

Fakultet tehničkih nauka, DRA, Novi Sad

Predmet:  
Organizacija podataka

Dr Slavica Kordić,  
Milan Čeliković,  
Vladimir Dimitrieski,  
Branko Terzić,  
Aleksa Mirković,  
Ivana Zeljković

# Comma-Separated Values CSV

# CSV

- Comma-separated values
  - Datoteke zarezom odvojenih vrednosti
- Character-separated values
  - Datoteke znakom odvojenih vrednosti
- Tabelarni podaci predstavljani u obliku običnog teksta
  - Čist tekst bez formatiranja
  - Datoteke sa ekstenzijom \*.txt ili \*.csv

# CSV

- Prva upotreba
  - 1967. u IBM kompajleru Fortran programskog jezika na operativnom sistemu OS/360
- Kasnija upotreba
  - Baze podataka, pregledači teksta
- Danas
  - Za razmenu podataka između baza podataka različitih arhitektura

# CSV

- Prednosti
  - Čitljivost podataka putem tekstualnog editora

# CSV

- Ne postoji standard za definisanje CSV datoteke, ali postoje pravila koja treba poštovati
  - CSV datoteka sadrži tabelarni zapis podataka
  - Svaki red u datoteci, definisan prelomom reda, predstavlja red u tabeli
  - Poslednji zapis na kraju može, ali ne mora, sadržati prelom reda
  - Prva linija može biti, ali ne mora, zaglavlje koje definiše naziv pojedine kolone

# CSV

- U svakom redu datoteke se može nalaziti jedno ili više polja odvojeno znakom za odvajanje
  - svaki red treba da sadrži jednak broj polja
  - poslednje polje u redu ne sme biti popraćeno znakom za razdvajanje
- Razmaci su deo zapisa u polju i ne smeju se ignorisati
- Podaci u poljima reda mogu biti u dvostrukim navodnicima
- Polja koja sadrže podatke koji u sebi imaju znak za odvajanje, dvostruke navodnike i prelom reda moraju biti u dvostrukim navodnicima.
  - ako se dvostruki navodnici nalaze u podatku zapisanom u polju oni se moraju propratiti sa još jednim dvostrukim navodnicima

# CSV

- Neki primeri odstupanja od pravila:
  - Razni znakovi se koriste za odvajanje npr. zarez (,), tačka zarez (;), tab ( ), razmak ( ) i sl
  - Zapis praznog reda koji se može interpretirati kao zapis praznih i 0 vrednosti u svim poljima ili kao zapis u samo jednom polju
  - Korišćenje jednostrukih navodnika oko zapisa



# Primer 1.

- Napisati *Java* program koji čita sadržaj datoteke *drzave\_gradovi.csv* i ispisuje ga na ekran. Zadatak uraditi koristeći:
  - Klasu *java.util.Scanner*
  - Klasu *java.io.BufferedReader*
  - Klasu *CSVReader*

# Primer 1. - *java.util.Scanner*

```
Scanner scanner = new Scanner(new File("src\\drzave_gradovi.csv"));

scanner.useDelimiter(",");

while (scanner.hasNext()) {
    System.out.print(scanner.next() + " ");
}

scanner.close();
```

# Primer 1. - *java.io.BufferedReader*

```
String csvFilename = "src\\drzave_gradovi.csv";
BufferedReader fileReader = null;

final String DELIMITER = ",";
try
{
    String line = "";
    fileReader = new BufferedReader(new FileReader(csvFilename));
    while ((line = fileReader.readLine()) != null) {
        String[] tokens = line.split(DELIMITER);
        for(String token : tokens)
        {
            System.out.print(token + " | ");
        }
        System.out.println("");
    }
} catch (Exception e) {
    e.printStackTrace();
}
finally{
    try {
        fileReader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

# Primer 1. - *CSVReader*

```
String csvFilename = "src\\drzave_gradovi.csv";
CSVReader csvReader = new CSVReader(new FileReader(csvFilename));
String[] row = null;

while((row = csvReader.readNext()) != null) {
    String rowData = "";
    for(int i = 0; i < row.length; i++){
        rowData += row[i];
        if(i != row.length - 1)
            rowData += ",";
    }

    System.out.println(rowData);
}
csvReader.close();
```

# Primer 1. - *CSVReader*

```
String csvFilename = "src\\drzave_gradovi.csv";
CSVReader csvReader = new CSVReader(new FileReader(csvFilename));
String[] row = null;

List<String[]> content = csvReader.readAll();

for (Object object : content) {
    row = (String[]) object;
    String rowData = "";
        for(int i = 0; i<row.length; i++){
            rowData+=row[i];
            if(i!=row.length-1)
                rowData+=",";
        }
    System.out.println(rowData);
}
csvReader.close();
```

# Primeri zadavanja delimiter znakova

- Delimiter znak tačka zarez (;)
  - `CSVReader reader = new CSVReader(new FileReader(file), ';');`
- Delimiter znak #
  - `CSVReader reader = new CSVReader(new FileReader(file), '#');`
- Delimiter znak zarez (,), a sadržaj polja unutar znaka ('')
  - `CSVReader reader = new CSVReader(new FileReader(file), ',', '\"');`
- Delimiter znak zarez (,), sadržaj polja unutar znaka (') i prvih 5 redova u datoteci se izostavljaju prilikom čitanja
  - `CSVReader reader = new CSVReader(new FileReader(file), ',', '\"', 5);`

## Primer 2.

- Napisati *Java* program koji formira izlaznu datoteku *drzave\_gradovi2.csv* sa sledećim sadržajem:

Australia,Canberra

Canada,Ottawa

China,Beijing

France,Paris

# Primer 2.

```
String csv = "src\\drzave_gradovi2.csv";  
CSVWriter writer = new CSVWriter(new FileWriter(csv));  
  
String [] country = "Australia,Canberra".split(",");  
writer.writeNext(country);  
  
country = "Canada,Ottawa".split(",");  
writer.writeNext(country);  
  
country = "China,Beijing".split(",");  
writer.writeNext(country);  
  
country = "France,Paris".split(",");  
writer.writeNext(country);  
  
writer.close();
```



# Primer 2.

```
String csv = "src\\drzave_gradovi2.csv";
CSVWriter writer = new CSVWriter(new FileWriter(csv));

List<String[]> data = new ArrayList<String[]>();
data.add(new String[] { "Australia", "Canberra" });
data.add(new String[] { "Canada", "Ottawa" });
data.add(new String[] { "China", "Beijing" });
data.add(new String[] { "France", "Paris" });

writer.writeAll(data);

writer.close();
```

# Primer 3.

- Pročitati sadržaj datoteke *drzave\_gradovi.csv* u odgovarajuću listu *bean-ova* i ispisati nazive glavnih gradova.

```
String csvFilename = "src\\drzave_gradovi.csv";
CSVReader csvReader = new CSVReader(new FileReader(csvFilename));

ColumnPositionMappingStrategy strat = new ColumnPositionMappingStrategy();
strat.setType(Country.class);

//column names from the bean (case insensitive)
String[] columns = new String[] {"countryName", "capitalName", "capitalLatitude",
    "capitalLongitude", "countryCode", "continentName"};

strat.setColumnMapping(columns);

CsvToBean csv = new CsvToBean();

List list = csv.parse(strat, csvReader);
for (Object object : list) {
    Country country = (Country) object;
    System.out.println(country.getCapitalName());
}
csvReader.close();
```

# Zadatak 1.

- Data je ulazna csv datoteka, u kojoj svaki red datoteke sadrži x, y, z koordinate u prostoru za jednu tačku. Podaci su delimitirani znakom zarez.
  - Formirati izlaznu csv datoteku koja u svakom redu sadrži x, y, z koordinate u prostoru za jednu tačku, kao i rastojanje te tačke od koordinatnog početka. Podaci su delimitirani znakom #.
  - Formirati izlaznu csv datoteku koja u svakom redu sadrži x, y, z koordinate u prostoru za jednu tačku, kao i rastojanje te tačke od koordinatnog početka. Tačke su sortirane u opadajućem redosledu prema rastojanju tačaka od koordinatnog početka. Podaci su delimitirani znakom #.

# Zadatak 1.

- input.csv

x,y,z

1,2,3

3,4,5

5,6,7

....

output.csv

x#y#z#d

1#2#3#3.74

3#4#5#7.07

5#6#7#10.49

....

# Zadatak 2.

- Data je ulazna csv datoteka, u kojoj svaki red datoteke sadrži broj indeksa, ime i prezime studenta, naziv predmeta koji je student polagao i ocenu koju student ima iz predmeta.
  - Za svaku ocenu (5,6,7,8,9,10), formirati izlazne datoteke koje sadrže u svakom redu broj indeksa, ime i prezime studenta, i broj predmeta na kojima je student ostvario tu ocenu.
  - Formirati izlaznu datoteku koja sadrži u svakom redu broj indeksa, ime i prezime studenta, broj predmeta koje je student položio, kao i prosečnu ocenu na položenim predmetima.

# Zadatak 3.

- Data je ulazna csv datoteka, u kojoj svaki red datoteke sadrži broj indeksa, ime i prezime studenta, naziv predmeta koji je student polagao i ocenu koju student ima iz predmeta. Na početku datoteke nalazi se zaglavlje koje sadrži sledeće podatke:  
Datum: Fakultet: Stud. Program:  
Skolska godina: Semestar:
- Napisati *Java* program koji čita podatke iz datoteke i ispisuje kompletan sadržaj na ekran.

# Zadatak 4.

- Napisati *Java* program putem kojeg će korisnik moći da opiše format csv datoteke: zaglavlje, broj kolona i nazivi kolona. Nakon toga omogućiti korisniku popunjavanje datoteke konkretnim podacima.