

Praktikum iz programiranja 1



2024/25



○ kursu

- **Model 100 poena**

- Redovno prisustvo nastavi = 4 poena
- 2 kolokvijuma = 96 poena
- 40+56

Kontakti

- Microsoft Teams ([eng5l88](#))

Nastavnik:

- dr Marina Svičević
marina.svicevic@pmf.kg.ac.rs
marina.svicevic@gmail.com

Asistenti:

- Ana Vidosavljević
4-2019@pmf.kg.ac.rs
- Milica Vasović
30-2020@pmf.kg.ac.rs
- Jana Brzaković
27-2021@pmf.kg.ac.rs

Najvažnije teme

- Cilj predmeta: programerska pismenost
- Uvod u programiranje
- Osnove programskog jezika Python
 - aritmetičke operacije
 - standardni ulaz/izlaz
 - ugrađene funkcije
 - upravljačke komande, grananje, ponavljanja
 - funkcije, rekurzivne funkcije
 - stringovi
- Strukture - liste

Kompjuterski programi

Rešavanje problema korišćenjem kompjuterskih programa se može razložiti na više etapa:

1. Definisanje problema
2. Analiza problema
3. Definisanje algoritma
4. Projektovanje programa
5. Kodiranje
6. Testiranje
7. Analiza rezultata
8. Isporuka programa
9. Održavanje

Šta je algoritam?

- Algoritmi su korišćeni davno pre nego što je iko pokušao da definiše šta je to algoritam.

Primeri:

- Euklidov postupak za konstruisanje pomoću lenjira i šestara
- Euklidov postupak za određivanje NZD
- Vavilonski iterativni postupak za određivanje kvadratnog korena (1500 g.p.n.e.).

Šta je algoritam?

Postupak za određivanje kvadratnog korena

- Da biste odredili kvadratni koren bilo kog pozitivnog broja S :
 - 1. Napravite početnu pretpostavku.** Pretpostavite da je vrednost kvadratnog korena od S neki pozitivni broj x_0 .
 - 2. Poboljšajte pretpostavljeni rezultat.** Primените формулу $x_1 = (x_0 + S / x_0) / 2$. Broj x_1 je bolja aproksimacija za $\text{sqrt}(S)$.
 - 3. Ponavljajte korak 2 dok ne konvergirate ka rešenju.** Primенjuјte формулу $x_{n+1} = (x_n + S / x_n) / 2$ sve dok se cifre brojeva x_{n+1} i x_n ne budu poklapale do u želјenu decimalu.

Algoritmi

Reč algoritam, onako kako je mi danas poznajemo, prvi je upotrebio persijski matematičar Musa Al Horezmi pre više od 1200 godina.

- Konačan uređen niz precizno formulisanih pravila kojima se rešava jedan ili čitava klasa problema

Može se opisati tekstualno na prirodnom jeziku ili grafički



Zadatak

- Opisati algoritam za zamenu probušene gume na autu



Zadatak

Algoritam

- Probušena guma na autu:

Potrebno: auto

Ako je guma probušena onda

Potrebno: rezervni točak, dizalica, ključ

Podići auto dizalicom

Ako ima neodšrafljenih šrafova

Odšrafiti šraf

Skinuti točak

Staviti rezervni točak

Ako ima nezašrafljenih šrafova

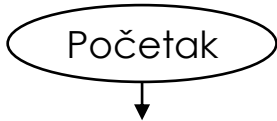
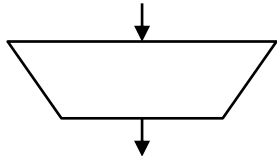
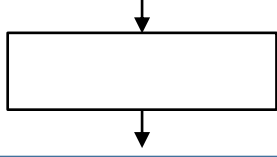
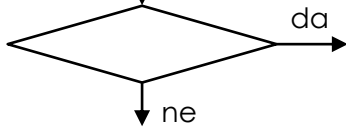
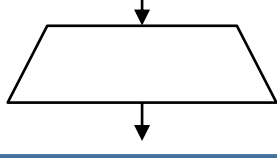
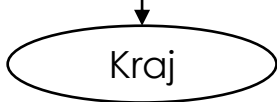
Zašrafiti šraf

Spustiti auto

Spakovati alat

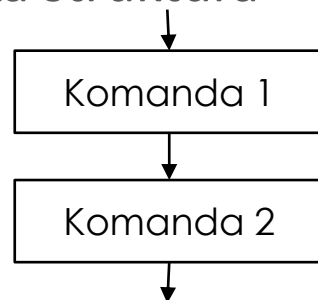
Srećan put

Alogitmi – grafički simboli

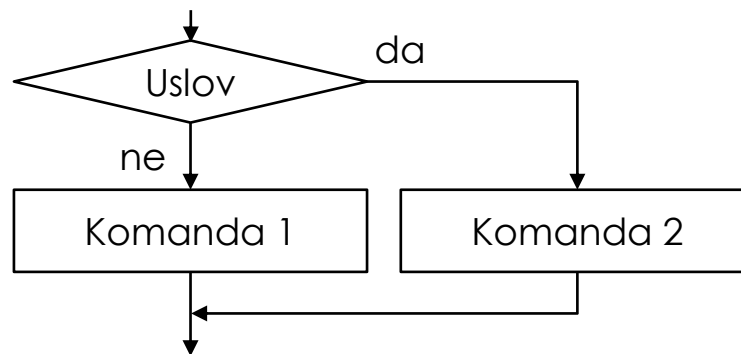
	Početak algoritma
	Ulazne veličine
	Izračunavanje
	Uslovno grananje
	Izlazne veličine
	Kraj algoritma

Alogitmi – osnovne strukture

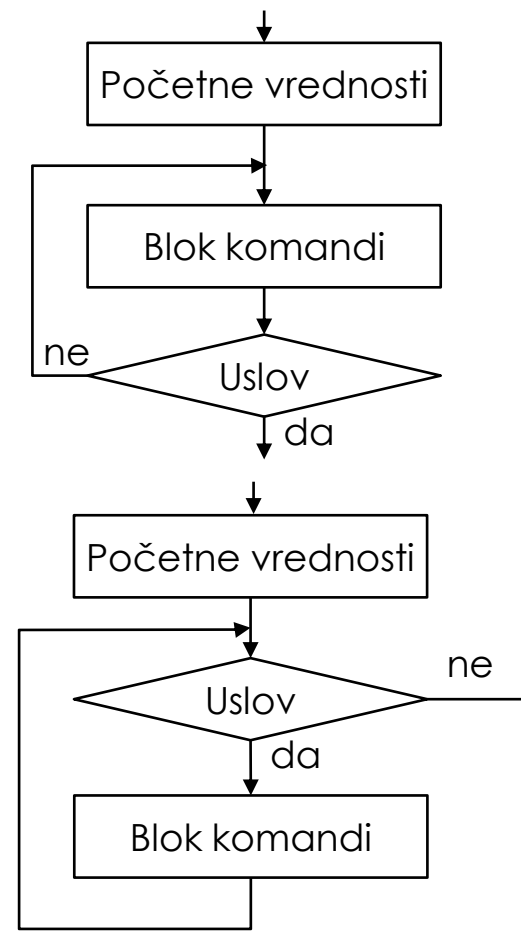
■ Linijska struktura



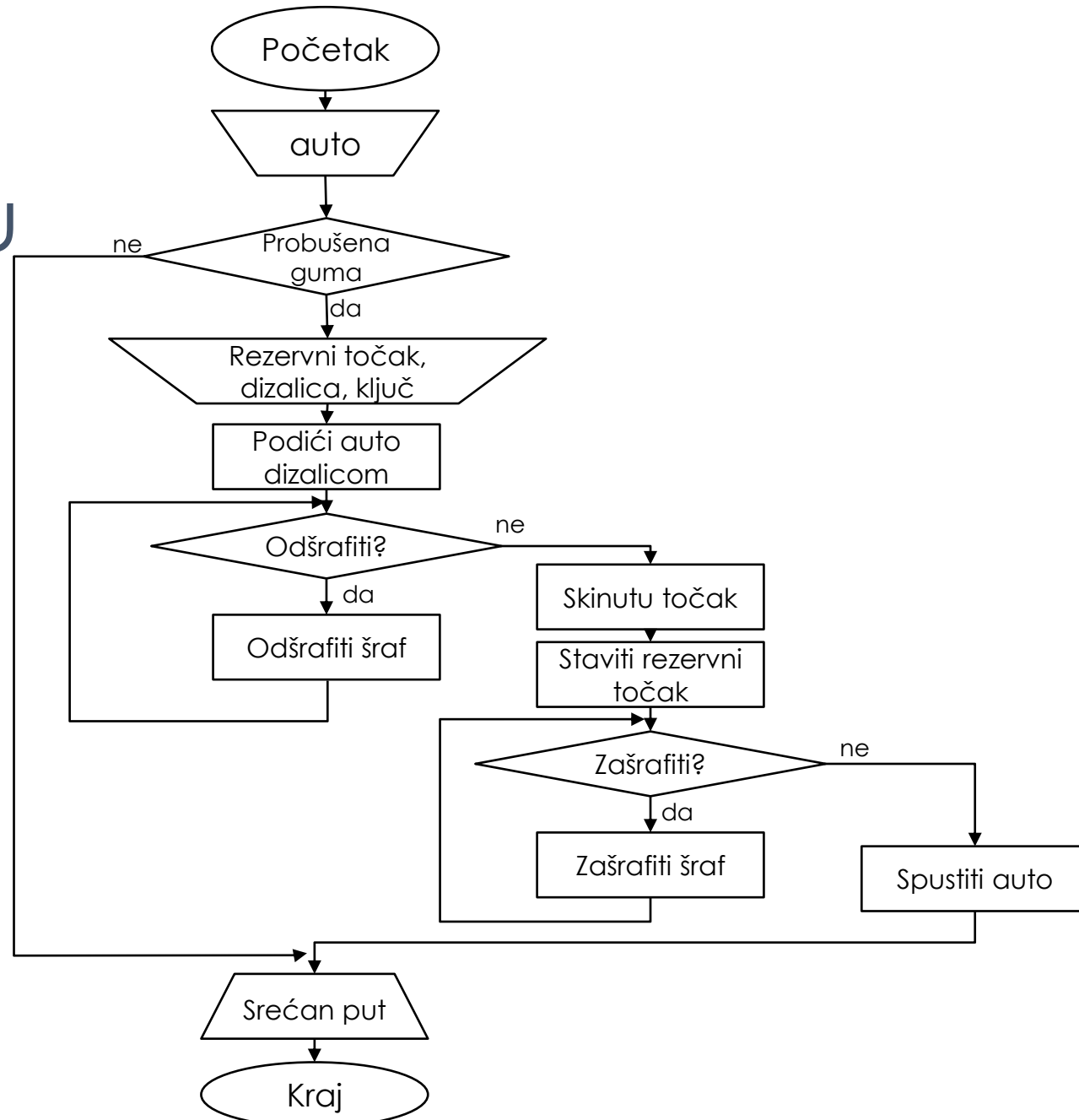
■ Razgranata struktura



■ Ciklička struktura



Algoritmi: probušena guma na autu



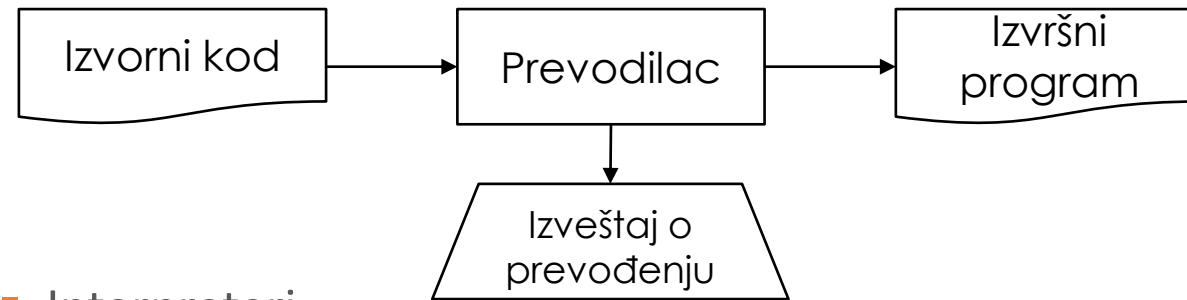
Vrste programskih jezika

- Algoritmi se u računaru realizuju odgovarajćim **programom**
- Program može biti realizovan korišćenjem različitih simbola (jezika), kojima se reprezentuju pojedine operacije podržane od strane računarskog sistema
- Prema stepenu bliskoski sa arhitekturom računara
 - Jezici niskog nivoa (mašinski jezik, Assembler)
 - Jezici visokog nivoa (FORTRAN, Pascal, C, C++, C#, Java...)

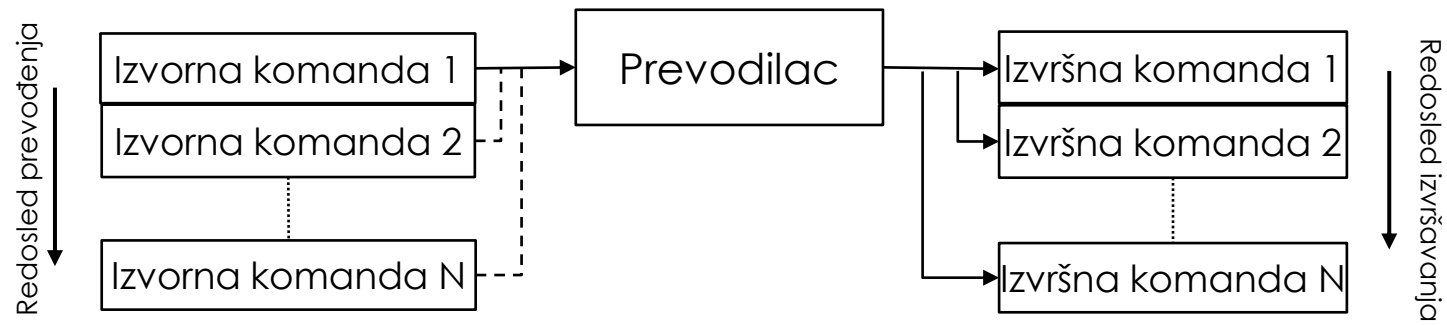
Vrste programskih jezika

- Prema načinu prevođenja

- Kompajleri



- Interpreteri



Vrste programskih jezika

- Viši programski jezici dele se prema načinu kako se program formira:
 - Imperativni (opisuju kako se nešto izračunava)
 - Deklarativni (opisuju šta treba izračunati) – Prolog i SQL
- Program prilikom izvršavanja prolazi kroz određena stanja koja su definisana promenljivim u memoriji. Programer u imperativnom jeziku određuje način na koji pojedine promenljive menjaju svoja stanja, dok se kod deklarativnih jezika ovim pitanjem bave prevodilac ili interpreter, na osnovu opisa željenog rezultata.
- Imperativni jezici se razlikuju prema organizaciji izvornog koda:
 - Proceduralni – Basic, Fortran, Pascal, C
 - Objektno orijentisani – C++, Java, C#
- Funkcionalni pristup programiranju je bliži deklarativnom, gradi se od funkcija - Haskell