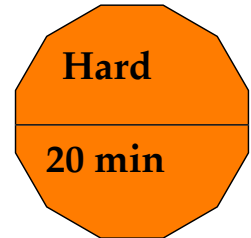
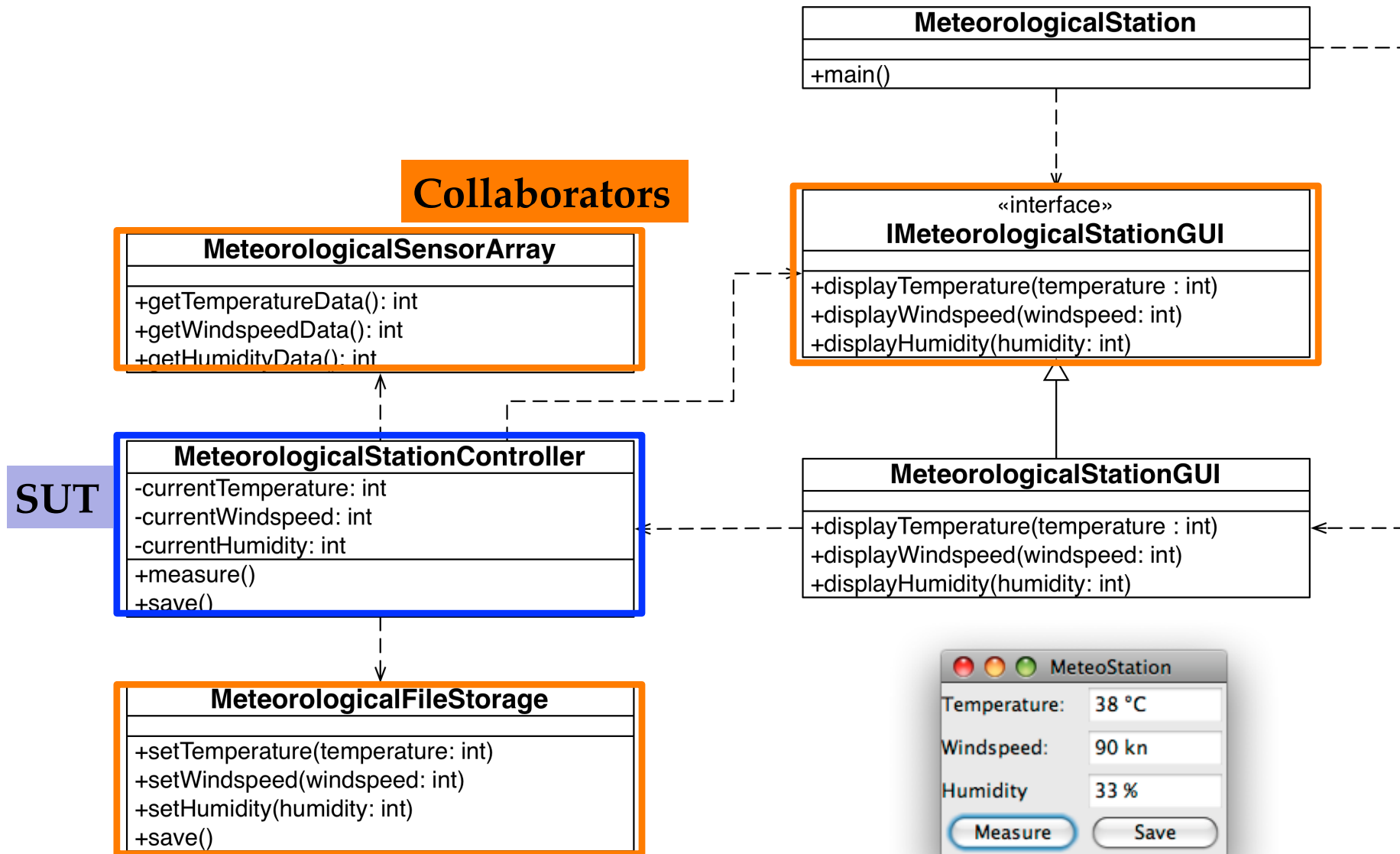


# Task #2 (In-Class Exercise) using JUnit, Easy Mock and Guice

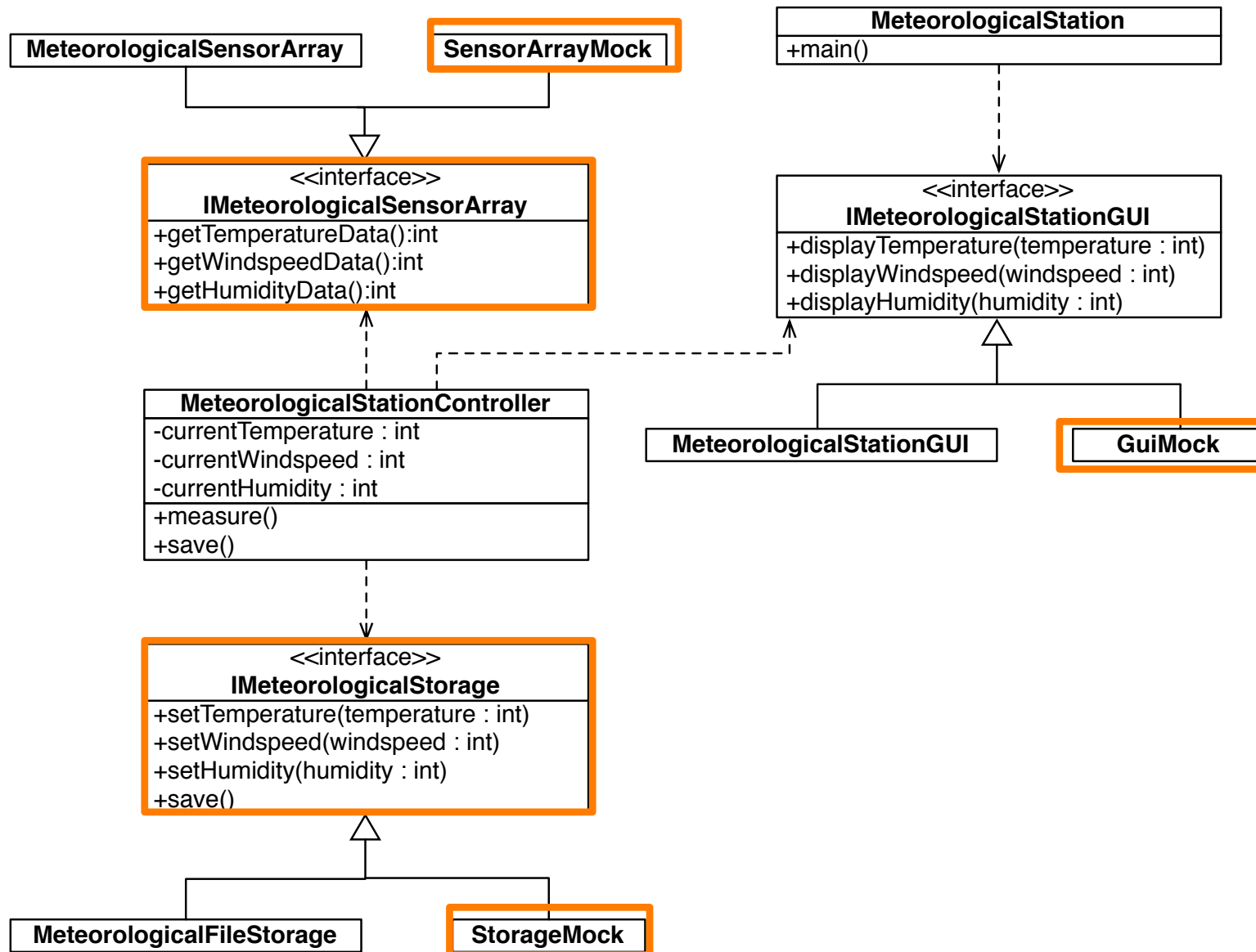


- Task #2:
  - Use the refactored code from Task #1 to unit test **MeteorologicalStationController** (both the `measure()` and the `save()` method)
  - Use EasyMock to mock the collaborators of `MeteorologicalStationController` (remember to introduce interfaces where necessary)
  - Create a "Test" Module to configure the test bindings
- Post your solution in Moodle.

# Meteorological Station Application Model



# Introduce Interfaces and Create Mock



# Solution to Task#2 Test Module

```
import org.easymock.EasyMock;  
import com.google.inject.Binder;  
import com.google.inject.Module;  
...
```

```
public class TestModule implements Module {  
    public void configure(Binder binder) {
```

create mock  
object

bind mock  
object

```
        IMeteorologicalStationGUI guiMock =  
        EasyMock.createMock(IMeteorologicalStationGUI.class);  
        binder.bind(IMeteorologicalStationGUI.class).toInstance(guiMock);
```

```
        IMeteorologicalStorage storageMock =  
        EasyMock.createMock(IMeteorologicalStorage.class);  
        binder.bind(IMeteorologicalStorage.class).toInstance(storageMock);
```

```
        IMeteorologicalSensorArray sensorArrayMock =  
        EasyMock.createMock(IMeteorologicalSensorArray.class);  
        binder.bind(IMeteorologicalSensorArray.class).toInstance(sensorArrayMock);
```

```
    }
```

```
}
```

# Solution to Task#2

## MeteorologicalStationTest

```
import org.easymock.EasyMock;
import org.junit.Before;
import org.junit.Test;
import com.google.inject.Guice;
import com.google.inject.Injector;
...
public class MeteorologicalStationTest {
```

```
    private static final int testHumidity = 42;
    private static final int testWindspeed = 0;
    private static final int testTemperature = 13;
    private Injector injector;
    private MeteorologicalStationController controller;
```

define test oracles

```
@Before
```

```
    public void setUp() {
        injector = Guice.createInjector(new TestModule());
        controller = injector.getInstance(MeteorologicalStationController.class);
    }
```

Instantiate an  
Injector and tell it  
which module to use

Instantiate an  
instance of the class  
needing injection

# Solution to Task#2

## MeteorologicalStationTest – testMeasure()

```
...  
public class MeteorologicalStationTest {  
...  
    @Test  
    public void testMeasure() {  
        IMeteorologicalSensorArray sensorArray =  
injector.getInstance(IMeteorologicalSensorArray.class);  
  
        EasyMock.expect(sensorArray.getTemperatureData()).andReturn(testTemperature);  
  
        EasyMock.expect(sensorArray.get WindspeedData()).andReturn(test Windspeed);  
  
        EasyMock.expect(sensorArray.getHumidityData()).andReturn(testHumidity);  
        EasyMock.replay(sensorArray);  
        controller.measure();  
        EasyMock.verify(sensorArray);  
    }  
...  
}
```

JUnit test case

Record expected behaviors of the mock object

Replay recorded behaviors

Execute the SUT

Verify that the specified behavior has been used