Diagram Test

1. public void addRelationship()
    a. Testing that our addRelationship() method in Diagram.java adds a Relation r to the list of all relationships in the diagram
    b. Passes
2. public void deleteRelationship()
    a. Testing that our deleteRelationship() method in Diagram.java deletes a Relation r from the list of all relationships in the diagram
    b. Passes
3. public void addClass()
    a. Testing that our addClass() method in Diagram.java adds a ClassComponent c to the list of all classes in the diagram
    b. Passes
4. public void deleteClass()
    a. Testing that our deleteClass() method in Diagram.java deletes a ClassComponent c from the list of all classes in the diagram
    b. Passes
5. public void getAllClasses()
    a. Testing that our getAllClasses() method in Diagram.java returns the list of all classes in the diagram
    b. Passes
6. public void getAllRelationships()
    a. Testing that our getAllRelationships() method in Diagram.java returns the list of all relationships in the diagram
    b. Passes

Relation Test

1. Public Void construct()
    a. Testing that after a relation is added, the source and destination components of the relation are assigned correctly.
    b. Passes
2. public void componentTest1()
    a. Testing that when a Relation of type Subtyping is added that's it's type really is.
    b. Passes
3. public void componentTest2()
    a. Testing that when a Relation of type Delegation is added that's it's type really is.
    b. Passes
4. public void componentTest3()
    a. Testing that when a Relation of type Containment is added that's it's type really is.

b. Passes

Class Test

1. public void testName()
    a. Testing with getter if String classname can be added for DesignClass and DesignInterface
    b. Passes
2. public void testPosition()
    a. Testing with with getter if Point position can be added for DesignClass and DesignInterface
    b. Passes
3. public void testSter()
    a. Testing with getter if String stereotype can be added for DesignClass and DesignInterface
    b. Passes
4. public void testMethods()
    a. Testing with getter if String method can be added for DesignClass and DesignInterface
    b. Passes
5. public void testinstanceVariables()
    a. Testing with getter if String instanceVariables can be added for DesignClass

DiagramView overview

1. We coded the DiagramView in small chunks and used the "gradle run" command very often to check how DiagramView was working
2. Used Zoom's share screen feature to show each other what the DiagramView class looked like in the editor after making local changes
3. By doing those two things we know that our diagram editor successfully:
    a. Displays the top menu bar with "file" → "open", "save as", and "export" options as well as "edit" → "undo" and "redo" options
    b. Displays the right-hand side toolbar with "add class", "add design class", "add interface", "add stereotype", "add delegation", "add containment", and "add subtype" buttons
    c. When a user selects "add class" or "add design class" a pop up window appears where they can enter the name of the class and then the class rectangle appears
    d. If the user clicks and drags within the top left hand corner of the class rectangle, the class can move around within the window