

# Programiranje 1 — deseta domača naloga

Rok za oddajo: torek, 16. januarja 2024

## Urejanje preglednice

### Naloga

Za potrebe te naloge naj bo *preglednica* skupek podatkov, urejenih v  $m$  vrstic in  $n$  stolpcev, pri čemer so vsi podatki znotraj istega stolpca istega tipa, podatki v različnih stolpcih pa so lahko različnih tipov. Takšne preglednice lahko tvorijo podatkovno bazo.<sup>1</sup>

V vsakem stolpcu bodo podatki pripadali enemu od treh tipov:

- Tip 1: celo število z intervala  $[-10^6, 10^6]$ .
- Tip 2: niz, sestavljen iz največ 20 znakov (ti so lahko le velike in male črke angleške abecede ter podčrtaji).
- Tip 3: element množice  $\{\text{da}, \text{ne}, \text{morda}\}$ .

Primer preglednice je prikazan na sliki 1.

Stolpec 1 (tip 2)	Stolpec 2 (tip 2)	Stolpec 3 (tip 1)	Stolpec 4 (tip 2)	Stolpec 5 (tip 3)
Ana	Vidmar	3	Kamnik	ne
Marija	Krajnc	3	Koper	da
Janez	Krajnc	5	Novo_mesto	da
Marija	Kralj	4	Ptuj	morda
Marija	Bogataj	2	Velenje	ne
Marija	Krajnc	5	Kranj	da
Peter	Kralj	4	Maribor	morda
Ana	Vidmar	5	Trbovlje	ne
Marija	Kralj	4	Ljubljana	morda
Peter	Krajnc	3	Celje	da
Marija	Bogataj	4	Jesenice	morda

Slika 1: Primer preglednice.

Napišite program, ki preglednico uredi po podanem zaporedju kriterijev. Vsak kriterij je podan kot  $s$  ali  $-s$ , pri čemer je  $s \in \{1, 2, \dots, n\}$  zaporedna številka stolpca. Če je kriterij podan kot  $s$ , je treba stolpec  $s$  urediti naraščajoče, kriterij  $-s$  pa zahteva padajoče urejanje. Vrednosti tipa 1 urejamo tako, kot urejamo števila, vrednosti tipa 2 urejamo glede na naravno urejenost tipa `String` (leksikografsko po abecedi ASCII, če urejamo naraščajoče, in obratno leksikografsko, če urejamo padajoče), vrednosti tipa 3 pa uredimo kot  $(\text{da}, \text{ne}, \text{morda})$ , če jih urejamo naraščajoče, oziroma kot  $(\text{morda}, \text{ne}, \text{da})$ , če jih urejamo padajoče.

Program naj preglednico primarno uredi po prvem kriteriju. Vrstice, ki se po prvem kriteriju med seboj ne razlikujejo, naj med seboj uredi po drugem kriteriju. Vrstice, ki se

<sup>1</sup>Pri podatkovnih bazah se takim preglednicam reče enostavno *tabele*. Ker pa pri našem predmetu izraz *tabela* označuje zaporedje elementov istega tipa, smo v izogib zmedi uporabili drugačen izraz.

ne razlikujejo niti po prvem niti po drugem kriteriju, naj uredi po tretjem itd. Če se vrstici med seboj ne razlikujeta po nobenem kriteriju, ju program med seboj ne sme zamenjati.

Na primer, pri zaporedju kriterijev (5, -3, 2, 1) bi preglednico na sliki 1 uredili tako, da bi dobili preglednico na sliki 2. Preglednica je primarno urejena po petem stolpcu (naraščajoče), sekundarno po tretjem (padajoče), terciarno po drugem (naraščajoče), kvartarno pa po prvem stolpcu (naraščajoče). Vrstici (Marija, Kralj, 4, Ptuj, morda) in (Marija, Kralj, 4, Ljubljana, morda) se med seboj ne razlikujeta po nobenem od podanih kriterijev, zato se njun medsebojni vrstni red ni spremenil.

Stolpec 1 (tip 2)	Stolpec 2 (tip 2)	Stolpec 3 (tip 1)	Stolpec 4 (tip 2)	Stolpec 5 (tip 3)
Janez	Krajnc	5	Novo_mesto	da
Marija	Krajnc	5	Kranj	da
Marija	Krajnc	3	Koper	da
Peter	Krajnc	3	Celje	da
Ana	Vidmar	5	Trbovlje	ne
Ana	Vidmar	3	Kamnik	ne
Marija	Bogataj	2	Velenje	ne
Marija	Bogataj	4	Jesenice	morda
Marija	Kralj	4	Ptuj	morda
Marija	Kralj	4	Ljubljana	morda
Peter	Kralj	4	Maribor	morda

Slika 2: Preglednica s slike 1, urejena po zaporedju kriterijev (5, -3, 2, 1).

## Vhod

Vsa števila na vhodu so cela. V vseh vrsticah so podatki med seboj ločeni s po enim presledkom.

Prva vrstica vsebuje števili  $m \in [1, 500]$  in  $n \in [1, 20]$ . Druga vrstica vsebuje  $n$  števil iz množice  $\{1, 2, 3\}$ , ki podajajo tipe posameznih stolpcev ( $i$ -to število podaja tip  $i$ -tega stolpca). Tretja vrstica vsebuje število kriterijev v zaporedju ( $k \in [1, n]$ ), četrta pa števila  $s_1, s_2, \dots, s_k$ , pri čemer za vsak  $i \in \{1, \dots, k\}$  velja  $s_i \in [1, n]$  ali  $s_i \in [-n, -1]$ . Poleg tega za vse pare  $(i, j)$  velja  $i \neq j \implies |s_i| \neq |s_j|$ .

V skritih testnih primerih 1–25 velja  $k = 1$  (en sam kriterij), v primerih 1–10 pa poleg tega še  $n = 1$  (en sam stolpec).

## Izhod

Po vrsticah izpišite urejeno preglednico. V vsaki vrstici naj bodo podatki med seboj ločeni z znakom |. Ne izpisujte presledkov ali praznih vrstic.

## Javni testni primer

Vhod:

```
11 5
2 2 1 2 3
4
5 -3 2 1
Ana Vidmar 3 Kamnik ne
```

```
Marija Krajnc 3 Koper da
Janez Krajnc 5 Novo_mesto da
Marija Kralj 4 Ptuj morda
Marija Bogataj 2 Velenje ne
Marija Krajnc 5 Kranj da
Peter Kralj 4 Maribor morda
Ana Vidmar 5 Trbovlje ne
Marija Kralj 4 Ljubljana morda
Peter Krajnc 3 Celje da
Marija Bogataj 4 Jesenice morda
```

Izhod:

```
Janez|Krajnc|5|Novo_mesto|da
Marija|Krajnc|5|Kranj|da
Marija|Krajnc|3|Koper|da
Peter|Krajnc|3|Celje|da
Ana|Vidmar|5|Trbovlje|ne
Ana|Vidmar|3|Kamnik|ne
Marija|Bogataj|2|Velenje|ne
Marija|Bogataj|4|Jesenice|morda
Marija|Kralj|4|Ptuj|morda
Marija|Kralj|4|Ljubljana|morda
Peter|Kralj|4|Maribor|morda
```

## Opombe

Pri tej nalogi lahko uporabite katerikoli »razumen« algoritem urejanja, le da je *stabilen*, kar pomeni, da med seboj ne menjuje enakovrednih vrstic. Lahko uporabite navadno vstavljanje, katero od javinih metod (`Arrays.sort`, `Collections.sort` ali `seznam.sort`, kjer je *seznam* objekt tipa `List`) ali pa kaj svojega. Poudarek pri tej nalogi namreč ni na samem urejanju, pač pa na izdelavi primerjalnikov. Poskusite podatke urediti samo enkrat. Če boste delali tako, si boste namreč morali že vnaprej pripraviti celovit primerjalnik.

## Oddaja naloge

Program oddajte v obliki ene same datoteke z nazivom `DN10_vvvvvvvv.java`, pri čemer niz `vvvvvvvv` zamenjajte s svojo vpisno številko. Morebitne lastne razrede definirajte kot statične notranje razrede v okviru razreda `DN10_vvvvvvvv` (tako, kot ste to storili pri osmi domači nalogi).