

4. DEFINISANJE ULAZA I PRIKAZIVANJE IZLAZA

Naredbe za definisanje ulaza i izlaza u programu su temeljni elementi za obezbjeđenje interakcije programa s korisnicima.

4.1 Komanda Print

U Liberty Basicu komanda Print šalje izabrane sadržaje (podatke, poruke, rezultate izračunavanja itd.) na izlazni prozor.

Postoji nekoliko načina za korištenje Print naredbe:

- štampanje teksta na izlaznom prozoru ekrana izvodi se tako da se odabrani tekst koji treba da se štampa stavi u navodnike poslije naredbe Print. Tako će naredbom:

Print "Institut "Kemal Kapetanović" Zenica"

na ekranu biti odštampan tekst:

Institut "Kemal Kapetanović" Zenica

Sve što se poslije naredbe Print navede između znakova navoda, identično se štampa na ekranu.

Naredba:

Print "5+7"

neće štampati na ekranu broj 12, nego će kao tekst odštampati sve što je navedeno između navodnika, a u ovom slučaju to je tekst:

5+7

- naredba:

Print 5+7

kod koje se ne koriste navodnici, na ekranu štampa rezultat izračunatog aritmetičkog izraza. U ovom slučaju je to broj 12.

Poslije naredbe Print se može naći bilo koji aritmetički izraz, čiju vrijednost je potrebno izračunati.

Primjer 4.1:

```
`Filename: Povrsinakruga.bas
`Program za izracunavanje površina kruga i polovine kruga
Print "Površina kruga radijusa 3 je:"
Print 3 * 3 * 3.1416
Print "Površina polovine ovog kruga je:"
Print (3 * 3 * 3.1416) / 2
End
```

End naredba u LB programima je opcijaska naredba. Obično se ova naredba koristi da se označi fizički kraj programa, ali se može koristiti i na mjestima logičkog kraja programa, kada se na primjer u programu ima grana na čijem kraju se program završava.

Svaka Print naredba uzrokuje štampanje nove linije.

Napomena

U LBU se za pisanje naredbi mogu koristiti mala, velika, ili i mala i velika slova (može se pisati na primjer: **print**, **PRINT**, ili **Print**). Kod nekih programskih jezika se mora voditi računa o tome koja slova se koriste pri pisanju naredbi. Tako C++ zahtijeva da sve naredbe budu pisane malim slovima.

4.2 Čišćenje izlaznog prozora

Naredba **Cls** (clear screen) obezbjeđuje čišćenje izlaznog prozora, odnosno ekrana.

4.3 Napredno štampanje

Print naredba.

4.3.1 Korištenje znaka tačka-zarez

Primjer:

```
Print 15
Print 20
Print 25
```

Rezultat:

10
20
25

Primjer:

Print 15; 20; 25

Rezultat:

152025

Primjer:

Print 15; " "; 20; " "; 25

Rezultat:

15 20 25

Primjer 4.2: Štampanje rezultata proračuna

```
`Filename: Povrsinakruga.bas
`Program za izracunavanje površina kruga i polovine kruga
Print "Površina kruga radijusa 3 je: ";
Print 3 * 3 * 3.1416
Print "Površina polovine ovog kruga je: ";
Print (3 * 3 * 3.1416) / 2
End
```

Rezultat će biti:

Površina kruga radijusa 3 je: 28.2744
Površina polovine ovog kruga je: 14.1372

Primjer:

Print "Sarajevo";

Kod naredne Print naredbe, štampanje izlaza će se nastaviti u istoj liniji.

4.3.2 Štampanje praznih linija

Print

4.4 Definisanje promjenljivih

U LB programima, kao i kod većine drugih programskih jezika, moraju se podaci čuvati u memorijskim lokacijama centralne memorije (RAM memorija), koje su rezervisane za u programu definisane promjenljive.

Vrijednost svake promjenljive smještena je u odgovarajućoj memorijskoj lokaciji pomenute RAM memorije.

Vrijednost može da bude broj, karakter, ili string sa karakterima.

Pri tome programer ne treba da zna koje memorijske lokacije drže njegove podatke. Umjesto pamćenja specifične memorijske lokacije (odnosno njene adrese), potrebno je pamtititi samo ime promjenljive koja je kreirana. Ime promjenljive je veza sa odgovarajućom memorijskom lokacijom i vrijednošću promjenljive koja je smještena u njoj.

Kod viših programskih jezika, broj promjenljivih je ograničen samo veličinom RAM memorije.

Svakoj promjenljivoj se dodjeljuje odgovarajuće ime. Pri tome ova imena treba dodjeljivati tako da se što lakše zapamte i da korisnika asociraju na to o kakvoj se promjenljivoj radi, odnosno opisuju vrstu podataka koju sadrže.

Kod definisanja imena promjenljivih, treba poštovati slijedeća pravila:

- imena promjenljivih moraju počinjati slovom,
- poslije prvog slova, promjenljive mogu da sadrže brojeve i tačke,
- dužina imena promjenljive nije ograničena,
- promjenljive s malim i velikim slovima se razlikuju; ime promjenljive MyName i MYNAME su dvije različite promjenljive,
- promjenljivim se ne smiju davati imena LB naredbi, jer će LB upozoriti na grešku o nepravilnom izboru imena promjenljive.

Promjenljive mogu da sadrže brojeve ili stringove. String se sastoji od jednog ili više karaktera, pri čemu to može biti: riječ, ime, rečenica, ili adresa.

Promjenljive koje sadrže stringove, nazivaju se *string promjenljive*. String promjenljive se definišu tako da se na kraju izabranog imena promjenljive stavi znak \$ (na primjer; Company\$, Ime\$, Opcina\$).

4.5 Dodjeljivanje vrijednosti promjenljivim

Vrijednost promjenljivim se dodjeljuje putem izraza za dodjeljivanje, koji uključuje znak jednakosti =. Primjer:

```
cijena = 956.34
```

Opcijski se može koristiti i naredba Let:

```
Let cijena = 956.34
```

koja postoji samo zato što je kod starih verzija Basica bila obavezna.

O znaku jednakosti se može misliti kao o strelici usmjerenoj nalijevo. Strelica pokazuje da se sve ono što je na desnoj strani znaka jednakosti pridjeljuje lijevoj strani, odnosno imenu promjenljive koja se nalazi na tom mjestu.

Kod string-promjenljivih, u izrazu za dodjeljivanje se vrijednost promjenljive stavlja među navodnike:

```
adresa$ = "Ferhadija 34"
```

Nakon što se "stave" vrijednosti u promjenljive, one ostaju tu za vrijeme izvršenja programa, ili dok se u njih ne stave druge vrijednosti. Promjenljiva može da "drži" samo jednu vrijednost. Prema tome, naredbe:

```
godine = 67  
godine = 27
```

dovode do toga da promjenljiva godine sadrži vrijednost 27.

Promjenljivoj se takođe može dodijeliti vrijednost druge promjenljive, kao i vrijednost aritmetičkog izraza u kome su argumenti broćane promjenljive:

```
pi = 3.1416  
radius = 3  
povrsina = radius * radius * pi  
polapovrsine = povrsina / 2
```

LB postavlja vrijednosti promjenljivih na nulu. To znači da nakon imenovanja promjenljive LB pretpostavlja da je njena vrijednost jednaka nuli, sve dok joj se tokom izvršenja programa ne dodijeli neka druga vrijednost. Prema tome, ako se želi da promjenljiva počne nulom, ne mora joj se dodjeljivati vrijednost nula. Kod string-promjenljivih, LB inicijalno skladišti nul stringove, odnosno prazne stringove u njih.

Kod slijedećeg primjera, program prvo pohranjuje rezultate proračuna u promjenljive, a zatim štampa vrijednosti promjenljivih:

Primjer 4.3

```
`Filename: Povrsinakruga2.bas
`Program za izracunavanje povrsina kruga i polovine kruga
pi = 3.1416
radius = 3
povrsina = radius * radius * pi
polapovrsi = povrsina / 2
Print "Povrsina kruga radijusa 3 je: ";
Print povrsina
Print "Povrsina polovine ovog kruga je: ";
Print polapovrsi
End
```

4.6 Dodjeljivanje ulaznih vrijednosti naredbom Input putem tastature

Primjer:

```
Print "Unesi godinu rođenja radnika ";  
Input godrod
```

Kada računar dođe do prve print naredbe, on štampa poruku kao i obično.

Znak tačka-zarez na kraju Print naredbe, kod izvršenja ove naredbe drži kursor u istom redu sa odštampanom porukom, tako da računar u istom redu, iza znaka pitanja očekuje unos vrijednosti promjenljive godrod.

Program se nastavlja tek kada se unese vrijednost promjenljive godrod i pritisne taster <Enter>.

Primjer 4.4

```
'Filename: Input.bas  
'Demonstracija Input naredbe  
'  
Print "Prije naredbe Input, varijabla x ima vrijednost□";x  
Print  
Print "Unesi vrijednost promjenljive x□";  
Input x  
Print  
Print "Sada promjenljiva x ima vrijednost:□";x  
End
```

Kada se program starta, LB svim promjenljivim dodjeljuje nule. To znači da će promjenljiva x u prethodnom primjeru imati vrijednost nula sve dok joj se putem Input naredbe ne dadne nova vrijednost.

4.6.1 Unošenje stringova

Putem `Input` naredbe mogu se unijeti vrijednosti za sve tipove promjenljivih. U slijedećem primjeru je pokazano kako se putem `Input` naredbe unosi vrijednost jedne string promjenljive.

Promjer 4.5

```
Print "Unesi prezime radnika";  
Input prezime$  
Print prezime$
```

Napomena:

Ako korisnik ne ukuca ništa i pritisne `Enter` na mjestu gdje je program očekivao unos vrijednosti string-promjenljive, vrijednost ove promjenljive će biti `null string`, ili prazan string, koji će sadržavati praznine.

4.6.2 Unos vrijednosti varijabli putem jedne `Input` naredbe

Umjesto korištenja kombinacije naredbi `Print` i `Input`, kod unosa vrijednosti promjenljivih može se koristiti samo jedna `Input` naredba, kako to pokazuje slijedeći primjer:

```
Input "Unesi vrijednost promjenljive x:"; x
```

Kod ove naredbe se kod starta programa ne pojavljuje znak pitanja na kraju teksta. Želi li se imati i ovaj znak, potrebno ga je unijeti u tekst-pitanje `Input` naredbe.

Neka pitanja ne mogu stati u jedan red s jednom `Input` naredbom. Tada treba kombinovati `Print` i `Input` naredbe, tako da se prvi dio teksta pitanja `Print` naredbom odštampa u prvom redu, a ostatak pitanja se odštampa uz `Input` naredbu u drugom redu.

Rezime i zaključak

Naš sadašnji cilj je da naučimo osnove LBa. Učenjem ovih osnova, ne stičemo samo znanja iz programiranja u LBU, nego se pripremamo za sve programske jezike. Svi jezici imaju promjenljive, proračune i kontrolne programske izraze (o istim će

biti riječi kasnije). Ipak, treba istaći da većina jezika stupa u interakciju s Windows kontrolama na totalno različite načine.

Sadržaj tradicionalnog programiranja za sada je u centru naše pažnje, pogotovo za one koji ranije nisu nikako, ili nisu ozbiljno programirali. Kada se stekne više sigurnosti i iskustva u pomenutom tradicionalnom programiranju, biće se u stanju vidjeti kako jezici podržavaju razne Windows kontrole.

4.7 Štampanje na štampaču