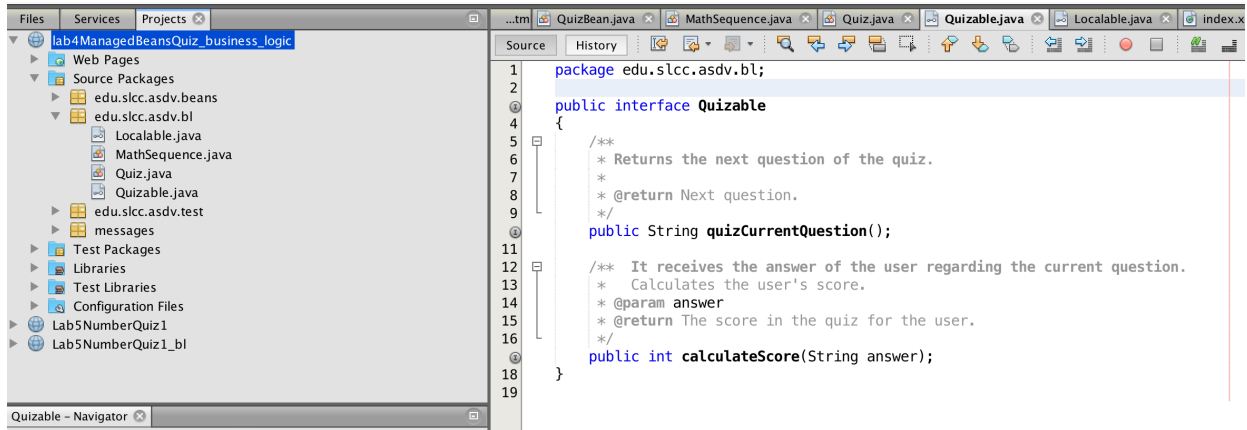


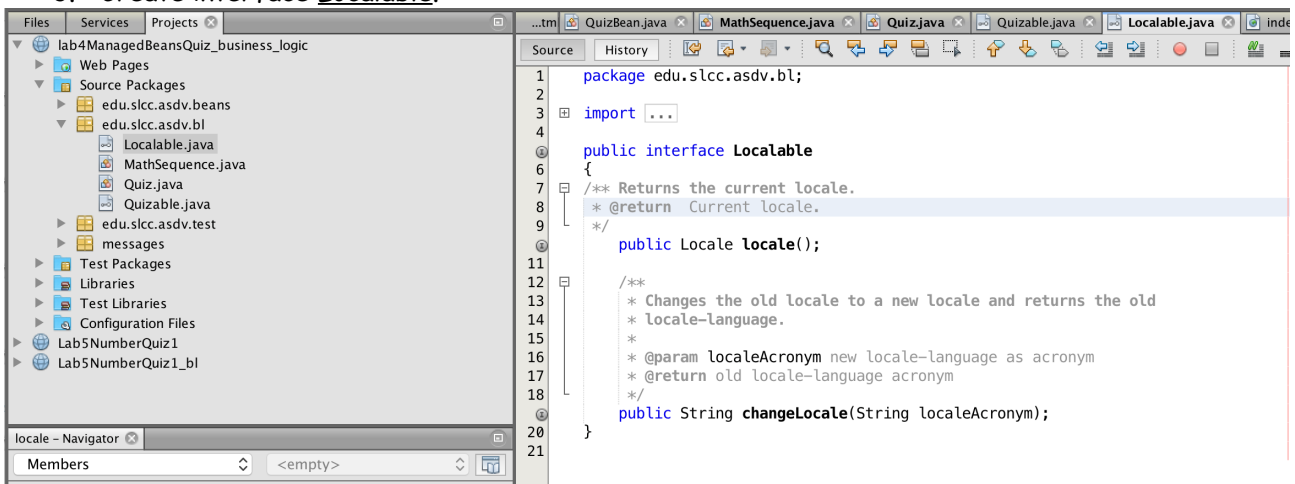
ASDV 2620, Web App II

Modularity(Isolation) of Business Logic in Web Apps

1. Copy and paste the Quiz project of chapter 2 of the textbook (Horstmann). Rename by adding to existing name, "BusinessLogic".
2. Create the package `edu.slcc.asdv.bl` and interface `Quizable`.



3. Create interface `Localable`.



- Rename the existing class Problem into MathSequence as shown. Use List, not ArrayList. The reason for List is that if we change later the implementation of MathSequence from ArrayList to another Collection type we do not have to change the classes that use MathSequence.

```

1 package edu.slcc.asdv.bl;
2 import java.io.Serializable;
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class MathSequence implements Serializable
7 {
8     private List<Integer> sequence;
9     private int solution;
10
11     public MathSequence(int[] values, int solution)
12     {
13         sequence = new ArrayList<Integer>();
14         for (int i = 0; i < values.length; i++)
15         {
16             sequence.add(values[i]);
17         }
18         this.solution = solution;
19     }
20
21     public List<Integer> getSequence(){return sequence;}
22
23     public void setSequence(List<Integer> newValue){sequence = newValue;}
24
25     public int getSolution(){return solution;}
26
27     public void setSolution(int newValue){solution = newValue;}
28 }
29

```

- Implement the business logic in class Quiz. Most of the implementation is transferring the existing code from the bean QuizBean into Quiz.

```

1 package edu.slcc.asdv.bl;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5 import java.util.List;
6 import java.util.Locale;
7 import javax.faces.component.UIViewRoot;
8 import javax.faces.context.FacesContext;
9
10 public class Quiz
11     implements Quizable,
12     Localable,
13     Serializable
14 {
15     private List<MathSequence> problems = new ArrayList<MathSequence>();
16     private int currentIndex = 0;
17     private int score = 0;
18     private String userCurrentAnswer = "";
19
20     public Quiz()
21     {
22         //...23 lines
23     }
24     public int getScore(){return score;}
25     public void setScore(int score) {this.score = score;}
26     public String getUserCurrentAnswer(){return userCurrentAnswer;}
27     public void setUserCurrentAnswer(String userCurrentAnswer){this.userCurrentAnswer = userCurrentAnswer;}
28
29     @Override
30     public String quizCurrentQuestion()
31     {
32         //...4 lines
33     }
34
35     @Override
36     public int calculateScore(String answer)
37     {
38         //...13 lines
39     }
40
41     @Override
42     public Locale locale()
43     {
44         //...3 lines
45     }
46
47     @Override
48     public String changeLocale(String localeAcronym)
49     {
50         //...8 lines
51     }
52 }
53

```

- Modify your existing `QuizBean` by taking out its business logic and creating a place holder for the business logic (line 15) of type `Quiz`. `QuizBean`'s complete implementation is shown below:

The screenshot shows an IDE with the following components:

- Project Explorer (Left):** Shows a project structure with packages like `edu.slcc.asdv.beans`, `edu.slcc.asdv.bl`, and `edu.slcc.asdv.test`. The `QuizBean.java` file is selected.
- Members (Bottom Left):** Lists the methods of the `QuizBean` class, including `clone()`, `equals()`, `finalize()`, `getAnswer()`, `getClass()`, `getQuestion()`, `getScoreOut()`, `hashCode()`, `notify()`, `notifyAll()`, `setAnswer()`, `toString()`, and `wait()`.
- Source Editor (Right):** Displays the source code of `QuizBean.java`. The code is as follows:


```

1 package edu.slcc.asdv.beans;
2
3 import edu.slcc.asdv.bl.Quiz;
4 import java.io.Serializable;
5 import javax.inject.Named;
6 import javax.enterprise.context.SessionScoped;
7
8 @Named (value="quizBean")
9 @SessionScoped
10 public class QuizBean implements Serializable
11 {
12     private int scoreOut; //displayed in JSF index
13     private String answer; // answer the user entered
14     private String question; //question( MathSequence) displayed in JSF index
15     Quiz quiz; // placeholder for Bussiness Logic
16
17     public QuizBean()
18     {
19         quiz = new Quiz();
20     }
21     public String getQuestion()
22     {
23         question = quiz.quizCurrentQuestion();
24         return question;
25     }
26     public String getAnswer()
27     {
28         return answer;
29     }
30
31     public void setAnswer(String answer)
32     {
33         this.answer = answer;
34         scoreOut = quiz.calculateScore(answer);
35     }
36
37     public int getScoreOut()
38     {
39         return scoreOut;
40     }
41
42 }

```

- Modify you `index.xhtml`, so its EL uses the new properties and methods of the `QuizBean`.
Clean build and run.