

Prevodioci i interpretatori - Jun 2006. - praktični deo

1. Na lokalnom računaru je smešten sadržaj web sajta u obliku skupa html datoteka raspoređenih u okviru direktorijuma `www` i njegovih poddirektorijuma. Ime početne strane sajta se navodi kao argument komandne linije perl skripta, a podrazumeva se da je `index.html` ukoliko ovaj argument nije naveden. Datoteke su povezane hiperlinkovima. Napisati perl skript koji generiše izveštaj o svim datotekama do kojih se može stići praćenjem lokalnih linkova, krenuši od početne strane sajta. Uz ime svake datoteke potrebno je navesti broj linkova koji vode od nje kao i broj linkova koji vode do nje. Izveštaj bi trebalo da bude uređen opadajući po "popularnosti" datoteke tj. drugom od pomenutih brojeva.
2. Programski jezik dopušta definisanje funkcija čiji su argumenti, lokalne promenljive i rezultat isključivo celi brojevi. Definicija funkcije počinje ključnom rečju `function` za kojom (opciono) sledi ime rezultatske promenljive i znak `=`, zatim ime funkcije i u zagradama lista argumenata funkcije. Telo funkcije se navodi u okviru vitičastih zagrada i sadrži niz naredbi dodele, izraza i/ili poziva ugrađene funkcije `print` koja služi za ispis. Jezik podržava aritmetičke izraze nad celim brojevima kao i ternarni operator `?:`. Izvršavanje programa počinje od funkcije sa imenom `main`. Napisati interpretator za dati jezik. Npr.

```
function c = proizvod(a, b) {
    c = a*b;
}

function f = faktorijel(n) {
    f = n==0 ? 1 : n*faktorijel(n-1);
}

function ispisi_kvadrat(n) {
    a = n*n;
    print(a);
}

function main(){
    a = 5; b = 6;
    ispisi_kvadrat(a);
    print(proizvod(a, b));
    print(faktorijel(a));
}
```

Predlog rešenja zadatka:

- (a) Gramatikom opisati dati jezik i napisati leksički analizator za njega.
- (b) Definirati hijerarhiju klasa za predstavljanje sintaksnog stabla kojim se predstavlja kôd u telu funkcija. Za početak dopustiti samo naredbe dodele aritmetičkih izraza promenljivim i korišćenje ugrađene funkcije `print`.
- (c) Definirati klasu `Funkcija`. Klasa bi trebalo da čuva imena ulaznih parametara, ime izlaznog parametra i sintakšno stablo koje predstavlja kôd same funkcije. Interpretator bi globalno trebalo da čuva mapiranje imena definisanih funkcija u objekte klase `Funkcija`.
- (d) Kroz hijerarhiju čvorova sintaksnog stabla definisati metod `Interpret` koji vrši interpretaciju programa predstavljenog stablom. Argument ovog metoda bi trebalo da bude mapiranje imena promenljivih u njihove vrednosti.
- (e) U klasi `Funkcija` definisati metod `Call` kojim se poziva sama funkcija i vraća vrednost njenog rada. Ovaj metod bi trebalo da iskoristi napravljeni metod za interpretaciju kôda predstavljenog sintaksnim stablom. Argument metoda `Call` je niz brojeva koji predstavljaju ulazne parametre funkcije, a rezultat je ceo broj koji predstavlja rezultat funkcije. U glavnom programu interpretatora pozvati metod `Call` funkcije `main`.

Testirati sa

```
function main() {
    print(2+3*5);
}
```

- (f) U izrazima omogućiti i pozive funkcija kreiranjem klase `PozivFunkcije`.

```
function a = f(b) {
    a = b+1;
}
function main()
{
    print(f(3));
}
```

- (g) Omogućiti korišćenje ternarnog uslovnog operatora `?:`.