

# Vaje pri predmetu Programiranje 1

## Teden 6: Tabele 2

### Volitve

#### Naloga

V Butalah potekajo volitve prav vsako leto. Odkar pomnijo, se za naklonjenost volilcev potegujejo iste stranke, pa tudi volilci vsakokrat glasujejo na istih voliščih. Butalci so programerjem prijazen narod: butalsko štetje se je pričelo z letom 0, stranke in volišča pa so označeni z indeksi od 0 naprej.

Profesor Butnik, eden najuglednejših Butalcev, je izide volitev za vso butalsko zgodovino zapisal v tabelo tipa `int[][][]` in velikosti  $L \times S \times V$ , kjer je  $L$  število let,  $S$  število strank,  $V$  pa število volišč (velja  $L > 0$ ,  $S > 0$  in  $V > 0$ ). Element  $(l, s, v)$  v tej tabeli vsebuje število glasov, ki jih je v letu  $l$  stranka  $s$  prejela na volišču  $v$ .

Razred `Volitve` dopolnite s sledečimi metodami (Butnikovi podatki so zbrani v tabeli `t`):

- `public static int steviloGlasov(int[][][] t, int leto, int stranka)`

Vrne skupno število glasov (preko vseh volišč), ki jih je v podanem letu prejela podana stranka.

- `public static int[][] glasovi(int[][][] t)`

Za vsako leto in za vsako stranko izračuna število glasov, ki jih je stranka prejela v tistem letu, in rezultate vrne v obliki tabele velikosti  $L \times S$ .

- `public static int najVolisce(int[][][] t, int stranka)`

Vrne indeks volišča, na katerem je podana stranka v skupnem seštevku preko vseh let prejela največ glasov. Če je takih volišč več, naj metoda vrne prvo od njih.

- `public static int vsotaUvrstitev(int[][][] t, int stranka, int volisce)`

Za podano stranko vrne vsoto uvrstitev, ki jih je v posameznih letih dosegla na podanem volišču. Na primer, če je  $L = 3$  in je stranka na podanem volišču v letu 0 dosegla tretje mesto, v letu 1 četrto, v letu 2 pa prvo mesto, je rezultat enak  $3 + 4 + 1 = 8$ . Uvrstitev stranke v podanem letu na podanem volišču je enaka  $(1 + n_{>})$ , kjer je  $n_{>}$  število strank, ki so v tistem letu na tistem volišču prejele strogo več glasov.

#### Testiranje

Doslej ste svoj program testirali z množico vhodov in pripadajočih izhodov, tokrat pa ga boste z množico testnih razredov in pripadajočih izhodov. Vsak testni razred izdela tabelo izidov volitev, vsaj po enkrat pokliče eno od metod iz razreda `Volitve` in izpiše rezultate klicev.

Program `tj.exe` tokrat poženemo takole:

```
tj.exe <imenik_z_razredom_Volitve> <imenik_s_testi> <imenik_z_rezultati>
```

Drugi in tretji argument sta enaka kot doslej, prvi argument pa je *imenik* (mapa), ki vsebuje datoteko `Volitve.java`, ne datoteka `Volitve.java` sama. V našem primeru se bodo tako testni razredi kot razred `Volitve` nahajali v isti mapi. Če nas ne moti, da program `tj.exe` tudi datoteke z rezultati shrani v isto mapo, ga lahko znotraj te mape poženemo takole ...

```
tj.exe . . .
```

... oziroma kar takole:

```
tj.exe
```