



Distance Learning System

Uvod u programiranje

Python and programming fundamentals

vladimir.maric@link.co.rs

aleksandra.lazarevic@link.co.rs

Uvod u programiranje

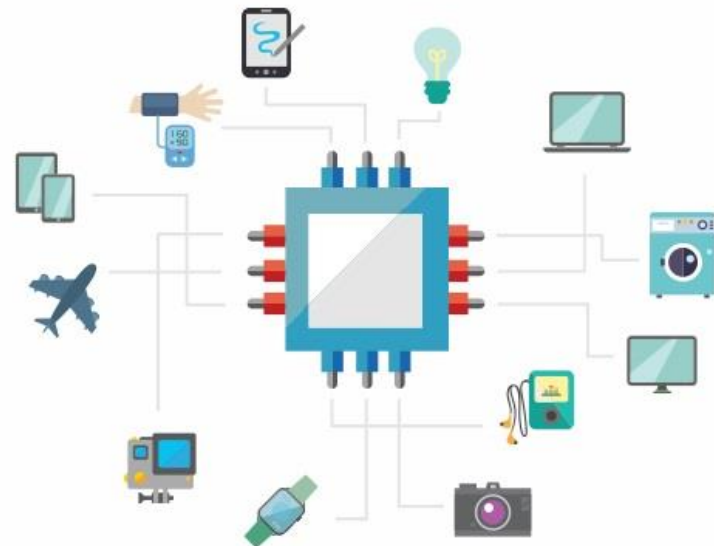
- Ono što čoveka izdvaja od drugih vrsta jesu alati koje je razvio kako bi proširio i pojačao moć svog intelekta. Prvi takav „alat“ jeste jezik (govor i pismo).
- Govor je čoveku omogućio prenos saznanja i artikulaciju intelekta, a pismo uvid u znanja koja su formirana i razvijana od strane desetina prethodnih generacija. Redovi koje vi upravo čitate idealan su primer moći pisane reči, koja vam omogućava da saznate nešto što do sada niste znali, te da na taj način pokušate da moć svog intelekta iskoristite na potpuno novim poljima.

Artikulisane govora i pojava pisama omogućili su čovečanstvu da još većom brzinom nastavi napred. Poslednje u nizu velikih dostignuća ljudske vrste jesu **kompjuteri** – mašine koje su čoveku omogućile proširivanje umne snage do neslućenih granica.

Svet kompjutera

Kompjuteri su danas svuda oko nas, unutar desktop i laptop računara, pametnih telefona, tableta, pametnih časovnika, televizora, automobila, aviona, veš-mašina, šporeta, klima-uređaja, bankomata...

Većina svakodnevnih aktivnosti danas se ne može zamisliti bez kompjutera. Iako toga možda niste ni svesni, veliki broj uobičajenih dnevnih operacija u potpunosti zavisi od kompjutera. Tako se sa pravom može reći da je današnji svet – svet kompjutera.



Šta je kompjuter?

- Kompjuter je uređaj koji je sposoban da izvrši proizvoljan broj aritmetičkih i logičkih operacija
- Kompjuter je uređaj koji može da:
 - prihvati neke ulazne podatke,
 - izvrši obradu podataka,
 - proizvede izlaz koji može biti razumljiv čoveku ili nekoj drugoj mašini.

Iako se pojam kompjutera poistovećuje sa desktop i laptop računarima, veoma je bitno razumeti da kompjuteri postoje u veoma širokom spektru oblika. Na hiljade kompjutera skriveno je u uređajima oko nas iako toga u nekim situacijama nismo ni svesni.

Šta su programi

Svaki kompjuter kontroliše program, što je zapravo skup instrukcija koje kompjuteru govore šta je potrebno da uradi. Bez takvih instrukcija, kompjuter bi bio mašina koja ne bi imala nikakvu praktičnu upotrebu

S obzirom na to da se kompjuteri rađaju kao uređaji bez ikakve svesti, jasno je da se procesom pisanja instrukcija, objedinjenih unutar programa, kompjuterima udahnuje život. Takav proces drugačije se naziva **programiranje**.

without program
computer is
dead machine



program



computer

Šta je programiranje

Ono što kompjutere izdvaja od ostalih mašina koje je čovek tokom vremena napravio jeste mogućnost programabilnosti. To praktično znači da je kompjuter, uopšteno posmatrano, jedna univerzalna alatka koja u zavisnosti od instrukcija koje joj se proslede može obavljati različite poslove. Upravo u ovoj činjenici krije se i najveća moć ovih mašina – potpuna proizvoljnost programabilnosti.

Proces stvaranja kompjuterskih programa naziva se **programiranje**. Tako je programiranje čin pisanja instrukcija koje omogućavaju da kompjuter obavi neku operaciju. Iako uopšteno vlada drugačije mišljenje, programiranje je veoma kreativan proces, koji u velikoj meri sadrži razne aspekte umetnosti, inženjerstva i nauke.

Ko su programeri?

Kompjuterske programe kreiraju programeri, ljudi koji poseduju neophodna znanja kako bi bili u mogućnosti da napišu instrukcije koje diriguju kompjuterom. Moderno kompjutersko programiranje može se smatrati relativno novom profesijom, starom tek nešto više od pedesetak godina. Ipak, prvi oblici programiranja zabeleženi su znatno ranije, u vreme postojanja mehaničkih računara, značajno drugačije konstrukcije od onih kakve danas poznajemo.

Prvim kompjuterskim programerom na svetu smatra se **Ada Lovelace**, ćerka pesnika Lorda Bajrona (engl. Lord Byron).



Izvor https://en.wikipedia.org/wiki/Ada_Lovelace

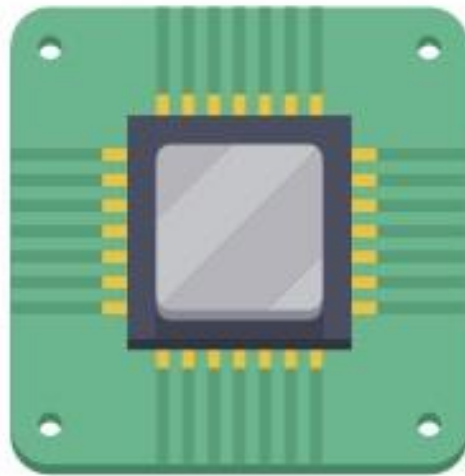
Programski jezici

U svakodnevnom životu, ljudi se međusobno sporazumevaju korišćenjem prirodnih jezika (engl. natural languages). Jezik je ništa drugo d vanapred utvrđen skup pravila, koja se primenjuju kako bi se misli artikulisale i prezentovale nekoj drugoj osobi.

Na identičan način obavlja se i proces komunikacije sa kompjuterima. Ipak, kompjuteri ne razumeju nijednu vrstu prirodnih jezika koje ljudi koriste, te je stoga neophodno kompjuteru se obratiti na njegovom maternjem jeziku.

Srce svakog kompjutera jeste centralna procesorska jedinica (engl. Central Processing Unit, CPU) ili jednostavno procesor. Tako je komunikacija sa kompjuterom zapravo razgovor sa centralnom procesorskom jedinicom.

Kompjuterski procesor nije u stanju da razume jezik kojim se ljudi svakodnevno služe. Sve što procesor razume jeste serija bitova, odnosno nula i jedinica, koji se prevode u određene instrukcije. Takva serija bitova, koja se interpretira u različite naredbe, naziva se **mašinski jezik**.



Mašinski jezik

Mašinski jezik je maternji jezik kompjutera, odnosno jedini jezik koji kompjuter razume. Programski kod ovakvog jezika sačinjen je iz niza nula i jedinica, a svaka od takvih cifara naziva se **bit**.

Korišćenje mašinskog jezika za upućivanje instrukcija računaru, može biti veoma zahtevan i dugotrajan posao.

10111001	00000000
11010010	10100001
00000100	00000000
10001001	00000000
00001110	10001011
00000000	00011110
00000000	00000010
10111001	00000000
11100001	00000011
00010000	11000011
10001001	10100011
00001110	00000100
00000010	00000000

Asembler

Asembler, često skraćeno imenovan kao **asm**, predstavlja programski jezik niskog nivoa, kod kojeg se osnovne programske naredbe pišu korišćenjem unapred utvrđenih reči, koje simbolizuju naredbu koju predstavljaju

Takve asemblerske jezičke konstrukcije veoma često se nazivaju i simboli, a sam assembler simboličkim jezikom

Ipak, i pored toga što je značajno jednostavniji za razumevanje, asemblerski jezik funkcioniše na identičan način kao i mašinski, s obzirom na to da između asemblerskih i mašinskih naredbi skoro uvek postoji odnos jedan prema jedan.

```
MOV CX,1234
MOV DS:[0],CX

MOV CX,4321
MOV DS:[2],CX

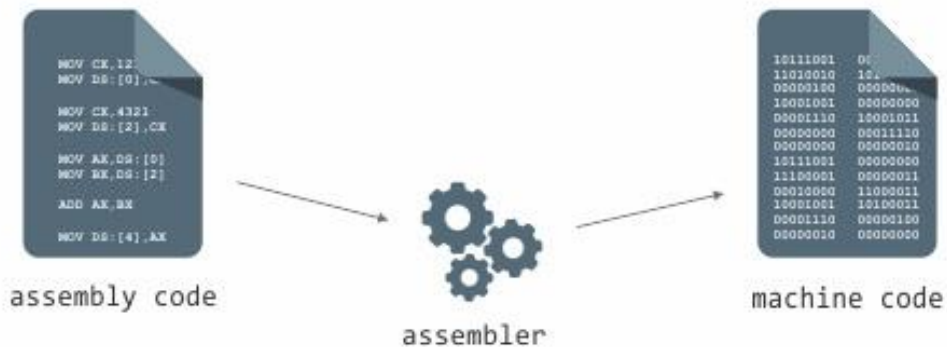
MOV AX,DS:[0]
MOV BX,DS:[2]

ADD AX,BX

MOV DS:[4],AX
```

Asembler

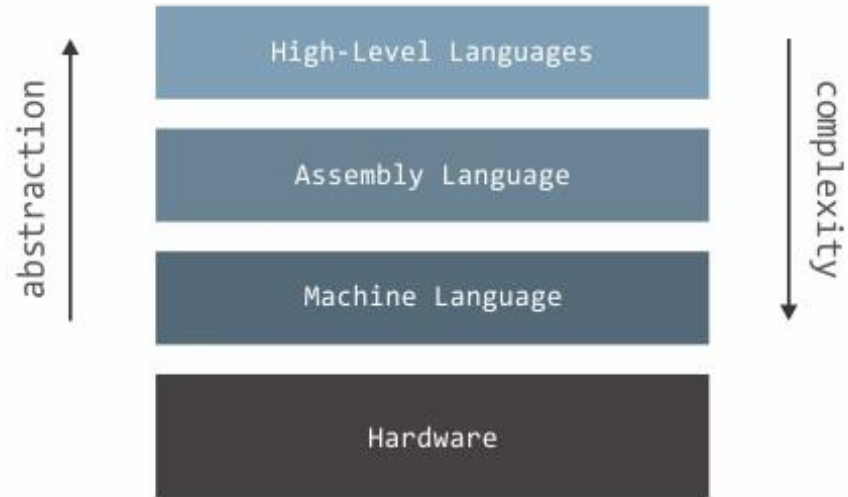
Kompjuter nije naučio asemblerske naredbe, već se one, kako bi procesor bio u stanju da ih razume, pre izvršavanja prevode u oblik mašinskog koda. Svaka asemblerska naredba prevodi se u odgovarajući skup nula i jedinica. Takav posao obavlja program koji se naziva asembler. On je zadužen da asemblerske naredbe prevede u mašinski kod



Jezici višeg nivoa

Programski jezici višeg nivoa, u znatno većoj meri koriste elemente prirodnih jezika, uglavnom reči i skraćenice preuzete iz engleskog jezika, te na taj način unapređuju čitljivost i olakšavaju kreiranje programskog koda.

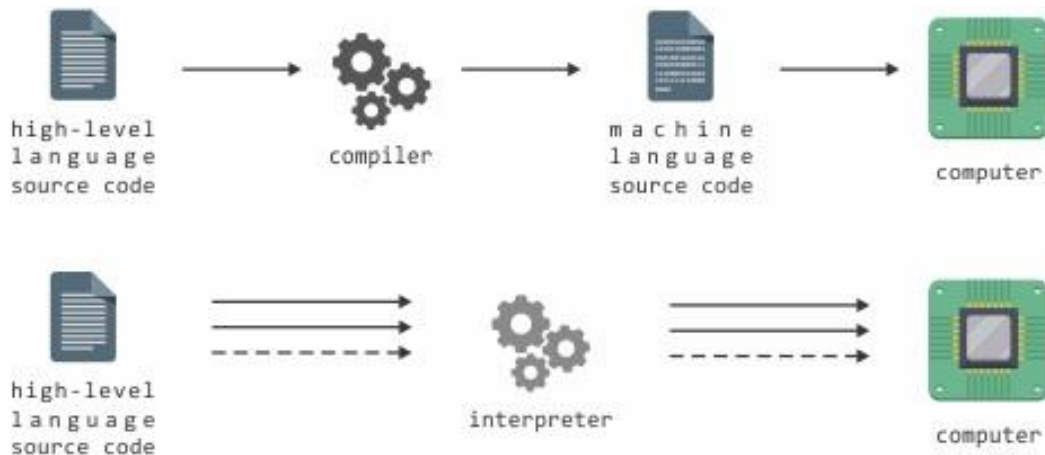
Programi napisani jezicima višeg nivoa na neki način moraju biti prevedeni u mašinski jezik pre nego što ih kompjuter izvrši, baš kao što je to bio slučaj i sa asemblerskim kodom. Programski jezici višeg nivoa za obavljanje takvog posla koriste **kompajlere** i **interpretere**.



Kompajler i interpreter

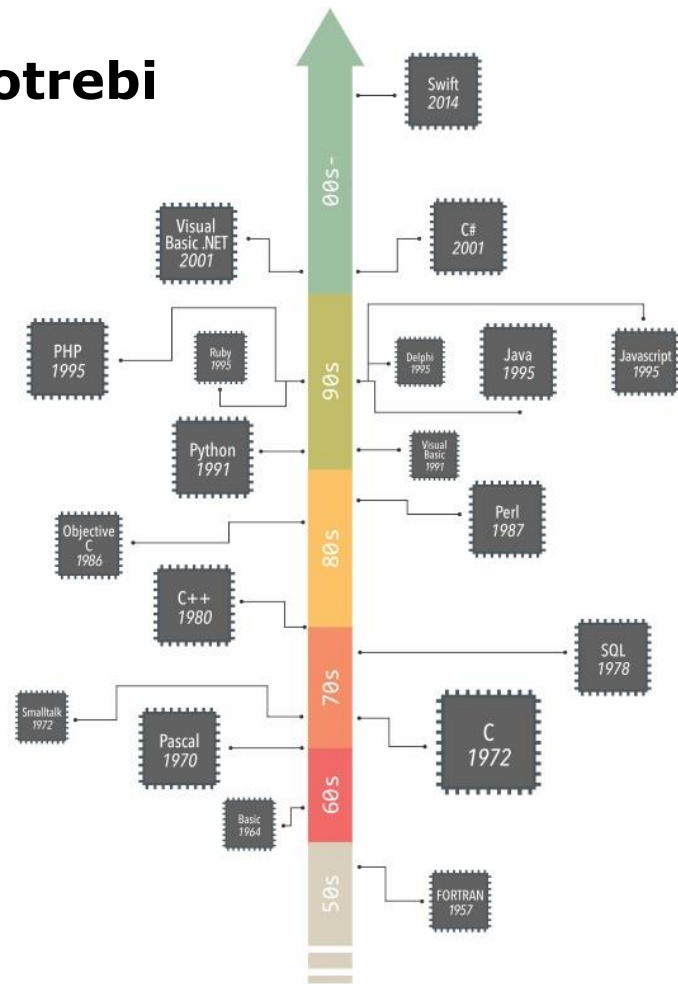
Kompajler prevodi izvorni kod programa napisanih jezicima višeg nivoa u mašinski, kako bi takav kod mogao kompjuter da izvrši. Posao koji kompajler obavlja drugačije se naziva kompajliranje, a jezici koji koriste kompajler - kompajlirani jezici

Interpreter obavlja prevođenje manjih delova izvornog koda, i to u toku njegovog izvršavanja. Tako korišćenjem interpretera nije potrebno prevesti kompletan program, već se prevođenje obavlja u delovima, prilikom izvršavanja.



Moderni programski jezici u širokoj upotrebi

Veoma zahtevan posao jeste izmeriti popularnost nekog jezika, s obzirom na to da se takva karakteristika može posmatrati kroz prizmu mnogih različitih parametara (broja napisanih programa, veličine programerske zajednice, količine dostupnih pozicija novih radnih mesta itd.)



Šta je algoritam

Algoritam se definiše kao ispravno opisana procedura za rešavanje nekog problema, najčešće problema u matematici ili kompjuterskom programiranju.

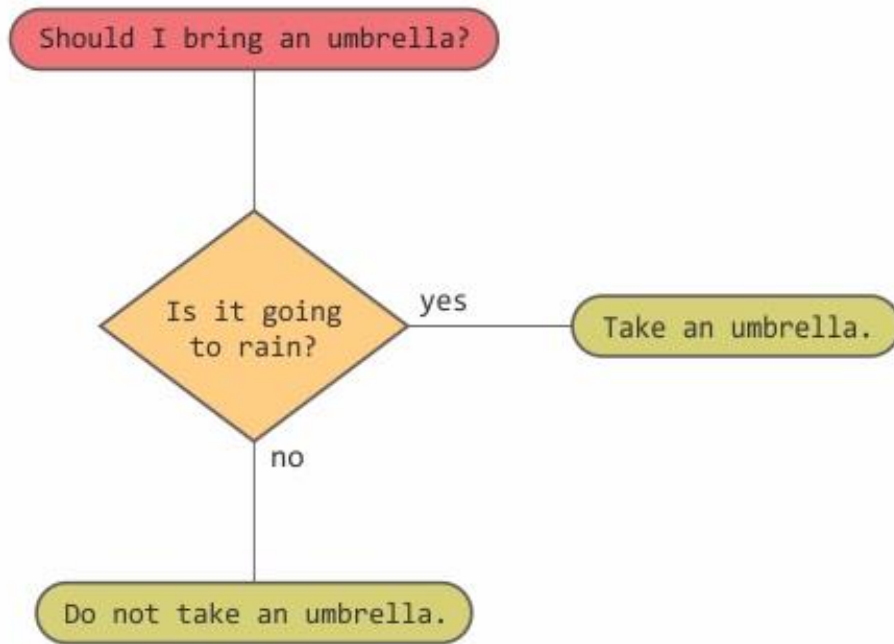
Algoritmi kompjuterskog programiranja u potpunosti su nezavisni od programskog jezika, odnosno tehnologije koja će biti korišćena za realizaciju takvih algoritama.



Dijagram toka





Dijagram toka jeste jedan od načina na koji je grafički moguće predstaviti korake nekog procesa.

Dijagrami tokova sačinjeni su iz niza grafičkih simbola, koji predstavljaju osnovne gradivne blokove ovakvih dijagrama.



Simboli dijagrama toka

Simbole koji se mogu koristiti prilikom kreiranja dijagrama tokova standardizovali su Američki nacionalni institut za simbole (engl. American National Symbols Institute, ANSI) i Internacionalna organizacija za standardizaciju (engl. The International Organizations for Standardization, ISO)

Simbol	Naziv
	Terminal
	Operacija
	Ulaz/Izlaz
	Grananje

Simbol	Naziv
	Predefinisani proces
	Linija toka
	Konektor
	Komentar

Rešavanje problema u kompjuterskom programiranju

Svaki problem može se definisati kroz prizmu **ulaza** i očekivanih osobina **izlaza**. Koraci koji bi učestvovali u procesu rešavanja problema mogli bi se nazvati **procedurom**.



Rešavanje problema u kompjuterskom programiranju

Još 1966. godine kompjuterski naučnici Corrado Böhm i Giuseppe Jacopini uvideli su i demonstrirali da se gotovo svi kompjuterski algoritmi mogu predstaviti korišćenjem tri osnovne kontrolne strukture:

sekvenca
selekcija
ponavljanje



sequence



selection

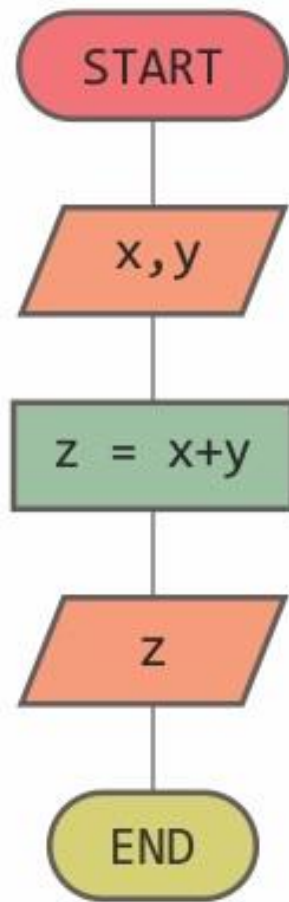


repetition

Sekvenca

Sekvencijalna struktura je konstrukcija kod koje se koraci algoritma izvršavaju jedan za drugim

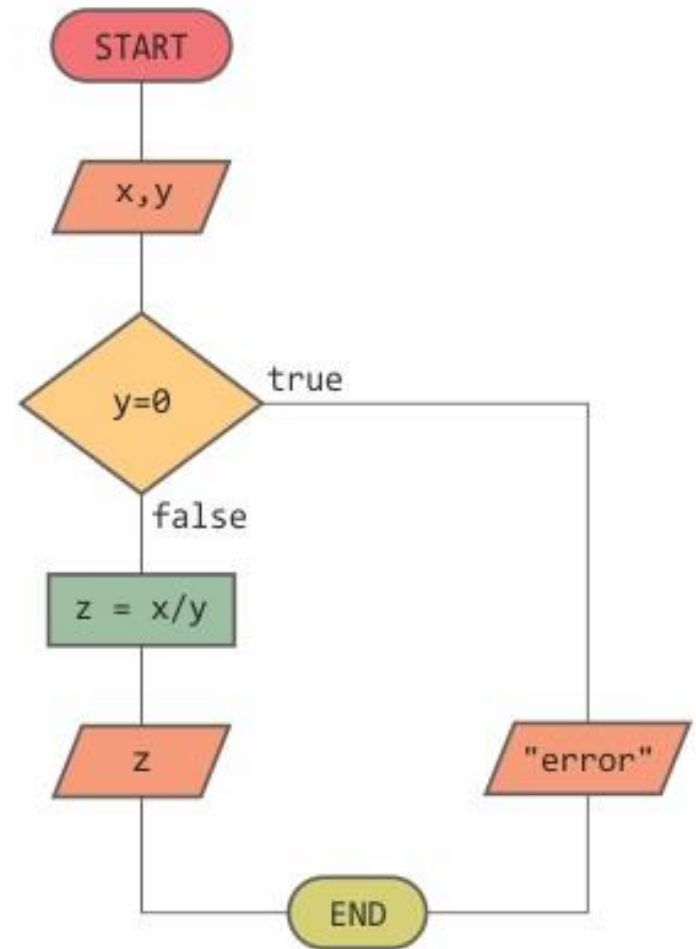
Sekvencijalnom strukturom algoritma mogu se rešiti samo najjednostavniji problemi prilikom razvoja softvera. Stoga se prilikom programiranja ovakva struktura sprovođenja koraka programske logike veoma retko koristi za rešavanje kompletnog problema. Mnogo se češće pribegava kombinovanju sekvencijalne (linijske) strukture sa jednim od dva kontrolna toka opisana u nastavku lekcije.



Selekcija

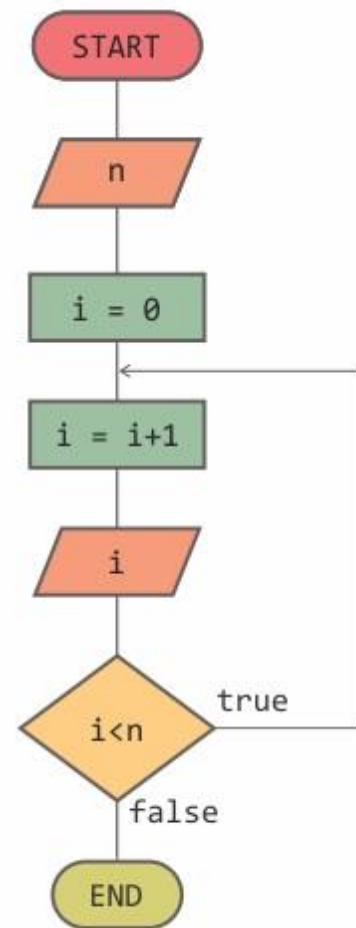
Selekcija je jedna od struktura izvršavanja programske logike, koja omogućava da se određeni koraci uopšte ne izvrše. Tako selekcija omogućava modelovanje logike koja treba da sadrži neku odluku

Kada se govori o kompjuterskom programiranju, algoritam koji u sebi sadrži strukturu selekcije može se ilustrovati rešavanjem problema deljenja dva broja



Ponavljanje

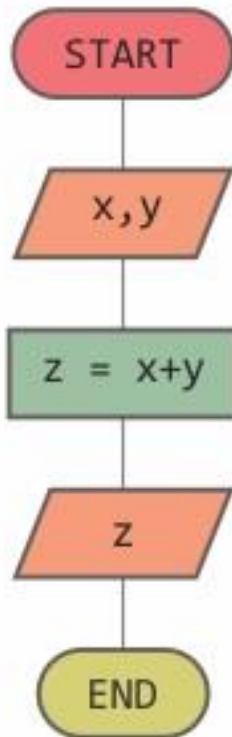
Struktura ponavljanja omogućava da se određeni koraci algoritma, samim tim, i kompjuterskog programa izvrše više puta, sve dok za tim ima potrebe. Idealni primer situacije koja bi zahtevala jedan takav scenario bio bi kompjuterski program čiji bi cilj bio ispisivanje prvih n celih pozitivnih brojeva.



Pseudokod

Pseudokod je oblik za izražavanje algoritama, koji je najbliži pisanju izvornog koda nekog programskog jezika.

Tako pseudokod omogućava da se najviše približimo postupku pisanja izvornog koda, ali bez potrebe da razmišljamo o sintaksnim pravilima konkretnog jezika



```
GET x, y
COMPUTE z
      z = x+y
DISPLAY z
```

Vežba 1

Problem: Na ulazu u lokalni noćni klub instaliran je čitač ličnih karata. Cilj takvog uređaja jeste provera starosti osobe koja želi da uđe u klub. Ulaz u klub je zabranjen muškim osobama mlađim od 18 godina i ženskim osobama mlađim od 16 godina, kao i svim osobama starijim od 70 godina (primer sa godinama je isključivo hipotetički radi vežbe).

Potrebno je napraviti algoritam na osnovu kojeg bi čitač ličnih karata utvrđivao kome je ulaz u klub dozvoljen, a kome nije.

Rešenje 1

GET age, gender

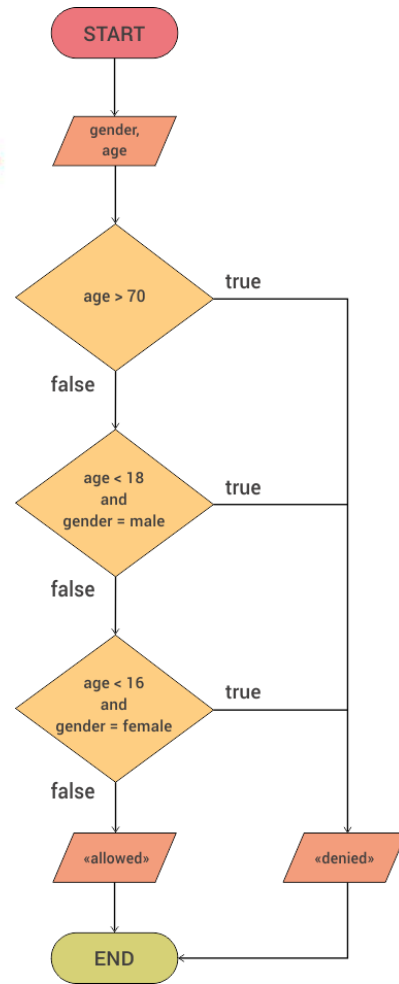
IF age > 70 THEN DISPLAY "denied"

ELSE IF age < 18 AND gender == male THEN DISPLAY "denied"

ELSE IF age < 16 AND gender == female THEN DISPLAY "denied"

ELSE

DISPLAY "allowed"



Vežba 2

Problem: Naći najveći od 3 uneta broja.

Rešenje 2

GET a, b, c

ASSIGN max = a

IF b > max THEN max = b

IF c > max THEN max = c

DISPLAY max

