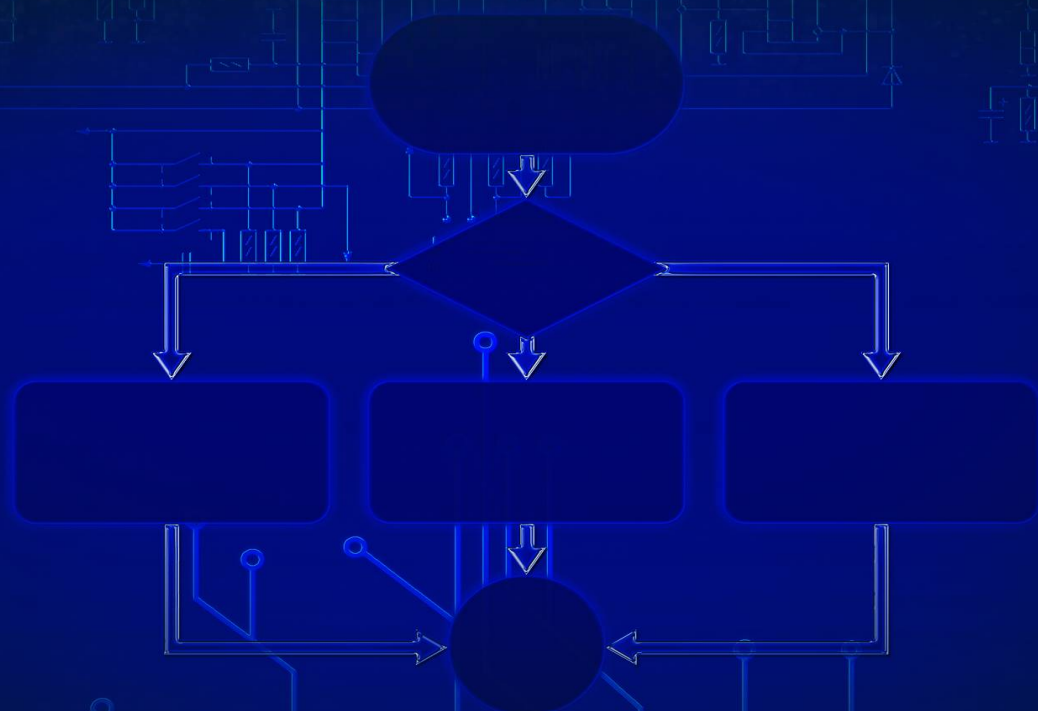


NAUČITE PROGRAMIRANJE UZ FlowRun

VIZUELNI UVOD U SVIJET
PROGRAMIRANJA



SAKIB HADŽIAVDIĆ

Naučite programiranje uz FlowRun

Vizuelni uvod u svijet programiranja

Sakib Hadžiavdć

Ova knjiga je dostupna na <https://leanpub.com/nauciteprogramiranjeuzflowrun>

Ova verzija je objavljena 2025-04-24



Ovo je [Leanpub](#) knjiga. Leanpub omogućava autorima i izdavačima korištenje Lean Publishing procesa. [Lean Publishing](#) je proces objavljivanja knjige u nastajanju koristeći jednostavne alate i više iteracija kako bi se dobila povratna informacija od čitatelja, prilagođavao sadržaj dok ne dobijete pravu knjigu i izgradili popularnost kada to postignete.

© 2025 Sakib Hadžiavdć

Contents

Poruka za Vas	1
Uvod	2
FlowRun editor	3
Kako vam FlowRun pomaže da bolje učite?	3
Osnove	4
Hello World!	4
Brojevi	5
Variable	7
Mijenjanje varijabli	9
Korisnički unos	10
Grananje pomoću If	12
Vježbe	17
Petlje	18
While petlja	18
Do While petlja	19
For petlja	20
Petlja u petlji	22
Vježbe	25
Funkcije	26
Predefinisane funkcije	26
Korisnički definisane funkcije	26
Povratna vrijednost funkcije	26
Rekurzivne funkcije	26
Opseg važenja funkcije (scope)	26
Vježbe	26
Nizovi	28

CONTENTS

Izračunavanje prosječne ocjene	28
Pronalaženje elementa u nizu	28
Frekvencija brojeva	28
Sortiranje niza	28
Stringovi su nizovi	28
Vježbe	28
Matrice	29
Sabiranje matrica	29
Provjera jedinične matrice	29
Vježbe	29
Primjeri iz stvarnog svijeta	30
Statistika	30
Iks Oks	30
Vježbe	30
Generisanje pravog koda	31
Najbolje prakse	32
Koristite smisljena imena	32
Koristite privremene varijable	32
Koristite funkcije	32
Recite korisniku šta da radi	32
Razgovarajte s korisnikom	32
Graciozno istretirajte greške	32
Česti problemi i rješenja	33
Varijabla nije deklarirana	33
Varijabla nije inicijalizovana	33
Pogrešan format unosa	33
Dijeljenje nulom	33
Pristup indeksu niza izvan granica	33
Ispravljanje problema	33
Šta dalje?	34
Resursi	35
FlowRun.io dokumentacija	35
FlowRun YouTube kanal	35
FlowRun diskusija uživo	35

FlowRun izvorni kod	35
Referenca predefinisanih funkcija	36
Rječnik pojmova	37

Poruka za Vas

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Uvod

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

FlowRun editor

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Kako vam FlowRun pomaže da bolje učite?

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Osnove

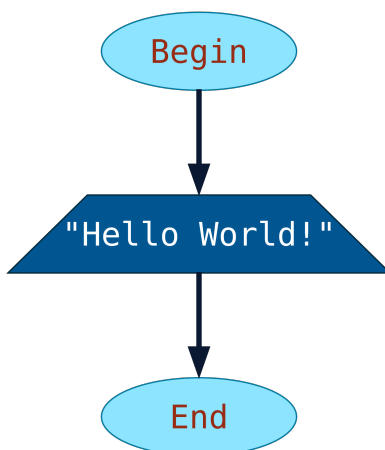
U ovom poglavlju ćete naučiti osnovne blokove potrebne za izradu jednostavnih programa u FlowRun-u.

Hello World!

Program “Hello World!” je tradicionalno prvi program koji napravite kada učite novi programski jezik. On ispisuje tekst “Hello World!” na ekranu.

Koraci:

- otvorite <https://flowrun.io/scratchpad>
- kliknite na strelicu koja pokazuje od Begin (en. “početak”) do End (en. “kraj”)
- odaberite Output (en. “izlaz”, “ispis”)
- kliknite na novi blok za izlaz
- promijenite njegov tekst u "Hello World!" (potrebni su navodnici)
- kliknite na dugme Pokreni



Slika 1. Primjer Hello World

Isprobajte sada!



Slika 2. QR kod za Hello World

Kada ga pokrenete, vidjet ćete ovakav izlaz:

```
1 Hello World!
```



Potrebni su nam navodnici oko `Hello World!` jer želimo običan tekst. Riječ `Hello` bi za FlowRun mogla značiti nešto posebno: ime varijable, ime funkcije itd. (o tome ćete učiti kasnije u ovoj knjizi).



Ovaj program možete sačuvati na flowrun.io ako želite, samo se prvo trebate Prijaviti. Kliknite na dugme “Sačuvaj”. Možete imati vlastitu kolekciju programa na flowrun.io, dijeliti ih sa prijateljima i nastavnicima, exportati kao sliku itd.

Brojevi

Imamo dvije vrste brojeva:

- cijeli brojevi (bez decimala) kao što su 5, -23, 0, 19
- realni brojevi (brojevi s decimalnim zarezom) kao što su 3.14, 0.004, -3.33

Možemo vršiti osnovne operacije nad njima:

- + sabiranje
- - oduzimanje
- * množenje
- / dijeljenje

- % ostatak

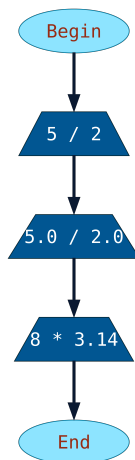
Zašto su nam potrebne dvije vrste brojeva? Zato što matematičke operacije nad njima mogu dati *različite rezultate*. Potrebne su nam obe vrste, ovisno o problemu koji pokušavamo riješiti.

Glavna razlika je kod dijeljenja:

- dijeljenje cijelih brojeva daje cijeli broj: $5 / 2 = 2$
- dijeljenje realnih brojeva daje realan broj: $5.0 / 2.0 = 2.5$

Koristite blok Output koji ste naučili u prethodnoj sekciji i ispišite ove izraze:

- $5 / 2$
- $5.0 / 2.0$
- $8 * 3.14$



Slika 3. Primjer Brojevi

Isprobajte sada!



Slika 4. QR kod za brojeve

Kada ga pokrenete, vidjet ćete ovakav izlaz:

```
1  2
2  2.5
3  25.12
```



Decimalni zarez se piše s tačkom (.), na primjer 3 . 14. Zarez (,) se koristi u druge svrhe, o čemu ćete učiti kasnije.

Modulo operator

Modulo operator (%) vraća *ostatak nakon dijeljenja*. Na primjer, 11 podijeljeno sa 4 je 2, s ostatkom 3. Ili drugačije rečeno, $4 * 2 + 3$ jednako je 11. Ovaj ostatak (3) možemo dobiti pomoću modulo operatora: $11 \% 4$.

Ovo je korisno za:

- utvrđivanje je li broj *djeljiv* nekim drugim brojem, ako jeste, onda je ostatak nula
- utvrđivanje je li broj paran (djeljiv sa 2) ili neparan (nije djeljiv sa 2)
- dobijanje zadnje cifre broja, dijeljenjem sa 10 (npr. $49 \% 10$ je 9)

Varijable

Varijable koristimo za **pohranu podataka**. Varijabla ima **ime** i **tip**.

Na primjer, možemo deklarirati varijablu broj: `Integer`. Ime varijable je broj, a njen tip je `Integer`. U notaciji koja se koristi u ovoj knjizi, odvojeni su dvotačkom (:).

Recimo da varijabla broj pohranjuje vrijednost 10. Imena su važna jer se u programu možemo pozivati na broj umjesto da svugdje ponavljamo broj 10. Lako ga je ažurirati s vrijednosti 10 na neku drugu vrijednost.

Ako programu kažemo da je tip varijable `Integer`, to će nam pomoći u nekim situacijama. Na primjer, može zabraniti pohranjivanje `Real` nih brojeva u tu varijablu (ili bilo kog drugog tipa), može osigurati da se neke funkcije mogu izvršavati nad određenim tipovima itd (o njima ćete učiti kasnije u knjizi).

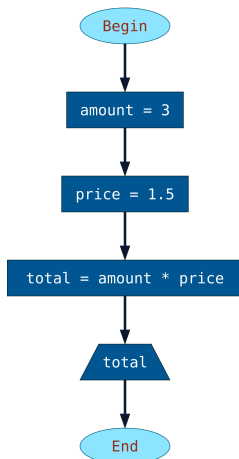
* * *

Prije nego što upotrijebimo varijablu, moramo je **deklarirati**, reći programu da ćemo je koristiti. Možemo joj odmah dodijeliti **početnu vrijednost**, što je dobra programerska praksa. Ako joj ne dodijelimo vrijednost, program bi se mogao srušiti. (Ništa se neće dogoditi vašem kompjuteru, ne brinite!)

U sljedećem primjeru deklariramo 3 varijable (koristeći blok `Declare`):

- `amount`: `Integer`, s početnom vrijednošću 3
- `price`: `Real`, s početnom vrijednošću 1.5
- `total`: `Real`, s početnom vrijednošću `amount * price`

Zatim ispisujemo `total` na ekranu.



Slika 5. Primjer Varijable

Isprobajte sada!



Slika 6. QR kod za varijable

Kada ga pokrenete, vidjet ćete ovakav izlaz:

1 4.5

Ovo je trivijalan primjer i sigurno biste ga lakše izračunali s kalkulatorom nego ovako. Ali, negdje se mora početi, a varijable su vrlo bitan gradivni blok!



Na varijable možete gledati kao na posude za hemikalije. Prvo, posude mogu biti prazne. Drugo, svakoj hemikaliji treba posebna vrsta posude. Vodu možete staviti u plastičnu posudu, ali ne možete staviti sumpornu kiselinu (rastopila bi plastiku). Na isti način ne možete staviti vrijednost tipa `Real` u varijablu koja je tipa `Integer`. Ali ponekad možete kombinirati te vrijednosti s operatorima i funkcijama (baš kao hemijske reakcije). U gornjem primjeru pomnožili smo `Real`ni broj s `Integer`om i to je radilo.

Mijenjanje varijabli

Značenje riječi “varijabla” je “varirati”, “mijenjati se”. Varijablu možete ažurirati **dodjeljivanjem nove vrijednosti**.

Uradite sljedeći primjer:

- deklarirate varijablu `x`: `Integer` s vrijednošću 5
- ispišite `x`
- dodajte blok `Assign`
- postavite ime varijable bloka za dodjelu na `x`, a vrijednost na 19
- ponovo ispišite `x`



Slika 7. Primjer Mijenjanje Varijabli

Isprobajte sada!



Slika 8. QR kod za mijenjanje varijabli

Kada ga pokrenete, vidjet ćete ovakav izlaz:

```
1 5
2 19
```

Početna vrijednost je 5, pa se ona ispiše prvi put. Nakon bloka za dodjelu vrijednost se promijenila u 19.

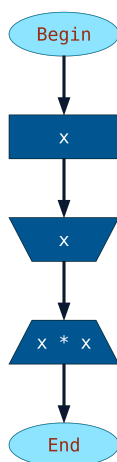
Korisnički unos

U prethodnim primjerima koristili smo fiksne (hardkodirane) vrijednosti u programima. Ali obično nam je potrebna vrijednost kao **unos od korisnika**. Kada se vrijednost pročita od korisnika, FlowRun je **pohranjuje u varijablu**.

Uradite sljedeći primjer:

- deklarirate varijablu x : Integer
- dodajte blok Input s imenom varijable x
- ispišite $x * x$

Upravo ste implementirali program koji izračunava kvadrat unesenog broja!



Slika 9. Primjer Korisnički Unos

Isprobajte sada!



Slika 10. QR kod za korisnički unos

Kada ga pokrenete, tražit će od vas da unesete broj. Ako unesete 5 na primjer, vidjet ćete ovakav izlaz:

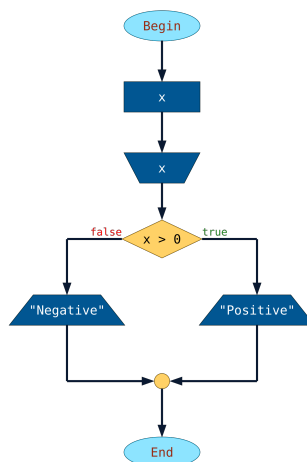
1 25

Grananje pomoću If

Vidjeli smo da program izvršava blok za blokom, od vrha prema dnu. Ali obično trebamo donijeti neke odluke (grananje) tokom izvršavanja programa ("ako ovo onda uradi x inače uradi y"). To radimo pomoću bloka `If`.

Napravimo program koji će nam reći je li broj pozitivan ili negativan:

- deklarirate varijablu `x`: `Integer`
- unesite `x`
- dodajte blok `If` s vrijednošću `x > 0`
- u grani `true` ispišite "Positive" (linija s oznakom `true`)
- u grani `false` ispišite "Negative" (linija s oznakom `false`)



Slika 11. Primjer Grananje

Isprobajte sada!



Slika 12. QR kod za grananje

Ako korisnik unese 9, izlaz će biti "Positive". Ako korisnik unese -55, izlaz će biti "Negative".



Izraz $x > 0$ je tipa Boolean. Boolean varijabla može biti ili `true` ili `false`.

Koristimo operatore poređenja (`<`, `<=`, `==`, `!=`, `>`, `>=`) za poređenje brojeva, koji vraćaju Boolean vrijednost. Operatore jednakosti i nejednakosti (`==` i `!=`) koristimo za poređenje drugih tipova.

Imajte na umu da se jednakost piše s dvostrukim znakom jednakosti (`==`). Znak jednakosti (`=`) koristi se za ažuriranje varijable (u većini programskih jezika).

Booleani

Vrijednost tipa Boolean može biti samo `true` ili `false`. Imamo 3 osnovna operatora za njihovo kombiniranje:

- `!`, operator negacije
- `&&`, operator I (AND)
- `||`, operator ILI (OR)

Operator negacije

Operator negacije obrće vrijednost. Piše se *ispred Boolean vrijednosti/varijable*. Recimo da imamo varijablu `b`: Boolean. Ako je `b == true`, onda je `!b == false`. Ako je `b == false`, onda je `!b == true`.

Operator I (AND)

Operator I (&&) kombinira dvije Boolean vrijednosti logičkim I. Koristimo ga kada želimo izvršiti neki dio koda samo ako su **obje vrijednosti true**. Ovdje je tabela sa svim mogućim kombinacijama:

a	b	a && b
false	false	false
false	true	false
true	false	false
true	true	true

Operator ILI (OR)

Operator ILI (||) kombinira dvije Boolean vrijednosti logičkim ILI. Koristimo ga kada želimo izvršiti neki dio koda ako je **bilo koja vrijednost true**. Ovdje je tabela sa svim mogućim kombinacijama:

a	b	a b
false	false	false
false	true	true
true	false	true
true	true	true

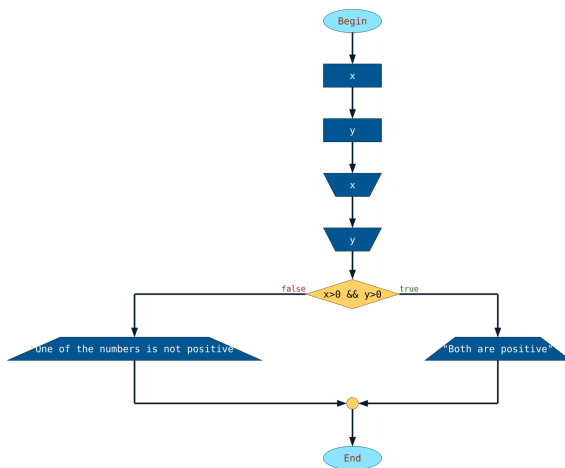
* * *

Napravimo sada program koji od korisnika uzima dva broja. Treba nam reći jesu li oba unesena broja pozitivna.

Koraci:

- deklarirate dva cijela broja: x i y
- unesite x i y

- dodajte blok If s izrazom $x > 0 \ \&\& \ y > 0$
- dodajte blok Output u grani true, s vrijednošću "Oba su pozitivna"
- dodajte blok Output u grani false, s vrijednošću "Jedan od brojeva nije pozitivan"



Slika 13. Primjer Booleani

Isprobajte sada!



Slika 14. QR kod za Booleane



Ako niste sigurni kojim redoslijedom se operacije izvršavaju, možete koristiti zagrade kako biste osigurali da se izvršava onako kako želite. Zadnji primjer bi bio isti kao $(x > 0) \ \&\& \ (y > 0)$. Dakle, kada imate mnogo podizraza, zagrade su vrlo korisne.



Operator $\&\&$ ima jaču vezu od $||$, tako se $a \ || \ b \ \&\& \ c$ izvršava kao $a \ || \ (b \ \&\& \ c)$. Možete zamisliti $\&\&$ kao množenje, a $||$ kao zbrajanje.

Stringovi

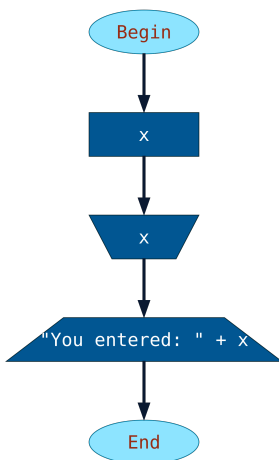
Tip `String` smo upoznali već u sekciji “Hello World!”. `String` je u suštini samo tekst, niz slova.

Postoji jedan vrlo čest operator koji se koristi nad `String`ovima, a zove se “operator spajanja”. Sintaksa koja se koristi u `FlowRun`-u je `"abc" + "def"`, koristimo `+` da spojimo dva `String`a u jedan. Rezultat je `String` vrijednosti `"abcdef"`.

Također možete spajati druge tipove sa `String`om. Ovo je korisno za ljepši ispis vrijednosti. Na primjer, možete ispisati `"Unijeli ste: " + x` (gdje je `x` broj ili nešto drugo). To bi rezultiralo `String`om koji sadrži tekst “Unijeli ste: ” i vrijednost varijable `x` dodanu na njega.

Napravimo program koji radi upravo to:

- deklarirate `x`: `Integer`
- unesite `x`
- ispišite `"You entered: " + x`



Slika 15. Primjer Stringovi

Isprobajte sada!



Slika 16. QR kod za Stringove

Vježbe

1. unesite ime: String od korisnika i pozdravite ga, npr. "Zdravo Bob!"
2. izračunajte obim kvadrata, koristeći od korisnika unesen a: Integer (dužina stranice)
3. izračunajte površinu kvadrata, koristeći od korisnika unesen a: Integer
4. izračunajte površinu pravougaonika, koristeći od korisnika unesene dužine stranica a: Integer i b: Integer
5. izračunajte površinu trougla, koristeći od korisnika unesene osnovica: Real i visina: Real
6. unesite dva ugla trougla i izračunajte treći
7. ispišite zbir, proizvod, razliku, količnik i ostatak za dva cijela broja unesena od korisnika
8. zamijenite dva cijela broja unesena od korisnika x i y koristeći treću varijablu t
9. neka korisnik unese svoju godinu rođenja, a zatim ispišite "Imate xyz godina", gdje je xyz njihova dob
10. unesite x: Integer i procenat: Real (0-100) i izračunajte taj procenat x
11. unesite x: Integer i ispišite njegovu apsolutnu vrijednost
12. unesite tri broja i pronađite najveći
13. ispišite je li cijeli broj paran ili neparan
14. ispišite "da" ako je cijeli broj unesen od korisnika *veći ili jednak 0* i manji od 100, inače ispišite "ne"
15. ispišite "da" ako je cijeli broj unesen od korisnika djeljiv sa 3 ili 5, inače ispišite "ne"
16. izračunajte sumu cifara za cijeli broj unesen od korisnika
17. unesite cijeli broj n i ispišite ga s obrnutim redoslijedom cifara, npr. 123 -> 321
18. unesite miligrami: Integer i izračunajte broj kilograma, grama i miligrama
19. unesite milisekunde: Integer i izračunajte broj sati, minuta, sekundi i milisekundi
20. unesite bajtovi: Integer i izračunajte broj MB, KB i bajta B

Petlje

Kada želimo izvršiti neki dio koda *više puta*, uvijek možemo kopirati i zalijepiti te blokove. Ali to nije baš efikasno niti čitljivo, stoga imamo **blokove petlji** kao što su `While`, `Do While` i `For`.

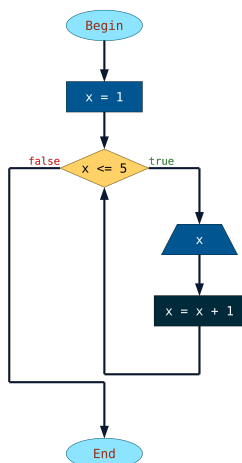
While petlja

`While` petlja ima dva dijela:

- `condition` - Boolean koji se provjerava *svaki put prije* nego što se `body` izvrši
- `body` - niz blokova koji se izvršavaju kada je uslov `true`

Napravite program koji ispisuje brojeve od 1 do 5:

- deklarirate varijablu `x`: `Integer = 1`
- dodajte `While` blok s uslovom `x <= 5`
- ispisati `x` u tijelu `While` petlje
- dodijelite `x = x + 1` u tijelu `While` petlje



Slika 17. While petlja

Isprobajte sada!



Slika 18. QR kod za While petlju

Trebali biste vidjeti ovaj ispis:

```
1 1
2 2
3 3
4 4
5 5
```

Obratite pažnju na blok Assign: $x = x + 1$. Možete koristiti prethodnu vrijednost x (*prije* nego što je dodjela izvršena), dok izračunavate njegovu novu vrijednost! Zar to nije super? :)



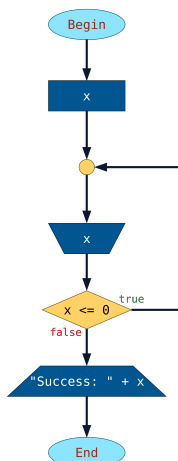
Sve petlje moraju imati uslov za zaustavljanje kako bi se izbjegle **beskonačne petlje**. Provjerite da li uslov smisla i da ažurirate varijable uslova tako da se petlja na kraju zaustavi. Ako dobijete beskonačnu petlju, ne brinite, kliknite na **Stop button**, popravite program i pokušajte ponovo.

Do While petlja

Blok `Do While` je vrlo sličan `While`. Glavna razlika je u tome što `Do While` **provjerava uslov nakon što izvrši tijelo**. To znači da izvršava blok *barem jednom*.

Uradimo program koji čita broj od korisnika. Ako uneseni broj nije pozitivan, ponovo tražimo od korisnika da unese broj. Koraci:

- deklarirajte `x`: `Integer`
- dodajte `Do While` blok s uslovom `x <= 0`
- unesite `x` u tijelo `Do While` petlje (strelica pokazuje od kruga do dijamanta `Do While`)



Slika 19. Do While petlja

Isprobajte sada!



Slika 20. QR kod za Do While petlju

Čitamo ga ovako: “Unesi broj x, sve dok je manji ili jednak nuli”.

Pokrenite program nekoliko puta, isprobajte različite unose: 1, 17, -55, 6. Program će izaći iz bloka `Do While` samo kada unesete pozitivan broj.

For petlja

For petlja je malo složenija. Sastoji se od:

- imena varijable petlje
- početne vrijednosti varijable petlje
- krajnje vrijednosti varijable petlje

- **koraka** (inkrementa), za koliko se varijabla petlje povećava u svakoj iteraciji

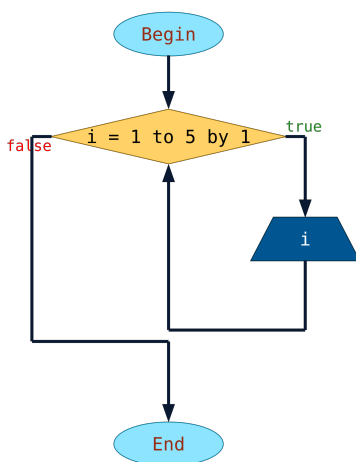
For petlja zamjenjuje sljedeći kod:

- deklarisanje `i`: Integer s *početnom vrijednošću*
- `While` s uslovom `i <= end`
- dodijela `i = i + step` na kraju `While` petlje

* * *

Lakše je razumjeti petlju u akciji. Uradimo isti program kao prije, ispišimo prvih 5 brojeva:

- deklarirate For petlju s imenom varijable `i`
- postavite početnu vrijednost `i` na 1
- postavite krajnju vrijednost `i` na 5
- postavite korak na 1
- ispišite `i` u tijelu For petlje



Slika 21. Primjer For Petlje

Isprobajte sada!



Slika 22. QR kod za For petlju

Dobili smo isti rezultat kao i prije s `While` petljom, ali s puno manje koda. Osim toga, čitljivije je nego prije!



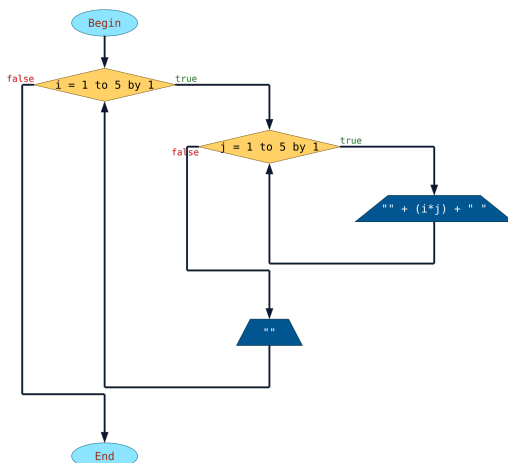
U FlowRun-u, `For` varijabla (obično nazvana `i`) se automatski deklarira. Ako deklarirate varijablu `i` prije petlje, FlowRun će je koristiti.

Petlja u petlji

Kada koristite petlju *unutar druge petlje*, ta unutrašnja petlja se naziva “unutrašnja petlja” (ili “ugniježdena petlja”). Vanjska petlja je naravno “vanjska petlja”.

Uradimo tablicu množenja 5x5, koristeći ugniježdene `For` petlje:

- dodajte `For` petlju s `i` kao varijablom petlje, počevši od 1 do 5
- dodajte `For` petlju s `j` kao varijablom petlje, počevši od 1 do 5
- ispišite `" " + (i*j) + " "` unutar tijela unutrašnje petlje i provjerite je li “New Line” onemogućeno
- ispišite prazan string u vanjskoj petlji, nakon unutrašnje petlje, i provjerite je li “New Line” omogućeno



Slika 23. Tablica množenja

Isprobajte sada!



Slika 24. QR kod tablice množenja

Kada ga pokrenete, trebali biste vidjeti ovo u izlazu:

```

1  1 2 3 4 5
2  2 4 6 8 10
3  3 6 9 12 15
4  4 8 12 16 20
5  5 10 15 20 25
  
```



Pazite da **ugniježdenim petljama** date različita imena varijabli. U našem primjeru imali smo *i* i *j*, što je u redu. Ako slučajno stavite *i* u obje petlje, FlowRun će koristiti *i* deklarisanu u vanjskoj petlji i vjerovatno ćete dobiti pogrešan rezultat (rijetko to želite učiniti).

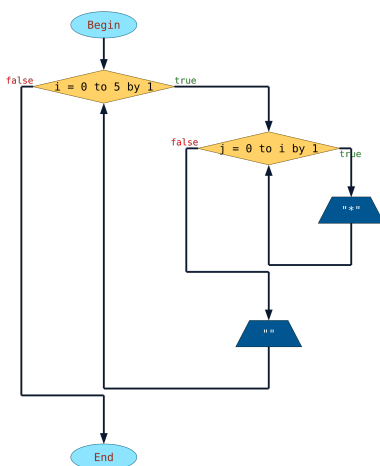
Uradimo još jedan program, ispisat će “trokut zvjezdica” poput ovog:

```
1 *
2 **
3 ***
4 ****
5 *****
6 *****
```

Koraci:

- dodajte For petlju s i kao varijablom petlje, počevši od 0 do 5
- dodajte For petlju s j kao varijablom petlje, počevši od 0 do i
- ispišite "*" unutar tijela unutrašnje petlje i provjerite je li “New Line” onemogućeno
- ispišite prazan string u vanjskoj petlji, nakon unutrašnje petlje, i provjerite je li “New Line” omogućeno

Kada ga pokrenete, vidjet ćete gornji uzorak.



Slika 25. Trokut zvjezdica

Isprobajte sada!



Slika 26. QR kod za trokut zvjezdica

Vježbe

1. prođite kroz cijele brojeve od 1 . . n i ispišite svaki broj koji je djeljiv i sa 5 i sa 3
2. pronađite sumu prvih n cijelih brojeva
3. unosite brojeve od korisnika sve dok su > 0 i ispišite najveći uneseni broj
4. unosite brojeve od korisnika sve dok su > 0 i ispišite njihov prosjek
5. izračunajte faktorijel koristeći `while` petlju ($n! = n * (n-1) * (n-2) \dots * 1$)
6. izračunajte faktorijel koristeći `for` petlju ($n! = n * (n-1) * (n-2) \dots * 1$)
7. ispišite obrnuti trokut, prvo 5 zvjezdica, zatim 4 i tako dalje...
8. ispišite kvadrat 10x10 napravljen od zvjezdica (samo vanjske zvjezdice naravno)
9. ispišite svoje ime u sredini kvadrata 10x10 zvjezdica

Funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Predefinisane funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Korisnički definisane funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Povratna vrijednost funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Rekurzivne funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Rekurzivno računanje faktoriijela

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Opseg važenja funkcije (scope)

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Vježbe

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Nizovi

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Izračunavanje prosječne ocjene

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Pronalaženje elementa u nizu

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Frekvencija brojeva

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Sortiranje niza

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Stringovi su nizovi

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Vježbe

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Matrice

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Sabiranje matrica

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Provjera jedinične matrice

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Vježbe

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Primjeri iz stvarnog svijeta

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Statistika

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Iks Oks

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Vježbe

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Generisanje pravog koda

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Najbolje prakse

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Koristite smislena imena

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Koristite privremene varijable

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Koristite funkcije

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Recite korisniku šta da radi

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Razgovarajte s korisnikom

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Graciozno istretirajte greške

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Česti problemi i rješenja

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Varijabla nije deklarirana

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Varijabla nije inicijalizovana

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Pogrešan format unosa

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Dijeljenje nulom

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Pristup indeksu niza izvan granica

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Ispravljanje problema

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Šta dalje?

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Resursi

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

FlowRun.io dokumentacija

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

FlowRun YouTube kanal

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

FlowRun diskusija uživo

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

FlowRun izvorni kod

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Referenca predefinisanih funkcija

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.

Rječnik pojmova

Ovaj sadržaj nije dostupan u probnoj verziji knjige. Knjiga se može kupiti na Leanpub-u na <https://leanpub.com/nauciteprogramiranjeuzflowrun>.