

Šta je programiranje?

Programiranje je proces kreiranja instrukcija koje računari koriste da bi obavili određeni zadatak. Te instrukcije, koje nazivamo kodom, pišemo koristeći programski jezik. Kada se sve instrukcije izvrše na računaru, stvaramo ono što zovemo softver.

Zašto je programiranje važno?

Inovacija: Programiranje je osnova za stvaranje novih tehnologija. Od aplikacija za mobilne telefone do kompleksnih sistema za prepoznavanje slike, sve je napravljeno pomoću programiranja.

Problem Solving: Učenje programiranja razvija vaše vještine rješavanja problema. Da biste napisali efikasan kod, morate prvo razumjeti problem, pronaći najbolje rešenje, a zatim ga implementirati koristeći programski jezik.

Profesionalna prilika: Programiranje je jedna od najtraženijih vještina u 21. vijeku. Bez obzira na to da li želite da budete programer, analitičar podataka, naučnik ili čak preduzetnik, vještine programiranja su neophodne.

Zašto programski jezik C?

C je jedan od najstarijih i najčešće korišćenih programskih jezika. Evo nekoliko razloga zašto biste trebali da učite C:

Osnova za druge jezike: Veliki broj modernih programskih jezika, uključujući C++, Java, i Python, temelji se na jeziku C. Znanje C-a može olakšati učenje ovih jezika.

Efikasnost: Jezik C se direktno prevodi u mašinski jezik, što ga čini izuzetno efikasnim. To je razlog zašto se koristi za razvoj sistema u realnom vremenu i za programiranje hardvera.

Razumijevanje računarske nauke: Kroz C možete razumjeti osnovne koncepte računarske nauke, kao što su memorija, pokazivači i izuzeci.

Varijable, tipovi podataka i operatori u programskom jeziku C

Nakon što ste upoznati sa osnovama programiranja i programskim jezikom C, sljedeći korak u vašem učenju je da se detaljnije upoznate sa osnovnim elementima jezika C. To uključuje varijable, tipove podataka i operatore.

1. Varijable

Varijable su osnovni elementi svakog programskog jezika. One su kontejneri u kojima se čuvaju vrijednosti koje koristimo u našim programima. U ovoj lekciji ćete naučiti kako da deklarirate varijable, kako da im dodijelite vrednosti i kako da ih koristite u vašim programima.

2. Tipovi podataka

Svaka varijabla ima svoj tip podataka. Ovo određuje kakvu vrstu vrijednosti varijabla može da sadrži i koliko prostora zauzima u memoriji. C ima nekoliko osnovnih tipova podataka, uključujući `int` (za cjelobrojne vrednosti), `float` (za decimalne brojeve), `char` (za karaktere) i `double` (za decimalne brojeve sa dvostrukom preciznošću). U ovoj lekciji ćete se detaljnije upoznati sa svim ovim tipovima podataka.

3. Operatori

Operatori su simboli koje koristimo za izvođenje operacija na vrijednostima i varijablama. To uključuje aritmetičke operatore kao što su `+`, `-`, `*` i `/`, ali i druge kao što su operatori poređenja (`==`, `<`, `>`, itd.) i logički operatori (`&&`, `||`, `!`). Ova lekcija će vas upoznati sa svim ovim operatorima i pokazati vam kako da ih koristite u vašim programima.

Varijable u C

Varijable su jedan od osnovnih elemenata programiranja. One predstavljaju mjesta u memoriji na kojima se čuvaju vrijednosti koje koristimo u našim programima.

1. Deklaracija varijabli

Prije nego što koristimo varijablu u programu, moramo je prvo deklarirati. To činimo tako što navedemo tip podatka varijable, a zatim i njeno ime. Na primjer:

```
int broj;
```

Ovdje smo deklarirali varijablu tipa `int` (integer, tj. cjelobrojni tip) sa imenom `broj`.

2. Inicijalizacija varijabli

Inicijalizacija varijable znači davanje početne vrijednosti varijabli u trenutku njene deklaracije. Na primjer:

```
int broj = 10;
```

Ovde smo inicijalizovali varijablu `broj` vrijednošću 10.

3. Dodjeljivanje vrijednosti varijabli

Možemo dodijeliti vrijednost varijabli nakon njene deklaracije koristeći operator dodjele (`=`). Naprimjer:

```
int broj;
```

```
broj = 10;
```

Ovdje smo prvo deklarirali varijablu `broj`, a zatim smo joj dodijelili vrijednost 10.

4. Imenovanje varijabli

Prilikom imenovanja varijabli važno je da imena budu smisljena i jasna. Također, imena varijabli u jeziku C mogu sadržavati slova (velika i mala), brojeve i donju crtu (`_`), ali moraju početi slovom ili donjom crtom.

Naprimjer, sledeće su ispravna imena varijabli:

```
int broj1;
```

```
int _broj;
```

```
int broj_korisnika;
```

Sljedeće nisu ispravna imena varijabli:

int 1broj; // nevalidno: ime počinje brojem

int broj-korisnika; // nevalidno: ime sadrži crticu

U svakom programu, korištenje varijabli je neizostavno. Razumijevanje kako deklarirati, inicijalizovati i dodijeliti vrijednosti varijablama je osnovna vještina koju svaki programer mora da savlada.

Tipovi podataka u C

Svaka varijabla koju deklariramo u programu mora imati svoj tip podatka. Tip podatka varijable određuje kakvu vrstu vrijednosti varijabla može da sadrži i koliko prostora zauzima u memoriji. C ima nekoliko osnovnih tipova podataka, a mi ćemo se fokusirati na četiri najčešće korišćena: `int`, `float`, `char` i `double`.

1. `int`

Tip `int` koristi se za predstavljanje cjelobrojnih vrijednosti. Naprimjer:

```
int broj = 10;
```

Ova linija deklarira varijablu `broj` tipa `int` i dodjeljuje joj vrijednost 10.

2. `float`

Tip `float` koristi se za predstavljanje decimalnih brojeva (brojeva sa pokretnim zarezom). Naprimjer:

```
float prosek = 4.5;
```

Ova linija deklarira varijablu `prosek` tipa `float` i dodjeljuje joj vrijednost 4.5.

3. `char`

Tip `char` koristi se za predstavljanje pojedinačnih karaktera. Naprimjer:

```
char inicijal = 'S';
```

Ova linija deklarira varijablu `inicijal` tipa `char` i dodjeljuje joj vrijednost 'S'. Važno je napomenuti da se karakteri u C-u stavljaju u jednostruke navodnike (apostrofi ili polunavodnik).

4. `double`

Tip `double` je sličan tipu `float`, ali može da čuva decimalne brojeve sa dvostrukom preciznošću. Naprimjer:

```
double precizan_prosjek = 4.53;
```

Ova linija deklarira varijablu `precizan_prosjek` tipa `double` i dodjeljuje joj vrijednost 4.53.

Osim ovih osnovnih tipova podataka, C takođe podržava nekoliko drugih, uključujući `long`, `short`, `unsigned int`, `signed int` i druge, ali za sada je važno razumjeti ove četiri osnovna tipa podataka i znati kada i kako ih koristiti.

Primjer programa za ispisivanje teksta sa detaljnim objašnjenjem naredbi

```
#include <stdio.h>

int main() {

    printf("Dobrodošli u svijet programiranja!\n");

    return 0;

}
```

Objašnjenje:

#include <stdio.h>: Ova linija je direktiva za preprocesor da uključi standardnu biblioteku za ulaz/izlaz `stdio.h`. Ova biblioteka se koristi za rad sa ulazom (input) i izlazom (output), kao što je `printf()` funkcija.

int main(): Ova linija predstavlja početak glavne funkcije `main`. Svaki C program mora imati glavnu funkciju kao početnu tačku.

printf("Dobrodošli u svijet programiranja!\n");: Ova linija koristi funkciju `printf()` da bi ispisala tekst "Dobrodošli u svijet programiranja!" na ekranu. `\n` je specijalni karakter koji predstavlja novi red.

return 0;: Ova linija označava kraj `main` funkcije. Vraća vrijednost 0, što ukazuje na to da je program uspješno izvršen.

Program u C-u za izračunavanje zbira dva broja, sa detaljnim objašnjenjem

```
#include <stdio.h>

int main() {

    int a, b, zbir;

    printf("Unesite prvi broj: ");

    scanf("%d", &a);

    printf("Unesite drugi broj: ");

    scanf("%d", &b);

    zbir = a + b;

    printf("Zbir brojeva %d i %d je: %d\n", a, b, zbir);

    return 0;

}
```

Objašnjenje:

#include <stdio.h>: Ova linija je direktiva za preprocesor da uključi standardnu biblioteku za ulaz/izlaz stdio.h. Ova biblioteka se koristi za rad sa ulazom (input) i izlazom (output), kao što su printf() i scanf() funkcije.

int main(): Ova linija predstavlja početak glavne funkcije main. Svaki C program mora imati glavnu funkciju kao početnu tačku.

int a, b, zbir;: Ova linija deklarira tri cjelobrojne (integer) promjenljive a, b i zbir.

printf("Unesite prvi broj: ");: Ova linija ispisuje poruku koja traži od korisnika da unese prvi broj.

scanf("%d", &a);: Ova linija koristi funkciju **scanf()** da preuzme broj koji korisnik unese sa tastature i dodeli ga promjenljivoj a. Specifikator formata **%d** se koristi za cjelobrojne vrijednosti. **&** se koristi da se pristupi memorijskoj lokaciji promjenljive a.

printf("Unesite drugi broj: "); i scanf("%d", &b);: Ove linije rade isto kao prethodne dvije, ali za drugi broj, koji se dodjeljuje promjenljivoj b.

zbir = a + b;: Ova linija računa zbir promjenljivih a i b i dodjeljuje ga promjenljivoj **zbir**.

printf("Zbir brojeva %d i %d je: %d\n", a, b, zbir); Ova linija ispisuje rezultat. Svaki **%d** je specifikator formata koji se koristi za ispis integera. Oni su mjesta na kojima će biti postavljene vrijednosti promjenljivih a, b i zbir.

return 0; Ova linija označava kraj main funkcije. Vraća vrijednost 0, što ukazuje na to da je program uspješno izvršen.

OVE I DRUGE PROGRAME U PROGRAMSKOM JEZIKU C MOŽETE PISATI PREKO RAZVOJNOG OKRUŽENJA KOJE MOŽETE PREUZETI I INSTALIRATI PREKO SLJEDEĆEG LINKA:

[DEV C, C++](#)