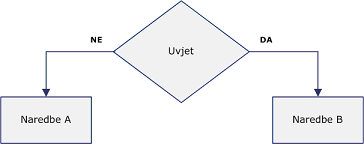
**Računarstvo II1 21 20.04 5 i 6h**

**Struktura grananja**

Grananje je programska struktura koja omogučuje različit tijek programa, ovisan o rezultatu postavljenog uvjeta. To je važna struktura bez koje bi mogućnost rješavanja zadatka računalom bila vrlo ograničena. Rezultat postavljenog uvjeta mora biti jedno od dvaju stanja: true/false, da/ne, istina/laž, 1/0. Vrijednost rezultata uvjeta bit će 1 ako je uvjet zadovoljen i 0 ako uvjet nije zadovoljen.

Dijelovi programa koji se uvjetno izvode, grupiraju se u skupine ili blokove naredbi. Svaki se **blok naredbi** u ovoj inačici pseudojezikom omeđuje parom otvorena i zatvorena vitičasta zagrada( {} ). Ako blok naredbi ima samo jednu naredbu, zagrade se mogu izostaviti. Blokovi naredbi se obično pišu uvučeno zbog preglednosti.

U dijagramu tijeka se za grananje koristi geometrijski lik romb:



Slika 1. Dijagram tijeka grananja (romb je lik za grananje)

Iz ove inačice pseudojezika za strukturu grananja su navedeni u tablici 11.

|  |  |
| --- | --- |
| Opis | Pseudojezik |
| Grananje | ako je uvjet onda  naredba1  inače  naredba2; |
| Blok naredbi | {  } |

Zadatak

Algoritam zadatka napišite pseudojezikom i nacrtajte dijagram tijeka.

Potrebno je unijeti prirodan broj pa odrediti je li paran ili neparan. Rezultat ispisati.

ulaz (x);

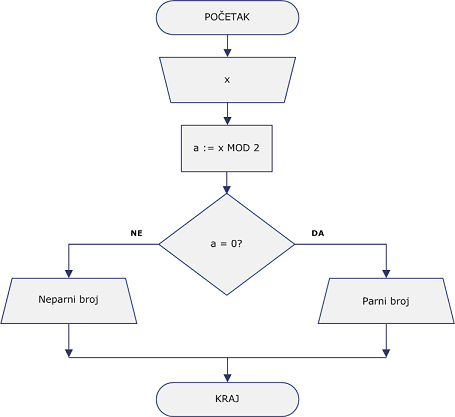
a = x MOD 2;

ako je a=0 onda

izlaz („Broje je paran“);

inače

ulaz („Broj je neparan“);



Slika 2. Dijagram tijeka s grananjem za zadatak određivanja parnosti broja

**Struktura petlje**

Često je u programu potrebno **ponoviti neku radnju vise puta**. Da se ne bi ista naredba pisala vise puta, postoji programska struktura petlje, koja omogućava ponavljanje jedne ili vise naredbi. Petlja može biti bezuvjetna, pri čemu se izvršava unaprijed zadan broj broj puta ili uvjetna pri čemu broj ponavljanja petlje ovisi o postavljenom uvjetu.

|  |  |
| --- | --- |
| **Opis** | **Pseudojezik** |
| Petlja s unaprijed poznatim brojem ponavljanja | za b := p do k činiti  naredba; |
| Petlja kod koje nije unaprijed poznat broj ponavljanja, a uvjet se provjerava na početku petlje | dok je uvjet činiti  naredba; |
| Petlja kod koje nije unaprijed poznat broj ponavljanja, a uvjet se provjerava na kraju petlje | ponavljati  naredba;  dok je uvjet; |

Zadatak

Algoritam zadatka napišite psudojezikom i nacrtajte dijagram tijeka:

Potrebno je unijeti 100 cijelih brojeva. Prije svakog unosa broja ispisati tekst “Unesi broj”.

za k := 1 do 100 činiti

{

izlaz(“Unesi broj”);

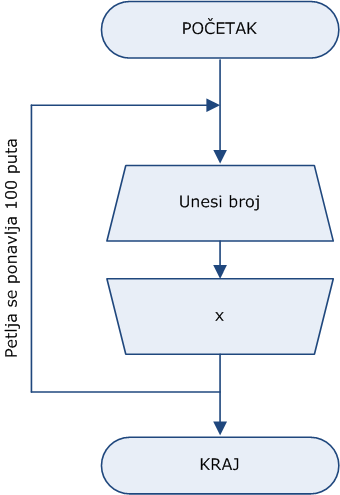
ulaz(x);

}

U ovom primjeru riječ o **bezuvjetnoj petlji** jer će se ona uvijek ponoviti 100 puta. Naredba:

za k := do 100 činiti

govori da petlja treba ponoviti 100 puta. Pri svakom prolazu petlje vrijednost varijable k se uvećava za jedan (od početne vrrijednosti do krajne vrijednosti 100).



Slika . Dijagram tijeka bezuvjetne petlje