Collaborative development exercise

- Further develop a Figure editor available here: https://github.com/LASER-UMASS/cs520-Spring2020.git
- Form pairs that will collaboratively work on specification, design, implementation, and testing

Figure editor (v1): MVC architecture

Separates data representation (Model), visualization (View), and client interaction (Controller)
Figure editor (v1): Model design

<table>
<thead>
<tr>
<th>Method</th>
<th>Method and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>setCaption(java.lang.String newCaption) Sets the caption to the given non-null String.</td>
</tr>
<tr>
<td>void</td>
<td>setImage(java.awt.Image newImage) Sets the image to the given non-null Image.</td>
</tr>
</tbody>
</table>

NOTE: The violation state is implied (not shown).

Model (v1): Specification

• Each pair collaboratively works on the behavioral specification (written as an FSA)

![FSM Diagram]

NOTE: The violation state is implied (not shown).

Figure editor (v2): MVC architecture

Separates data representation (Model), visualization (View), and client interaction (Controller).

Model (v1): Implementation and testing

• Use the Model class diagram and FSA to do the following:
  – One developer implements the model’s class
  – One tester implements its test suite
  – Then integrate

![Class Diagram]
Figure editor (v1): Observer pattern

```java
Observable (abstract)
- observers: Set<Observer>
  + register(o: Observer)
  + unregister(o: Observer)
  + stateChanged()

MyObservable
- state: State
  + getState(): State
  + setState(state: State)

Observer
- <<interface>>
  + update()

MyObserver
- state: State
  + update()
```

Figure editor (v2): Implementation

- Use the Java PropertyChangeSupport class: https://docs.oracle.com/javase/7/docs/api/java/beans/PropertyChangeSupport.html
- Key steps:
  - Modify the Model class to have to have a propertyChangeSupport field and use its methods
  - Modify the View class to be a PropertyChangeListener and implement the propertyChange method
  - Modify the Controller class to only invoke the Model class methods

```
Figure editor (v2): Implementation

Applications:
- Use the Java PropertyChangeSupport class: https://docs.oracle.com/javase/7/docs/api/java/beans/PropertyChangeSupport.html
- Key steps:
  - Modify the Model class to have to have a propertyChangeSupport field and use its methods
  - Modify the View class to be a PropertyChangeListener and implement the propertyChange method
  - Modify the Controller class to only invoke the Model class methods
```
Figure editor (v2): Testing

- Manually test that after the user appropriately sets the image or caption the model and its view are appropriately updated
- If needed, modify the JUnit test suite until all tests pass again

Topics covered

- Documentation, e.g.,
  - README, javadoc, internal comments
- Specification, e.g.,
  - Natural language, FSAs, test suite
- Architecture & design, e.g.,
  - Patterns (MVC, Observer, Adapter)
  - Class diagrams
- Implementation
  - Individual & pair programming
  - Java, Swing
- Testing
  - Unit, integration, system
  - JUnit

Final project: Topic selection

- Form team of 4 or 5 students
- Select one of the following 4 topics:
  1. MSR 2020 mining challenge
  2. Replication study
  3. Model inference for inferring processes (i.e. specification mining)
  4. EleNa: Elevation-based navigation
- Due: Thursday March 5, 2020 12 PM EST

https://people.cs.umass.edu/~hconboy/class/2020Spring/CS520/finalProject.pdf