Software Modeling & Analysis

Traveler Digital Watch

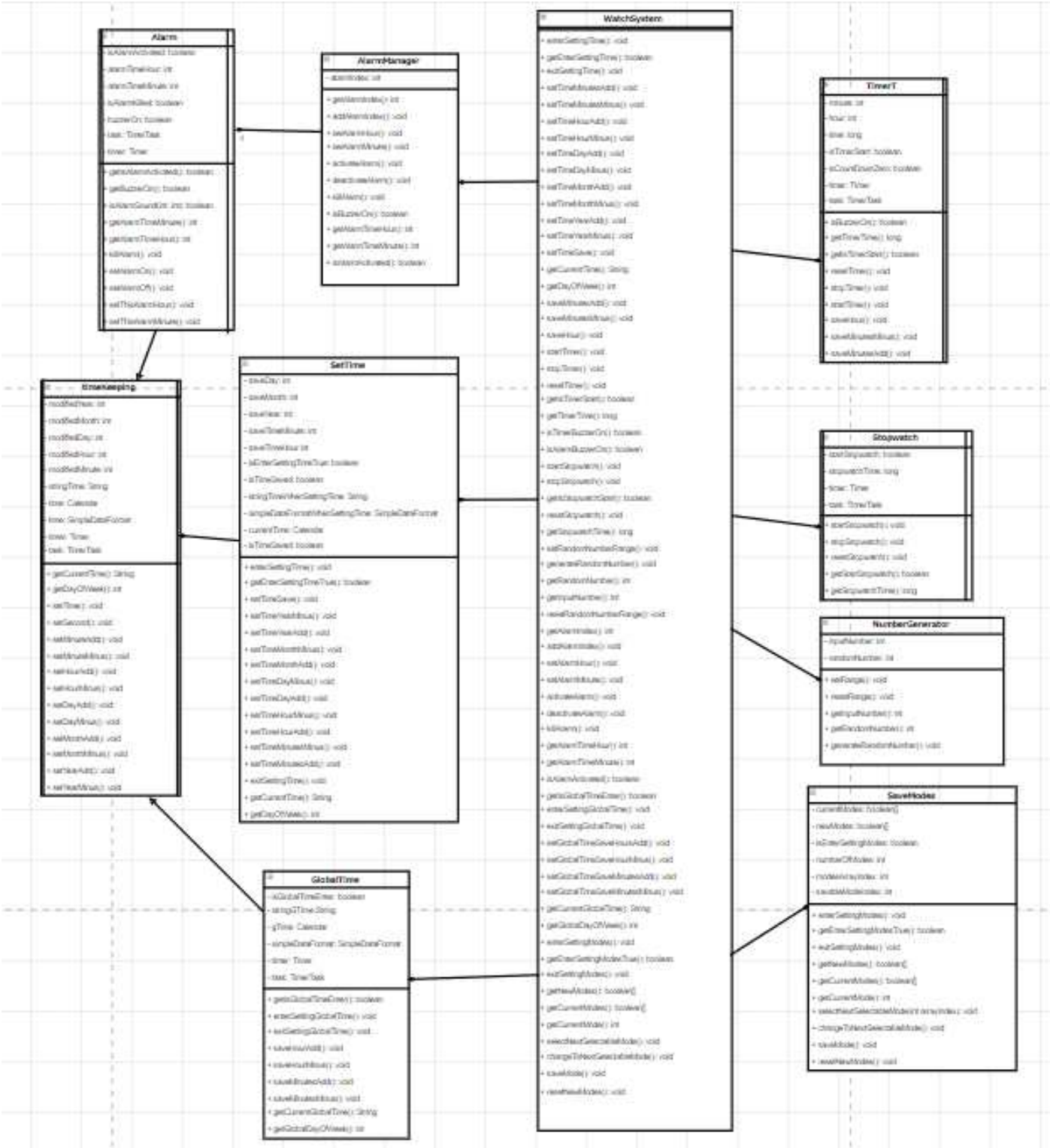
Project Team
7 Team

Date
2019-05-27

Team Information
201411295 이상훈
201711394 민하은
201711395 박성준
201711423 정종화

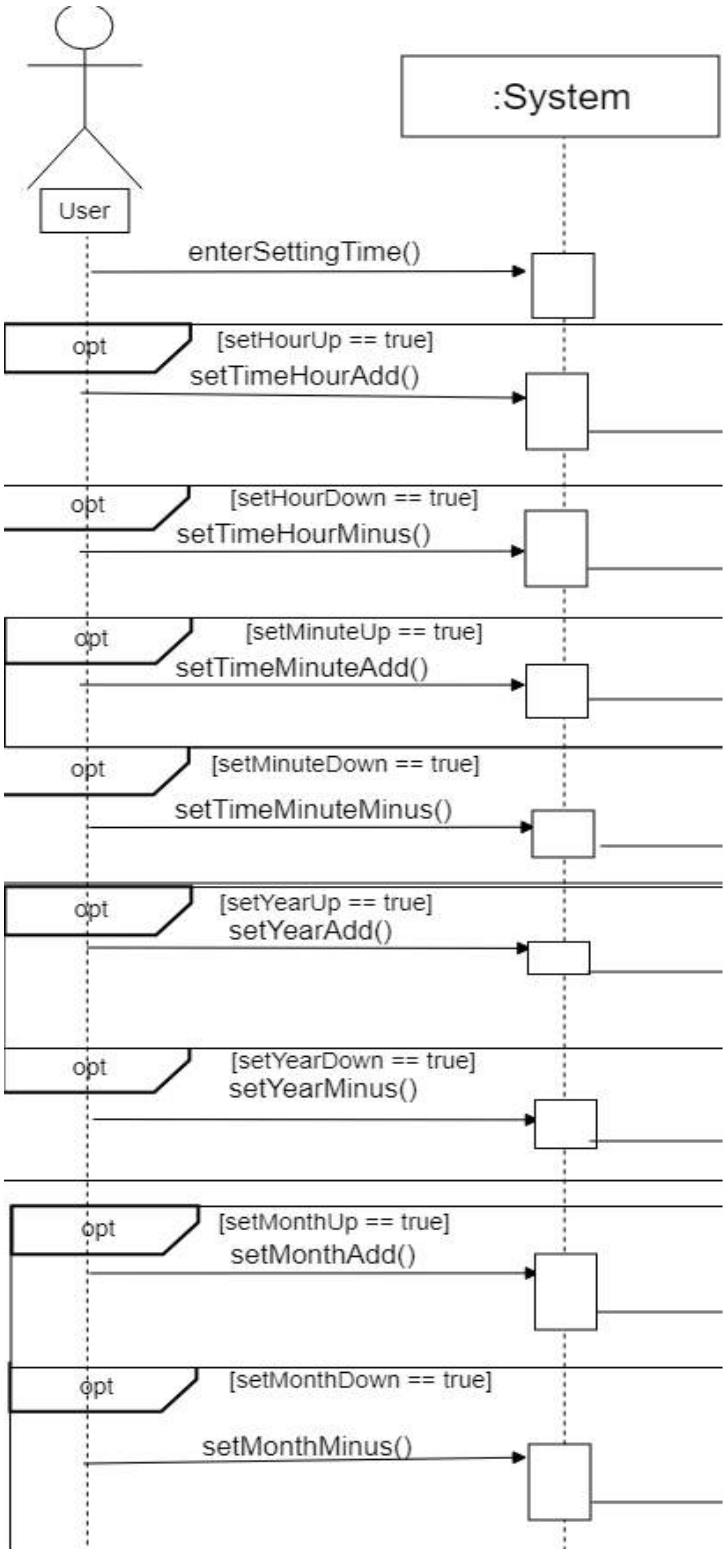
Index

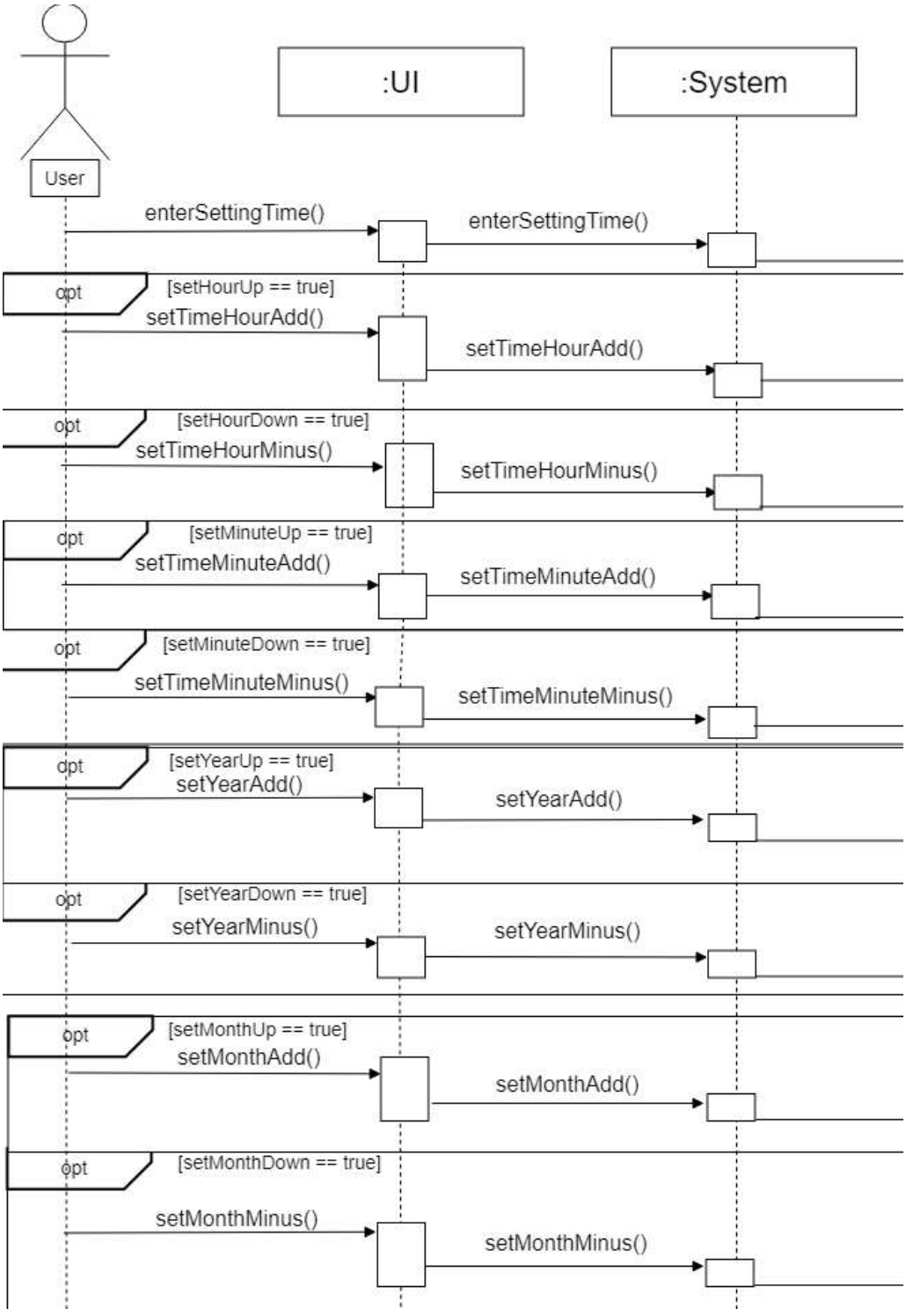
Activity 2052. Implement Windows



Class Diagram을 보면 중앙의 Watch System이 모든 기능들을 갖고 있어서, 모든 명령을 호출 할 수 있다.

SetTIme 기능에서 UI 삽입 전 후의 예시.




User
:UI
:System
enterSettingTime()
enterSettingTime()
opt
[setHourUp == true]
setTimeHourAdd()
setTimeHourAdd()
opt
[setHourDown == true]
setTimeHourMinus()
setTimeHourMinus()
opt
[setMinuteUp == true]
setTimeMinuteAdd()
setTimeMinuteAdd()
opt
[setMinuteDown == true]
setTimeMinuteMinus()
setTimeMinuteMinus()
opt
[setYearUp == true]
setYearAdd()
setYearAdd()
opt
[setYearDown == true]
setYearMinus()
setYearMinus()
opt
[setMonthUp == true]
setMonthAdd()
setMonthAdd()
opt
[setMonthDown == true]
setMonthMinus()
setMonthMinus()

1.1 Set Time(R.1.0)

Name	SetTime
Responsibilities	시계의 현재 시간을 조정한다.
Type	UI
Cross References	R.4.3, R.6.0
Note	입력된 버튼에 따라 연, 월, 일, 시, 분을 1씩 조정할 수 있다.
Pre-Conditions	현재 시간을 보는 상태
Post-Conditions	조정된 시간을 사용자에게 보여준다.

1.2 Set Timer(R.2.0)

Name	SetTimer
Responsibilities	타이머의 시간을 조정한다.
Type	UI
Cross References	R.2.1
Note	입력된 버튼에 따라 타이머 시간 1분 혹은 1시간씩 조정한다.
Pre-Conditions	타이머 모드 상태
Post-Conditions	타이머에 조정된 시간이 저장되고, 사용자에게 보여준다.

1.3 Start Timer(R.2.1)

Name	StartTimer
Responsibilities	타이머 시간이 줄어들기 시작한다.
Type	UI
Cross References	R.2.0, R.2.2
Note	버튼이 입력되면 타이머에 저장되어있던 시간이 줄어들기 시작한다.
Pre-Conditions	타이머 모드 상태
Post-Conditions	타이머 시간이 1초마다 줄어들고, 사용자에게 보여준다.

1.4 Stop Timer(R.2.2)

Name	StopTimer
Responsibilities	타이머 시간이 멈춘다.
Type	UI
Cross References	R.2.1
Note	버튼이 입력되면 타이머 시간이 더 이상 줄어들지 않는다,
Pre-Conditions	타이머 모드 상태, 타이머 시작 후.
Post-Conditions	타이머 시간이 줄어들지 않고, 사용자에게 보여준다.

1.5 Start Stopwatch(R.3.0)

Name	StartStopwatch
Responsibilities	스톱워치 시간이 증가하기 시작한다.
Type	UI
Cross References	
Note	버튼이 입력되면 스톱워치 시간이 최대 시간 1시간 59분 59초까지 증가한다.
Pre-Conditions	스톱워치 모드 상태.
Post-Conditions	스톱워치 시간이 증가하고, 사용자에게 보여준다.

1.6 Stop Stopwatch(R.3.1)

Name	StopStopwatch
Responsibilities	스톱워치 시간이 멈춘다.
Type	UI
Cross References	R.3.0
Note	버튼이 입력되면 스톱워치 시간이 버튼이 입력된 순간의 시간으로 정지된다.
Pre-Conditions	스톱워치 모드 상태, 스톱워치 시작 후.
Post-Conditions	스톱워치 시간이 멈추고, 사용자에게 보여준다.

1.7 Reset Stopwatch(R.3.2)

Name	ResetStopwatch
Responsibilities	스톱워치 시간이 0분 0초 0.00초로 초기화 된다.
Type	UI
Cross References	R.3.1
Note	버튼이 입력되면 스톱워치 시간이 0분 0초 0.00초으로 초기화된다.
Pre-Conditions	스톱워치 모드 상태, 스톱워치 멈춤 후.
Post-Conditions	스톱워치 시간이 초기화 되고, 사용자에게 보여준다.

1.8 Activate / Deactivate Alarm(R.4.0)

Name	Activate/Deactiate Alarm
Responsibilities	해당 알람이 활성화 되어있으면 비활성화 시키고, 비활성화 되어있으면 활성화 시킨다.
Type	UI
Cross References	R.4.2
Note	버튼이 입력되면 해당 알람이 활성화 되거나 비활성화 된다. 사용자가 알람을 선택하여 원하는 알람에 적용할 수 있다.
Pre-Conditions	알람 모드 상태. 원하는 알람 선택 후.
Post-Conditions	해당 알람이 활성화 또는 비활성화 되고, 사용자에게 보여준다.

1.9 Turn off Alarm(R.4.1)

Name	Turn Off Alarm
Responsibilities	어떤 알람이 울릴 때, 알람을 꺼버린다.
Type	UI
Cross References	
Note	버튼이 입력되면 울리고 있는 모든 알람이 1분간 꺼지고 재시작한다.
Pre-Conditions	알람이 울리고 있는 상태.
Post-Conditions	울리고 있는 모든 알람이 꺼진다.

1.10 Change Alarm(R.4.2)

Name	Change Alarm
Responsibilities	4가지 알람들을 차례대로 보여준다.
Type	UI
Cross References	R.4.0, R.4.3
Note	버튼이 입력되면, 알람 0~3 까지 4개의 알람을 차례대로 볼 수 있다.
Pre-Conditions	알람 모드 상태.
Post-Conditions	차례로 알람을 넘기고, 사용자에게 보여준다.

1.11 Set Alarm(R.4.3)

Name	Set Alarm
Responsibilities	해당 알람 시간을 설정 할 수 있다.
Type	UI
Cross References	R.4.2
Note	버튼을 입력하여, 알람 시간을 1시간 혹은 1분씩 조정할 수 있다. 사용자가 알람을 선택하여 원하는 알람에 적용할 수 있다.
Pre-Conditions	알람 모드 상태.
Post-Conditions	해당 알람의 시간이 설정되고, 사용자에게 보여준다.

1.12 Set Number Range(R.5.2)

Name	Set Number Range
Responsibilities	생성될 난수의 범위를 조정한다.
Type	UI
Cross References	
Note	버튼을 입력하여, 생성될 난수 범위를 1씩 증가시킨다. default 난수 범위는 0이다.
Pre-Conditions	RNG(난수생성)모드 상태.
Post-Conditions	생성될 난수 범위가 저장되고, 사용자에게 보여준다.

1.13 Generate Random Number(R.5.1)

Name	Generate Number Range
Responsibilities	설정된 난수 범위 내에서 난수를 발생한다.
Type	UI
Cross References	
Note	버튼을 입력하여, 설정된 난수 범위 내에서 난수를 생성한다.
Pre-Conditions	RNG(난수생성)모드 상태.
Post-Conditions	난수를 생성하여 사용자에게 보여준다.

1.14 Reset Number Range(R.5.2)

Name	Reset Number Range
Responsibilities	난수 생성 범위를 0으로 초기화한다.
Type	UI
Cross References	
Note	버튼을 입력하여, 난수 생성 범위를 0으로 초기화한다.
Pre-Conditions	RNG(난수생성)모드 상태.
Post-Conditions	난수를 0으로 초기화하여 사용자에게 보여준다.

1.15 Set Global Time(R.6.0)

Name	Set Global Time
Responsibilities	Global time을 조정한다.
Type	UI
Cross References	
Note	버튼을 입력하여, Global Time을 15분씩 혹은 1시간씩 조정한다. default 시간은 현재 시간과 동일하다.
Pre-Conditions	Global Time 모드 상태.
Post-Conditions	조정된 Global time을 사용자에게 보여준다.

1.16 Change Mode Display(R.7.0)

Name	Change Mode Display
Responsibilities	선택 가능한 다음 모드로 넘어간다.
Type	UI
Cross References	R.8.0
Note	버튼을 입력하여, 지정된 4개의 모드가 차례대로 넘어가게 한다.
Pre-Conditions	Set Mode, Set Time, Set Global Time 모드가 아닌 상태.
Post-Conditions	다음 지정된 모드를 사용자에게 보여준다.

1.17 Set Mode(R.8.0)

Name	Set Mode
Responsibilities	사용자가 원하는 모드 4개를 선택할 수 있게 한다.
Type	UI
Cross References	R.7.0
Note	버튼을 입력하여, 총 6개의 모드 중 4개의 모드를 선택할 수 있게 한다. 이 때 선택된 모드는 Change Mode Display 기능으로 보여질 수 있다.
Pre-Conditions	Set Mode 버튼 입력 상태
Post-Conditions	사용자가 선택한 4개 모드가 Change Mode Display로 선택 될 수 있다.

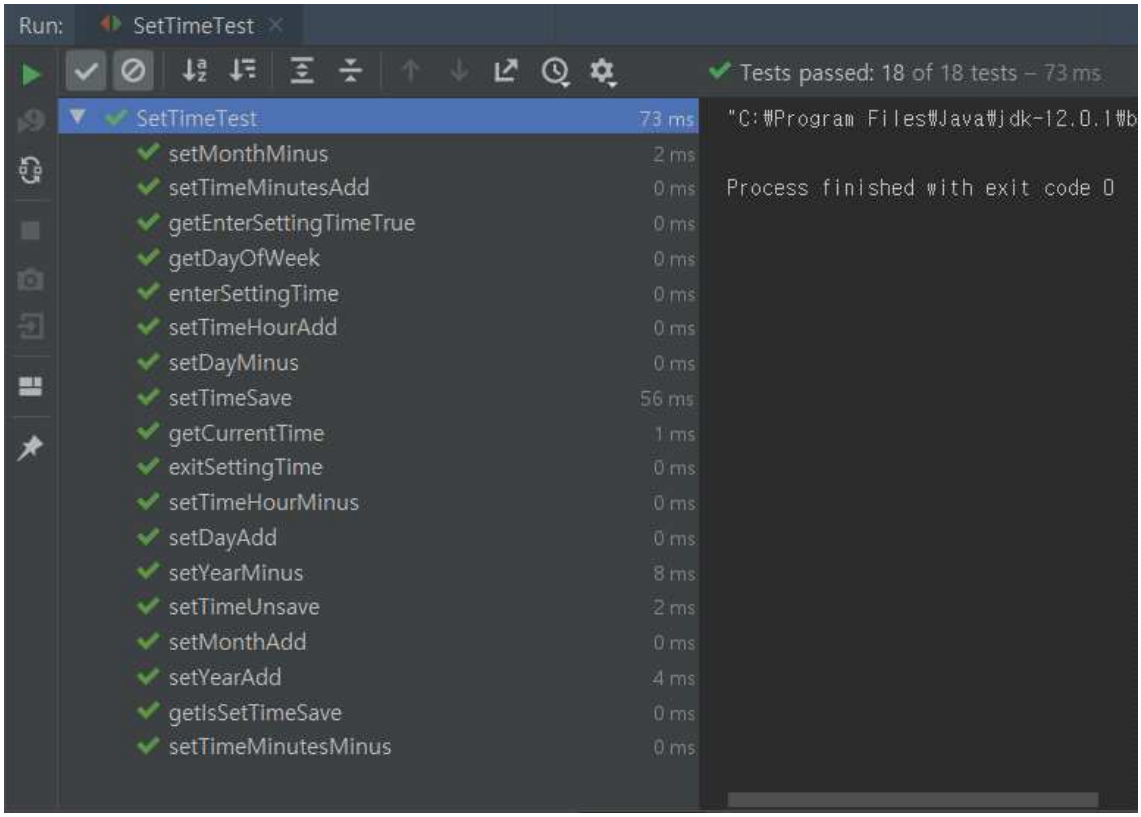
Activity 2055. Write Unit Test Code

Set Time

```
public void setMonthMinus() {
}

@Test
public void setYearAdd() {
    SetTime st = new SetTime();
    String testTime;
    StringTokenizer st0, st1;
    Calendar time = Calendar.getInstance();
    SimpleDateFormat simpleDateFormat;

    st.setYearAdd();
    time.add(Calendar.YEAR, amount: 1);
    simpleDateFormat = new SimpleDateFormat( pattern: "yy-MM-dd HH:mm:ss");
    testTime = simpleDateFormat.format(time.getTime());
    st0= new StringTokenizer(st.getCurrentTime(), delim: " :-");
    st1 = new StringTokenizer(testTime, delim: " :-");
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    //assertEquals(st0.nextToken(),st1.nextToken());
}
```

```java
@Test
public void setTimeUnsave() {
    SetTime st = new SetTime();
    String testTime;
    StringTokenizer st0, st1;
    Calendar time = Calendar.getInstance();
    SimpleDateFormat simpleDateFormat;

    st.setYearAdd();
    st.setMonthMinus();
    st.setTimeMinutesMinus();
    st.setTimeMinutesMinus();
    st.setTimeMinutesMinus();
    st.setTimeUnsave();


    simpleDateFormat = new SimpleDateFormat( pattern: "yy-MM-dd HH:mm:ss");
    testTime = simpleDateFormat.format(time.getTime());
    st0= new StringTokenizer(st.getCurrentTime(), delim: " :-");
    st1 = new StringTokenizer(testTime, delim: " :-");
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
  //  assertEquals(st0.nextToken(),st1.nextToken());
}
```

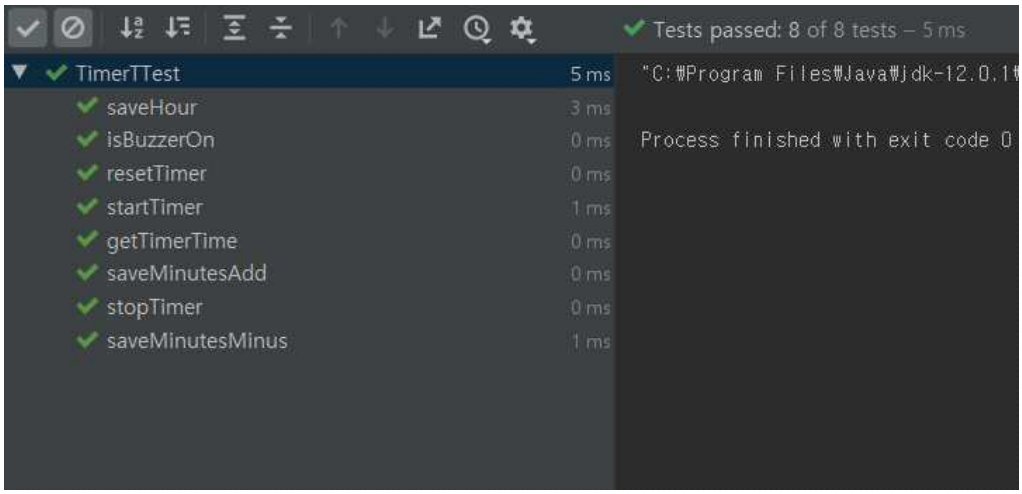
```java
@Test
public void getCurrentTime() {
    SetTime st = new SetTime();
    String testTime;
    StringTokenizer st0, st1;
    Calendar time = Calendar.getInstance();
    SimpleDateFormat simpleDateFormat;


    simpleDateFormat = new SimpleDateFormat( pattern: "yy-MM-dd HH:mm:ss");
    testTime = simpleDateFormat.format(time.getTime());
    st0= new StringTokenizer(st.getCurrentTime(), delim: " :-");
    st1 = new StringTokenizer(testTime, delim: " :-");
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
    assertEquals(st0.nextToken(),st1.nextToken());
  //  assertEquals(st0.nextToken(),st1.nextToken());
}
```


Run:
SetTimeTest
Tests passed: 18 of 18 tests – 73 ms
SetTimeTest 73 ms
setMonthMinus 2 ms
setTimeMinutesAdd 0 ms
getEnterSettingTimeTrue 0 ms
getDayOfWeek 0 ms
enterSettingTime 0 ms
setTimeHourAdd 0 ms
setDayMinus 0 ms
setTimeSave 56 ms
getCurrentTime 1 ms
exitSettingTime 0 ms
setTimeHourMinus 0 ms
setDayAdd 0 ms
setYearMinus 8 ms
setTimeUnsave 2 ms
setMonthAdd 0 ms
setYearAdd 4 ms
getIsSetTimeSave 0 ms
setTimeMinutesMinus 0 ms
"C:₩Program Files₩Java₩jdk-12.0.1₩b
Process finished with exit code 0

Timer

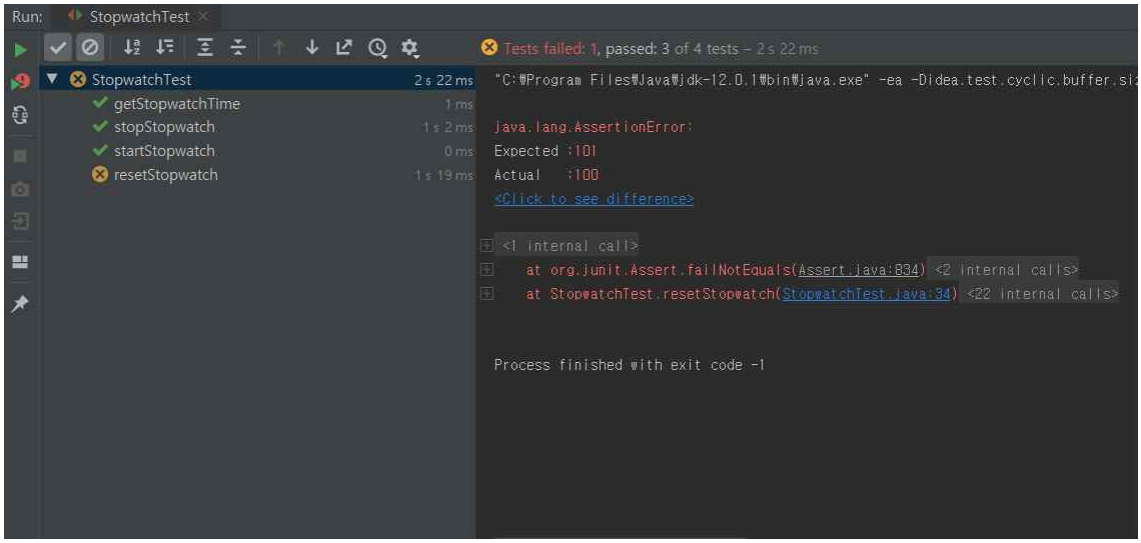
```
@Test
public void saveMinutesAdd() {
    TimerT tImerT = new TimerT();
    tImerT.saveMinutesAdd();
    assertEquals(tImerT.getTimerTime(),  actual: 60);
    tImerT.saveMinutesAdd();
    assertEquals(tImerT.getTimerTime(),  actual: 120);
}

@Test
public void saveMinutesMinus() {
    TimerT tImerT = new TimerT();
    tImerT.saveMinutesAdd();
    assertEquals(tImerT.getTimerTime(),  actual: 60);
    tImerT.saveMinutesAdd();
    assertEquals(tImerT.getTimerTime(),  actual: 120);
    tImerT.saveMinutesMinus();
    assertEquals(tImerT.getTimerTime(),  actual: 60);
}

@Test
public void saveHour() {
    TimerT timerT =new TimerT();
    timerT.saveHour();
    assertEquals(timerT.getTimerTime(),  actual: 3600);
    timerT.saveHour();
    assertEquals(timerT.getTimerTime(),  actual: 0);
}
```

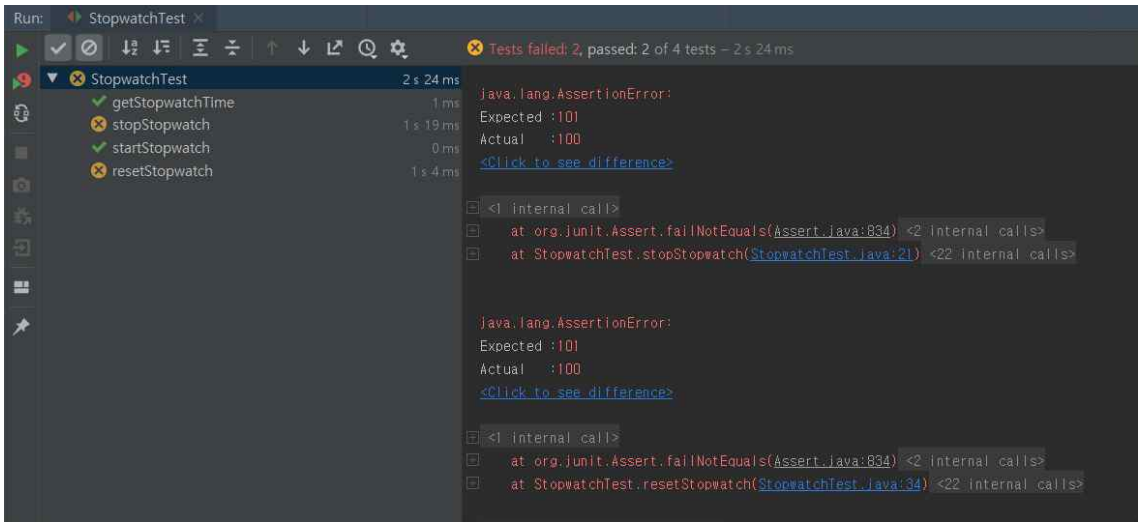
```
@Test
public void resetTimer() {
    TimerT timerT =new TimerT();
    timerT.saveHour();
    assertEquals(timerT.getTimerTime(),  actual: 3600);
    timerT.saveMinutesAdd();
    assertEquals(timerT.getTimerTime(),  actual: 3660);
    timerT.resetTimer();
    assertEquals(timerT.getTimerTime(),  actual: 0);
}
```

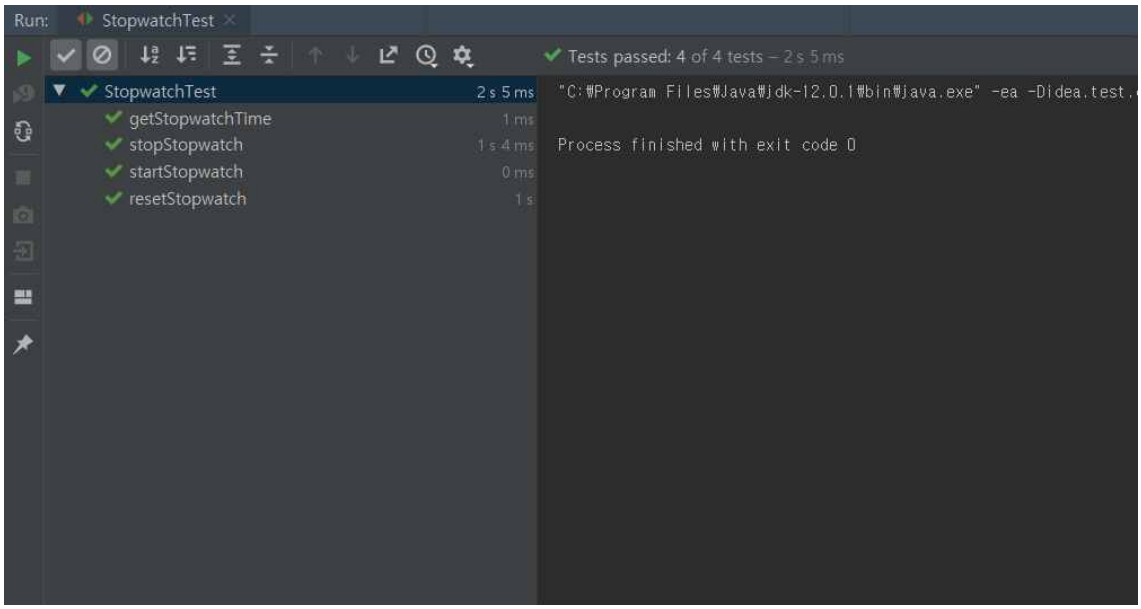


Stopwatch

```
@Test
public void stopStopwatch() {
    Stopwatch st = new Stopwatch();
    st.startStopwatch();
    try {
        Thread.sleep( millis: 1000);
    }catch (InterruptedException e){
        e.getMessage();
    }
    st.stopStopwatch();
    assertEquals(st.getStopwatchTime(), actual: 100);
}

@Test
public void resetStopwatch() {
    Stopwatch st = new Stopwatch();
    st.startStopwatch();
    try {
        Thread.sleep( millis: 1000);
    }catch (InterruptedException e){
        e.getMessage();
    }
    st.stopStopwatch();
    assertEquals(st.getStopwatchTime(), actual: 100);
    st.resetStopwatch();
    assertEquals(st.getStartStopwatch(), actual: false);
    assertEquals(st.getStopwatchTime(), actual: 0);

}
```


Run: StopwatchTest
Tests failed: 1, passed: 3 of 4 tests – 2 s 22 ms
StopwatchTest 2 s 22 ms
getStopwatchTime 1 ms
stopStopwatch 1 s 2 ms
startStopwatch 0 ms
resetStopwatch 1 s 19 ms
"C:₩Program Files₩Java₩jdk-12.0.1₩bin₩java.exe" -ea -Didea.test.cyclic.buffer.si
java.lang.AssertionError:
Expected :101
Actual :100
<Click to see difference>
<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
at StopwatchTest.resetStopwatch(StopwatchTest.java:34) <22 internal calls>
Process finished with exit code -1

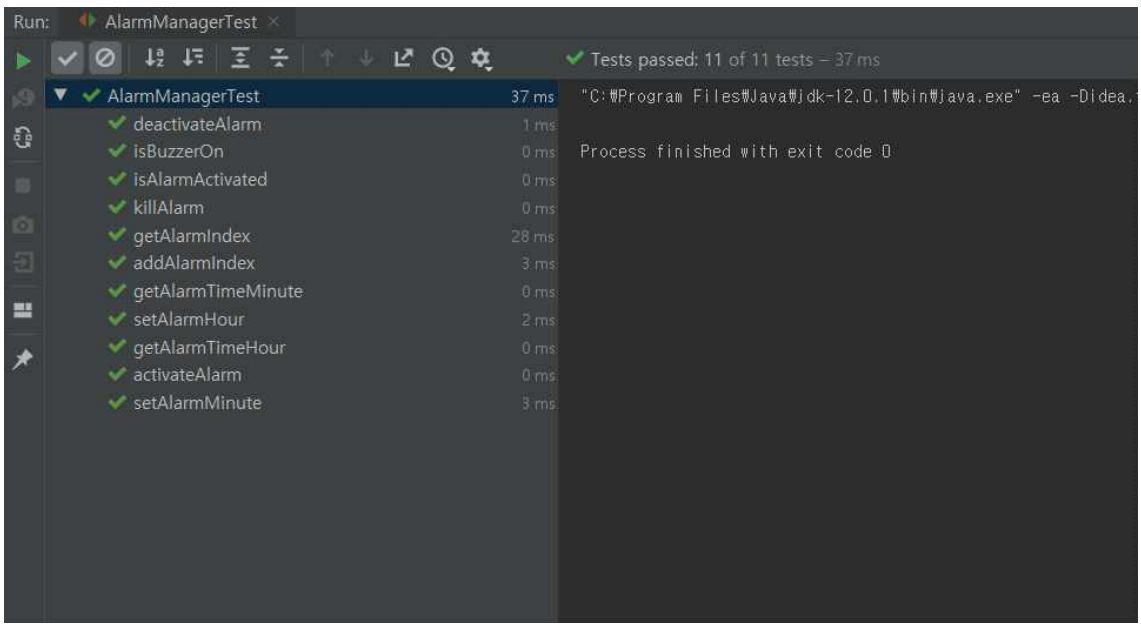

Run: StopwatchTest
Tests failed: 2, passed: 2 of 4 tests – 2 s 24 ms
StopwatchTest 2 s 24 ms
getStopwatchTime 1 ms
stopStopwatch 1 s 19 ms
startStopwatch 0 ms
resetStopwatch 1 s 4 ms
java.lang.AssertionError:
Expected :101
Actual :100
<Click to see difference>
<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
at StopwatchTest.stopStopwatch(StopwatchTest.java:21) <22 internal calls>
java.lang.AssertionError:
Expected :101
Actual :100
<Click to see difference>
<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
at StopwatchTest.resetStopwatch(StopwatchTest.java:34) <22 internal calls>


Run: StopwatchTest
Tests passed: 4 of 4 tests – 2 s 5 ms
StopwatchTest 2 s 5 ms
getStopwatchTime 1 ms
stopStopwatch 1 s 4 ms
startStopwatch 0 ms
resetStopwatch 1 s
"C:₩Program Files₩Java₩jdk-12.0.1₩bin₩java.exe" -ea -Didea.test.
Process finished with exit code 0

Alarm Manager

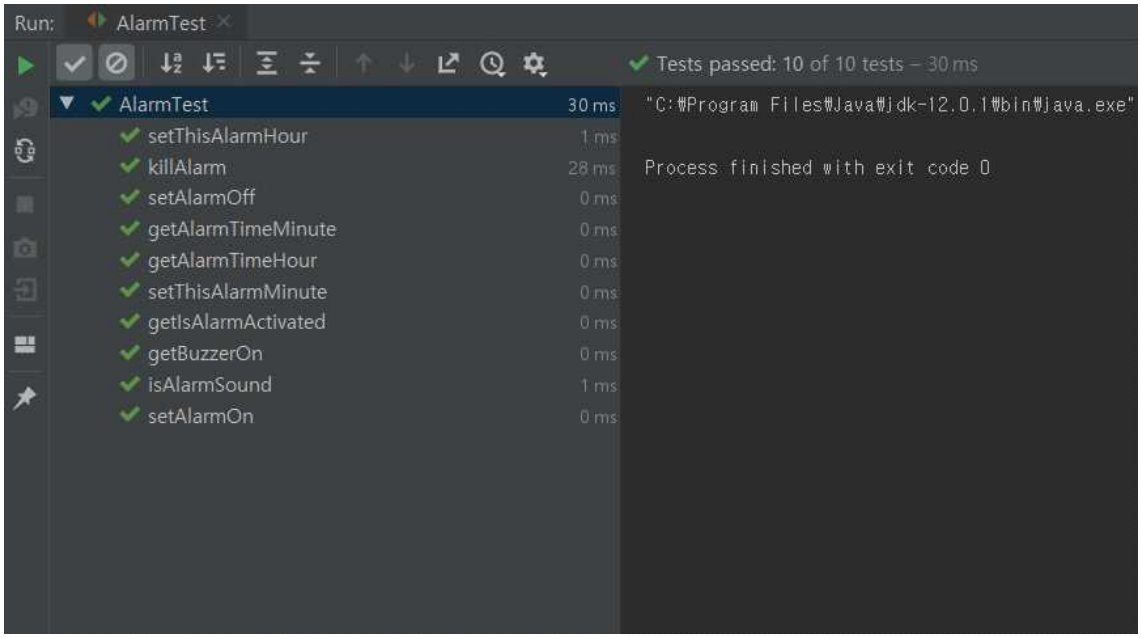
```
@Test
public void getAlarmIndex() {
    AlarmManager am = new AlarmManager();
    assertEquals(am.getAlarmIndex(), actual: 0);
}

@Test
public void addAlarmIndex() {
    AlarmManager am = new AlarmManager();
    am.addAlarmIndex();
    assertEquals(am.getAlarmIndex(), actual: 1);
    am.addAlarmIndex();
    assertEquals(am.getAlarmIndex(), actual: 2);
    am.addAlarmIndex();
    assertEquals(am.getAlarmIndex(), actual: 3);
    am.addAlarmIndex();
    assertEquals(am.getAlarmIndex(), actual: 0);
}
```

```java
public void setAlarmHour() {
    AlarmManager am = new AlarmManager();
    int i=0;
    am.alarm0.setThisAlarmHour();
    assertEquals(am.getAlarmTimeHour(),  actual: 1);
    i++;
    while(i<24){
        am.alarm0.setThisAlarmHour();

        i++;
        if(i != 24) {
            assertEquals(am.getAlarmTimeHour(), i);
        }
    }
    assertEquals(am.getAlarmTimeHour(),  actual: 0);
}

@Test
public void setAlarmMinute() {
    AlarmManager am = new AlarmManager();
    int i=0;
    am.alarm0.setThisAlarmMinute();
    assertEquals(am.getAlarmTimeMinute(),  actual: 1);
    i++;
    while(i<60){
        am.alarm0.setThisAlarmMinute();
        i++;
        if(i != 60){
            assertEquals(am.getAlarmTimeMinute(), i);
        }
    }
    assertEquals(am.getAlarmTimeMinute(),  actual: 0);
```



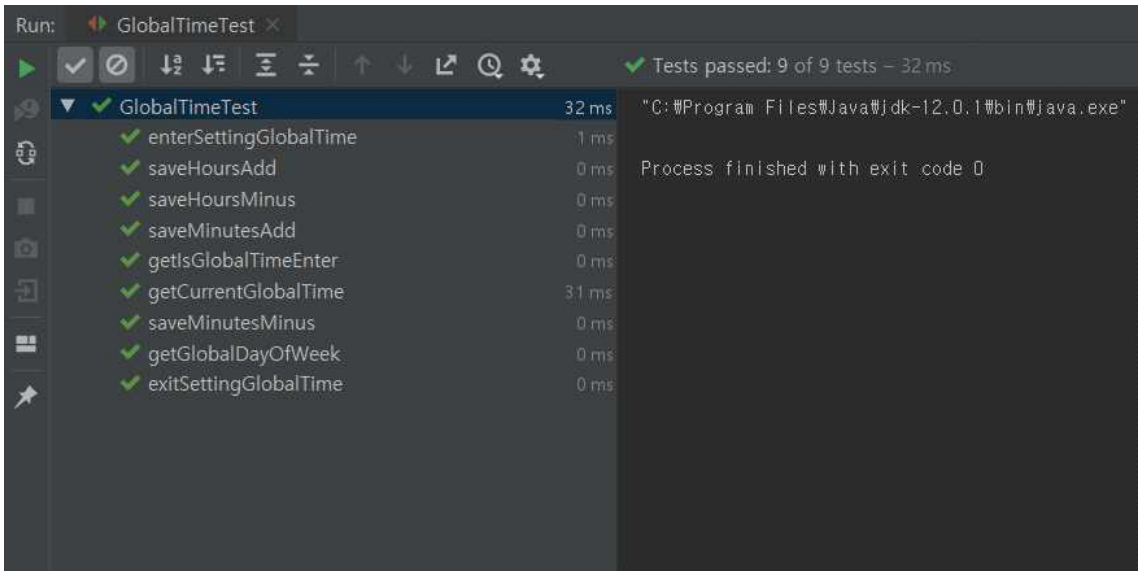
Alarm

```java
@Test
public void isAlarmSound() {
    Alarm am = new Alarm();
    am.setThisAlarmHour();
    am.setThisAlarmMinute();
    am.isAlarmSound( currentHour: 1, currentMinute: 1);
    assertEquals(am.getBuzzerOn(),  actual: true);
}

@Test
public void setThisAlarmHour() {
}

@Test
public void setThisAlarmMinute() {
}

@Test
public void setAlarmOn() {
}

@Test
public void setAlarmOff() {
}

@Test
public void killAlarm() {
    Alarm am = new Alarm();
    am.setThisAlarmHour();
    am.setThisAlarmMinute();
    am.isAlarmSound( currentHour: 1, currentMinute: 1);
    am.killAlarm();
    assertEquals(am.getBuzzerOn(),  actual: false);
}
```



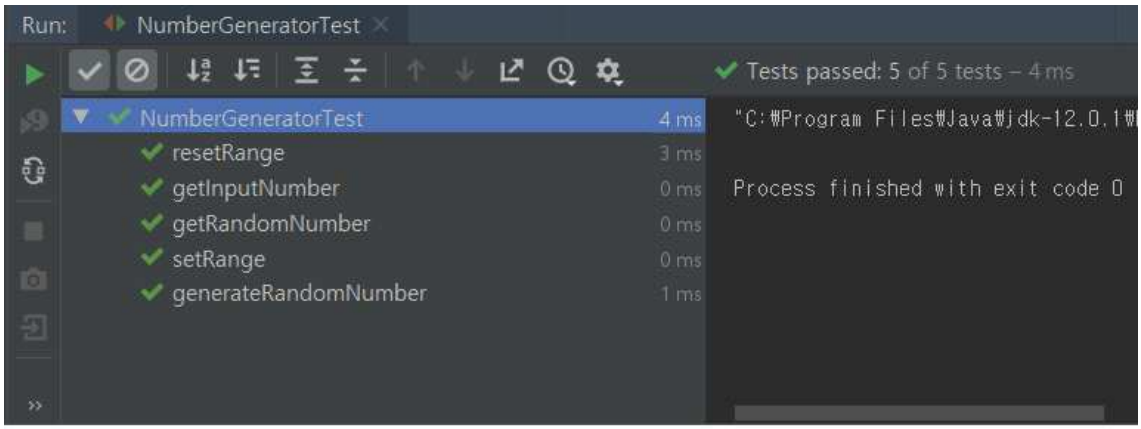
Global Time

```
@Test
public void getCurrentGlobalTime() {
    GlobalTime gt = new GlobalTime();
    String testTime;
    StringTokenizer st, st1;
    Calendar time = Calendar.getInstance();

    SimpleDateFormat simpleDateFormat;
    simpleDateFormat = new SimpleDateFormat( pattern: "yy-MM-dd HH:mm:ss");
    testTime = simpleDateFormat.format(time.getTime());
    st= new StringTokenizer(gt.getCurrentGlobalTime(), delim: " :-");
    st1 = new StringTokenizer(testTime, delim: " :-");
    assertEquals(st.nextToken(),st1.nextToken());
    assertEquals(st.nextToken(),st1.nextToken());
    assertEquals(st.nextToken(),st1.nextToken());
    assertEquals(st.nextToken(),st1.nextToken());
    assertEquals(st.nextToken(),st1.nextToken());
    //assertEquals(st.nextToken(),st1.nextToken());          // 초가 2초정도 차이가 난다...!!
}

@Test
public void getGlobalDayOfWeek() {
}
}
```


Run:
GlobalTimeTest
Tests passed: 9 of 9 tests – 32 ms
GlobalTimeTest 32 ms
enterSettingGlobalTime 1 ms
saveHoursAdd 0 ms
saveHoursMinus 0 ms
saveMinutesAdd 0 ms
getIsGlobalTimeEnter 0 ms
getCurrentGlobalTime 31 ms
saveMinutesMinus 0 ms
getGlobalDayOfWeek 0 ms
exitSettingGlobalTime 0 ms
"C:₩Program Files₩Java₩jdk-12.0.1₩bin₩java.exe"
Process finished with exit code 0

Random Number Generator Test Code

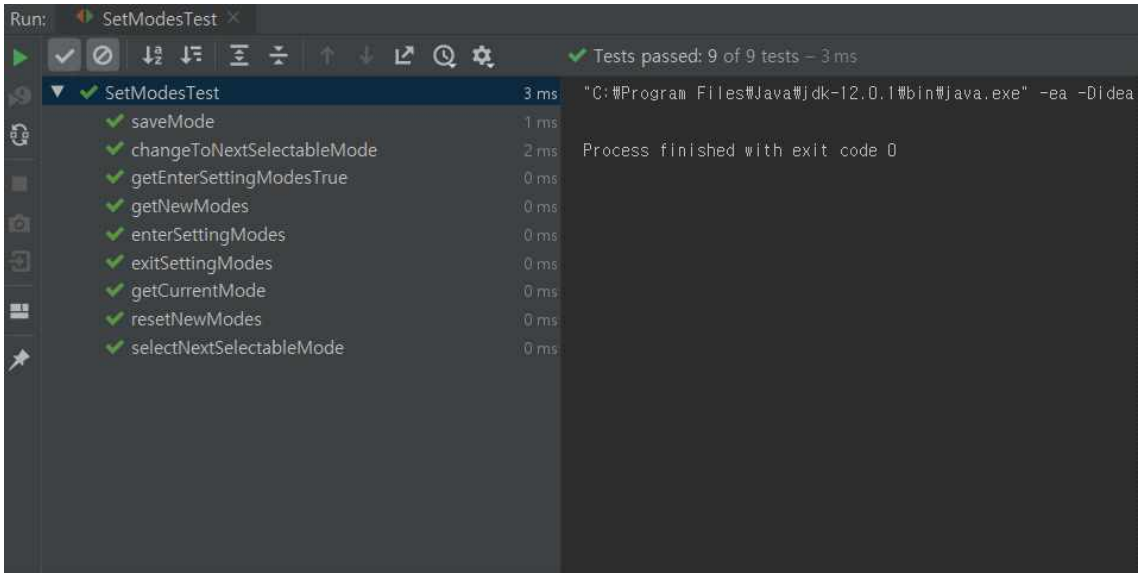
```java
@Test
public void resetRange() {
    NumberGenerator ng = new NumberGenerator();
    ng.resetRange();
    assertEquals(ng.getInputNumber(),  actual: 0);
}

@Test
public void getInputNumber() {
}

@Test
public void getRandomNumber() {
}

@Test
public void generateRandomNumber() {
    NumberGenerator ng = new NumberGenerator();
    int i=0;
    while(i<11) {
        ng.setRange();
        i++;
    }
    ng.generateRandomNumber();
    testRandomNumber=ng.getRandomNumber();
    assertTrue( condition: testRandomNumber<10);
}
```



Set Mode

```
@Test
public void selectNextSelectableMode() {
    SetModes sm = new SetModes();
    assertEquals(sm.getCurrentMode(), actual: 0);
    sm.selectNextSelectableMode(sm.getCurrentMode());
    assertEquals(sm.getCurrentMode(), actual: 1);
    sm.selectNextSelectableMode(sm.getCurrentMode());
    assertEquals(sm.getCurrentMode(), actual: 2);
    sm.selectNextSelectableMode(sm.getCurrentMode());
    assertEquals(sm.getCurrentMode(), actual: 3);
    sm.selectNextSelectableMode(sm.getCurrentMode());
    assertEquals(sm.getCurrentMode(), actual: 0);
}

@Test
public void changeToNextSelectableMode() {
    SetModes sm = new SetModes();
    sm.saveMode();
    sm.changeToNextSelectableMode();
    sm.saveMode();
    sm.changeToNextSelectableMode();
    sm.saveMode();
    sm.changeToNextSelectableMode();

    assertArrayEquals(sm.getNewModes(), new boolean[]{true, true, true, true, false, false});
}
```

```
@Test
public void resetNewModes() {
    SetModes sm = new SetModes();
    sm.resetNewModes();
    assertArrayEquals(sm.getNewModes(), new boolean[]{true, false, false, false, false, false});
}
```


Run:
SetModesTest
Tests passed: 9 of 9 tests – 3 ms
SetModesTest 3 ms
saveMode 1 ms
changeToNextSelectableMode 2 ms
getEnterSettingModesTrue 0 ms
getNewModes 0 ms
enterSettingModes 0 ms
exitSettingModes 0 ms
getCurrentMode 0 ms
resetNewModes 0 ms
selectNextSelectableMode 0 ms
"C:₩Program Files₩Java₩jdk-12.0.1₩bin₩java.exe" -ea -Idea
Process finished with exit code 0