

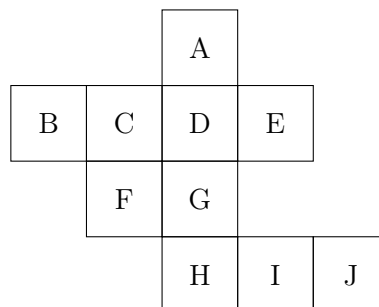
# Programiranje 1 — sedma domača naloga

Rok za oddajo: torek, 19. decembra 2023

## Stanovanjski blok

### Naloga

V soseski Megleni dvori so stanovanja v posameznih blokih razporejena v ploski pokončni pravokotni mreži, sami bloki pa nimajo nujno oblike kvadra. Blok lahko izgleda, denimo, tako, kot prikazuje slika 1.



Slika 1: Primer razporeditve stanovanj v meglenodvorskem bloku.

Vsako stanovanje ima tako najmanj eno in največ štiri sosednja stanovanja (levo, zgoraj, desno in spodnje). V primeru na sliki 1 ima stanovanje A samo spodnjega soseda (stanovanje D), stanovanje I levega in desnega (H in J), stanovanje D pa ima vse štiri sosede.

Za vsako stanovanje poznamo podatke o osebah, ki živijo v njem, in njegova sosednja stanovanja. Razporeditev stanovanj v bloku je povsem natančno določena že s podatki o enem samem (kateremkoli!) stanovanju v bloku. Na podlagi podatkov o sosednjih stanovanjih lahko namreč v celoti rekonstruiramo celotno zgradbo bloka. Recimo, da za primer na sliki 1 vemo le to, da blok vsebuje stanovanje G. Potemtakem morajo v bloku biti tudi sosedje stanovanja G (stanovanja D, F in H), sosedje le-teh (A, C, E, I) itd.

V skladu s sledečimi navodili napišite razrede `Oseba`, `Stanovanje` in `Blok`. V razrede lahko dodate poljubne lastne attribute, konstruktorje in metode.

### Razred `Oseba`

Objekt razreda `Oseba` predstavlja stanovalca nekega stanovanja. Za vsako osebo želimo hraniti njen »IP« (ime in priimek), spol in starost. Razred naj vsebuje sledeče javno dostopne konstruktorje in metode:<sup>1</sup>

- `public Oseba(String ip, char spol, int starost) [1–50]`

Inicializira objekt, ki predstavlja osebo s podanim IP-jem, spolom (M: moški, Z: ženski) in starostjo.

- `public String toString() [1–5]`

<sup>1</sup>V oglatih oklepajih so navedene številke skritih testnih primerov, v katerih se lahko pripadajoči element pokliče.

Vrne niz sledeče oblike:

*ip, □ spol, □ starost*

Primer:

Jože Gorišek, M, 56

- `public boolean jeStarejsaOd(Oseba os)` [6–10]

Vrne `true` natanko v primeru, če je oseba `this` starejša od osebe `os`. Parameter `os` ni v nobenem testnem primeru enak `null`.

## Razred Stanovanje

Objekt razreda `Stanovanje` predstavlja posamezno stanovanje v bloku. Za vsako stanovanje želimo hraniti njegove stanovalce in sosednja stanovanja. Razred naj vsebuje sledeče javno dostopne konstruktorje in metode:

- `public Stanovanje(Oseba[] stanovalci)` [11–50]

Inicializira objekt, ki predstavlja stanovanje s podanimi stanovalci in (zaenkrat) brez sosednjih stanovanj. Parameter `stanovalci` lahko kaže tudi na prazno tabelo (tabela dolžine 0), v nobenem testnem primeru pa ni `null`.

- `public int steviloStanovalcev()` [11–15]

Vrne število oseb v stanovanju `this`.

- `public int steviloStarejsihOd(Oseba os)` [16–20]

Vrne število oseb v stanovanju `this`, starejših od osebe `os`. Parameter `os` ni nikoli `null`.

- `public int[] mz()` [21–25]

Vrne tabelo z dvema elementoma: prvi vsebuje število moških, drugi pa število žensk v stanovanju `this`.

- `public Oseba starosta()` [26–30]

Vrne najstarejšo osebo v stanovanju `this`. Če je stanovanje prazno, vrne `null`. Lahko predpostavite, da v stanovanju biva največ ena najstarejša oseba.

- `public void nastaviSosedo(Stanovanje levi, Stanovanje zgornji,  
Stanovanje desni, Stanovanje spodnji)` [31–50]

Nastavi vse štiri sosede stanovanja `this`. Če katerega od sosedov ni, ima pripadajoči parameter vrednost `null`. Metoda naj **ne** nastavi stanovanja `this` kot soseda stanovanj `levi`, `zgornji`, `desni` ali `spodnji`!

- `public Oseba starostaSosescine()` [31–35]

Vrne najstarejšo osebo v skupini stanovanj, ki jo sestavljajo stanovanje `this` in vsi njegovi sosedje (torej največ 5 stanovanj). Če je celotna skupina stanovanj prazna, naj metoda vrne `null`. Lahko predpostavite, da v skupini obstaja največ ena najstarejša oseba.

- `public Oseba[] sosedjeSosedov()` [36–40]

Vrne tabelo, ki vsebuje vse stanovalce, ki živijo v *sosedih sosedov* stanovanja `this`. Stanovanje `this` ni samo svoj sosed soseda! Tabela mora biti dolga natanko toliko,

kolikor je oseb v sosedih sosedov stanovanja `this`, vrstni red njenih elementov pa je lahko poljuben.

## Razred Blok

Objekt razreda `Blok` predstavlja celoten blok. Razred naj vsebuje sledeče javno dostopne konstruktorje in metode:

- `public Blok(Stanovanje stanovanje)` [41–50]

Inicializira objekt, ki predstavlja blok, ki vsebuje podano stanovanje (ter vse njegove sosedne, sosedne sosedov itd.). Parameter `stanovanje` ni nikoli `null`.

- `public Oseba starosta()` [41–45]

Vrne najstarejšo osebo v bloku `this`. Če je blok povsem prazen, vrne `null`. Lahko predpostavite, da v celotnem bloku obstaja največ ena najstarejša oseba.

- `public int[][] razporeditev()` [46–50]

Vrne tabelo velikosti  $M \times N$ , kjer je  $M$  število etaž (»vrstic«) bloka `this`,  $N$  pa število njegovih vertikal (»stolpcev«). V primeru na sliki 1, denimo, velja  $M = 4$  in  $N = 5$ . Vsak element vrnjene tabele podaja število stanovalcev v pripadajočem stanovanju bloka, element z vrednostjo  $-1$  pa pove, da na pripadajočem mestu v bloku ni stanovanja.

Za primer s slike 1 bi vrnjena tabela izgledala tako:

$$\begin{bmatrix} -1 & -1 & a & -1 & -1 \\ b & c & d & e & -1 \\ -1 & f & g & -1 & -1 \\ -1 & -1 & h & i & j \end{bmatrix}$$

Pri tem  $a$  predstavlja število oseb v stanovanju A,  $b$  število oseb v stanovanju B itd.

## Predpostavke

Lahko se zanesete na sledeče predpostavke:

- V vseh testnih razredih so klici metod `nastaviSosedo` med seboj konsistentni: če obstaja klic, ki stanovanje  $P$  nastavi kot levega (zgornjega/desnega/spodnjega) sosedo stanovanja  $Q$ , potem obstaja tudi klic, ki stanovanje  $Q$  nastavi kot desnega (spodnjega/levega/zgornjega) sosedo stanovanja  $P$ .
- Število stanovanj v bloku v vseh testnih primerih pripada intervalu  $[1, 10^3]$ .
- Skupno število oseb v bloku v vseh testnih primerih pripada intervalu  $[0, 10^4]$ .
- Vsaka oseba živi v največ enem stanovanju.

## Oddaja naloge

Pri izdelavi razredov `Oseba`, `Stanovanje` in `Blok` izhajajte iz datotek `Oseba.java`, `Stanovanje.java` in `Blok.java`, ki jih najdete na Učilnici v pripadajočem arhivu zip. Imen datotek ne spreminjajte; ostanejo naj `Oseba.java`, `Stanovanje.java` in `Blok.java`.

Oddajte vse tri datoteke, tudi če katere od njih morebiti ne boste spreminjali. Datotek ne »zipajte«, ampak vsako posebej oddajte na Učilnico.