```
1
2 import java.io.*;
4
5 public class UserData {
6     private ArrayList<User> users = new ArrayList<User>();
7     private User currentUser;
8     private File file = new File("data.save");
9     private boolean isLogin = false;
10
11     public UserData() {
12         InputStreamReader read;
13         try {
14             read = new InputStreamReader(new
  FileInputStream(file));
15             BufferedReader bufferedReader = new
  BufferedReader(read);
16             String lineT = null;
17             while ((lineT = bufferedReader.readLine()) != null)
  {
18                 String[] str = lineT.split(" ");
19                 User userReg = new User(str[0], str[1]);
20                 userReg.setScore(Integer.parseInt(str[2]));
21                 users.add(userReg);
22             }
23         } catch (FileNotFoundException e) {
24             e.printStackTrace();
25         } catch (NumberFormatException e) {
26             e.printStackTrace();
27         } catch (IOException e) {
28             e.printStackTrace();
29         }
30     }
31
32     public void saveScore(int score) {
33         try {
34             FileWriter fileWriter = new FileWriter(file);
35             BufferedWriter bufferWriter = new
  BufferedWriter(fileWriter);
36             for (User user : users) {
37                 if (user.getUserName() ==
  currentUser.getUserName()) {
38                     if (user.getScore() < score)
```

```
39                          user.setScore(score);
40                      }
41                      bufferWriter.write(user.getUserName() + " " +
  user.getPassword() + " " + user.getScore() + "\n");
42                  }
43                  bufferWriter.close();
44                  fileWriter.close();
45              } catch (IOException e) {
46                  e.printStackTrace();
47              }
48          }
49
50          public String getName(int index) {
51              return users.get(index).getUserName();
52          }
53
54          public User[] getList() {
55              ArrayList<User> usersTemp = users;
56              for (int i = 0; i < usersTemp.size() - 1; ++i) {
57                  for (int j = i; j < usersTemp.size() - 1 - i; ++j) {
58                      if (usersTemp.get(j).getScore() <
  usersTemp.get(j + 1).getScore()) {
59                          User temp = usersTemp.get(j);
60                          usersTemp.set(j, usersTemp.get(j + 1));
61                          usersTemp.set(j + 1, temp);
62                      }
63                  }
64              }
65              int rankSize = usersTemp.size() <= 10 ? usersTemp.size()
  : 10;
66              User[] userList = new User[rankSize];
67              for (int i = 0; i < rankSize; i++)
68                  userList[i] = usersTemp.get(i);
69              return userList;
70          }
71
72          public boolean register(User user) {
73              for (User u : users)
74                  if (u.getUserName().equals(user.getUserName())) {
75                      return false;
76                  }
77              try {
```

```
78              users.add(user);
79              FileWriter fileWriter = new FileWriter(file, true);
80              BufferedWriter bufferWriter = new
  BufferedWriter(fileWriter);
81              bufferWriter.write(user.getUserName() + " " +
  user.getPassword() + " " + user.getScore() + "\n");
82              bufferWriter.close();
83              fileWriter.close();
84          } catch (IOException e) {
85              e.printStackTrace();
86          }
87          return true;
88      }
89
90      public boolean login(String userName, String password) {
91          boolean flag = false;
92          for (User user : users) {
93              if (userName.equals(user.getUserName()) &&
  password.equals(user.getPassword())) {
94                  flag = true;
95                  currentUser = user;
96                  isLogin = true;
97                  break;
98              }
99          }
100         return flag;
101     }
102
103     public boolean isLogin() {
104         return isLogin;
105     }
106 }
107
```