

Programski jezici i strukture podataka

2

Programiranje

- Aktivnost koja počinje i završava se u glavi programera.
- Kao izvor programiranja označena je potreba za automatizacijom umnog rada.
- Rešavanje problema automatizacije umnog rada zahteva upoznavanje prirode umnog rada.

Programiranje kroz primer pravljenja pregleda ispitnih rezultata studenata

- Pregled ispitnih rezultata sadrži spiskove sa imenima i ocenama studenata.
 - Jedan uređen, odnosno **sortiran**, u alfabetskom redu imena studenata,
 - drugi sortiran u opadajućem redu njihovih ocena.
- Pored toga, pregled ispitnih rezultata sadrži:
 - brojeve studenata koji su dobili pojedine prolazne ocene,
 - ukupan broj studenata koji su dobili prolaznu ocenu,
 - ukupan broj studenata koji su dobili neprolaznu ocenu.
- Na kraju se u pregledu ispitnih rezultata javlja i srednja prolazna ocena.

Pravljenje pregleda ispitnih rezultata se svodi na **obradu podataka**

1. Preuzimanje sa ocenjenih studentskih radova imena studenata i njihovih ocena.
2. Sortiranje spiskova u alfabetskom redu imena i u opadajućem redu ocena.
3. Određivanje broja studenata koji su dobili pojedine od prolaznih ocena.
4. Ustanovi se ukupan broj studenata koji su dobili prolaznu ocenu, kao i ukupan broj studenata koji su dobili neprolaznu ocenu.
5. Na kraju se izračuna srednja prolazna ocena.

Računar

- Obrada ovih podataka se sastoji od obavljanja **operacija** kao što su, na primer, aritmetičke operacije.
- Mašina, sposobna da obavlja obrade podataka na koje se svodi umna aktivnost, omogućuje automatizaciju umnog rada se naziva **računar**.

Redosled obavljanja operacija

- **Opis obrade podataka** u kome su precizirane i operacije od kojih se obrada sastoji i redosled njihovog obavljanja se naziva **algoritam**.
- Obavljanje obrade podataka se svodi na **izvršavanje** njenog algoritma.
- **Izvršilac** algoritma može da bude i čovek ili računar.

Formiranje nesortiranog spiska

Milan 9

Predrag 5

Miodrag 8

Milica 6

Dragan 8

Milena 8

Dragojlo 7

Dragana 9

Milinka 8

Dragica 5

Milenka 7

Milisav 6

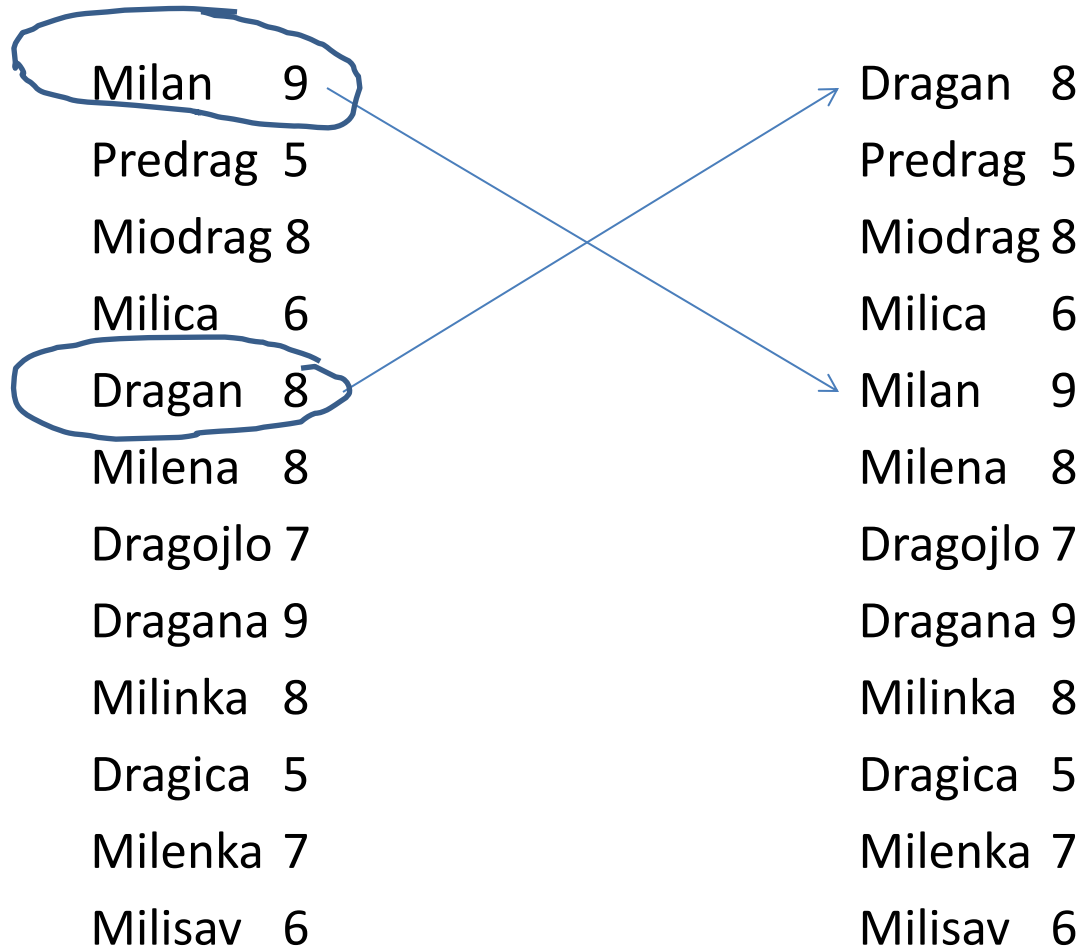
Pravilo sortiranja

1. Spisak se sortira u alfabetskom redu imena studenata.
2. Spisak se sortira u opadajućem redu ocena.

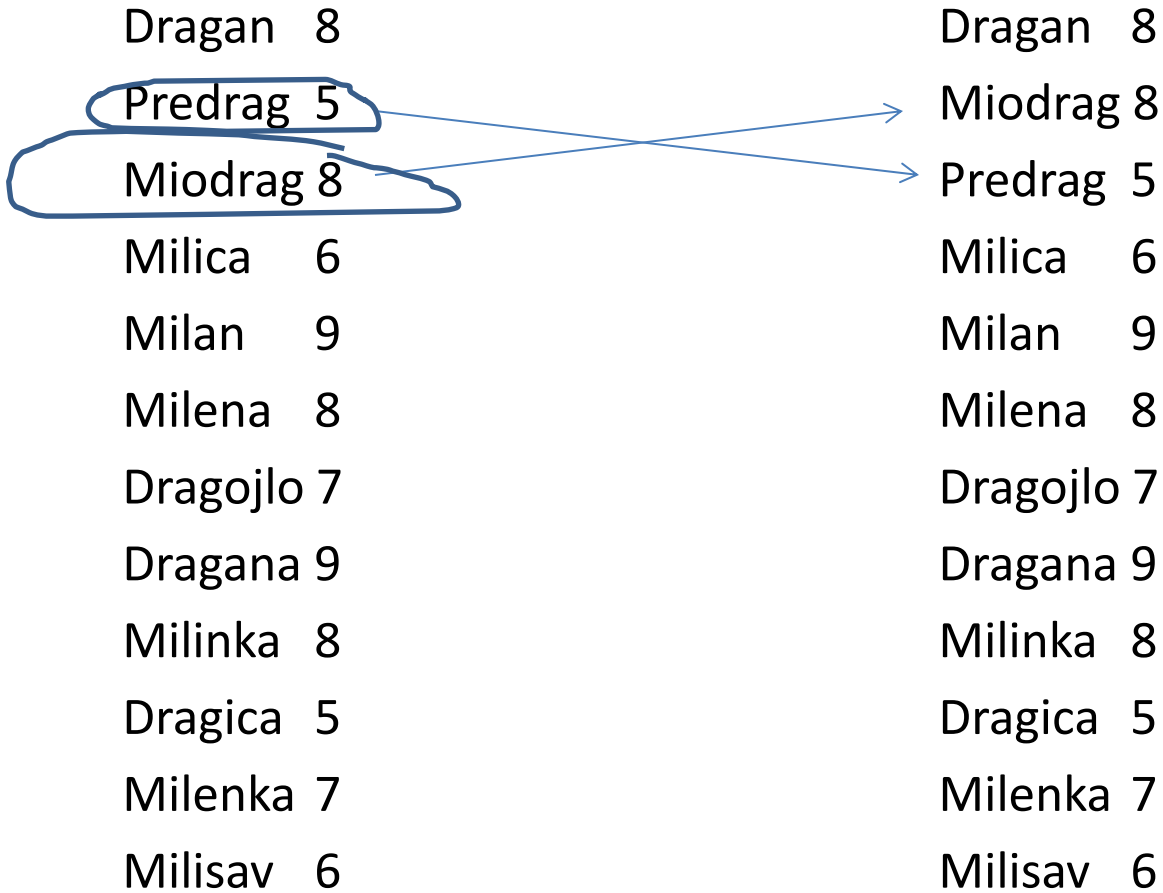
Sortiranje spiska u alfabetskom redu imena studenata

- Da li prvo ime u nesortiranom spisku po pravilu sortiranja prethodi svim preostalim imenima.
 - poređenje prvog imena sa svim preostalim, idući po redu od drugog do poslednjeg imena.
- Ako poređenje pokaže da prvo ime ne prethodi nekom od preostalih imena, tada linije sa ova dva imena menjaju mesta.

Zamena mesta

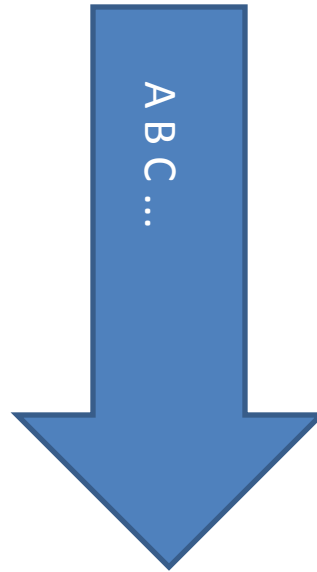


...



Spisak sortiran u alfabetskom redu imena

Dragan 8
Dragana 9
Dragica 5
Dragojlo 7
Milan 9
Milena 8
Milenka 7
Milica 6
Milinka 8
Milisav 6
Miodrag 8
Predrag 5



Spisak sortiran u opadajućem redu ocena

| | |
|----------|---|
| Dragana | 9 |
| Milan | 9 |
| Dragan | 8 |
| Milena | 8 |
| Milinka | 8 |
| Miodrag | 8 |
| Dragojlo | 7 |
| Milenka | 7 |
| Milisav | 6 |
| Milica | 6 |
| Dragica | 5 |
| Predrag | 5 |

Utvrdjivanje broja studenata
koji su dobili pojedine prolazne
ocene:

$$2+4+2+2=10$$



10
9
8
8
...

Određivanje srednje prolazne ocene

Ukupna suma prolaznih ocena se odredi kada se saberu proizvodi pojedinih prolaznih ocena i broja studenata koji su ove ocene dobili:

$$10*0+9*2+8*4+7*2+6*2=76$$

Delenjem ove sume brojem studenata koji su dobili prolaznu ocenu:

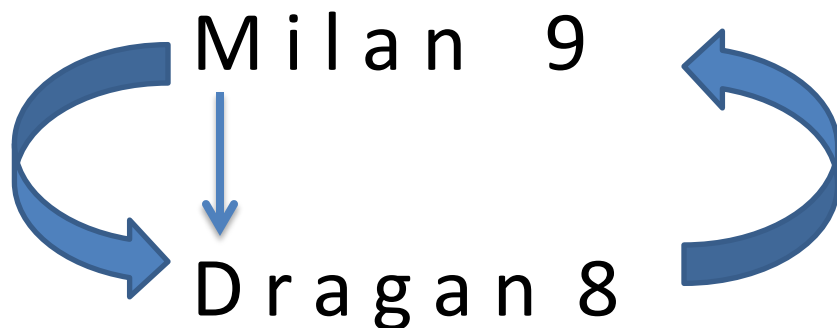
$$76/10=7.6$$

nastaje **srednja prolazna ocena.**

Poređenje imena

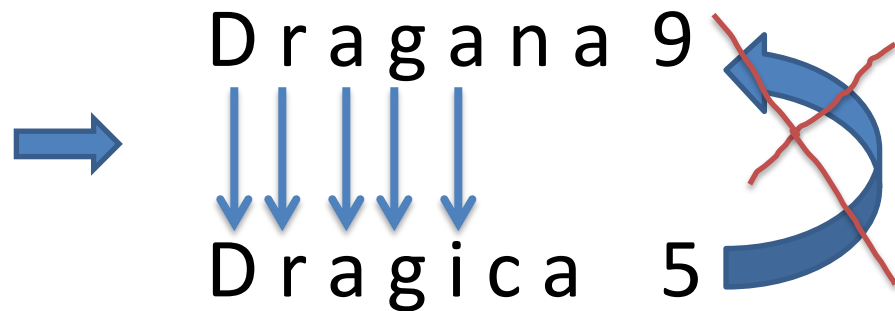
- Nije dovoljno precizan da bi poslužio kao algoritam pomenute aktivnosti.
- U opisu se poređenje imena pominje bez objašnjenja.

Korespondentni znakovi imena se porede znak po znak



Poređenje se završava čim se ustanovi da su dva znaka različita.

U toku poređenja imena korespondentni znakovi imena se porede znak po znak.



Poređenje imena

- Ako su svi znakvi imena jednaki, tada su i imena jednaka.
- U slučaju kada su imena različite dužine, ali su korespondentni znakovi kraćeg imena i početka dužeg imena jednaki, kao u primeru imena **Dragan i Dragana**, podrazumeva se da kraće ime prethodi dužem.

Programski jezik, program, programiranje

- Usvojeni pristup ne vodi ka preciznom opisu, odnosno algoritmu, jer se zasniva na korištenju govornog jezika.
- U govornom jeziku to je teško postići zbog višeznačnosti reči, nepreciznosti i sl.
- Za potrebe preciznog i konciznog izražavanja algoritama razvije poseban jezik.
- Taj jezik se naziva **programski jezik**.
- Programski jezik omogućuje opisivanje tačnog redosleda, odnosno programa, obavljanja operacija neke obrade podataka.
- Algoritam izražen u programskom jeziku se naziva **program**.
- Pravljenje programa se naziva **programiranje**.
- Program izvršava izvršilac programa.

Osobine prethodno izloženog opisa

- vrednost
- lokacija
- identifikator
- promenljiva
- redosled obavljanja operacija
- ponavljanje operacija
- uslovljenost obavljanja operacija
- grupisanje operacija

Osobine programskog jezika

Iz prethodnog razmatranja osobina programskog jezika sledi da je neophodno da programski jezik omogući označavanje:

1. podataka (konstanti)
2. promenljivih
3. operacija
4. (sekvencijalnog) redosleda obavljanja operacija
5. grupisanja operacija
6. ponavljanja (repeticije) obavljanja operacija i
7. uslovljenosti (alternacije) obavljanja operacija.

U jeziku C

- Program se sastoji od jedne ili više funkcija.
- Funkcije su gradivne jedinice programa.
- Funkcije mogu pozivati jedna drugu.
- Sve funkcije su ravnopravne.
- Od funkcije `main()` počinje izvršavanje programa, ona "upravlja programom", iz nje se mogu pozivati ostale funkcije.
- Program mora imati bar jednu funkciju: `main()`
- "Zdravo svete"

Funkcije u jeziku C

- Funkcija služi određenoj svrsi.
- Funkcije se sastoje od naredbi.
 - Naredbe određuju sekvencijalno izvršavanje programa.
 - Naredbe se grupišu u blokove.
 - U jeziku C blok se označava srednjim zagradama { }.

Program u jeziku C

- Program u jeziku C se sastoji od proizvoljno mnogo izvornih datoteka i datoteka zaglavlja.
- Manje zahtevni programi (u smislu količine koda i broja funkcija) pišu se u jednoj izvornoj datoteci.
- **Izvorna datoteka** se uobičajeno sastoji iz:
 - pretprocesorskih direktiva,
 - globalnih deklaracija i
 - definicija funkcija.

Izvorna datoteka

- Izvorna datoteka se uobičajeno označava imenom sa sufiksom `.c`. (**z1.c**)
- Ako se program sastoji od više izvornih datoteka uobičajeno je da se deklaracije funkcija, globalne promenljive, makroi i konstante izdvoje u posebnu datoteku "zaglavlja", koja se pretprocesorskom direktivom `#include` poziva u svakoj izvornoj datoteci.
- Datoteke zaglavlja se označavaju imenom i sufiksom `.h`.

Izvorni kod

- Izvorna datoteka se sastoji od izvornog koda.
- Izvorni kod započinje pretprocesorskim direktivama.
- U pojedinačnom redu se može nalaziti samo jedna predprocesorska direktiva, red započinje znakom #.
- Između dva susedna tokena može biti proizvoljan broj razmaka i tabulatora.
- Uobičajeno je:
 - novu deklaraciju i naredbu započeti u novom redu,
 - uvući ugneždenu strukturu ili blok naredbi.