



Лабораторийн ажил 8 ТАЙЛАН

Лаборатори гүйцэтгэсэн:

Оюутан:
Н. Хангал

B222270032

Unit Testing

Албан бус тестийг төлөвлөгөөг томъёолох

- а. Өгөгдсөн код болон функциональ дээр алдаагүй ажиллах тохиолдлын нэгжийн тестүүдийг төлөвлөсөн.

Meeting class tests

№	Тест нэр	Тайлбар	Оролт	Хүлээгдэж буй үр дүн
1.1	testCreateMeeting_basically	Зөв сар, өдөр, цаг, тайлбартай уулзалт үүсгэх	month=5, day=10, start=9, end=11, desc="Team Meeting"	Meeting объект үүсэх, getMonth() → 5, getDay() → 10
1.2	testToString_includesRoomAndAttendees	Уулзалт хэвлэхэд өрөө ба оролцогчид гарах	Room="2A01", Attendees=["Greg"]	Текстэд "Room 2A01" ба "Greg" багтсан байх

Calendar class test

№	Тест нэр	Тайлбар	Оролт	Хүлээгдэж буй үр дүн
2.1	testAddMeeting_validDate	Хүчинтэй огноотой уулзалт нэмэх	(5, 20, 10, 12)	isBusy(5, 20, 10, 12) → true
2.2	testIsBusy_falseForEmptyDay	Уулзалт байхгүй өдөр	(6, 1, 9, 10)	FALSE
2.3	testRemoveMeeting_valid	Уулзалт устгах	add → remove	isBusy → false

Room class test

№	Тест нэр	Тайлбар	Оролт	Хүлээгдэж буй үр дүн
3.1	testAddMeetingToRoom	Өрөөнд уулзалт нэмэх	Room("2A01"), Meeting(5, 12, 14, 15)	isBusy(5, 12, 14, 15) → true
3.2	testRoomAgendaPrint	Хөтөлбөр хэвлэх	printAgenda(5)	Хариу нь уулзалтын мэдээлэл агуулна

Person class test

№	Тест нэр	Тайлбар	Оролт	Хүлээгдэж буй үр дүн
4.1	testAddMeetingToPerson	Ажилтанд уулзалт нэмэх	Person("Greg"), Meeting(7, 8, 10, 11)	isBusy(7, 8, 10, 11) → true

4.2	testPrintAgenda	Ажилтны хөтөлбөр хэвлэх	printAgenda(7)	Хариу нь "Greg" ба уулзалтын мэдээлэл агуулна
-----	-----------------	-------------------------	----------------	---

Organization class test

№	Тест нэр	Тайлбар	Оролт	Хүлээгдэж буй үр дүн
5.1	testGetExistingEmployee	Байгаа ажилтныг авах	"Greg Gay"	getName() → "Greg Gay"
5.2	testGetExistingRoom	Байгаа өрөөг авах	"2A01"	getID() → "2A01"

jUnit framework ашиглан тест код бичих

Буруу ажиллах тохиолдлуудыг аль болох олон тохиолдлоор төлөвлөж, Ж: 2 сарын 35-ны өдөр уулзалт товлож болох уу гм. Үүнийг тестлэх код бичсэн.

Хэрэв тест тохиолдол нь exception шидэх ёстой бол шидэж байгаа эсэхийг шалгасан.

Хэрэв програм fail болох ёстой бол зөв fail болж байгаа эсэхийг шалгах нэгжийн тест бичсэн.

```

import org.junit.Test;

public class CalendarTest {

    private Calendar calendar;

    @Before
    public void setUp() {
        calendar = new Calendar();
    }

    @Test
    public void testAddMeeting_holiday() {
        try {
            Meeting midsommar = new Meeting(6, 26, "Midsommar");
            calendar.addMeeting(midsommar);
            Boolean added = calendar.isBusy(6, 26, 0, 23);
            assertTrue("Midsommar should be marked as busy on the calendar", added);
        } catch (TimeConflictException e) {
            fail("Should not throw exception: " + e.getMessage());
        }
    }

    @Test
    public void testInvalidDay_Feb35() {
        try {
            Meeting invalid = new Meeting(2, 35, 10, 11);
            calendar.addMeeting(invalid);
        }
    }
}

```

```

package edu.sc.csce747.MeetingPlanner;

import static org.junit.Assert.assertEquals;

public class MeetingTest {

    @Test
    public void testInvalidTimes() {
        try {
            Meeting m = new Meeting();
            m.setStartTime(15);
            m.setEndTime(10);
            assertTrue("Энд буруу цагийн утга байна", m.getStartTime() > m.getEndTime());
        } catch (Exception e) {
            fail("Exception гарч болошгүй: " + e.getMessage());
        }
    }

    @Test
    public void testFullConstructor() {
        try {
            ArrayList<Person> people = new ArrayList<>();
            people.add(new Person("Test User"));
            Room room = new Room("1A01");

            Meeting m = new Meeting(5, 20, 9, 10, people, room, "Team Meeting");
            assertEquals("Team Meeting", m.getDescription());
            assertEquals("1A01", m.getRoom().getID());
            assertEquals("Test User", m.getAttendees().get(0).getName());
        }
    }
}

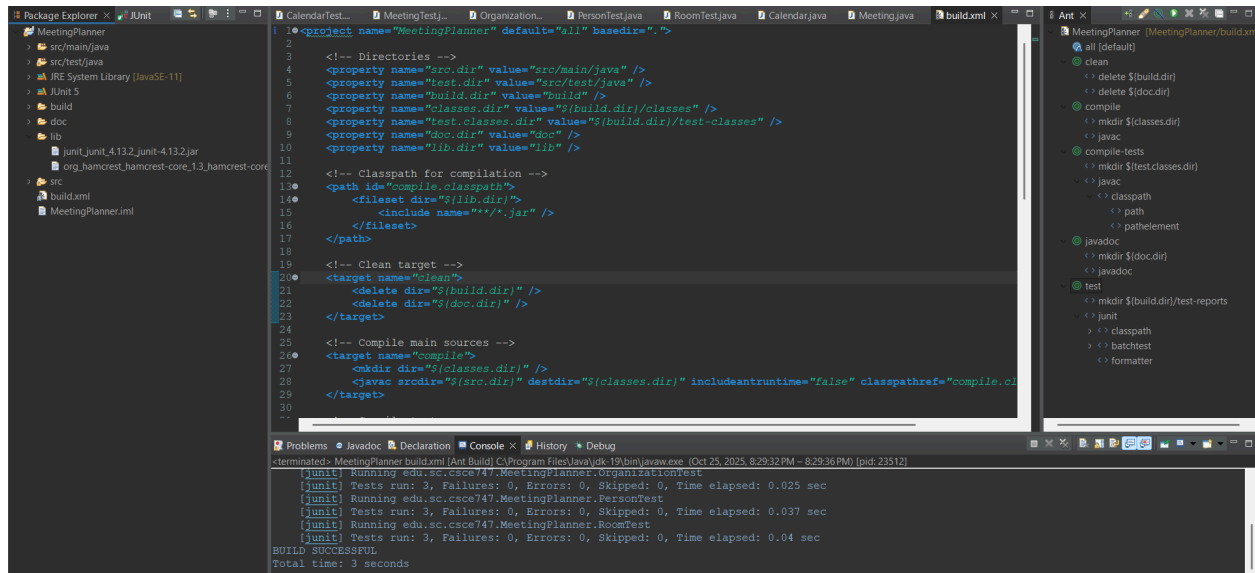
```



Build tool тохируулах

ant болон ижилт төстэй build tool тохируулж javadoc-р баримт үүсгэх target үүсгэсэн.

unit test ажиллуулах target үүсгэх.



Өөрийн ажлыг github дээр байршуулах

[khangal5/batalgaajuulalt.Lab8](#)

Тайлан

src/test/java/ доторх 5 class тус бүрд unit test бичсэн.

1. CalendarTest.java - 9
2. MeetingTest.java - 3
3. OrganizationTest.java - 3
4. PersonTest.java - 3
5. RoomTest.java - 3 нийт 21 unit test бичсэн.

Нийт 13 failure гарсан. Алдаа гарсан testCase үүд:

CalendarTest:

1. Хуурамч (оргүй) өдөр — **2 сарын 35-ны өдөр** — зөвшөөрөгдөхгүй байхыг шалгах.
2. 13-р сар гэж байхгүй тул, **буруу сар**-ыг шалгах.
3. **Буруу эхлэх цаг** (-2 цаг) байх үед exception гаргах ёстойг шалгах.
4. **Дуусах цаг** 25 гэж буруу утга өгвөл exception гаргах ёстойг шалгах.
5. Уулзалтын **эхлэх ба дуусах цаг ижил** байвал (**9–9**) буруу гэж үзэх ёстой.
6. **Давхцаж буй уулзалтуудыг** зөв илрүүлж exception гаргах эсэхийг шалгах.
7. **Байхгүй индексээр уулзалт авах** оролдлого (**index = 0**, харин жагсаалт хоосон) → exception гарах ёстойг шалгах.
8. **Устгах индекс байхгүй үед** exception гаргах ёстойг шалгах.

OrganizationTest:

1. Байхгүй өрөөг ("**999X**") авах гэж оролдож, системийн хариу үйлдлийг шалгана
2. Байхгүй ажилтныг ("Unknown Person") авах гэж оролдох үед системийн хариу үйлдэл шалгах.

PersonTest:

1. Хэрвээ давхцаж байгаа цагийн хоёр уулзалт (9–11 ба 10–12) нэмбэл `TimeConflictException` гарч байгаа эсэхийг шалгах.
2. Хэрвээ хүн 3-р сарын 15-нд 13–14 цагийн уулзалт нэмбэл `isBusy()` функц үнэн (`true`) буцаах ёстой.

RoomTest:

1. Хоёр уулзалт 2-р сарын 20-нд 9–11 ба 10–12 цаг давхцаж байгаа үед `Room` объект `TimeConflictException` шидэх ёстойг шалгана.