



Лабораторийн ажил 7 ТАЙЛАН

Сэдвийн нэр: MeetingPlanner Java project дээр unit testing хийх

Лаборатори гүйцэтгэсэн:

Оюутан:

Г. Алтай

B222270028

Github repo link:

<https://github.com/altai861/batalgaajuulalt-lab7-unittesting-MeetingPlanner>

Танилцуулга

Энэхүү ажлаар **MeetingPlanner** java project дээр unit test бичсэн. MeetingPlanner project нь дараах классууд болон үйлдлийг хийх боломжтой.

Calendar, Meeting, Organization, Person, PlannerInterface, Room, TimeConflictException

- Уулзалт товллох
- Амралт товллох
- Өрөөний сул эсэхийг шалгах
- Хүний завтай эсэхийг шалгах
- Өрөөний уулзалтын хөтөлбөр хэвлэх
- Хүний уулзалтын хөтөлбөр хэвлэх

Нийт CalendarTest, MeetingTest, OrganizationTest, PersonTest, RoomTest гэсэн 5 unit test class үүсгэж нийтдээ **24 unit test** бичсэн.

Тест кэйсүүд болон үр дүн

CalendarTest: 14 test — 3 errors — 2 failures — 9 success

1. **testAddAndGetMeeting():** Өдөр болон цагийг зөв оруулан (1 сарын 15, 10–11 ийн хооронд) уулзалт бүртгэсэн. - success
2. **testIsBusyTrueAndFalse():** Зөв өгөгдлөөр уулзалт бүртгэн Calendar классийн isBusy функцийг шалгасан. — error
3. **testClearSchedule():** Уулзалт бүртгээд араас нь тухайн өдрийн хуваарийг цэвэрлэж хуваарийг үлдсэн эсэхийг шалгах — success
4. **testRemoveMeeting():** Calendar классийн remove функцийг шалгах — success
5. **testAddMeetingInvalidDay():** 2 сарын 35–ын өдөр уулзалт бүртгэх оролдлого. TimeConflictException буцаах ёстой. — success
6. **testAddMeetingInvalidMonth():** 13 сарын 10 ны өдөр уулзалт бүртгэх оролдлого. TimeConflictException буцаах ёстой. — success

7. **testAddMeetingValidMonth():** 12 сарын 10 уулзалт бүртгэх оролдлого. Асуудалгүй ажиллах ёстой. — **error**
8. **testAddMeetingIllegalTime():** Уулзалтын цагийг буруу утга өгөх. Сөрөг утга эсвэл эхлэх цаг нь дуусах цагаасаа хойш гэх мэт — **success**
9. **testAddMeetingIllegalMonth():** 13 сард уулзалт захиалах. Алдаа заах ёстой. — **success**
10. **testGetMeetingInvalidIndex():** Тухайн өдрийн уулзалтаас байхгүй уулзалт авах оролдлого. IndexOutOfBoundsException өгөх ёстой. — **success**
11. **testGetMeetingOnInvalidDate():** 1 сарын 40 ийн өдрийн эхний уулзалтыг авах. TimeConflictException буцаах ёстой. — **failure**
12. **testRemoveMeetingInvalidIndex():** Тухайн өдрийн уулзалтаас байхгүй уулзалт хасах оролдлого. IndexOutOfBoundsException өгөх ёстой. — **success**
13. **testAddMeetingToNov30th():** 11 сарын 30 ын өдөр уулзалт бүртгэх. Алдаагүй бүртгэх ёстой. — **failure**
14. **testMeetingThatEndsWithinTheHour():** Тухайн эхэлсэн цагтаа дуусдаг уулзалт бүртгэх. Асуудалгүй болох ёстой. — **error**

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with a 'JUnit' folder containing test classes.
- JUnit Runner:** Displays the results of a test run. It shows 14 runs, 3 errors, and 2 failures. The test 'testAddMeetingToNov30th()' is highlighted in blue.
- Failure Trace:** Shows the stack trace for the failed test:


```
org.opentest4j.AssertionFailedError: expected: <edu.sc.csce747.MeetingPlanner.N>
    at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:1)
    at edu.sc.csce747.MeetingPlanner.CalendarTest.testAddMeetingToNov30th(CalendarTest.java:1596)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```
- Source Code:** Shows the source code for 'CalendarTest.java'. The code includes imports for JUnit and the MeetingPlanner classes, and defines several test methods:


```
package edu.sc.csce747.MeetingPlanner;
import static org.junit.jupiter.api.Assertions.*;

class CalendarTest {

    /**
     * Test: Successfully adding a meeting and retrieving it
     * Description: Checks that addMeeting correctly stores a meeting and getMeeting retrieves it.
     */
    @Test
    void testAddAndGetMeeting() throws TimeConflictException {
        Calendar cal = new Calendar();
        Meeting m = new Meeting(1, 15, 10, 11);
        cal.addMeeting(m);
        assertEquals(m, cal.getMeeting(1, 15, 0));
    }

    /**
     * Test: Check isBusy returns correct busy status
     * Description: Ensures isBusy returns true during the meeting and false outside.
     */
    @Test
    void testIsBusyTrueAndFalse() throws TimeConflictException {
        Calendar cal = new Calendar();
        Meeting m = new Meeting(1, 15, 10, 11);
        cal.addMeeting(m);
        assertTrue(cal.isBusy(1, 15, 10));
        assertFalse(cal.isBusy(1, 15, 12));
    }

    /**
     * Test: Clear schedule removes all meetings for a day
     * Description: Ensures that after clearing, the day has no meetings.
     */
    @Test
    void testClearSchedule() throws TimeConflictException {
        Calendar cal = new Calendar();
        Meeting m = new Meeting(1, 15, 10, 11);
        cal.addMeeting(m);
        cal.clearSchedule(1, 15);
        assertEquals(0, cal.printAgenda(1, 15).length() - ("Agenda for 1/15:\n".length()));
    }

    /**
     * Test: Remove meeting successfully
     * Description: Checks that a meeting can be removed and is no longer in the calendar.
     */
    @Test
    void testRemoveMeeting() throws TimeConflictException {
        Calendar cal = new Calendar();
        Meeting m = new Meeting(1, 15, 10, 11);
        cal.addMeeting(m);
        cal.removeMeeting(1, 15, 10, 11);
        assertEquals(0, cal.printAgenda(1, 15).length());
    }
}
```

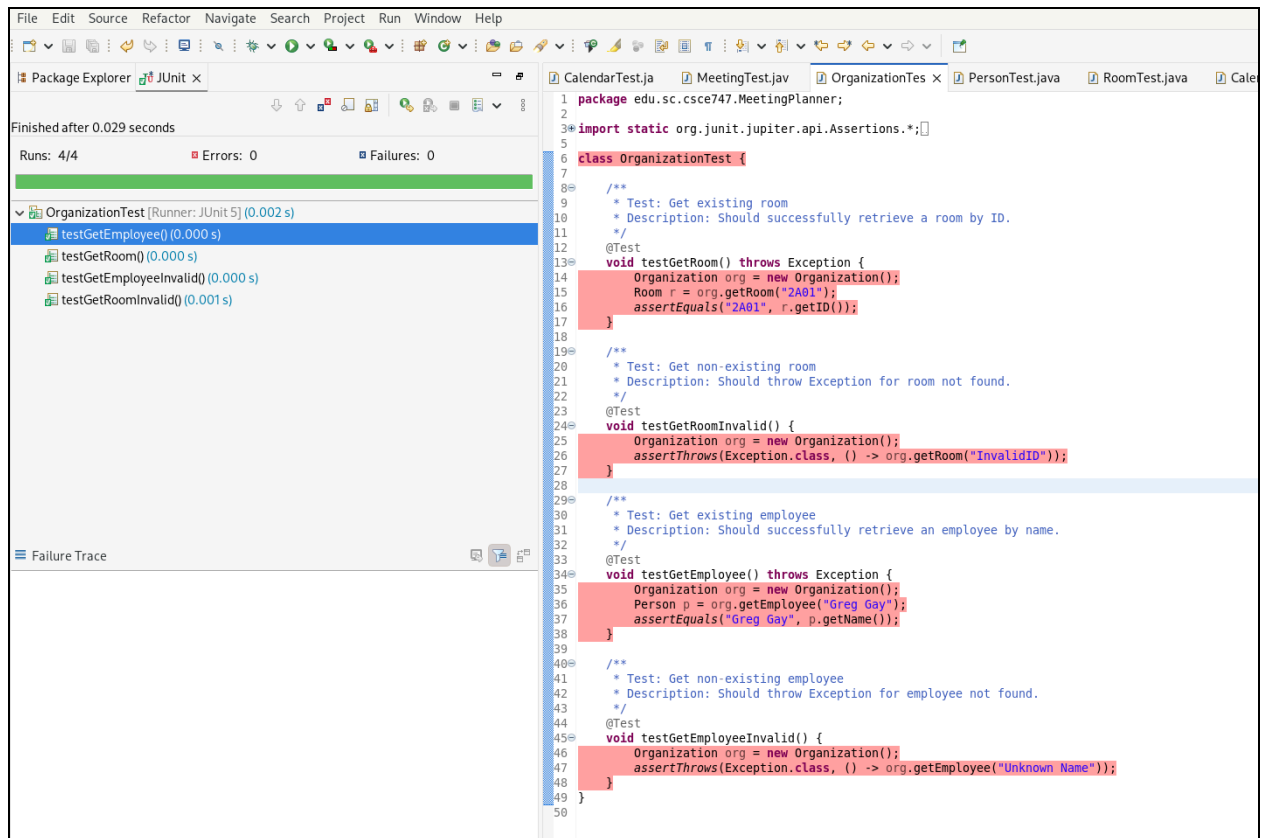
MeetingTest: 2 test — 2 success

1. **testMeetingConstructorAndGetters():** Meeting class-ийн constructor зөв ажиллаж байгаа эсэхийг шалгах — **success**
2. **testAddRemoveAttendee():** Уулзалтад хүн нэмж хасах үйлдлийн тест — **success**

```
1 package edu.sc.csce747.MeetingPlanner;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7 class MeetingTest {
8
9     /**
10      * Test: Meeting constructor and getters
11      * Description: Verifies all fields are correctly set and retrieved.
12      */
13     @Test
14     void testMeetingConstructorAndGetters() {
15         ArrayList<Person> attendees = new ArrayList<>();
16         Room room = new Room("2A01");
17         Meeting m = new Meeting(1, 15, 10, 11, attendees, room, "Team Meeting");
18         assertEquals(1, m.getMonth());
19         assertEquals(15, m.getDay());
20         assertEquals(10, m.getStartTime());
21         assertEquals(11, m.getEndTime());
22         assertEquals(room, m.getRoom());
23         assertEquals("Team Meeting", m.getDescription());
24         assertEquals(attendees, m.getAttendees());
25     }
26
27     /**
28      * Test: Add and remove attendee
29      * Description: Ensures attendees can be added and removed from a meeting.
30      */
31     @Test
32     void testAddRemoveAttendee() {
33         Person p = new Person("John Doe");
34         ArrayList<Person> attendees = new ArrayList<>();
35         Meeting m = new Meeting(1, 15, 10, 11, attendees, new Room(), "Meeting");
36         m.addAttendee(p);
37         assertTrue(m.getAttendees().contains(p));
38         m.removeAttendee(p);
39         assertFalse(m.getAttendees().contains(p));
40     }
41 }
42
```

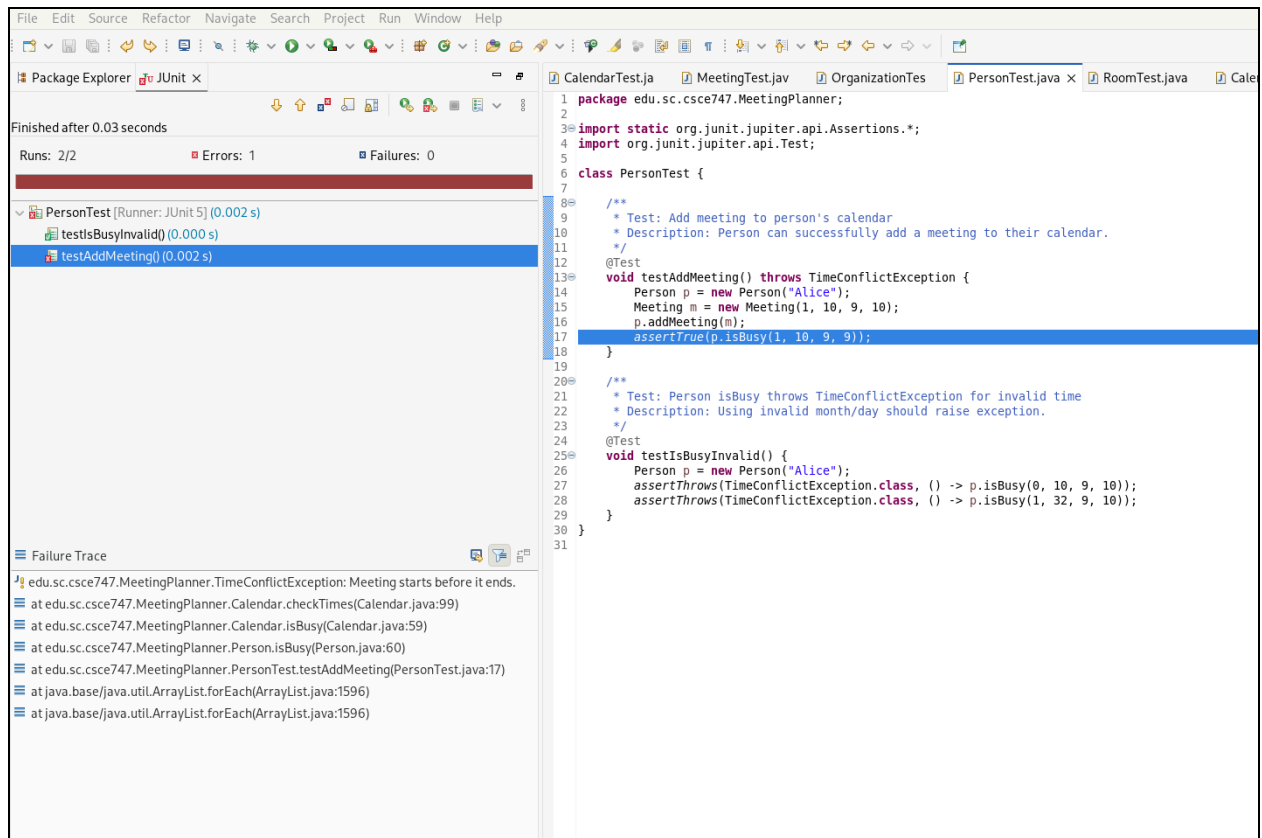
OrganizationTest: 4 tests — 4 success

1. **testGetRoom():** getRoom функцийг шалгах — **success**
2. **testGetRoomInvalid():** Байхгүй өрөөний нэр шалгах. Алдаа буцаах ёстой — **success**
3. **testGetEmployee():** getEmployee функцийг шалгах — **success**
4. **testGetEmployeeInvalid():** Байхгүй ажилтны нэр шалгах. Алдаа буцаах ёстой — **success**



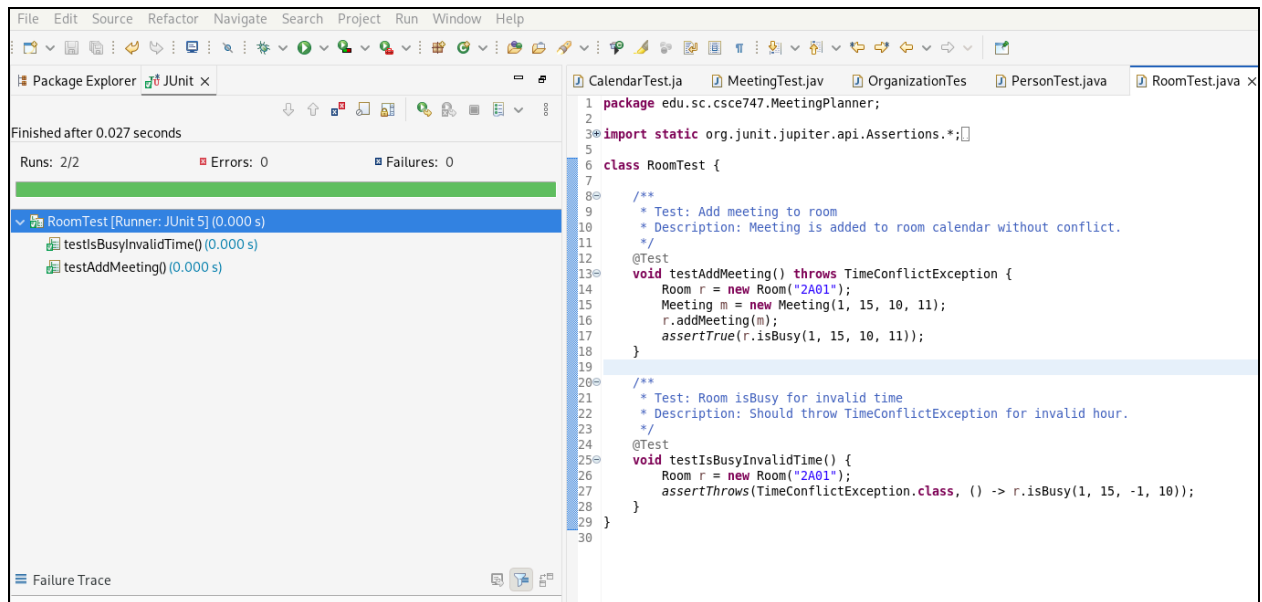
PersonTest – 2 tests – 1 success - 1 error

1. **testAddMeeting():** Тухайн хүн дээр уулзалт үүсгээд тухайн уулзалтын эхлэх цаг дээр тухайн хэрэглэгч завтай завгүй эсэх шалгалт – **error** (Дээр гарсан эхлэлийн болон төгсгөлийн цаг давхцаж болохгүй байгаа алдаанаас болж гарсан)
2. **testIsBusyInvalid():** Хэрэглэгчийг буруу утгатай цагууд дээр завтай эсэхийг шалгах. `TimeConflictException` буцаах ёстой. — **success**



RoomTest – 2 tests – 1 success - 1 error

1. **testAddMeeting():** Өрөө рүү уулзалт нэмээд тухайн өрөө хувааль дээр busy гэсэн тодотголтой болсон эсэхийг шалгана. – **success**
2. **testIsBusyInvalid():** Өрөөн дээр буруу цагаар isBusy функцийг дуудах. TimeConflictException буцаах – **success**



CalendarTest-ээс олдсон алдаанууд !

1. Calendar-аас уулзалт авах getMeeting функц эхлээд сар, өдөр гэсэн утгуудыг шалгадаггүй шууд array-аас get функцээ ашиглан авдаг. Үүн дээр буруу утга оруулахад IndexOutOfBoundsException ирж байгаа. TimeConflictException буцаах ёстой.
2. Уулзалтын цагийн эхлэх болон төгсөх цаг ижил байх боломжгүй байна. Уг нь бол тухайн цагтаа дуусах боломжтой.
3. 12 дугаар сар дээр уулзалт нэмэх боломжгүй байна.
4. Мөн 11 сарын 30нд уулзалт нэмэх боломжгүй байна.