

**Name : Hasnat Ahmad ,
Afaq-Ur-Rehman
Roll No : 20P-0079, 20P-0034
Course : SCD**

REFACTORING

Hasnat Ahmad

```
1+ import java.io.*;
7  /*****/
8  class MyServer
9  {
10 ArrayList al=new ArrayList();
11 ArrayList users=new ArrayList();
12 ServerSocket ss;
13 Socket s;
14

8  class MyServer
9  {
10 ArrayList al=new ArrayList();
11 ArrayList users=new ArrayList();
12 ServerSocket serverSocket;
```

Hasnat Ahmad

```
20 try{
21     ss=new ServerSocket(PORT);
22     System.out.println("Server Started "+ss);
23     while(true)
24     {
25         s=ss.accept();
26         Runnable r=new MyThread(s,al,users);
27         Thread t=new Thread(r);
28         t.start();
29         // System.out.println("Total alive clients : "+ss.);
30     }
31 }
```

```
20 try{
21     serverSocket=new ServerSocket(PORT);
22     System.out.println("Server Started "+serverSocket);
23     while(true)
24     {
25         s=serverSocket.accept();
26         Runnable r=new MyThread(s,al,users);
27         Thread t=new Thread(r);
28         t.start();
29         // System.out.println("Total alive clients : "+ss.);
30     }
31 }
```

```

13     Socket socket;
14
15     public final static int PORT=10;
16     public final static String UPDATE_USERS="updateuserslist: ";
17     public final static String LOGOUT_MESSAGE="@@logoutme@@: ";
18     public MyServer()
19     {
20         try{
21             serverSocket=new ServerSocket(PORT);
22             System.out.println("Server Started "+serverSocket);
23             while(true)
24             {
25                 socket=serverSocket.accept();
26                 Runnable r=new MyThread(socket,al,users);
27                 Thread t=new Thread(r);
28                 t.start();

```

Afaq Ahmad

```

27         Thread t=new Thread(r);
28         t.start();

```

```

27         Thread thread=new Thread(r);
28         thread.start();

```

```

25         socket=serverSocket.accept();
26         Runnable r=new MyThread(socket,al,users);
27         Thread thread=new Thread(r);
28         thread.start();

26         Runnable runnable=new MyThread(socket,al,users);
27         Thread thread=new Thread(runnable);
28         thread.start();

```

Hasnat Ahmad

```

51 Socket s;
52 ArrayList al;
53 ArrayList users;
54 String username;

```

```

51 Socket socket;
52 ArrayList al;
53 ArrayList users;
54 String username;

```

Afaq Ahmad

```

59 this.s=s;
60 this.al=al;
61 this.users=users;
62 try{
63     DataInputStream dis=new DataInputStream(s.getInputStream());
64     username=dis.readUTF();
65     al.add(s);
66     users.add(username);
67     tellEveryone("***** " + username+" Logged in at "+(new Date())+" *****");
68     sendNewUserList();
69 }
70 catch(Exception e){
71     System.err.println("MyThread constructor "+e);
72 }
73 }
74

```

```

69 MyThread (Socket socket, ArrayList al, ArrayList users){
70     this.socket=socket;
71     this.al=al;
72     this.users=users;
73     try{
74         //creating dataInputStream object which returns output stream for this socket
75         DataInputStream dataInputStream=new DataInputStream(socket.getInputStream());
76
77         //convert bytes to UTF8
78         username=dataInputStream.readUTF();
79         al.add(socket);
80
81         //add user to ArrayList
82         users.add(username);
83
84         tellEveryone("***** " + username+" Logged in at "+(new Date())+" *****");
85         sendNewUserList();
86     }
87     catch(Exception e){
88         System.err.println("MyThread constructor "+e);
89     }
90 }

```

Hasnat Ahmad

```

76 public void run()
77 {
78     String s1;
79     try{
80         DataInputStream dis=new DataInputStream(s.getInputStream());
81         do
82         {
83             s1=dis.readUTF();
84             if(s1.toLowerCase().equals(MyServer.LOGOUT_MESSAGE)) break;
85             // System.out.println("received from "+s.getPort());
86             tellEveryone(username+" said: "+" : "+s1);
87         }
88         while(true);
89         DataOutputStream tdos=new DataOutputStream(s.getOutputStream());
90         tdos.writeUTF(MyServer.LOGOUT_MESSAGE);
91         tdos.flush();
92         users.remove(username);
93         tellEveryone("***** "+username+" Logged out at "+(new Date())+" *****");
94         sendNewUserList();
95         al.remove(s);
96         s.close();
97     }
98 }
99 catch(Exception e){System.out.println("MyThread Run"+e);}
100 }

```

```

93 public void run()
94 {
95     String message;
96     try{
97         //creating dataInputStream object which will returns input stream this socket
98         DataInputStream dataInputStream=new DataInputStream(socket.getInputStream());
99         do
100         {
101             //convert bytes to UTF8
102             message=dataInputStream.readUTF();
103
104             if(message.toLowerCase().equals(MyServer.LOGOUT_MESSAGE)) break;
105             // System.out.println("received from "+s.getPort());
106             tellEveryone(username+" said: "+" : "+message);
107         }
108         while(true);
109
110         //creating dataInputStream object which will returns output stream for this socket
111         DataOutputStream dataOutputStream=new DataOutputStream(socket.getOutputStream());
112
113         //write message on socket
114         dataOutputStream.writeUTF(MyServer.LOGOUT_MESSAGE);
115         //Flush data
116         dataOutputStream.flush();
117         users.remove(username);
118         tellEveryone("***** "+username+" Logged out at "+(new Date())+" *****");
119         sendNewUserList();
120         al.remove(socket);
121
122         //close socket
123         socket.close();

```

Afaq Ahmad

```
110 public void tellEveryOne(String s1)
111 {
112     Iterator i=al.iterator();
113     while(i.hasNext())
114     {
115         try{
116             Socket temp=(Socket)i.next();
117             DataOutputStream dos=new DataOutputStream(temp.getOutputStream());
118             dos.writeUTF(s1);
119             dos.flush();
120             //System.out.println("sent to : "+temp.getPort()+" : "+ s1);
121         }
122         catch(Exception e){System.err.println("TellEveryOne "+e);}
123     }
124 }
125 ///////////////////////////////////////////////////
126 }
```

```
110 public void tellEveryOne(String message)
111 {
112     Iterator iterator=al.iterator();
113     while(iterator.hasNext())
114     {
115         try{
116             Socket temp=(Socket)iterator.next();
117             DataOutputStream dataOutputStream=new DataOutputStream(temp.getOutputStream());
118             dataOutputStream.writeUTF(message);
119             dataOutputStream.flush();
120             //System.out.println("sent to : "+temp.getPort()+" : "+ s1);
121         }
122         catch(Exception e){System.err.println("TellEveryOne "+e);}
123     }
124 }
```

Afaq Ahmad

```
128 class MyClient implements ActionListener
129 {
130 Socket s;
131 DataInputStream dis;
132 DataOutputStream dos;
133
134 JButton sendButton, logoutButton, loginButton, exitButton;
135 JFrame chatWindow;
136 JTextArea txtBroadcast;
137 JTextArea txtMessage;
138 JList usersList;
139
```

```
130 Socket socket;
131 DataInputStream dis;
132 DataOutputStream dos;
133
134 JButton sendButton, logoutButton, loginButton, exitButton;
135 JFrame chatWindow;
136 JTextArea txtBroadcast;
137 JTextArea txtMessage;
138 JList usersList;
```


Afaq Ahmad

```
202 if(s!=null)
203 {
204 JOptionPane.showMessageDialog(chatWindow,"u r logged out right now. ", "Exit",JOptionPane.INFORMATION_MESSAGE);
205 logoutSession();
206 }
207 System.exit(0);
208 }
209 });
210 }
211 //////////////////////////////////
212 public void actionPerformed(ActionEvent ev)
213 {
214 JButton temp=(JButton)ev.getSource();
215 if(temp==sendButton)
216 {
217 if(s==null)
218 {JOptionPane.showMessageDialog(chatWindow,"u r not logged in. plz login first"); return;}
219 try{
220 dos.writeUTF(txtMessage.getText());
221 txtMessage.setText("");
222 }
223 catch(Exception excp){txtBroadcast.append("\nsend button click :"+excp);}
224 }
225 if(temp==loginButton)
226 {
227 String uname=JOptionPane.showInputDialog(chatWindow,"Enter Your lovely nick name: ");
228 if(uname!=null)
229 clientChat(uname);
230 }
231 if(temp==logoutButton)
232 {
233 if(s!=null)
234 logoutSession();
235 }
```

Hasnat Ahmad

```
93 public void run()
94 {
95     String message;
96     try{
97         //creating dataInputStream object which will returns input stream this socket
98         DataInputStream dataInputStream=new DataInputStream(socket.getInputStream());
99         do
100         {
101             //convert bytes to UTF8
102             message=dataInputStream.readUTF();
103
104             if(message.toLowerCase().equals(MyServer.LOGOUT_MESSAGE)) break;
105             // System.out.println("received from "+s.getPort());
106             tellEveryone(username+" said: "+" : "+message);
107         }
108         while(true);
109
110         //creating dataInputStream object which will returns output stream for this socket
111         DataOutputStream dataOutputStream=new DataOutputStream(socket.getOutputStream());
112
113         //write message on socket
114         dataOutputStream.writeUTF(MyServer.LOGOUT_MESSAGE);
115         //Flush data
116         dataOutputStream.flush();
117         users.remove(username);
118         tellEveryone("***** "+username+" Logged out at "+(new Date())+" *****");
119         sendNewUserList();
120         al.remove(socket);
121
122         //close socket
123         socket.close();
124     }
125 }
```

Hasnat Ahmad

```
264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dos=new DataOutputStream(socket.getOutputStream());
268     ClientThread ct=new ClientThread(dataInputStream,this);
269     Thread t1=new Thread(ct);
270     t1.start();
271     dos.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }
```

```
264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dataOutputStream=new DataOutputStream(socket.getOutputStream());
268     ClientThread ct=new ClientThread(dataInputStream,this);
269     Thread t1=new Thread(ct);
270     t1.start();
271     dataOutputStream.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }
```

Afaq Ahmad

```
264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dos=new DataOutputStream(socket.getOutputStream());
268     ClientThread ct=new ClientThread(dataInputStream,this);
269     Thread t1=new Thread(ct);
270     t1.start();
271     dos.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }
```

```
264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dataOutputStream=new DataOutputStream(socket.getOutputStream());
268     ClientThread clientthread=new ClientThread(dataInputStream,this);
269     Thread t1=new Thread(clientthread);
270     t1.start();
271     dataOutputStream.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }
```

Hasnat Ahmad

```
264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dos=new DataOutputStream(socket.getOutputStream());
268     ClientThread ct=new ClientThread(dataInputStream,this);
269     Thread t1=new Thread(ct);
270     t1.start();
271     dos.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }

264 try{
265     socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266     dataInputStream=new DataInputStream(socket.getInputStream());
267     dataOutputStream=new DataOutputStream(socket.getOutputStream());
268     ClientThread clientthread=new ClientThread(dataInputStream,this);
269     Thread thread=new Thread(clientthread);
270     thread.start();
271     dataOutputStream.writeUTF(uname);
272     chatWindow.setTitle(uname+" Chat Window");
273 }
```

Afaq Ahmad

```
249 if(s==null) return;
250 try{
251 dos.writeUTF(MyServer.LOGOUT_MESSAGE);
252 Thread.sleep(500);
253 s=null;
254 }
255 catch(Exception e){txtBroadcast.append("\n inside logoutSession Method"+e);}
256
257 logoutButton.setEnabled(false);
258 loginButton.setEnabled(true);
259 chatWindow.setTitle("Login for Chat");
260 }
261 ///////////////////////////////////////////////////
262 public void clientChat(String uname)
263 {
264 try{
265 s=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266 dis=new DataInputStream(s.getInputStream());
267 dos=new DataOutputStream(s.getOutputStream());
268 ClientThread ct=new ClientThread(dis,this);
269 Thread t1=new Thread(ct);
270 t1.start();
271 dos.writeUTF(uname);
272 chatWindow.setTitle(uname+" Chat Window");
273 }
274 catch(Exception e){txtBroadcast.append("\nClient Constructor " +e);}
275 logoutButton.setEnabled(true);
276 loginButton.setEnabled(false);
277 }
.....
262 public void clientChat(String username)
263 {
264 try{
265 socket=new Socket(InetAddress.getLocalHost(),MyServer.PORT);
266 dataInputStream=new DataInputStream(socket.getInputStream());
267 dataOutputStream=new DataOutputStream(socket.getOutputStream());
268 ClientThread clientthread=new ClientThread(dataInputStream,this);
269 Thread thread=new Thread(clientthread);
270 thread.start();
271 dataOutputStream.writeUTF(username);
272 chatWindow.setTitle(username+" Chat Window");
273 }
274 catch(Exception e){txtBroadcast.append("\nClient Constructor " +e);}
275 logoutButton.setEnabled(true);
276 loginButton.setEnabled(false);
277 }
```

Afaq Ahmad

```

296 class ClientThread implements Runnable
297 {
298     DataInputStream dis;
299     MyClient client;
300
301 ClientThread(DataInputStream dis, MyClient client)
302 {
303     this.dis=dis;
304     this.client=client;
305 }
.....
298 DataInputStream dataInputStream;
299 MyClient client;
300
301 ClientThread(DataInputStream dataInputStream, MyClient client)
302 {
303     this.dataInputStream=dataInputStream;
304     this.client=client;
305 }
-----

```

Hasnat Ahmad

```
307 public void run()
308 {
309     String s2="";
310     do
311     {
312         try{
313             s2=dis.readUTF();
314             if(s2.startsWith(MyServer.UPDATE_USERS))
315                 updateUsersList(s2);
316             else if(s2.equals(MyServer.LOGOUT_MESSAGE))
317                 break;
318             else
319                 client.txtBroadcast.append("\n"+s2);
320             int lineOffset=client.txtBroadcast.getLineStartOffset(client.txtBroadcast.getLineCount()-1);
321             client.txtBroadcast.setCaretPosition(lineOffset);
322         }
323         catch(Exception e){client.txtBroadcast.append("\nClientThread run : "+e);}
324     }
325     while(true);
326 }

307 public void run()
308 {
309     String message="";
310     do
311     {
312         try{
313             message=dataInputStream.readUTF();
314             if(message.startsWith(MyServer.UPDATE_USERS))
315                 updateUsersList(message);
316             else if(message.equals(MyServer.LOGOUT_MESSAGE))
317                 break;
318             else
319                 client.txtBroadcast.append("\n"+message);
320             int lineOffset=client.txtBroadcast.getLineStartOffset(client.txtBroadcast.getLineCount()-1);
321             client.txtBroadcast.setCaretPosition(lineOffset);
322         }
323         catch(Exception e){client.txtBroadcast.append("\nClientThread run : "+e);}
324     }
325     while(true);
326 }
```


DEFENSIVE PROGRAMMING

Hasnat Ahmad

```
253 if(temp==loginButton)
254 {
255 String username=JOptionPane.showInputDialog(chatWindow,"Enter Your lovely nick name: ");
256 |
257 if(username!=null)
258     clientChat(username);
259 }
```

```
253 if(temp==loginButton)
254 {
255
256 String username=JOptionPane.showInputDialog(chatWindow,"Enter Your lovely nick name: ");
257
258 JFrame f;
259 f=new JFrame();
260 try
261 {
262     if(username.length()==0) {
263         JOptionPane.showMessageDialog(f,"UserName Cannot be Empty");
264         return ;
265     }
266
267     Integer.parseInt(username);
268     JOptionPane.showMessageDialog(f,"UserName Cannot be Integer");
269     return ;
270 }
271 catch (NumberFormatException e)
272 {
273     System.out.println();
274 }
275 }
```

Hasnat Ahmad

```
242 public void actionPerformed(ActionEvent ev)
243 {
244     JButton temp=(JButton)ev.getSource();
245     if(temp==sendButton)
246     {
247
248
249
250     if(socket==null)
251         {JOptionPane.showMessageDialog(chatWindow,"u r not logged in. plz login first"); return;
252         }
253     try{
254         dataOutputStream.writeUTF(txtMessage.getText());
255         txtMessage.setText("");
256     }
257     catch(Exception excp){txtBroadcast.append("\nsend button click :"+excp);}
258 }
259 if(temp==loginButton)
260 {
261     String username=JOptionPane.showInputDialog(chatWindow,"Enter Your lovely nick name: ");
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
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
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Afaq Ahmad

```
290 if(username!=null)
291
292     clientChat(username);
293 }

290 if(username!=null)
291     sendButton.setEnabled(true);
292     clientChat(username);
```

Afaq Ahmad

```
310 public void logoutSession()
311 {
312     if(socket==null) return;
313     try{
314         dataOutputStream.writeUTF(MyServer.LOGOUT_MESSAGE);
315         Thread.sleep(500);
316         socket=null;
317     }
318     catch(Exception e){txtBroadcast.append("\n inside logoutSession Method"+e);}
319
320     logoutButton.setEnabled(false);
321     loginButton.setEnabled(true);
322     chatWindow.setTitle("Login for Chat");
323 }
324 //////////////////////////////////////

310 public void logoutSession()
311 {
312     if(socket==null) return;
313     try{
314         dataOutputStream.writeUTF(MyServer.LOGOUT_MESSAGE);
315         Thread.sleep(500);
316         socket=null;
317     }
318     catch(Exception e){txtBroadcast.append("\n inside logoutSession Method"+e);}
319
320     logoutButton.setEnabled(false);
321     loginButton.setEnabled(true);
322     sendButton.setEnabled(false);
323     chatWindow.setTitle("Login for Chat");
324 }
```

Hasnat Ahmad

```
216 sendButton.addActionListener(this);
217 logoutButton.addActionListener(this);
218 loginButton.addActionListener(this);
219 exitButton.addActionListener(this);
220 logoutButton.setEnabled(false);
221 loginButton.setEnabled(true);
222 txtMessage.addFocusListener(new FocusAdapter()

216 sendButton.addActionListener(this);
217 logoutButton.addActionListener(this);
218 loginButton.addActionListener(this);
219 exitButton.addActionListener(this);
220 sendButton.setEnabled(false);
221 logoutButton.setEnabled(false);
222 loginButton.setEnabled(true);
223 txtMessage.addFocusListener(new FocusAdapter()
```

Comments have been added in file.