

Vaje - Vrsta

Ogledali si bomo strukturo `Vrsta` in njeno uporabo. Dobili boste že pripravljeno implementacijo vrste `Vrsta.class`, ki pozna osnovne funkcije za delo z vrsto. Ta implementacija vrste pozna konstruktorja

- `Vrsta()` – naredi prazno vrsto
- `Vrsta(int[] t)` – naredi vrsto iz tabele, `t[0]` je začetni element

in metode:

- `public void vstavi(int x)` – vstavi celo število v vrsto.
- `public void odstrani()` – odstrani začetni element iz vrste.
- `public boolean prazna()` – ali je vrsta prazna.
- `public int vrh()` – vrne začetni element vrste, vrsta se ne spremeni.
- `public String toString()` – vrne vrsto v obliki niza `»# - elt1 - elt2 - elt 3 - ...«` (prazna vrsta se izpiše kot `»#«`).
- `public void izpis()` – izpiše vrsto v obliki `»Vrsta: # - elt1 - elt2 - elt 3 - ...«`.

Če metode uporabimo narobe (ERROR v definiciji + vstavljanje v polno vrsto), dobimo `IllegalStateException`. Večino rešitev pišemo v datoteki `Resitev.java`. **Osnutek** testnega programa za te naloge je `TestResitev.java`.

1. Oglej si predstavitev vrste (`VrstaPrav.java`). Vrsta je predstavljena s krožno tabelo. Nariši predstavitev, označi ustrezna indeksa začetka in konca. Nariši si primer, kako se vrsta polni in prazni.
2. V razredu (`VrstaNapak.java`) sta dve (2) NAPAKI (no, datoteka se mora seveda imenovati `Vrsta.java` – tega ne štejem za napako – ker pa že imaš rešitev (drugačno) `Vrsta.java`, pazi, da boš ustrezno naredil podimenike)!!! Sestavi testni program, ki preveri, če vse operacije nad vrsto delujejo pravilno in s pomočjo rezultatov poišči napaki! Najprej premisli, kaj vse bi bilo smiselno preveriti. **Osnutek** testnega programa je `TestVrstaNapak.java`.
3. V razredu `Resitev` sestavi statično metodo `int nti(Vrsta v, int n)`, ki vrne n-ti element vrste. Velja $n > 0$. Če ima vrsta manj kot n elementov, vrni `Integer.MIN_VALUE`. Vrste `v` kasneje ne potrebujemo več.
4. Sestavi statično metodo `Vrsta brisi_nti(Vrsta v, int n)`, ki vrne vrsto iz enakih elementov kot `v`, le da izbriše n-ti element vrste `v`. Velja $n > 0$. Če n-tega elementa ni, vrnemo celotni `v`! Vrste `v` kasneje ne potrebujemo več. Pazi na vrstni red elementov v vrsti!
5. Sestavi statično metodo `void izloci(Vrsta v, int x)`, ki iz vrste izloči vse podatke, ki so enaki `x`.
6. Napiši statično metodo `void obrni(Vrsta v)`, ki obrne vrsto `v`!
7. Sestavi statično metodo `Vrsta zlij(Vrsta v1, Vrsta v2)`, ki dve nepadajoče urejeni vrsti zlije v novo nepadajoče urejeno vrsto!
8. Iz razreda `Vrsta` izpelji razred `VrstaPoljubno`, kjer je vrsta lahko poljubno velika, ne da bi uporabnik sam skrbel zato. Namig: če porabiš ves prostor v tabeli, avtomatsko naredi dvakrat večjo novo tabelo in vanjo preloži elemente (spremeniti bo torej potrebno metodo `vstavi`). Ali bi to šlo, če bi komponente v razredu `Vrsta` imele dostop `private`? V datoteki `VrstaPoljubno.java` je POSKUS narediti tak razred, a ima nekaj napak! Namig: napaka je v enem od konstruktorjev, pri vstavi pa malo preveč površno obravnavamo indekse, da o pozabljenem elementu sploh ne govorimo! **Osnutek** testnega programa je `TestVrstaPoljubno.java`