Antonio Scalfaro

CMSC 215 / 6381

Assignment 4 – Time Interval Checker

10/09/2023

**UML Diagram**

|  |
| --- |
| **Time** |
| hours:int  minutes:int  meridian:String |
| Time()  Time()  getHours()  getMinutes()  getMeridian()  to24Hours()  compareTo()  toString()  validateTime() |

|  |
| --- |
| **InvalidTime extends Exception** |
| message:String |
| InvalidTime()  getMessage() |

|  |
| --- |
| **Interval<TypeTime extends Time>** |
| start:TypeTime  end:TypeTime |
| Interval()  getStart()  getEnd()  within()  subinterval()  overlaps() |

|  |
| --- |
| **Project4 extends Application(JavaFX)** |
| grid:GridPane  scene:Scene  int1:Interval<Time>  int2:Interval<Time> |
| start()  createNumericTextField()  isValidInput()  main() |

|  |
| --- |
| **Project4\_UnitTest (junit)** |
|  |
| testInterval()  testTimeTo24Hours()  testTimeToString()  testTimeCompareTo() |

**Test Plan:**

Testing the Time Interval Checker requires both JUnit tests and manual testing of the GUI. The JUnit tests are as follows:

testInterval(): Tests the creation of Interval objects, then tests the Interval class methods. The within(), subinterval(), and overlaps() methods are tested. All tests pass.

testTimeTo24Hours(): Creates two Time objects, and converts them to 24 hour clock using the Time to24Hours() class method. All tests pass.

testTimeToString(): Tests the toString class method for Time and also tests the InvalidTime Exception class by creating an invalid Time and catching the exception. All tests pass.

testTimeCompareTo(): Tests the Time compareTo() class method, all tests pass.

The rest of the testing is done manually in the GUI. I test to ensure that an alert is sent to the GUI if an incorrectly formatted or blank textField is used. The buttons function and their event listeners work as designed. The correct answer is displayed in the output textField when the correct data is input and depending on which button is pressed.

**Lessons:**

There were many valuable learning experiences during this week’s assignment. First, continuing to learn the basics of JavaFX is very valuable and the more comfortable I get, the more functionality I can add. I had some trouble with making the buttons span across 3 columns but had no issue with the TextFields, so there is some investigating to still do there. I also had logical issues with calculating within() and subinterval() with certain times. I found that keeping the data in the 12 hour clock format made it very difficult to get the correct output when testing around 12 PM and 12 AM. This lead me to abandon the compareTo method for the most part and implement a to24Hours() class method that made the comparisons easier and gave the correct answer more often.