

## Firma.java

```
1 package pacOrg;
2
3 import java.nio.charset.Charset;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6 import java.nio.file.StandardOpenOption;
7 import java.time.LocalDate;
8 import java.time.format.DateTimeFormatter;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 import javax.swing.JOptionPane;
13
14 public class Firma {
15
16
17     private ArrayList<UserClass> listaKorisnika;
18     private ArrayList<RadnaStolica> listaStolica;
19     private ArrayList<Cena> listaCena;
20     private ArrayList<Historiat> listaIstorije;
21     private ArrayList<TerminiOrd> listaTermina;
22
23
24
25     public Firma() {
26         this.listaKorisnika = new ArrayList<UserClass>();
27         this.listaStolica = new ArrayList<RadnaStolica>();
28         this.listaCena = new ArrayList<Cena>();
29         this.listaIstorije = new ArrayList<Historiat>();
30         this.listaTermina = new ArrayList<TerminiOrd>();
31     }
32
33
34
35
36     public ArrayList<TerminiOrd> getListaTermina() {
37         return listaTermina;
38     }
39
40
41
42
43     public void setListaTermina(ArrayList<TerminiOrd> listaTermina) {
44         this.listaTermina = listaTermina;
45     }
46
47
48
49
50     public Firma(ArrayList<UserClass> listaKorisnika, ArrayList<RadnaStolica>
radnaStolica) {
51         super();
52         this.listaKorisnika = listaKorisnika;
53         this.listaStolica = radnaStolica;
54     }
55
56
57
58     public Firma(ArrayList<UserClass> listaKorisnika, ArrayList<RadnaStolica>
radnaStolica, ArrayList<Cena> listaCena) {
59         super();
60         this.listaKorisnika = listaKorisnika;
```

Firma.java

```
61         this.listaStolica = radnaStolica;
62         this.listaCena = listaCena;
63     }
64
65
66
67     public Firma(ArrayList<UserClass> listaKorisnika, ArrayList<RadnaStolica>
listaStolica, ArrayList<Cena> listaCena,
68         ArrayList<Historiat> listaIstorije) {
69         super();
70         this.listaKorisnika = listaKorisnika;
71         this.listaStolica = listaStolica;
72         this.listaCena = listaCena;
73         this.listaIstorije = listaIstorije;
74     }
75
76
77
78
79     public Firma(ArrayList<UserClass> listaKorisnika, ArrayList<RadnaStolica>
listaStolica, ArrayList<Cena> listaCena,
80         ArrayList<Historiat> listaIstorije, ArrayList<TerminiOrd> listaTermina) {
81         super();
82         this.listaKorisnika = listaKorisnika;
83         this.listaStolica = listaStolica;
84         this.listaCena = listaCena;
85         this.listaIstorije = listaIstorije;
86         this.listaTermina = listaTermina;
87     }
88
89
90
91
92     public ArrayList<RadnaStolica> getListaStolica() {
93         return listaStolica;
94     }
95
96
97     public void setListaStolica(ArrayList<RadnaStolica> listaStolica) {
98         this.listaStolica = listaStolica;
99     }
100
101
102     public ArrayList<Historiat> getListaIstorije() {
103         return listaIstorije;
104     }
105
106
107     public void setListaIstorije(ArrayList<Historiat> listaIstorije) {
108         this.listaIstorije = listaIstorije;
109     }
110
111
112     public ArrayList<Cena> getListaCena() {
113         return listaCena;
114     }
115
116
117     public void setListaCena(ArrayList<Cena> listaCena) {
118         this.listaCena = listaCena;
119     }
120
```

```

121
122 public ArrayList<UserClass> getListaKorisnika() {
123     return listaKorisnika;
124 }
125
126 public void setListaKorisnika(ArrayList<UserClass> listaKorisnika) {
127     this.listaKorisnika = listaKorisnika;
128 }
129
130 public ArrayList<RadnaStolica> getRadnaStolica() {
131     return listaStolica;
132 }
133
134 public void setRadnaStolica(ArrayList<RadnaStolica> radnaStolica) {
135     this.listaStolica = radnaStolica;
136 }
137
138
139
140 @Override
141 public String toString() {
142     return "Firma [listaKorisnika=" + listaKorisnika + ", listaStolica=" +
143     listaStolica + ", listaCena=" + listaCena
144     + ", listaIstorije=" + listaIstorije + ", listaTermina=" + listaTermina +
145     "]\n";
146 }
147
148
149 public boolean dodavanjePacijenta (UserClass userClass) {
150
151     for (int i = 0; i < listaKorisnika.size(); i++) {
152         UserClass korisnikLista = this.listaKorisnika.get(i);
153         if (korisnikLista.getId() == userClass.getId()) {
154             return false;
155         }
156     }
157
158     this.listaKorisnika.add(userClass);
159     save("pacijenti.txt");
160     return true;
161 }
162
163
164
165 public boolean dodavanjeTermina (TerminiOrd termin) {
166
167     for (int i = 0; i < listaTermina.size(); i++) {
168         TerminiOrd korisnikLista = this.listaTermina.get(i);
169         if (korisnikLista.getID() == termin.getID()) {
170             return false;
171         }
172     }
173
174     this.listaTermina.add(termin);
175     saveTermini("termini.txt");
176     return true;
177 }
178
179
180

```

```

181 public void lstTermina() {
182
183     loadTermini("termini.txt");
184     System.out.printf("%15s %15s %15s %15s %15s %15s \n", "ID", "ime", "prezime",
185         "Vreme Pocetno", "Vreme Krajnje", "Datum");
186
187     for (int i = 0; i < this.listaTermina.size(); i++) {
188         TerminiOrd ord = this.listaTermina.get(i);
189         System.out.println(ord);
190     }
191 }
192
193
194 public void lstImePrezimeTermini(String ime, String prezime) {
195
196     loadTermini("termini.txt");
197     System.out.printf("%15s %15s %15s %15s %15s %15s \n", "ID", "ime", "prezime",
198         "Vreme Pocetno", "Vreme Krajnje", "Datum");
199
200     for (int i = 0; i < this.listaTermina.size(); i++) {
201         if (this.listaTermina.get(i).getIme().equalsIgnoreCase(ime)&&
202             this.listaTermina.get(i).getPrezime().equalsIgnoreCase(prezime)) {
203
204             TerminiOrd ord = this.listaTermina.get(i);
205             System.out.println(ord);
206
207         }
208     }
209 }
210
211
212 public boolean unosCene (Cena cena) {
213
214     for (int i = 0; i < listaCena.size(); i++) {
215         Cena cenaLista = this.listaCena.get(i);
216         if (cenaLista.getId() == cena.getId()) {
217             return false;
218         }
219     }
220
221     this.listaCena.add(cena);
222     saveCena("cene.txt");
223     return true;
224 }
225
226
227
228 // save i load metoda historiata
229
230 public void saveHistoriat(String path) {
231
232     ArrayList<String> lines = new ArrayList<String>();
233
234     for (int i = 0; i < this.listaIstorije.size(); i++) {
235
236         Historiat podatakLista = this.listaIstorije.get(i);
237
238         int id = podatakLista.getID();
239         String imePacijenta = podatakLista.getImePacijenta();
240         String GornjiLeviZubi = podatakLista.getGornjiLeviZubi();
241         String GornjiDesniZubi = podatakLista.getGornjiDesniZubi();
242         String DonjiLeviZubi = podatakLista.getDonjiLeviZubi();

```

# Firma.java

```

243         String DonjiDesniZubi = podatakLista.getDonjiDesniZubi();
244         String Datum = podatakLista.getDatum();
245
246         String line = id + "/" + imePacijenta + "/" + GornjiLeviZubi + "/" + GornjiDesniZubi
+ "/" + DonjiLeviZubi + "/" + DonjiDesniZubi + "/" + Datum;
247
248         lines.add(line);
249     }
250
251     try {
252         Files.write(Paths.get(path), lines, Charset.defaultCharset(),
StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING,
StandardOpenOption.WRITE);
253     } catch (java.io.IOException e) {
254         System.out.println("Datoteka " + path + " nije pronadjena.");
255     }
256 }
257
258
259 public void loadHistoriat(String path) {
260
261
262     this.listaIstorije = new ArrayList<Historiat>();
263     List<String> lines;
264     try {
265         lines = Files.readAllLines(Paths.get(path), Charset.defaultCharset());
266         for (String line: lines) {
267             String[] attributes = line.split("/");
268
269             int id = Integer.parseInt(attributes[0]);
270             String imePacijenta = attributes[1];
271             String gornjiLeviZubi = attributes[2];
272             String gornjiDesniZubi = attributes[3];
273             String donjiLeviZubi = attributes[4];
274             String donjiDesniZubi = attributes[5];
275             String datum = attributes[6];
276
277             Historiat historiat = new Historiat(id, imePacijenta, gornjiLeviZubi,
gornjiDesniZubi,
278                 donjiLeviZubi, donjiDesniZubi, datum);
279             this.listaIstorije.add(historiat);
280         }
281     } catch (java.io.IOException e) {
282         System.out.println("Datoteka " + path + " nije pronadjena.");
283     }
284 }
285
286
287
288
289
290
291
292 // save i load metode
293
294 public void save (String path) {
295     ArrayList<String> lines = new ArrayList<String>();
296
297     DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd.MM.yyyy.");
298
299     for (int i = 0; i < this.listaKorisnika.size(); i++) {
300         UserClass userClass = this.listaKorisnika.get(i);

```

# Firma.java

```

301
302         int id = userClass.getId();
303         String ime = userClass.getIme();
304         String prezime = userClass.getPrezime();
305         String alergije = userClass.getAlergije();
306         String oboljenja = userClass.getHronicnaOboljenja();
307         String godiste = userClass.getGodiste();
308         String jmbg = userClass.getJmbg();
309         int telefon = userClass.getTelefon();
310         String adresa = userClass.getAdresa();
311         LocalDate datum = userClass.getDatum();
312         String formiraniDatum = dtf.format(datum);
313
314         String line = id + "/" + ime + "/" + prezime + "/" + alergije
315                     + "/" + oboljenja + "/" + godiste + "/" + jmbg + "/" + telefon
316                     + "/" + adresa + "/" + formiraniDatum;
317         lines.add(line);
318     }
319
320     try {
321         Files.write(Paths.get(path), lines, Charset.defaultCharset(),
StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING,
StandardOpenOption.WRITE);
322     } catch (java.io.IOException e) {
323         System.out.println("Datoteka " + path + " nije pronadjena.");
324     }
325 }
326
327 /*
328  * private int id;    // broj kartona
329  private String ime;
330  private String prezime;
331  private String alergije;
332  private String hronicnaOboljenja;
333  private String godiste;
334  private int jmbg;
335  private int telefon;
336  private String adresa;
337  private LocalDate datum;
338  private double trenutnoStanje;
339  private double dug;
340  private double placeno;
341  */
342
343 public void load (String path) {
344
345     this.listaKorisnika = new ArrayList<UserClass>();
346     DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd.MM.yyyy.");
347     List<String> lines;
348
349     try {
350
351         lines = Files.readAllLines(Paths.get(path), Charset.defaultCharset());
352         for(String line: lines) {
353             String[] attributes = line.split("/");
354
355             int id = Integer.parseInt(attributes[0]);
356             String ime = attributes[1];
357             String prezime = attributes[2];
358             String alergije = attributes[3];
359             String hronicna = attributes[4];
360             String godiste = attributes[5];

```

# Firma.java

```

361         String jmbg = attributes[6];
362         int telefon = Integer.parseInt(attributes[7]);
363         String adresa = attributes[8];
364         String datum = attributes[9];
365         LocalDate datumZaCuvanje = null;
366
367         try {
368             datumZaCuvanje = LocalDate.parse(datum, dtf);
369         } catch (Exception e) {
370             e.printStackTrace();
371         }
372
373
374         UserClass userClass = new UserClass(id, ime, prezime, alergije,
375             hronicna, godiste, jmbg, telefon, adresa, datumZaCuvanje);
376         this.listaKorisnika.add(userClass);
377
378     }
379
380 } catch (java.io.IOException e) {
381     System.out.println("Datoteka " + path + " nije pronadjena.");
382 }
383
384 }
385
386
387 /*
388  * private int id;    // broj kartona
389  private String ime;
390  private String prezime;
391  private String alergije;
392  private String hronicnaOboljenja;
393  private String godiste;
394  private int jmbg;
395  private int telefon;
396  private String adresa;
397  private LocalDate datum;
398  private double trenutnoStanje;
399  private double dug;
400  private double placeno
401  */
402
403
404
405 // metoda za listanje korisnika
406
407 public void lstCena() {
408     loadCena("cene.txt");
409     System.out.printf("%15s %15s %15s %15s %15s %15s \n", "id", "ime", "prezime",
410 "dug", "placeno", "trenutnoStanje");
411     for (int i = 0; i < this.listaCena.size(); i++) {
412         Cena cena = this.listaCena.get(i);
413         System.out.println(cena);
414     }
415 }
416
417 public void lstKorisnikaID(int id) {
418
419     load("pacijenti.txt");
420     System.out.printf("%15s %15s %15s %25s %25s %10s %14s %12s %30s %10s \n", "ID",
421 "Ime", "Prezime",

```

```

421         "Alergije", "Oboljenja", "Godiste", "JMBG", "Telefon", "Adresa", "Datum");
422
423     for (int i = 0; i < this.listaKorisnika.size(); i++) {
424         if (this.listaKorisnika.get(i).getId() == id) {
425             UClass userClass = this.listaKorisnika.get(i);
426             System.out.println(userClass);
427         }
428     }
429 }
430
431 public void lstKorisnikaImePrezime(String ime, String prezime) {
432     load("pacijenti.txt");
433     System.out.printf("%15s %15s %15s %25s %25s %10s %14s %12s %30s %10s \n", "ID",
434 "Ime", "Prezime",
435         "Alergije", "Oboljenja", "Godiste", "JMBG", "Telefon", "Adresa", "Datum");
436
437     for (int i = 0; i < this.listaKorisnika.size(); i++) {
438
439         if (this.listaKorisnika.get(i).getIme().equalsIgnoreCase(ime) &&
440             this.listaKorisnika.get(i).getPrezime().equalsIgnoreCase(prezime)) {
441             UClass userClass = this.listaKorisnika.get(i);
442             System.out.println(userClass);
443         }
444     }
445 }
446
447 public void lstKorisnika() {
448
449     load("pacijenti.txt");
450     System.out.printf("%15s %15s %15s %25s %25s %10s %14s %12s %30s %10s \n", "ID",
451 "Ime", "Prezime",
452         "Alergije", "Oboljenja", "Godiste", "JMBG", "Telefon", "Adresa", "Datum");
453
454     for (int i = 0; i < this.listaKorisnika.size(); i++) {
455
456         UClass userClass = this.listaKorisnika.get(i);
457         System.out.println(userClass);
458     }
459 }
460
461 }
462
463 public void lstCenaI(int id) {
464
465     loadCena("cene.txt");
466     System.out.printf("%15s %15s %15s %15s %15s %15s \n", "id", "ime", "prezime",
467 "dug", "placeno", "trenutnoStanje");
468     for (int i = 0; i < this.listaCena.size(); i++){
469         if (this.listaCena.get(i).getId() == id) {
470             Cena cena = this.listaCena.get(i);
471             System.out.println(cena);
472         }
473     }
474 }
475
476 public void lstCenaImePrezime(String ime, String prezime) {
477
478     loadCena("cene.txt");
479     System.out.printf("%15s %15s %15s %15s %15s %15s \n", "id", "ime", "prezime",

```



```

    "dug", "placeno", "trenutnoStanje");
480     for (int i = 0; i < this.listaCena.size(); i++) {
481
482         if (this.listaCena.get(i).getIme().equalsIgnoreCase(ime)&&
483             this.listaCena.get(i).getPrezime().equalsIgnoreCase(prezime)) {
484
485             Cena cena = this.listaCena.get(i);
486             System.out.println(cena);
487
488         }
489     }
490 }
491
492 public UserClass izmenaKorisnika(UserClass userClass) {
493     for(int i = 0; i < this.listaKorisnika.size(); i++) {
494         UserClass userLista = this.listaKorisnika.get(i);
495         int id = userLista.getId();
496         if (id == userClass.getId()) {
497             UserClass pacKojiSeMenja = this.listaKorisnika.set(i, userClass);
498             save("pacijenti.txt");
499             return pacKojiSeMenja;
500         }
501     }
502 }
503 return null;
504 }
505
506
507 /*
508 *
509 0 private String ID;
510 1 private String ime;
511 2 private String prezime;
512 3 private int vremePocetno;
513 4 private int vremeKrajnje;
514 5 private LocalDate datum;
515 */
516
517
518 public void saveTermini(String path) {
519
520     ArrayList<String>lines = new ArrayList<String>();
521     DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd.MM.yyyy.");
522
523     for(int i = 0; i < this.listaTermina.size(); i++) {
524
525         TerminiOrd ord = this.listaTermina.get(i);
526
527         int id = ord.getID();
528         String ime = ord.getIme();
529         String prezime = ord.getPrezime();
530         double VremePocetno = ord.getVremePocetno();
531         double VremeKrajnje = ord.getVremeKrajnje();
532         String datum = ord.getDatum();
533
534         String line = id + "/" + ime + "/" + prezime + "/" + VremePocetno + "/" + VremeKrajnje
535             + "/" + datum;
536         lines.add(line);
537     }
538
539     try {
540         Files.write(Paths.get(path), lines, Charset.defaultCharset(),

```

```

StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING,
StandardOpenOption.WRITE);
541     } catch (java.io.IOException e) {
542         System.out.println("Datoteka " + path + " nije pronadjena.");
543     }
544 }
545
546 /*
547  *
548  0 private int ID;
549  1 private String ime;
550  2 private String prezime;
551  3 private int vremePocetno;
552  4 private int vremeKrajnje;
553  5 private LocalDate datum;
554  */
555
556 public void loadTermini(String path) {
557
558     this.listaTermina = new ArrayList<TerminiOrd>();
559     List<String> lines;
560     try {
561         lines = Files.readAllLines(Paths.get(path), Charset.defaultCharset());
562         for (String line: lines) {
563             String[] attributes = line.split("/");
564             int id = Integer.parseInt(attributes[0]);
565             String ime = attributes[1];
566             String prezime = attributes[2];
567             double vremePocetno = Double.parseDouble(attributes[3]);
568             double vremeKranje = Double.parseDouble(attributes[4]);
569             String datum = attributes[5];
570
571             TerminiOrd ord = new TerminiOrd(id, ime, prezime, vremePocetno,
572                 vremeKranje, datum);
573             this.listaTermina.add(ord);
574         }
575
576     } catch (java.io.IOException e) {
577         System.out.println("Datoteka " + path + " nije pronađena.");
578     }
579
580
581 }
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

```

```

601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616 public UClass izmenaKorisnika2 (UClass userClass) {
617     for(int i = 0 ; i < this.listaKorisnika.size(); i++) {
618         UClass korisnikLista = this.listaKorisnika.get(i);
619         String ime = korisnikLista.getIme();
620         String prezime = korisnikLista.getPrezime();
621         if (ime.equalsIgnoreCase(userClass.getIme())
622             && prezime.equalsIgnoreCase(userClass.getPrezime())) {
623
624             UClass pacKojiSeMenja = this.listaKorisnika.set(i, userClass);
625             save("pacijenti.txt");
626             return pacKojiSeMenja;
627         }
628     }
629 }
630 }
631 return null;
632 }
633
634
635
636 // save i load metode za cenu.
637
638
639 public void saveCena (String path) {
640
641     ArrayList<String>lines = new ArrayList<String>();
642
643     for (int i = 0; i < this.listaCena.size(); i++) {
644
645         Cena cena = this.listaCena.get(i);
646         int id = cena.getId();
647         String ime = cena.getIme();
648         String prezime = cena.getPrezime();
649         double dug = cena.getDug();
650         double placeno = cena.getPlaceno();
651         double trenutnoStanje = cena.getTrenutnoStanje();
652
653         String line = id+"/"+ime+"/"+prezime+"/"+dug+"/"+placeno+"/"+trenutnoStanje;
654         lines.add(line);
655     }
656 }
657
658 try {
659     Files.write(Paths.get(path), lines, Charset.defaultCharset(),
StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING,
StandardOpenOption.WRITE);
660 } catch (java.io.IOException e) {

```

```

661         System.out.println("Datoteka " + path + " nije pronadjena.");
662     }
663
664 }
665
666
667
668 /*
669  * private int id;
670  private String ime;
671  private String prezime;
672  private double dug;
673  private double placeno;
674  private double trenutnoStanje;
675  */
676
677
678 public void loadCena (String path) {
679
680     this.listaCena = new ArrayList<Cena>();
681     List<String> lines;
682     try {
683         lines = Files.readAllLines(Paths.get(path), Charset.defaultCharset());
684         for (String line: lines) {
685             String[] attributes = line.split("/");
686
687             int id = Integer.parseInt(attributes[0]);
688             String ime = attributes[1];
689             String prezime = attributes[2];
690             double dug = Double.parseDouble(attributes[3]);
691             double placeno = Double.parseDouble(attributes[4]);
692             double trenutnoStanje = Double.parseDouble(attributes[5]);
693
694             Cena cena = new Cena(id, ime , prezime, dug, placeno, trenutnoStanje);
695             this.listaCena.add(cena);
696
697         }
698     } catch (Exception e) {
699         System.out.println("Datoteka " + path + " nije pronadjena.");
700     }
701
702
703 }
704
705
706 public Cena izmenaCenaID (Cena cena) {
707
708     for(int i = 0; i < this.listaCena.size(); i++) {
709
710         Cena cenaLista = this.listaCena.get(i);
711
712         int id = cenaLista.getId();
713         double dugLista = cenaLista.getDug();
714         double placenoLista = cenaLista.getPlaceno();
715         String ime = cenaLista.getIme();
716         String prezime = cenaLista.getPrezime();
717
718         if (id == cena.getId()) {
719
720             double dugProsledjeno = cena.getDug();
721             double placenoProsledjeno = cena.getPlaceno();
722

```

# Firma.java

```

723         double dugModified = (dugLista + dugProsledjeno);
724         double placenoModified = (placenoLista+ placenoProsledjeno);
725
726         double TrenutnoStanjeM = (dugModified-placenoModified);
727
728         Cena cenaX = new Cena(id, ime, prezime, dugModified, placenoModified,
TrenutnoStanjeM);
729         this.listaCena.set(i, cenaX);
730         saveCena("cene.txt");
731         return cenaX;
732     }
733
734
735 }
736 return null;
737
738 }
739
740 /* private int id;
741     private String ime;
742     private String prezime;
743     private double dug;
744     private double placeno;
745     private double trenutnoStanje;
746 */
747
748 public Cena izmenaCenaImePrezime (Cena cena) {
749
750     for(int i = 0; i < this.listaCena.size(); i++) {
751
752         Cena cenaLista = this.listaCena.get(i);
753
754         int id = cenaLista.getId();
755         double dugLista = cenaLista.getDug();
756         double placenoLista = cenaLista.getPlaceno();
757         String ime = cenaLista.getIme();
758         String prezime = cenaLista.getPrezime();
759
760         if (ime.equalsIgnoreCase(cena.getIme())&&
761             prezime.equalsIgnoreCase(cena.getPrezime())) {
762
763             double dugProsledjeno = cena.getDug();
764             double placenoProsledjeno = cena.getPlaceno();
765
766             double dugModified = (dugLista + dugProsledjeno);
767             double placenoModified = (placenoLista+ placenoProsledjeno);
768
769             double TrenutnoStanjeM = (dugModified-placenoModified);
770
771             Cena cenaX = new Cena(id, ime, prezime, dugModified, placenoModified,
TrenutnoStanjeM);
772             this.listaCena.set(i, cenaX);
773             saveCena("cene.txt");
774             return cenaX;
775         }
776
777     }
778
779     return null;
780
781 }
782

```

```

783
784 public Cena nuliranjeCeneID(Cena cenaProsledjena) {
785
786     for (int i = 0; i < this.listaCena.size(); i++) {
787         Cena cenaLista = this.listaCena.get(i);
788         String ime = cenaLista.getIme();
789         String prezime = cenaLista.getPrezime();
790         int id = cenaLista.getId();
791         if (id == cenaProsledjena.getId()) {
792
793             Cena cenaNew = new Cena(id, ime, prezime, cenaProsledjena.getDug(),
794 cenaProsledjena.getTrenutnoStanje());
795             this.listaCena.set(i, cenaNew);
796             saveCena("cene.txt");
797             return cenaNew;
798         }
799     }
800
801     }
802     return null;
803 }
804
805
806
807 // radne stolice
808
809 // proveru id stolica
810
811 public void dodavanjeHistoriata (Historiat historiata) {
812     loadHistoriat("Historiat.txt");
813     this.listaIstorije.add(historiata);
814     saveHistoriat("Historiat.txt");
815 }
816
817 public boolean dodavanjeStolice (RadnaStolica radnaStolica) {
818
819     for (int i = 0; i < this.listaStolica.size(); i++) {
820         RadnaStolica stolicaLista = this.listaStolica.get(i);
821         if (stolicaLista.getBrojstolice() == radnaStolica.getBrojstolice()) {
822             return false;
823         }
824     }
825
826     this.listaStolica.add(radnaStolica);
827     saveStolica("radnaStolica.txt");
828     return true;
829 }
830
831
832 public boolean dodavanjeStolice2 (RadnaStolica radnaStolica) {
833
834     for (int i = 0; i < this.listaStolica.size(); i++) {
835         RadnaStolica stolicaLista = this.listaStolica.get(i);
836         if (stolicaLista.getBrojstolice() == radnaStolica.getBrojstolice()) {
837             return false;
838         }
839     }
840
841     this.listaStolica.add(radnaStolica);
842     saveStolica("radnaStolica2.txt");
843     return true;

```

```

844
845     }
846
847     public boolean dodavanjeStolice3 (RadnaStolica radnaStolica) {
848
849         for (int i = 0; i < this.listaStolica.size(); i++) {
850             RadnaStolica stolicaLista = this.listaStolica.get(i);
851             if (stolicaLista.getBrojstolice() == radnaStolica.getBrojstolice()) {
852                 return false;
853             }
854         }
855
856         this.listaStolica.add(radnaStolica);
857         saveStolica("radnaStolica3.txt");
858         return true;
859     }
860
861
862
863     public void lstStolica() {
864
865
866         loadStolica("radnaStolica.txt");
867         System.out.printf("%5s %15s %15s %15s %25s %25s %25s %25s \n", "Br",
868             "Ime Pacijenta", "vreme Pocetno", "vreme Krajno", "gornji Levi", "gornji
Desni",
869             "donji Levi", "donji Desni");
870
871         for(int i = 0; i < this.listaStolica.size(); i++) {
872
873             RadnaStolica radnaStolica = this.listaStolica.get(i);
874             System.out.println(radnaStolica);
875
876         }
877     }
878
879
880     public void lstStolica2() {
881
882
883         loadStolica("radnaStolica2.txt");
884         System.out.printf("%5s %15s %15s %15s %25s %25s %25s %25s \n", "Br",
885             "Ime Pacijenta", "vreme Pocetno", "vreme Krajno", "gornji Levi", "gornji
Desni",
886             "donji Levi", "donji Desni");
887
888         for(int i = 0; i < this.listaStolica.size(); i++) {
889
890             RadnaStolica radnaStolica = this.listaStolica.get(i);
891             System.out.println(radnaStolica);
892
893         }
894     }
895
896     public void lstStolica3() {
897
898
899         loadStolica("radnaStolica3.txt");
900         System.out.printf("%5s %15s %15s %15s %25s %25s %25s %25s \n", "Br",
901             "Ime Pacijenta", "vreme Pocetno", "vreme Krajno", "gornji Levi", "gornji
Desni",
902             "donji Levi", "donji Desni");

```

```

903
904     for(int i = 0; i < this.listaStolica.size(); i++) {
905
906         RadnaStolica radnaStolica = this.listaStolica.get(i);
907         System.out.println(radnaStolica);
908
909     }
910
911 }
912
913
914
915 public void saveStolica (String path) {
916     ArrayList<String> lines = new ArrayList<String>();
917
918     for (int i = 0; i < this.listaStolica.size(); i++) {
919         RadnaStolica stolicaLista = this.listaStolica.get(i);
920
921
922         int brojStolice = stolicaLista.getBrojstolice();
923         String imeDoktora = stolicaLista.getImedoktora();
924         String vremePocetno = stolicaLista.getVremePocetno();
925         String vremeKrajnje = stolicaLista.getVremeKrajno();
926         String gornjiLevi = stolicaLista.getGornjiLevi();
927         String gornjiDesni = stolicaLista.getGornjiDesni();
928         String donjiLevi = stolicaLista.getDonjiLevi();
929         String donjiDesni = stolicaLista.getDonjiDesni();
930
931         String line = brojStolice + "/" + imeDoktora + "/" + vremePocetno + "/" +
vremeKrajnje
932             + "/" + gornjiLevi + "/" + gornjiDesni + "/" + donjiLevi + "/" + donjiDesni;
933
934         lines.add(line);
935     }
936     try {
937         Files.write(Paths.get(path), lines, Charset.defaultCharset(),
StandardOpenOption.CREATE, StandardOpenOption.TRUNCATE_EXISTING,
StandardOpenOption.WRITE);
938     } catch (java.io.IOException e) {
939         System.out.println("Datoteka " + path + " nije pronajena.");
940     }
941
942 }
943
944
945 /*
946  * private int brojstolice;
947  private String imedoktora;
948  private String vremePocetno;
949  private String vremeKrajno;
950  private String gornjiLevi;
951  private String gornjiDesni;
952  private String donjiLevi;
953  private String donjiDesni;
954  */
955
956 public void loadStolica(String path) {
957
958     this.listaStolica = new ArrayList<RadnaStolica>();
959     List<String> lines;
960     try {
961         lines = Files.readAllLines(Paths.get(path), Charset.defaultCharset());

```



```

962         for (String line: lines) {
963             String[] attributes = line.split("/");
964
965             int brojStolice = Integer.parseInt(attributes[0]);
966             String imeDoktora = attributes[1];
967             String vremePocetno = attributes[2];
968             String vremeKrajnje = attributes[3];
969             String gornjiLevi = attributes[4];
970             String gornjiDesni = attributes[5];
971             String donjiLevi = attributes[6];
972             String donjiDesni = attributes[7];
973
974             RadnaStolica radnaStolica = new RadnaStolica(brojStolice, imeDoktora,
975                 vremePocetno, vremeKrajnje, gornjiLevi, gornjiDesni, donjiLevi,
donjiDesni);
976
977             this.listaStolica.add(radnaStolica);
978
979         }
980     } catch (java.io.IOException e) {
981         System.out.println("Datoteka " + path + " nije pronajena.");
982     }
983 }
984
985
986
987 public void brisanjeSvihP() {
988
989     this.listaStolica.clear();
990     saveStolica("radnaStolica.txt");
991 }
992
993
994 public void brisanjeSvihP2() {
995
996     this.listaStolica.clear();
997     saveStolica("radnaStolica2.txt");
998 }
999
1000
1001 public void brisanjeSvihP3() {
1002
1003     this.listaStolica.clear();
1004     saveStolica("radnaStolica3.txt");
1005 }
1006
1007
1008
1009 // zanimljiva metoda
1010
1011 public RadnaStolica stolicaBrisanjeID(int id) {
1012
1013     int index = -1;
1014     for (int i = 0; i < this.listaStolica.size(); i++) {
1015         if (this.listaStolica.get(i).getBrojstolice() == id) {
1016             index = i;
1017         }
1018     }
1019     if (index != -1) {
1020         RadnaStolica radnaStolica = this.listaStolica.remove(index);
1021         System.out.println("Brisanje po broju je uspesnu uradjeno");
1022         saveStolica("radnaStolica.txt");

```

```

1023         return radnaStolica;
1024     }
1025     return null;
1026 }
1027
1028 public RadnaStolica stolicaBrisanjeID2(int id) {
1029
1030     int index = -1;
1031     for (int i = 0; i < this.listaStolica.size(); i++) {
1032         if (this.listaStolica.get(i).getBrojstolice() == id) {
1033             index = i;
1034         }
1035     }
1036     if (index != -1) {
1037         RadnaStolica radnaStolica = this.listaStolica.remove(index);
1038         System.out.println("Brisanje po broju je uspesnu uradjeno");
1039         saveStolica("radnaStolica2.txt");
1040         return radnaStolica;
1041     }
1042     return null;
1043 }
1044
1045 public RadnaStolica stolicaBrisanjeID3(int id) {
1046
1047     int index = -1;
1048     for (int i = 0; i < this.listaStolica.size(); i++) {
1049         if (this.listaStolica.get(i).getBrojstolice() == id) {
1050             index = i;
1051         }
1052     }
1053     if (index != -1) {
1054         RadnaStolica radnaStolica = this.listaStolica.remove(index);
1055         System.out.println("Brisanje po broju je uspesnu uradjeno");
1056         saveStolica("radnaStolica3.txt");
1057         return radnaStolica;
1058     }
1059     return null;
1060 }
1061
1062 public UserClass korisnikBrisanjeID(int id) {
1063
1064     int index = -1;
1065     for (int i = 0; i < this.listaKorisnika.size(); i++) {
1066         if (this.listaKorisnika.get(i).getId() == id) {
1067             index = i;
1068         }
1069     }
1070     if (index != -1) {
1071         UserClass radnaStolica = this.listaKorisnika.remove(index);
1072         save("pacijenti.txt");
1073         System.out.println("Brisanje po broju je uspesnu uradjeno");
1074         return radnaStolica;
1075     }
1076     return null;
1077 }
1078
1079
1080
1081 public TerminiOrd TerminBrisanjeID(int id) {
1082
1083     int index = -1;
1084     for (int i = 0; i < this.listaTermina.size(); i++) {

```

```

1085         if (this.listaTermina.get(i).getID() == id) {
1086             index = i;
1087         }
1088     }
1089     if (index != -1) {
1090         TerminiOrd radnaStolica = this.listaTermina.remove(index);
1091         saveTermini("termini.txt");
1092         JOptionPane.showMessageDialog(null, "Brisanje po broju je uspesno uradjeno !",
1093 "Brisanje ok", 1);
1094         //System.out.println("Brisanje po broju je uspesnu uradjeno");
1095         return radnaStolica;
1096     }
1097     return null;
1098 }
1099 public TerminiOrd TerminBrisanjeImePrezime(String ime, String prezime) {
1100
1101     int index = -1;
1102     for (int i = 0; i < this.listaTermina.size(); i++) {
1103         if (this.listaTermina.get(i).getIme().equalsIgnoreCase(ime)&&
1104             this.listaTermina.get(i).getPrezime().equalsIgnoreCase(prezime)) {
1105             index = i;
1106         }
1107     }
1108     if (index != -1) {
1109         TerminiOrd radnaStolica = this.listaTermina.remove(index);
1110         saveTermini("termini.txt");
1111         JOptionPane.showMessageDialog(null, "Brisanje po imenu i prezimenu je uspesno
1112 uradjeno !", "Brisanje ok", 1);
1113         //System.out.println("Brisanje po imenu i prezimenu je uspesno uradjeno !");
1114         return radnaStolica;
1115     }
1116     return null;
1117 }
1118
1119
1120 public Cena cenaBrisanjeID(int id) {
1121
1122     loadCena("cene.txt");
1123     int index = -1;
1124     for (int i = 0; i < this.listaCena.size(); i++) {
1125         if (this.listaCena.get(i).getId() == id) {
1126             index = i;
1127         }
1128     }
1129     if (index != -1) {
1130         Cena radnaStolica = this.listaCena.remove(index);
1131         saveCena("cene.txt");
1132         JOptionPane.showMessageDialog(null, "Brisanje po broju je uspesnu uradjeno",
1133 "Brisanje uspesno uradjeno !", 1);
1134         //System.out.println("Brisanje po broju je uspesnu uradjeno");
1135         return radnaStolica;
1136     }
1137     return null;
1138 }
1139 public void lstIstoriati() {
1140
1141     loadHistoriat("Historiat.txt");
1142
1143     System.out.printf("%10s %25s %25s %25s %25s %25s %10s \n",

```

```

1144         "ID", "ime Pacijenta", "Gornji Levi Zubi", "Gornji Desni Zubi", "Donji
Levi Zubi", "Donji Desni Zubi", "Datum");
1145
1146         for(int i = 0; i < this.listaIstorije.size(); i++) {
1147
1148             Historiat historiat = this.listaIstorije.get(i);
1149
1150             System.out.println(historiat);
1151
1152         }
1153
1154     }
1155
1156     public void lstIstoriatImePrez1String(String ime) {
1157
1158         loadHistoriat("Historiat.txt");
1159
1160         System.out.printf("%10s %25s %25s %25s %25s %25s %10s \n",
1161             "ID", "ime Pacijenta", "Gornji Levi Zubi", "Gornji Desni Zubi", "Donji
Levi Zubi", "Donji Desni Zubi", "Datum");
1162
1163         for (int i = 0; i < this.listaIstorije.size(); i++) {
1164
1165             if (this.listaIstorije.get(i).getImePacijenta().equalsIgnoreCase(ime)) {
1166
1167                 Historiat historiat = this.listaIstorije.get(i);
1168                 System.out.println(historiat);
1169
1170             }
1171         }
1172     }
1173
1174     public void lstIstoriatIDString(int ID) {
1175
1176         loadHistoriat("Historiat.txt");
1177
1178         System.out.printf("%10s %25s %25s %25s %25s %25s %10s \n",
1179             "ID", "ime Pacijenta", "Gornji Levi Zubi", "Gornji Desni Zubi", "Donji
Levi Zubi", "Donji Desni Zubi", "Datum");
1180
1181         for (int i = 0; i < this.listaIstorije.size(); i++) {
1182
1183             if (this.listaIstorije.get(i).getID()== ID) {
1184
1185                 Historiat historiat = this.listaIstorije.get(i);
1186                 System.out.println(historiat);
1187
1188             }
1189         }
1190     }
1191
1192     public Historiat brisanjeIstoriatIDString(int ID) {
1193
1194         loadHistoriat("Historiat.txt");
1195         int index = -1;
1196         System.out.printf("%10s %25s %25s %25s %25s %25s %10s \n",
1197             "ID", "ime Pacijenta", "Gornji Levi Zubi", "Gornji Desni Zubi", "Donji
Levi Zubi", "Donji Desni Zubi", "Datum");
1198
1199         for (int i = 0; i < this.listaIstorije.size(); i++) {
1200             if (this.listaIstorije.get(i).getID()== ID) {
1201

```

```

1202         index = i;
1203     }
1204 }
1205 if (index != -1) {
1206     this.listaIstorije.remove(index);
1207     saveHistoriat("Historiat.txt");
1208     JOptionPane.showMessageDialog(null, "Brisanje po Id je uspesno uradjeno !",
"The Help Me BUTTON :)", 1);
1209 }else {
1210     JOptionPane.showMessageDialog(null, "Brisanje po Id je neuspesno uradjeno !",
"The Help Me BUTTON :)", 1);
1211 }
1212 return null;
1213
1214 }
1215
1216 public Historiat brisanjeIstoriatIDString2(int ID) {
1217     loadHistoriat("Historiat.txt");
1218     int index = -1;
1219     System.out.printf("%10s %25s %25s %25s %25s %25s %10s \n",
1220         "ID", "ime Pacijenta", "Gornji Levi Zubi", "Gornji Desni Zubi", "Donji
Levi Zubi", "Donji Desni Zubi", "Datum");
1222     for (int i = 0; i < this.listaIstorije.size(); i++) {
1223         if (this.listaIstorije.get(i).getID()== ID) {
1224             index = i;
1225         }
1226     }
1227     if (index != -1) {
1228         this.listaIstorije.remove(index);
1229         saveHistoriat("Historiat.txt");
1230         JOptionPane.showMessageDialog(null, "Brisanje po Id je uspesno uradjeno !",
"The Help Me BUTTON :)", 1);
1232     }else {
1233         JOptionPane.showMessageDialog(null, "Brisanje po Id je neuspesno uradjeno !",
"The Help Me BUTTON :)", 1);
1234     }
1235     return null;
1236
1237 }
1238
1239
1240
1241
1242
1243
1244 }
1245

```