

Tema:**Homogena podatkovna struktura – enorazsežna/1D tabela****Vsebine:**

Linearni postopki na tabelaričnih podatkih
Linearno iskanje, filtriranje, izbiranje
Razredna knjižnica metod

Opomba: več ali manj je zahtevana izvedba po dveh metod; gre za isti mehanizem, le sredstvo manipulacije je drugo; torej : copy&paste.

Naloga 1

V zagonski(driver) metodi razreda MojeMetodeZaTabele obstaja delaracija tabele z inicializacijo elementov:

```
int[] ti = {12,3,5,9,-2,4};
```

Spišite razredni metodi vrniPrvega/1 in vrniZadnjega/1, ki bosta kot argument prejeli tabelo, kot je podana in: prva naj vrne vrednost prvega elementa tabele, druga vrednost zadnjega elementa tabele. Za vsak slučaj: vrniPrvega/1 interpretirate kot (verjetno static) int vrniPrvega(int[] tab)

Naloga 2

V razred dodajte še (dve) enakovredni metodi, ki bosta cilje iz naloge 1 izvedli na tabeli znakov, kot npr.:

```
char[] tc = new char[]{'a','h','H','j','X'};
```

(metodi opravljata enako delo kot tisti iz Naloga1. Zaslužita enako ime?)

Naloga 3

- Razredna metoda vrniVsoto/1 vrne vsoto elementov take tabele, kot je definirana v nalogi 1
- Razredna metoda vrniVsoto/2 vrne vsoto (prvih) n elementov tabele, kot je podana v nalogi 1

premislite : bo vrnjena vsota vrste int ali long ? Kdaj boste uporabili prvega, kdaj drugega. Pri vrniVsoto/2 je drugi argument število elementov, ki jih upoštevamo pri izračunu vsote. Če je ta nič, je vsota pač nič. Kaj pa če je njegova vrednost negativna? Hm, recimo, da potem upošteva v vsoti z-a-d-n-j-i-h n elementov v tabeli. In, če je elementov manj kot n, jih je v vsoti upoštevanih pač manj kot n.

- Razredna metoda vrniVsoto/3 vrne vsoto n zaporednih elementov tabele, začenši s k-tim elementom tabele. (glejte predhodne opombe)

Naloga 4

Metodi razreda `MojeMetodeZaTabele` z imenoma `getIntTab/1` in `genCharTab/1` vrneta tabelo ustreznega tipa (glej ime metode) in sicer tako dolgo, kot je vrednost argumenta tabele. Vrnjeni tabeli sta napolnjeni z naključnimi vrednostmi iz obsega `int` oz. iz množice znakov/simbolov za črke in številke.

Za kontrolo delovanja v razred dodajte še metodi za izpis vrednosti elementov tabel: `izpisiTab/1`. Pri prvi je tako argument tabela `int`-ov, pri drugi tabela `char`-ov.

Preskusite še : `System.out.println(java.util.Arrays.toString(tab));`

Naloga 5

(linearno oz. zaporedno iskanje) Spišite razredno metodo `aliJeVTabeli/2` oz. `static boolean aliJeVTabeli(int[] t, int iskana)`; . Metoda vrne vrednost `true`, če se vrednost argumenta iskana nahaja v tabeli, sicer vrne vrednost `false`; . Dodajte tudi metodo za enako iskanje po tabeli znakov. Metodi naj bosta del razreda iz Naloga 1.

Naloga 6

Poigrajte se z metod-`o/ama` iz Naloga 5; naredite kopijo metod v `aliJeVTabeliTest/2`. V kopijah dodajte števec, ki bo štel število primerjav pri iskanju (izvedb if stavkov) in število izpisal pred izhodom iz metode (pred stavkom `return`). Nato naredite tabelo 12-tih elementov in jo izpišite na zaslon. In:

- Izvedite iskanje elementa, ki se nahaja na prvem mestu v tabeli. Zapišite število primerjav.
- Izvedite iskanje elementa, ki se nahaja na zadnjem metu v tabeli. Zapišite število primerjav.
- Izvedite iskanje elementa, ki se nahaja na sredini tabele(recimo na mestu 6). Zapišite število primerjav.
- Spišite test (metodo?), ki bo 1000x izvedla iskanje naključnega elementa iz tabele in izračunajte povprečno število primerjav. Zapišite to število.

Vprašanja, na katera morate znati odgovoriti so:

Kateri element najdalj iščemo v zaporedju (najdalj je: za katerega 'porabimo' največ primerjav(if-ov)? Katerega iščemo najmanj časa? Kolikšno je povprečje pri iskanju ? Kako je število primerjav odvisno od števila elementov v zaporedju ? logaritemsko, linearno, kvadratično, eksponentialno

Naloga 7

Predelajte mehanizma metod iz Naloga5 v metodi `kjeVTabeli/2`, kjer pa ti nalogi vrneta poziciji iskanih elementov v tabeli. Metodi naj vrneta: -1 v primeru, da številskega elementa ni v tabeli, znak na poziciji 0 v kodni tabeli, v primeru iskanja znakovnega elementa.

Naloga 8

Metodi razreda iz Naloga 1 z imenom `skrajšajTabeloNa/2`, skrajša z argumentom podano tabelo na dolžino, podano z vrednostjo drugega argumenta in skrajšano vrne. Predpostavite, da v tabeli pri krajšanju zavržemo konec v tabelo zapisanega zaporedja (ohranimo prve elemente tabele, tiste na nižjih indeksih).

Metodi `zanemariPrviZadnji/1` vrneta tabelo brez prvega in zadnjega elementa z argumentom podane tabele.

Naloga 9

Metoda razreda `tableLeftTrim/2` naj bo kopija skrajševanja z 'leve' strani tabele. Dodajte razredu metodi `tableRightTrim/2`, ki bosta počeli enako, le z 'desnega' konca tabele, ter metodi `tableMidTrim/3`, ki bo ohranila zgolj n -elementov od k -tega elementa v tabeli dalje

Naloga 10

Spišite metode in jih dodajte v razred `MojeMetodeZaTabele`:

- Vrne tabelo lihih elementov podane tabele (lihi so tisti, ki so številsko lihi ali znaki na lihih mestih v kodni tabeli znakov).
- Vrne tabelo sodih elementov tabele.
- Iz znakovne tabele vrne tabelo brez samoglasnikov.

Naloga 11

Metoda razreda `statZnakov/1` na podani tabeli izvede statistiko pojavitev posameznih črk v tabeli. Statistiko izmed vseh znakov zapisanih v tabeli izvajate zgolj za črkovne znake, pri čemer 'a' in 'A' predstavljata isto črko. Rezultat izvedbe naj bo tabela, ki za vsako zahtevano črko vsebuje število njenih pojavitev. Metoda naj to metodo vrne kličočemu delu programa. Lahko uporabite poenostavitve in uporabite zgolj angleško abecedo.

/uporabite (direktno) razprševanje : črk je v ang. Abecedi 26, torej potrebujete tabelo števil te velikosti; A je prva črka in njene pojavitve zapisujemo na 1. el tabele (indeks 0), B druga itd. .. A in a preslikamo v isti element (0)... torej je logika : `if(crka=='A' || or crka=='a') tabelaStevil[0]++;`