

Interpretacija programa

Domagoj Dragaš, Mislav Martinić, Dora Parmać

29. svibnja 2020.

Sadržaj

1	Kako funkcioniра interpreter за osnovnu školu?	3
1.1	Varijable	3
1.2	Osnovne operacije	4
1.3	Petlje	5
1.4	Zadaci za vježbu	6

1 Kako funkcionira interpreter za osnovnu školu?

1.1 Varijable

Varijable možemo podijeliti na stringovne i brojčane varijable.

1. Sintaksa brojčnih varijabli:

```
IME VARIJABLE = VRIJEDNOST;
a = 2;
```

2. Sintaksa stringovnih varijabli:

```
IME VARIJABLE = "SADRZAJ";
b = "Ovo je moja poruka";
```

- Ukoliko sadržaj stringa pišemo unutar "...", taj string smije sadržavati znak " jedino ukoliko je ispred njega napisan znak \, tj. \".
- Na sličan način se piše string unutar '...'.
 - Oznaka \n je oznaka za novi red.
- Vidimo da ukoliko imamo samo \n, to označava prijelaz u novi red, a ukoliko je znak \ ispred, tj. \\n, imamo točno \n (kažemo da je \n escape-an znakom \)
- Na sličan način, ukoliko želimo baš znak \ unutar stringa, moramo ga escape-ati istim znakom, tj. napisati \\.
- Unutar navodnika, "... ili '...', \ nema nikakvo posebno značenje.
- Naredbe ispisujemo pomoću naredbe print
- Ako je kod upisa, upisan broj, onda to shvaćamo kao brojčanu varijablu, a ako je upisan string, onda to shvaćamo kao stringovnu varijablu, tj. nema deklariranja tipa varijable prije.

Pogledajmo primjer:

```
Ulaz:
primjer = '\nOvo \n je \n moja \n recenica\n'
print(primjer)
```

```
Izlaz:
Ovo \n je
moja \n recenica
```

Stringove možemo i konkatenerati (zalijepimo početak drugog stringa na kraj prvog).

```
Ulaz:
a = "Danas je ";
b = "petak.";
c = a+b;

Izlaz:
c = "Danas je petak."
```

Također, možemo pisati i komentare unutar znakova `# ... #`.

```
#Ovo je moj komentar.#
```

1.2 Osnovne operacije

Nad brojevnim ili int varijablama, možemo obavljati glavne matematičke operacije poput zbrajanja, oduzimanja, množenja, dijeljenja, te uspoređivati brojeve `s < (manje)`, `> (veće)`, `<= (manje ili jednako)`, `>= (veće ili jednako)`, `= (jednako)`, `!= (različito)`. U ovom slučaju znak `!` ispred `=` označava negaciju.

Pogledajmo primjer:

```
a = 9;
b = 3;
c = a+b;
d = a-b;
e = a*b;
f = a/b;
print(c);
print('\n');
print(d);
print('\n');
print(e);
print('\n');
print(f);
print('\n');
```

```
Izlaz:  
c = 12  
d = 6  
e = 27  
f = 3
```

1.3 Petlje

Podržavamo if i for petlje. Program u ovom slučaju je niz naredbi koje će se izvršiti unutar petlje.

```
AKO ( IME operator konst )  
{  
    program  
}
```

```
AKO ( IME operator IME )  
{  
    program  
}
```

```
AKO ( IME operator konst )  
{  
    program  
}  
INACE  
{  
    program  
}
```

```
AKO ( IME operator IME )  
{  
    program  
}  
INACE  
{  
    program  
}
```

```
FOR ( IME = IZRAZ ; IZRAZ )  
{  
    program  
}
```

FOR petlja iterira od prve granice, do druge. Npr.
Ako je FOR petlja ovog oblika:

```
FOR (i = 5; 10)
```

Tada i poprima redom vrijednosti [5, 6, 7, 8, 9]

Ako je FOR petlja ovog oblika:

```
FOR (i = 10; 5)
```

Tada i poprima redom vrijednosti [10, 9, 8, 7, 6]

Ne dopuštamo kvadriranje u obliku $BROJ^2$ ILI $BROJ^{**2}$. Umjesto toga pišemo $BROJ*BROJ$.

1.4 Zadaci za vježbu

1. zadatak: Upišite neki string i ispišite ga.

```
ispis("Upisi string: ");
upis(s);
ispis(s);
```

2. zadatak: Upišite dva broja i zbrojite ih.

```
ispis("Upisi dva broja: ");
upis(a);
upis(b);
c=a+b;
ispis(c);
```

3. zadatak: Upišite dva broja. Ako je prvi veći od drugog, ispišite prvi, inače ispišite drugi.

```
ispis("Upisi dva broja: ");
upis(a);
upis(b);
ako (a >= b){ #Ukoliko je prvi upisani veci od drugoga
    ispiši ga (Ako su jednaki svejedno je kojeg
    ispisujemo)#
ispis(a);
}
inae { #U suprotnom ispiši drugoga#
    ispis(b);
}
```

4. zadatak: Napisati program koji će unositi broj učenika te omogućiti unos visine učenika prema broju unesenih učenika. Program treba ispisati visinu najnižeg i najvišeg učenika.

```
ispis("Upisi n:");
upis(n);
```

```

ispis("Upisi visine učenika:");
upis(v);
max = v;
min = v;
for (i=0;n-1){
    upis(v);
    ako (v > max){ #Ako je zadnja upisana visina veća od
        maksimalne dosad upisane, onda je ta visina
        maksimalna.#
        max = v;
    }
    ako (v < min){ #Analogno prijasnjem#
        min = v;
    }
}
ispis("Najvisi učenik");
ispis(max);
ispis("Najnizi učenik");
ispis(min);

```

5. zadatak: Korisnik upisuje prirodan broj n. Program vraća string koji se sastoji od svih prostih brojeva do tog upisanog prirodnog broja n.

```

ispis("Upisi broj: ");
upis(n);
str = "";
for(i=2;n){
    #Pretpostavljamo da je broj prost dok se ne
    pokaže suprotno#
    prost=1;
    for(j=2;i){
        d=i/j;
        #Ako je broj djeljiv sa nekim brojem osim
        s njim ili 1 => nije prost#
        ako(d*j == i){
            prost=0;
        }
    }
    #Ako je prost konkatenujemo ga u string#
    ako(prost == 1){
        str = str + pretvori(i) + " ";
    }
}
ispis(str);

```
