

Teme:

Raba obstoječih struktur iz Java collection framework : ArrayList / Map-SET / HashMap-TreeSet
Primerjava in razvrščanje lastnih zapisov po izbranem kriteriju- primerljivost zapisov ; comparable
Razredni diagrami
Dokumentacija kode
Testiranje kode

Cilj nalog v nalogi 5 je realizacija (notranje)pomnilniške strukture za vzdrževanje spiska desetih najboljših igralcev igre X. Pri tem v sistem dobivamo podatke/zapise z identifikacijo uporabnika in številom točk, ki jih je dosegel v igri. Spisek vzdržuje le 10 najboljših igralcev igre, z njihovimi najboljšimi rezultati ter podatkom o tem, kolikokrat so igro igrali.

Struktura posameznega zapisa v spisku je dana kot:

```
class Zapis{
    private String upor;    // igralčev identifikator/nick name
    private int rez;        // njegov rezultat v igri (najboljši)
    private int st;         // število ponovitev igre
}
```

in celoten spisek kot:

```
class Zbirka{
    private Zapis tt[];    // omejena na velikost 10
}
```

Naloga 1

Spisek realiziramo kot (statično) tabelo desetih zapisov. Kot vajo lahko nalogo spišete sami ali pa uporabite implementacijo v dokumentu 05_RSO04_vaja_top10_01.zip. V vsakem primeru:

Preskusite delovanje za vse možne primere delovanja; analiza naj bo razvidna iz zapisnika testa.

Zapisnik je tabela, kot:

Predhodna situacija	Aktivnost, ki se preverja	Rezultat preverjanja	Ustreza zahtevam	opomba
Spisek je prazen	Dodam 1 zapis	V spisku je en sam element, postavljen na prvo mesto	da	-
...
....
Spisek je poln (10 el)	Dodam 1 zapis: uporabnik je že v spisku, njegov rezultat je manjši od obstoječega v spisku	V spisku sta 2 zapisa s tem uporabnikom	Ne ustreza zahtevam	popraviti

Spisek naj pokriva vse situacije, ki med izvajanjem lahko nastopijo.

Naloga 2

V skladu s (eventualno popravljeno/dopolnjeno) realizacijo iz naloge 1 :

- a) izrišite razredni diagram realizacije,
- b) zgenerirajte javansko programsko dokumentacijo .

Naloga 3

V realizaciji strukture razreda Zbirke statični mehanizem pomnjenja (tabelo) zamenjajte s kolekcijo iz Java Collection Framework, ki je po načinu rabe še najbolj podobna tabeli; to je ArrayList.

Pazite: kolekcije ne poznajo direktnega indeksiranja; ArrayList v ta namen implementira metode, ki vam omogočajo rabo indeksiranja z indeksom kot argumentom metode. Npr. :

- add(int index, E element)
- get(int index)
- remove(int index)

Naloga 4

Tako knjižnica Arrays, ko delate s tabelami ali pa Collections, kadar delate s kolekcijami, vsebujeta metode, ki vam omogočajo razvrščanje strukture. Primer naše realizacije je v tem, da želimo imeti zapise razvrščene po vrednosti rezultata, ki pa je le del zapisa. Java vnaprej ne more vedeti, po čem primerjamo take zapise, zato je potrebno tak mehanizem implementirati aka: povedati programskemu jeziku, kaj je merilo za razvrščanje. Izhodišče: zapisi naj bodo Comparable (interface, zagotavlja implementacijo metode compareTo).

Predalajte razred Zapis, da bo primerljiv/Comparable (+ implementirajte metodo compareTo). Dokažite, da tako program iz naloge 1, kot tisti iz naloge 3 delujeta z metodami sort, vgrajenimi v javanske razrede (Arrays.sort oz. Collections.sort).

Naloga 5

Struktura, ki jo potrebujemo za naš seznam mora biti urejena po določenem kriteriju. Tipično ArrayList ali seznam nista strukturi, ki sta optimalni na naš seznam (veliko sprememb in s tem veliko menjavanj vrstnih redov). Potrebujemo strukturo, ki je že v osnovi ordered&sorted (v info npr.: Choosing the right collection, Java Practices 3.01, 2018, <http://www.javapractices.com/topic/TopicAction.do?Id=65>, okt.2018). Recimo, da se iz nekih razlogov odločimo za implementacijo s TreeSet (ali TreeMap).

Realizirajte spisek.