano više naredbi.
6. Sve korišćene **promenljive se smatraju lokalne** osim ako se eksplicitno ne naglasi da je promenljiva globalna. U tom slučaju podrazumeva se da je ona već deklarirana u spoljašnjem okruženju.
7. **Operator dodeljivanja** se označava klasičnim simbolom za dodeljivanje (=). Na primer, izraz $a = b$ označava da je vrednost promenljive b dodeljuje promenljivoj a.

8. **Indeksiranje elemenata u nizu** se vrši **od 1** do broja tih elemenata, pojedinačni element niza se označava imenom niza i indeksom tog elementa u uglastim zagradama. Na primer, $a[5]$ označava 5-ti element niza a .
9. **Logički operatori** se označavaju simbolima **&&** (konjunkcija), **||** (disjunkcija) i **!** (negacija). Logički izrazi se obavezno omeđuju malim zagradama. Evaluacija logičkih izraza se vrši sleva na desno. Omogućena je i **skraćena evaluacija logičkih izraza**. Na primer, ako je u izrazu $a \&\& b$ istinitosna vrednost izraza a netačno, vrednost izraza b se ne izračunava jer će istinitosna vrednost složenog izraza u svakom slučaju biti netačno.
10. **Blok naredbi** se naznačava **uvlačenjem**. Ovaj način isticanja logička struktura algoritama doprinosi njegovoj boljoj čitljivosti.
11. **Neformalne naredbe sa elementima govornog jezika** mogu se koristiti kad god takve naredbe doprinose da algoritam bude razumljiviji. Na primer, naredba

for (svaki element v niza U) do
12. **Zaglavlje algoritma** počinje ključnom rečju *algorithm*, a iza nje se se navodi ime algoritma. Za imena algoritama se uglavnom koriste engleski nazivi jer su uobičajni u stručnoj literaturi.
13. Algoritmi se mogu koristiti i **kao funkcije i kao podprogrami**. U slučaju korišćenja kao funkcije naredba *return* se upotrebljava za vraćanje vrednosti i označavanja kraja izvršavanja algoritma. U slučaju da se algoritam koristi kao podprogram, naredba *return* nije obavezna, poslednja naredba u nizu predstavlja i kraj izvršavanja algoritma.

14. **Komunikacija između algoritama** je obezbeđena (a) **putem globalnih promenljivih** i (b) **navođenjem parametara algoritma**. Ulazni (ili ulazno/izlazni) parametri se zadaju u zaglavlju algoritma iza imena algoritma u zagrada, a izlazni parametri se navode iza naredbe return. Prilikom poziva algoritma umesti formalnih parametara navode se stvarni argumenti. Argumenti složenih tipova podataka se prenose samo po reference (pokazivač na stvarni argument).

15. Omogućeno je **korišćenje rekurzije**.