TESTING

Product name: UCSC Bus Buddy
Team name: SPARC
Date: 12/02/2013

1) Module: getBusTimes (String busNumber)
From file: BusStop.java

Equivalence class 1: “10”
Equivalence class 2: “15” – “16”
Equivalence class 3: “19” – “20”
Equivalence class 4: “nc”
Equivalence class 5: Any other String sequence

Tests Run:
EC1:
- Input “10” -> Test is successful

EC2:
- Input “15” -> Test is successful
- Input “16” -> Test is successful

EC3:
- Input “19” -> Test is successful
- Input “20” -> Test is successful

EC4:
- Input “nc” -> Test is successful

EC5:
- Input null -> Test is successful
- Input “11” -> Test is successful
- Input “oc” -> Test is successful
- Input “” -> Test is successful

2) Module: setTime (String time)
From file: BusStop.java

Equivalence class 1: Times representing in Strings between 00:00 and 23:59
Equivalence class 2: Any other String

Tests Run:
EC1:
- Input “01:30” -> Test is successful
- Input “13:45” -> Test is successful
- Input “00:00” -> Test is successful
- Input “23:59” -> Test is successful

EC2:

- Input null -> Test is successful
- Input “123:45” -> Test is successful
- Input “12:456” -> Test is successful
- Input “1:30” -> Test is successful
- Input “-1:45” -> Test is successful
- Input “25:45” -> Test is successful
- Input “26:62” -> Test is successful
- Input “null” -> Test is successful
- Input “” -> Test is successful

3) Module: addBusTime (String busNumber, Calendar busTime)
From file: MainActivity.java

Equivalence class 1: “10” with any busTime != null
Equivalence class 2: “15” – “16” with any busTime != null
Equivalence class 3: “19” – “20” with any busTime != null
Equivalence class 4: “nc” with any busTime != null
Equivalence class 5: Any other String sequence with any busTime input

Tests Run:
EC1:

- Input “10” with any busTime != null -> Test is successful

EC2:

- Input “15” with any busTime != null -> Test is successful
- Input “16” with any busTime != null -> Test is successful

EC3:

- Input “19” with any busTime != null -> Test is successful
- Input “20” with any busTime != null -> Test is successful

EC4:

- Input “nc” with any busTime != null -> Test is successful

EC5:

- Input null with any busTime != null -> Test is successful
- Input “10” with any busTime != null -> Test is successful
- Input “12” with any busTime != null -> Test is successful
- Input “ac” -> Test is successful
- Input “” with any busTime != null -> Test is successful
- Input null with busTime = null -> Test is successful
- Input “10” with busTime = null -> Test is successful
- Input “ac” with busTime = null -> Test is successful
- Input “” with busTime = null -> Test is successful

3) Module: printTime (Calendar time)
From file: BusStop.java

Equivalence class 1: Calendar objects with fields int HOUR_OF_DAY = 0-23
int MINUTE = 0-59
Equivalence class 2: Calendar objects set to null

Tests Run:
EC1:

- Calendar object: Calendar.HOUR_OF_DAY = 1
  Calendar.MINUTE = 30 -> Test is successful
- Calendar object: Calendar.HOUR_OF_DAY = 13
  Calendar.MINUTE = 45 -> Test is successful
- Calendar object: Calendar.HOUR_OF_DAY = 0
  Calendar.MINUTE = 0 -> Test is successful
- Calendar object: Calendar.HOUR_OF_DAY = 23
  Calendar.MINUTE = 59 -> Test is successful

EC2:

- Calendar object = null -> Test is successful

```
package com.example.ucscbusbuddy.test;

import java.util.Calendar;

import org.junit.Before;
import org.junit.Test;

import android.test.AndroidTestCase;

public class BusBuddyTest extends AndroidTestCase {

    public BusBuddyTest() {
        super();
    }

    @Before
    protected void setUp() throws Exception {
        super.setUp();
    }

    @Test
    public void testGetBusTimesValid() throws Exception {
        BBTestObject testObject = new BBTestObject ();
        assertNotNull (testObject.isAListOfTimes("10"));
        assertNotNull (testObject.isAListOfTimes("15"));
        assertNotNull (testObject.isAListOfTimes("16"));
        assertNotNull (testObject.isAListOfTimes("19"));
        assertNotNull (testObject.isAListOfTimes("20"));
        assertNotNull (testObject.isAListOfTimes("nc"));
    }

    @Test
    public void testGetBusTimesInvalid() throws Exception {
        BBTestObject testObject = new BBTestObject ();
        assertNull (testObject.isAListOfTimes(null));
        assertNull (testObject.isAListOfTimes("11"));
        assertNull (testObject.isAListOfTimes("oc"));
        assertNull (testObject.isAListOfTimes(""));
    }

    @Test
    public void testSetTimeValid () throws Exception {
        BBTestObject testObject = new BBTestObject ();
        assertNotNull (testObject.isAValidTime("01:30"));
        assertNotNull (testObject.isAValidTime("13:45"));
        assertNotNull (testObject.isAValidTime("00:00"));
        assertNotNull (testObject.isAValidTime("23:59"));
    }

    @Test
    public void testSetTimeInvalid () throws Exception {
        BBTestObject testObject = new BBTestObject ();
        assertNull (testObject.isAValidTime(null));
        assertNull (testObject.isAValidTime("123:45"));
        assertNull (testObject.isAValidTime("12:456"));
        assertNull (testObject.isAValidTime("1:30"));
        assertNull (testObject.isAValidTime("-1:45"));
```

```java
        assertNull (testObject.isAValidTime("25:45"));
        assertNull (testObject.isAValidTime("26:62"));
        assertNull (testObject.isAValidTime("null"));
        assertNull (testObject.isAValidTime(""));
    }

    @Test
    public void testAddBusTimeValid () throws Exception {
        Calendar busTime = Calendar.getInstance();
        BBTestObject testObject = new BBTestObject ();
        assertEquals (0, testObject.busTimeAdded("10", busTime));
        assertEquals (0, testObject.busTimeAdded("15", busTime));
        assertEquals (0, testObject.busTimeAdded("16", busTime));
        assertEquals (0, testObject.busTimeAdded("19", busTime));
        assertEquals (0, testObject.busTimeAdded("20", busTime));
        assertEquals (0, testObject.busTimeAdded("nc", busTime));
    }

    @Test
    public void testAddBusTimeInvalid () throws Exception {
        Calendar busTime = Calendar.getInstance();
        BBTestObject testObject = new BBTestObject ();
        assertEquals (1, testObject.busTimeAdded(null, busTime));
        assertEquals (1, testObject.busTimeAdded("12", busTime));
        assertEquals (1, testObject.busTimeAdded("ac", busTime));
        assertEquals (1, testObject.busTimeAdded("", busTime));

        busTime = null;
        assertEquals (1, testObject.busTimeAdded(null, busTime));
        assertEquals (1, testObject.busTimeAdded("10", busTime));
        assertEquals (1, testObject.busTimeAdded("11", busTime));
        assertEquals (1, testObject.busTimeAdded("ac", busTime));
        assertEquals (1, testObject.busTimeAdded("", busTime));
    }

    @Test
    public void testPrintTimeValid () throws Exception {
        BBTestObject testObject = new BBTestObject ();
        Calendar busTime = Calendar.getInstance();
        busTime.set (Calendar.HOUR_OF_DAY, 1);
        busTime.set (Calendar.MINUTE, 30);
        assertEquals ("01:30", testObject.printedTime(busTime));

        // Get a new instance of the Calendar object
        busTime = Calendar.getInstance();
        busTime.set (Calendar.HOUR_OF_DAY, 13);
        busTime.set (Calendar.MINUTE, 45);
        assertEquals ("01:45", testObject.printedTime(busTime));

        // Get a new instance of the Calendar object
        busTime = Calendar.getInstance();
        busTime.set (Calendar.HOUR_OF_DAY, 0);
        busTime.set (Calendar.MINUTE, 0);
        assertEquals ("12:00", testObject.printedTime(busTime));

        // Get a new instance of the Calendar object
        busTime = Calendar.getInstance();
        busTime.set (Calendar.HOUR_OF_DAY, 23);
        busTime.set (Calendar.MINUTE, 59);
        assertEquals ("11:59", testObject.printedTime(busTime));
    }

    @Test
    public void testPrintTimeInvalid () throws Exception {
        BBTestObject testObject = new BBTestObject ();
        Calendar busTime = null;
        assertNull (testObject.printedTime(busTime));
    }
}

```

BBTestObject.java

```java
package com.example.ucscbusbuddy.test;

import java.util.ArrayList;
import java.util.Calendar;

import com.example.ucscbusbuddy.BusStop;

public class BBTestObject {

    /*
     * Returns a list of bus times for a particular route passed.
     */
    public ArrayList<Calendar> isAListOfTimes (String busNumber) {
        BusStop stop = new BusStop();
        return stop.getBusTimes (busNumber);
    }

    /*
     * Given a 24-hour formatted HH:mm time, convert it to a Calendar
     * object and return it.
     */
    public Calendar isAValidTime (String time) {
        return BusStop.setTime (time);
    }

    /*
     * Adds a bus time to the list of times for a given bus route.
     * Calendar must have a valid date and time set, so there is no
     * check needed for that. An exception would be thrown if the
     * Calendar object was set to an invalid time before it was passed
     * to this method.
     */
    public int busTimeAdded (String busNumber, Calendar busTime) {
        BusStop stop = new BusStop();
        return stop.addBusTime (busNumber, busTime);
    }

    /*
     * Prints the time passed in 12-hour hh:mm format.
     */
    public String printedTime (Calendar time) {
        return BusStop.printTime(time);
    }
}

```