

Andri Friðriksson – [anf4@hi.is](mailto:anf4@hi.is) – 141190-2379  
Ása Júlía Aðalsteinsdóttir – [aja11@hi.is](mailto:aja11@hi.is) – 030896-2549  
Einar Andreas Helgason – [eah8@hi.is](mailto:eah8@hi.is) – 180282-7269  
Sigurlaug Þórðardóttir – [sth301@hi.is](mailto:sth301@hi.is) – 1502953279 (Presenting)

## HBV401G - Software Test

We created tests for our three sorting methods and our bookTrip method, we tested all methods both when they were called with a valid and invalid input. We used our MockDBTripManager and MockDBBookingManager to simulate a connection to a database. Below you can see the source code for our test class and our mock objects.

```
1. public class TripControllerTest {
2.     public TripControllerTest()
3.     { }
4.     @BeforeClass
5.     public static void setUpClass() {}
6.     @AfterClass
7.     public static void tearDownClass() {}
8.     @Before
9.     public void setUp() {}
10.    @After
11.    public void tearDown() {}
12.    /**
13.     * Tests of sortTripsByName method, of class TripController.
14.     */
15.    @Test
16.    public void testSortTripsByName_ValidInput() throws Exception {
17.        //Arrange
18.        TripController instance = new TripController();
19.        Trip[] trips = instance.searchTrips("", null, null, null, "", Boolean.FALSE,
20.                                           Boolean.FALSE, 0, 0, 0, 0);
21.        //Act
22.        Trip[] sortedTrips = instance.sortTripsByName(trips);
23.
24.        //Assert
25.        String currName = "";
26.        String prevName = "";
27.        for (Trip t: sortedTrips) {
28.            currName = t.getName();
29.            if (!"".equals(prevName)) assertTrue(currName.compareTo(prevName) >= 0);
30.            prevName = currName;
31.        }
32.    }
33.
34.    @Test
35.    public void testSortTripsByName_InvalidInput() throws Exception {
36.        //Arrange
37.        TripController instance = new TripController();
38.        Trip[] trips = new Trip[0];
39.        boolean thrown = false;
40.
41.        //Act
42.        try {
43.            Trip[] sortedTrips = instance.sortTripsByName(trips);
44.        } catch (IllegalArgumentException ex) {
45.            thrown = true;
46.        }
47.
48.        //Assert
49.        assertTrue(thrown);
50.    }
```

Andri Friðriksson – [anf4@hi.is](mailto:anf4@hi.is) – 141190-2379  
Ása Júlía Aðalsteinsdóttir – [aja11@hi.is](mailto:aja11@hi.is) – 030896-2549  
Einar Andreas Helgason – [eah8@hi.is](mailto:eah8@hi.is) – 180282-7269  
Sigurlaug Þórðardóttir – [sth301@hi.is](mailto:sth301@hi.is) – 1502953279 (Presenting)

```
51.  /**
52.   * Tests of sortTripsByTime method, of class TripController.
53.   */
54.  @Test
55.  public void testSortTripsByTime_ValidInput() throws Exception {
56.      //Arrange
57.      TripController instance = new TripController();
58.      Trip[] trips = instance.searchTrips("", null, null, null, "", Boolean.FALSE,
59.                                          Boolean.FALSE, 0, 0, 0, 0);
60.
61.      //Act
62.      Trip[] sortedTrips = instance.sortTripsByTime(trips);
63.
64.      //Assert
65.      Time currTime;
66.      Time prevTime = new Time(0, 0, 0);
67.      for (Trip t: sortedTrips) {
68.          currTime = t.getStartTime();
69.          assertTrue(currTime.after(prevTime));
70.          prevTime = currTime;
71.      }
72.  }
73.
74.  @Test
75.  public void testSortTripsByTime_InvalidInput() throws Exception {
76.      //Arrange
77.      TripController instance = new TripController();
78.      Trip[] trips = new Trip[0];
79.      boolean thrown = false;
80.
81.      //Act
82.      try {
83.          Trip[] sortedTrips = instance.sortTripsByTime(trips);
84.      } catch (IllegalArgumentException ex) {
85.          thrown = true;
86.      }
87.
88.      //Assert
89.      assertTrue(thrown);
90.  }
91.
92.  /**
93.   * Tests of sortTripsByPrice method, of class TripController.
94.   */
95.  @Test
96.  public void testSortTripsByPrice_ValidInput() throws Exception {
97.      //Arrange
98.      TripController instance = new TripController();
99.      Trip[] trips = instance.searchTrips("", null, null, null, "", Boolean.FALSE,
100.                                          Boolean.FALSE, 0, 0, 0, 0);
101.
102.      //Act
103.      Trip[] sortedTrips = instance.sortTripsByPrice(trips);
104.
105.      //Assert
106.      int currPrice = 0;
107.      int prevPrice = 0;
108.      for (Trip t: sortedTrips) {
109.          currPrice = t.getPrice();
110.          if (prevPrice != 0) assertTrue(currPrice >= prevPrice);
111.      }
112.      prevPrice = currPrice;
113.  }
```

Andri Friðriksson – [anf4@hi.is](mailto:anf4@hi.is) – 141190-2379

Ása Júlía Aðalsteinsdóttir – [aja11@hi.is](mailto:aja11@hi.is) – 030896-2549

Einar Andreas Helgason – [eah8@hi.is](mailto:eah8@hi.is) – 180282-7269

Sigurlaug Þórðardóttir – [sth301@hi.is](mailto:sth301@hi.is) – 1502953279 (Presenting)

```
3.  @Test public void testSortTripsByPrice_InvalidInput() throws Exception {
4.      //Arrange
5.      TripController instance = new TripController();
6.      Trip[] trips = new Trip[0];
7.      boolean thrown = false;
8.
9.      //Act
10.     try {
11.         Trip[] sortedTrips = instance.sortTripsByPrice(trips);
12.     } catch (IllegalArgumentException ex) {
13.         thrown = true;
14.     }
15.
16.     //Arrange
17.     assertTrue(thrown);
18. }
19.
20. /**
21.  * Tests of bookTrip method, of class TripController.
22.  */
23. @Test
24. public void testBookTrip_ValidInput() throws Exception {
25.     //Arrange
26.     TripController instance = new TripController();
27.     TourCompany tc = new TourCompany(1, "Fyrirtækið", 5675678, "háðær 15",
28.                                     "email@email.com");
29.     Trip trip = new Trip(1, "Ferð", new Date(117, 05, 22), new Time(9, 0, 0),
30.                         new Time(18, 0, 0), "", 1000, "Gamanferð", "Heima", "Rvk", 20,
31.                         false, false, tc, 20);
32.
33.     //Act
34.     Trip trip2 = instance.bookTrip("Jón Jónsson", 334234, "Jónsgeisli 5",
35.                                   "jon@jon.is", trip, 3, true, true);
36.
37.     //Assert
38.     assertTrue(trip2.getAvailablePlaces() == 17);
39. }
40.
41. @Test
42. public void testBookTrip_InvalidInput() throws Exception {
43.     //Arrange
44.     TripController instance = new TripController();
45.     TourCompany tc = new TourCompany(1, "Fyrirtækið", 5675678, "háðær 15",
46.                                     "email@email.com");
47.     Trip trip = new Trip(1, "Ferð", new Date(117, 05, 22), new Time(9, 0, 0),
48.                         new Time(18, 0, 0), "", 1000, "Gamanferð", "Heima", "Rvk", 20,
49.                         false, false, tc, 20);
50.     boolean thrown = false;
51.
52.     //Act
53.     try {
54.         Trip trip2 = instance.bookTrip("", 0, "", "", trip, 0, null, null);
55.     } catch (IllegalArgumentException ex) {
56.         thrown = true;
57.     }
58.
59.     //Assert
60.     assertTrue(thrown);
61. }
62. }
```

Andri Friðriksson – [anf4@hi.is](mailto:anf4@hi.is) – 141190-2379  
Ása Júlía Aðalsteinsdóttir – [aja11@hi.is](mailto:aja11@hi.is) – 030896-2549  
Einar Andreas Helgason – [eah8@hi.is](mailto:eah8@hi.is) – 180282-7269  
Sigurlaug Þórðardóttir – [sth301@hi.is](mailto:sth301@hi.is) – 1502953279 (Presenting)

```
1. public class MockDBBookingManager {
2.     private Connection myConn;
3.     private PreparedStatement myStmt;
4.     private ResultSet myRs;
5.     private final String dbname;
6.
7.     public MockDBBookingManager(String dbname)
8.     {
9.         this.dbname = dbname;
10.    }
11.
12.    public void bookTrip(Booking booking) throws ClassNotFoundException
13.    {}
14.
15.    public void cancelBooking(Booking booking) throws ClassNotFoundException
16.    {}
17. }
```

```
1. public class MockDBTripManager {
2.     private Connection myConn;
3.     private PreparedStatement myStmt;
4.     private ResultSet myRs;
5.     private final String dbname;
6.     public MockDBTripManager(String dbname) throws SQLException
7.     {
8.         this.dbname = dbname;
9.     }
10.    public Trip[] search(String tripName, Date date, Time startTime, Time endTime,
11.                        String description, Boolean familyFriendly, Boolean accessible,
12.                        int minPrice, int maxPrice, Integer type, Integer location)
13.                        throws SQLException, ClassNotFoundException {
14.        Trip[] trips = new Trip[10];
15.        TourCompany tc = new TourCompany(1, "Fyrirtækið", 5675678, "háðar 15",
16.                                           "email@email.com");
17.        for (int i = 0; i < 10; i++) {
18.            Trip trip = new Trip(i, "Ferð" + i, new Date(117, 05, 22 + i),
19.                                new Time(9 + i, 0, 0), new Time(18 + i, 0, 0), "",
20.                                1000 + (i * 100), "Gamanferð", "Heima", "Rvk", 20,
21.                                false, false, tc, 20 - i);
22.            trips[i] = trip;
23.        }
24.        return trips;
25.    }
26.
27.    public void updateAvailablePlaces(Trip t) throws ClassNotFoundException
28.    {}
29. }
```