디데이 시계

oopt Stage 2050, 2060

progect Team : 3 team
date : 2020/06/08

201611188 김동곤
201711337 이희광
201413146 양영준
201614150 김지현

2055. Write Unit Test Code

Unit Test <TimeKeeping> - 13개

```java
class TimeKeepingTest {

    @Test
    void showTimeKeepingtest() {
        TimeKeep tk = new TimeKeep();
        tk.showTimeKeeping();
        Calendar current_time = tk.gettime();
        assertNotNull(current_time);
    }

    @Test
    void getddaytest(){
        TimeKeep tk = new TimeKeep();
        assertEquals( expected: null,tk.getdday());
    }

    @Test
    void setddaytest(){
        TimeKeep tk = new TimeKeep();
        tk.setdday(null);
        assertEquals( expected: null,tk.getoriginTime());
        assertEquals( expected: null,tk.getddaymemoTime());
    }

    @Test
    void gettimetest(){
        TimeKeep tk = new TimeKeep();
        assertEquals( expected: null,tk.gettime());
    }
```

```java
@Test
void getRealTimetest(){
    TimeKeep tk = new TimeKeep();
    assertEquals( expected: null,tk.getRealTime());
}


@Test
void setCurrentTimetest(){
    TimeKeep tk = new TimeKeep();
    tk.showTimeKeeping();
    tk.setCurrentTime();
    assertEquals( expected: 1,tk.get_flag());
    assertEquals( expected: 0,tk.getCursor());
}

@Test
void setActiveFunctiontest(){
    TimeKeep tk = new TimeKeep();
    tk.setActiveFunction();
    assertEquals( expected: 2,tk.get_flag());
}

@Test
void get_flagtest(){
    TimeKeep tk = new TimeKeep();
    assertEquals( expected: null,tk.gettime());
}
```

```java
@Test
void getCursortest(){
    TimeKeep tk = new TimeKeep();
    assertEquals( expected: null,tk.getRealTime());
}


@Test
void addsecondstest(){
    TimeKeep tk = new TimeKeep();
    tk.showTimeKeeping();
    Calendar current_time = (Calendar) tk.gettime().clone();
    System.out.println("currenttiem " +current_time.get(Calendar.SECOND));
    tk.addseconds();
    assertEquals( expected: current_time.get(Calendar.SECOND)+1,
            tk.gettime().get(Calendar.SECOND));
}


@Test
void moveCursorTimertest(){
    TimeKeep tk = new TimeKeep();
    tk.moveCursor_time();
    int cur_cursor = tk.getCursor();
    assertEquals( expected: 0,cur_cursor);
}
```

```java
@Test
void plusTime_timetest(){
    TimeKeep tk = new TimeKeep();
    JButton button = new JButton();
    tk.showTimeKeeping();
    button.setText("Button1");//set모드
    tk.work(button);
    Calendar setting_time = (Calendar) tk.gettime().clone();
    button.setText("Button2");//1증가
    tk.work(button);
    button.setText("Button4");//확인
    tk.work(button);
    assertEquals( expected: setting_time.get(Calendar.MONTH)+1,
            tk.gettime().get(Calendar.MONTH));
}


@Test
void confirmTimetest(){
    TimeKeep tk = new TimeKeep();
    JButton button = new JButton();
    tk.showTimeKeeping();
    button.setText("Button1");//settime
    tk.work(button);
    button.setText("Button4");//confirm
    tk.work(button);
    assertEquals( expected: 0,tk.get_flag());
    assertEquals( expected: -1,tk.getCursor());
}
```

Unit Test <Alarm> - 5개

```java
@Test
void testCursor() {
    Alarm al = new Alarm();
    JButton btn1 = new JButton( text: "Button1");
    JButton btn4 = new JButton( text: "Button4");
    //초기 커서 위치 테스트
    assertEquals( expected: -1,al.getCursor());

    //커서 위치 이동 테스트
    al.work(btn1);
    assertEquals( expected: 3,al.getCursor());

    al.work(btn1);
    assertEquals( expected: 4,al.getCursor());

    al.work(btn1);
    assertEquals( expected: 5,al.getCursor());

    al.work(btn1);
    assertEquals( expected: 3,al.getCursor());

    //초기 커서 위치 돌아오는지 테스트
    al.work(btn4);
    assertEquals( expected: -1,al.getCursor());


}
```

```java
@Test
void testsetAlarm() {
    Alarm al = new Alarm();
    JButton btn1 = new JButton( text: "Button1");
    JButton btn4 = new JButton( text: "Button4");
    //초기 플래그값 확인
    assertEquals( expected: 0,al.get_flag());

    //set 플래그 확인
    al.work(btn1);
    assertEquals( expected: 1,al.get_flag());
    //설정화면 나왔을때.
    al.work(btn4);
    assertEquals( expected: 0,al.get_flag());

}
```

```java
@Test
void testconfirmAlarm(){
    Alarm al = new Alarm();
    JButton btn1 = new JButton( text: "Button1");
    JButton btn4 = new JButton( text: "Button4");

    //세팅 모드 진입
    al.work(btn1);
    assertEquals( expected: 1,al.get_flag());
    assertEquals( expected: 3,al.getCursor());

    //confirm 후
    al.work(btn4);
    assertEquals( expected: 0,al.get_flag());
    assertEquals( expected: -1,al.getCursor());
}
```

```java
@Test
void testplusTime_alarm(){
    Alarm al = new Alarm();
    JButton btn1 = new JButton( text: "Button1");
    JButton btn2 = new JButton( text: "Button2");
    //세팅 모드 진입
    al.work(btn1);
    assertEquals( expected: 0,al.getAlarm().get(Calendar.HOUR_OF_DAY));
    assertEquals( expected: 0,al.getAlarm().get(Calendar.MINUTE));
    assertEquals( expected: 0,al.getAlarm().get(Calendar.SECOND));
    //시 plus 후
    al.work(btn2);
    assertEquals( expected: 1,al.getAlarm().get(Calendar.HOUR_OF_DAY));
    assertEquals( expected: 0,al.getAlarm().get(Calendar.MINUTE));
    assertEquals( expected: 0,al.getAlarm().get(Calendar.SECOND));

    //분 plus 후
    al.work(btn1);
    al.work(btn2);
    assertEquals( expected: 1,al.getAlarm().get(Calendar.HOUR_OF_DAY));
    assertEquals( expected: 1,al.getAlarm().get(Calendar.MINUTE));
    assertEquals( expected: 0,al.getAlarm().get(Calendar.SECOND));

    //초 plus 후
    al.work(btn1);
    al.work(btn2);
    assertEquals( expected: 1,al.getAlarm().get(Calendar.HOUR_OF_DAY));
    assertEquals( expected: 1,al.getAlarm().get(Calendar.MINUTE));
    assertEquals( expected: 1,al.getAlarm().get(Calendar.SECOND));
}
```

```java
@Test
void testdeleteAlarm(){
    Alarm al = new Alarm();
    JButton btn1 = new JButton( text: "Button1");
    JButton btn4 = new JButton( text: "Button4");

    //세팅 호출 및 저장
    al.work(btn1);
    al.work(btn4);

    //저장 값  not null 확인
    assertNotNull(al.getAlarm());

    //리셋 호출
    al.work(btn4);
    //값 null 확인
    assertNull(al.getAlarm());
}
```

Unit Test <World Time> - 개

```java
@Test
void changeCountryTest(){
    WorldTime worldTime = new WorldTime();
    JButton jbutton = new JButton( text: "");
    jbutton.setText("Button1");

    // init
    int cursor = -1;
    assertEquals(cursor, worldTime.getCursor());

    // after changeCountry() Method
    worldTime.work(jbutton);
    cursor = 2;
    assertEquals(cursor, worldTime.getCursor());
}
```

```java
@Test
void nextCountryTest(){
    WorldTime worldTime = new WorldTime();
    JButton jbutton = new JButton( text: "");
    jbutton.setText("Button2");

    //init
    Queue<String> countriesQ = new LinkedList();
    countriesQ.offer( e: "LIS");
    countriesQ.offer( e: "LON");
    countriesQ.offer( e: "ROM");
    countriesQ.offer( e: "TYO");
    countriesQ.offer( e: "LAX");
    countriesQ.offer( e: "DEN");
    assertEquals(countriesQ, worldTime.get_countriesQ());

    String country = "SEL";
    assertEquals(country, worldTime.get_key());

    //after nextCountry() Method
    countriesQ.offer(country);
    country = countriesQ.poll();
    jbutton.setText("Button1");
    worldTime.work(jbutton);
    jbutton.setText("Button2");
    worldTime.work(jbutton);
    assertEquals(countriesQ, worldTime.get_countriesQ());
    assertEquals(country, worldTime.get_key());
}
```

```java
@Test
void changeModeTest(){
    WorldTime worldTime = new WorldTime();
    JButton jbutton = new JButton( text: "");
    jbutton.setText("Button3");

    worldTime.work(jbutton);
    Queue<String> countriesQ = new LinkedList();
    countriesQ.offer( e: "LIS");
    countriesQ.offer( e: "LON");
    countriesQ.offer( e: "ROM");
    countriesQ.offer( e: "TYO");
    countriesQ.offer( e: "LAX");
    countriesQ.offer( e: "DEN");
    assertEquals(countriesQ, worldTime.get_countriesQ());

    String country = "SEL";
    assertEquals(country, worldTime.get_key());

    int cursor = -1;
    assertEquals(cursor, worldTime.getCursor());
}
```

```java
@Test
void confirmCountryTest(){
    WorldTime worldTime = new WorldTime();
    JButton jbutton = new JButton( text: "");

    //changeCountry()
    jbutton.setText("Button1");
    worldTime.work(jbutton);

    //nexCountry()
    jbutton.setText("Button2");
    worldTime.work(jbutton);

    //confirmCountry
    jbutton.setText("Button4");
    worldTime.work(jbutton);

    int cursor = -1;
    assertEquals(cursor, worldTime.getCursor());

    Queue<String> countriesQ = new LinkedList();
    countriesQ.offer( e: "LON");
    countriesQ.offer( e: "ROM");
    countriesQ.offer( e: "TYO");
    countriesQ.offer( e: "LAX");
    countriesQ.offer( e: "DEN");
    countriesQ.offer( e: "SEL");
    assertEquals(countriesQ, worldTime.get_countriesQ_temp());

    String country = "LIS";
    assertEquals(country, worldTime.get_key());
}
```

```java
@Test
void get_keyTest(){
    WorldTime worldTime = new WorldTime();
    String initCountry = "SEL";
    assertEquals(initCountry, worldTime.get_key());
}
```

```java
@Test
void get_valueTest(){
    WorldTime worldTime = new WorldTime();
    Map<String, String> countries = new HashMap<~>();
    countries.put("SEL", "Asia/Seoul");
    String value = countries.get("SEL");

    assertEquals(value, worldTime.get_value());
}
```

```java
@Test
void getCursorTest(){
    WorldTime worldTime = new WorldTime();
    int cursor = -1;
    assertEquals(cursor, worldTime.getCursor());
}
```

Unit Test <StopWatch> - 7개

```java
class StopWatchTest {

    @Test
    void resetStopWatchTest() {
        Watch watch=new Watch();
        StopWatch st=new StopWatch();

        st.setStart(12);
        st.setEnd(22);
        st.setElapse(10);
        st.setElapsePrevious(1);
        st.setOn(true);

        Calendar cal=Calendar.getInstance();
        cal.set(Calendar.HOUR, 9);
        cal.set(Calendar.MINUTE, 2);
        cal.set(Calendar.SECOND, 5);

        st.setCal(cal);

        Calendar calLap=Calendar.getInstance();
        calLap.set(Calendar.HOUR, 5);
        calLap.set(Calendar.MINUTE, 2);
        calLap.set(Calendar.SECOND, 2);
```

```java
        st.setCalLap(calLap);
        assertEquals( expected: 12,st.getStart());
        assertEquals( expected: 22,st.getEnd());
        assertEquals( expected: 10,st.getElapse());
        assertEquals( expected: 1,st.getElapsePrevious());
        assertEquals( expected: true,st.isOn());

        assertEquals( expected: 9,st.getCal().get(Calendar.HOUR));
        assertEquals( expected: 2,st.getCal().get(Calendar.MINUTE));
        assertEquals( expected: 5,st.getCal().get(Calendar.SECOND));
        assertEquals( expected: 5,st.getCalLap().get(Calendar.HOUR));
        assertEquals( expected: 2,st.getCalLap().get(Calendar.MINUTE));
        assertEquals( expected: 2,st.getCalLap().get(Calendar.SECOND));

        JButton button=new JButton();
        button.setText("Button4");

        st.work(button);
        st.work(button);

        assertEquals( expected: 0,st.getStart());
        assertEquals( expected: 0,st.getEnd());
        assertEquals( expected: 0,st.getElapse());
        assertEquals( expected: 0,st.getElapsePrevious());
        assertEquals( expected: false,st.isOn());

            assertEquals( expected: 0,st.getCal().get(Calendar.HOUR));
            assertEquals( expected: 0,st.getCal().get(Calendar.MINUTE));
            assertEquals( expected: 0,st.getCal().get(Calendar.SECOND));
            assertEquals( expected: 0,st.getCalLap().get(Calendar.HOUR));
            assertEquals( expected: 0,st.getCalLap().get(Calendar.MINUTE));
            assertEquals( expected: 0,st.getCalLap().get(Calendar.SECOND));


    }


    @Test
    void startStopWatchTest(){

        Watch watch=new Watch();
        StopWatch st=new StopWatch();

        st.setOn(false);

        JButton button=new JButton();
        button.setText("Button2");
        st.work(button);

        assertEquals( expected: true,st.isOn());

    }
```

```java
@Test
void stopStopWatchTest(){
    Watch watch=new Watch();
    StopWatch st=new StopWatch();

    st.setOn(true);

    JButton button=new JButton();
    button.setText("Button2");
    st.work(button);


    assertEquals( expected: false,st.isOn());

}


@Test
void stopStopWatchTest(){
    Watch watch=new Watch();
    StopWatch st=new StopWatch();

    st.setOn(true);

    JButton button=new JButton();
    button.setText("Button2");
    st.work(button);


    assertEquals( expected: false,st.isOn());

}

@Test
void storeLapTimeTest(){
    Watch watch=new Watch();
    StopWatch st=new StopWatch();

    st.setOn(true);

    st.setElapse(36000);


    JButton button=new JButton();
    button.setText("Button1");
    st.work(button);

    assertEquals( expected: 10,st.getCalLap().get(Calendar.HOUR));
    assertEquals( expected: 0,st.getCalLap().get(Calendar.MINUTE));
    assertEquals( expected: 0,st.getCalLap().get(Calendar.SECOND));

}
```

```java
void getStopWatchTestInON(){

    Watch watch=new Watch();
    StopWatch st=new StopWatch();

    JButton button=new JButton();
    button.setText("Button4");
    st.work(button);

    st.setOn(true);

    //      this.end=calculateTime(this.end,this.cal);
    //          this.elapse=this.elapsePrevious+(this.end-this.start);
    //          splitElapse(elapse,this.cal);

    st.setElapsePrevious(1800);
    st.setStart(0);
    st.setEnd(1800);
    st.calculateTime(st.getEnd(),st.getCal());
    st.setElapse(st.getElapsePrevious()+(st.getEnd()-st.getStart()));
    st.splitElapse(st.getElapse(),st.getCal());

    assertEquals( expected: 1,st.getCal().get(Calendar.HOUR));
    assertEquals( expected: 0,st.getCal().get(Calendar.MINUTE));
    assertEquals( expected: 0,st.getCal().get(Calendar.SECOND));
}

@Test
void getStopWatchTestInOFF(){

    Watch watch=new Watch();
    StopWatch st=new StopWatch();


    st.setOn(false);


    Calendar cal=st.getStopWatch();

    assertEquals( expected: 0,cal.get(Calendar.HOUR));
    assertEquals( expected: 0,cal.get(Calendar.MINUTE));
    assertEquals( expected: 0,cal.get(Calendar.SECOND));


}
```

Unit Test <D-day> - 6개

```java
class DdayTest {

    @Test
    void setDdayTest() {
        Watch watch=new Watch();
        Dday d=new Dday();

        d.setFlag_set(0);
        d.setSetting_page(null);

        JButton button=new JButton();
        button.setText("Button1");
        d.work(button);

        assertEquals( expected: 1,d.getFlag_set());
        assertEquals( expected: 0,d.getCur_cursor());

        assertEquals( expected: "AAA",d.getSetting_page().get_memo());
    }
```

```java
@Test
void moveCursor_Dday_Test(){
    Watch watch=new Watch();
    Dday d=new Dday();

    //if && if
    d.setCur_cursor(2);

    d.setString_cur(2);

    d.setFlag_set(1);
    d.setSetting_page(null);

    JButton button=new JButton();
    button.setText("Button1");
    d.work(button);

    assertEquals( expected: 0,d.getString_cur());
    assertEquals( expected: 0,d.getCur_cursor());
    //if&&else
    d.setCur_cursor(2);
    d.setString_cur(1);
    d.setFlag_set(1);
    d.setSetting_page(null);

    d.work(button);

    assertEquals( expected: 2,d.getString_cur());
    assertEquals( expected: 2,d.getCur_cursor());

    d.setString_cur(0);

    //else

    d.setCur_cursor(7);
    d.setFlag_set(1);
    d.setSetting_page(null);


    d.work(button);

    assertEquals( expected: 1,d.getCur_cursor());

}
```

```java
    void showNextDdayTest(){
        Watch watch=new Watch();
        Dday d=new Dday();

        d.setFlag_set(0);
        d.setSetting_page(null);

        JButton button=new JButton();
        button.setText("Button1");
        d.work(button);

        dday_data current_page;

        current_page = new dday_data();

        current_page.set_memo(d.getSetting_page().get_memo());

        d.setCurrent_page(current_page);

         button=new JButton();
        button.setText("Button2");
        d.work(button);

        assertEquals( expected: "AAA",d.getCurrent_page().get_memo());

    }

    @Test
    void plusDayTest(){
        Watch watch=new Watch();
        Dday d=new Dday();

        d.setFlag_set(1);

        d.setCur_cursor(0);
        dday_data setting_page=new dday_data();
        Calendar cal= Calendar.getInstance();
        cal.set(Calendar.MONTH,10);

        setting_page.set_cal(cal);
        setting_page.set_memo("aaa");

        d.setSetting_page(setting_page);

        JButton button=new JButton();
        button.setText("Button2");
        d.work(button);

        assertEquals( expected: 11,d.getSetting_page().get_cal().get(Calendar.MONTH));
    }

    @Test
    void deleteDdayTest(){

        Watch watch=new Watch();
        Dday d=new Dday();

        d.setFlag_set(0);

        dday_data current_page;

        current_page = new dday_data();

        current_page.set_memo("111");

        d.setCurrent_page(current_page);
        d.getDdayQ().clear();

        JButton button=new JButton();
        button.setText("Button4");
        d.work(button);

        assertEquals( expected: null,d.getCurrent_page());
    }
```

```java
@Test
void confirmDdayTest(){

    Watch watch=new Watch();
    Dday d=new Dday();

    d.setFlag_set(1);


    dday_data setting_page=new dday_data();
    Calendar cal= Calendar.getInstance();
    cal.set(Calendar.MONTH,10);

    setting_page.set_cal(cal);
    setting_page.set_memo("no1");
    d.setSetting_page(setting_page);


    JButton button=new JButton();
    button.setText("Button4");
    d.work(button);

    assertEquals( expected: "no1",d.getCurrent_page().get_memo());
}
```

Unit Test <Timer> - 11개

```java
class TimerTest {

    @Test
    void getCursortest() {
        mTimer mtimer = new mTimer();
        int cursor = mtimer.getCursor();
        assertEquals( expected: -1,cursor);
    }

    @Test
    void getFlagtest(){
        mTimer mtimer = new mTimer();
        int flag = mtimer.get_flag();
        assertEquals( expected: 0,flag);
    }

    @Test
    void getPauseFlagtest(){
        mTimer mtimer = new mTimer();
        int flag = mtimer.getPauseFlag();
        assertEquals( expected: 0,flag);
    }

    @Test
    void getTimerTimetest(){
        mTimer mtimer = new mTimer();
        Calendar c = mtimer.getTimerTime();
        assertEquals( expected: null,c);
    }
```

```java
    @Test
    void setTimertest(){
        mTimer mtimer = new mTimer();
        JButton button = new JButton();
        button.setText("Button1");
        mtimer.work(button);
        int flag = mtimer.get_flag();
        assertEquals( expected: 1,flag);
    }

    @Test
    void moveCursorTimertest(){
        mTimer mtimer = new mTimer();
        mtimer.moveCursor_timer();
        int cur_cursor = mtimer.getCursor();
        assertEquals( expected: 3,cur_cursor);
    }
```

```java
@Test
void plusTimertest(){
    mTimer mtimer = new mTimer();
    JButton button = new JButton();
    button.setText("Button1");//set모드
    mtimer.work(button);
    button.setText("Button2");//1증가
    mtimer.work(button);
    button.setText("Button4");//확인
    mtimer.work(button);
    Calendar pre_time = mtimer.getTimer();
    assertEquals( expected: 1,pre_time.get(Calendar.HOUR_OF_DAY));
    assertEquals( expected: 0,pre_time.get(Calendar.MINUTE));
    assertEquals( expected: 0,pre_time.get(Calendar.SECOND));
}

@Test
void confirmTimertest(){
    mTimer mtimer = new mTimer();
    JButton button = new JButton();
    button.setText("Button1");//set모드
    mtimer.work(button);
    button.setText("Button4");//확인
    mtimer.work(button);
    int flag_set = mtimer.get_flag();
    int cur_cursor = mtimer.getCursor();
    assertEquals( expected: 0,flag_set);
    assertEquals( expected: -1,cur_cursor);
}
```

```java
@Test
void stopTimertest(){
    mTimer mtimer = new mTimer();
    JButton button = new JButton();
    button.setText("Button1");//set모드
    mtimer.work(button);
    button.setText("Button2");//1증가
    mtimer.work(button);
    button.setText("Button4");//확인
    mtimer.work(button);
    button.setText("Button4");//stop
    mtimer.work(button);
    Calendar pre_time = mtimer.getTimer();
    assertEquals( expected: null,pre_time);


}

@Test
void startTimertest(){
    mTimer mtimer = new mTimer();
    JButton button = new JButton();
    button.setText("Button1");//set모드
    mtimer.work(button);
    button.setText("Button2");//1증가
    mtimer.work(button);
    button.setText("Button4");//확인
    mtimer.work(button);
    button.setText("Button2");//start
    mtimer.work(button);
    int flag = mtimer.get_flag_sp();
    assertEquals( expected: 1,flag);
}
```

```java
@Test
void pauseTimertest(){
    mTimer mtimer = new mTimer();
    JButton button = new JButton();
    button.setText("Button1");//set모드
    mtimer.work(button);
    button.setText("Button2");//1증가
    mtimer.work(button);
    button.setText("Button4");//확인
    mtimer.work(button);
    button.setText("Button2");//start
    mtimer.work(button);
    button.setText("Button2");//pause
    mtimer.work(button);
    int flag = mtimer.get_flag_sp();
    assertEquals( expected: 0,flag);
}
```

Unit Test <Function Activator> - 8개

```java
@Test
void nextActivateFunctionTest() {
    Queue<Integer> modeQ = new LinkedList<~>();
    modeQ.offer(watch_Type.ALARM.ordinal());
    modeQ.offer(watch_Type.WORLDTIME.ordinal());
    modeQ.offer(watch_Type.STOPWATCH.ordinal());
    FunctionActivator functionActivator = new FunctionActivator(modeQ);

    JButton jButton = new JButton( text: "");
    jButton.setText("Button1");

    //init
    //position == 0
    int position = 0;
    assertEquals(position, functionActivator.get_position());

    //position 0 -> 1
    functionActivator.work(jButton);
    position = 1;
    assertEquals(position, functionActivator.get_position());

    // position 5 -> 0
    functionActivator.work(jButton);
    functionActivator.work(jButton);
    functionActivator.work(jButton);
    functionActivator.work(jButton);

    position = 0;
    assertEquals(position, functionActivator.get_position());
}
```

```java
@Test
void onOffFunctionTest() {
    Queue<Integer> modeQ = new LinkedList<~>();
    modeQ.offer(watch_Type.ALARM.ordinal());
    modeQ.offer(watch_Type.WORLDTIME.ordinal());
    modeQ.offer(watch_Type.STOPWATCH.ordinal());
    FunctionActivator functionActivator = new FunctionActivator(modeQ);

    JButton jButton = new JButton( text: "");
    jButton.setText("Button2");

    int position = 0;
    boolean active_function = true;
    assertEquals(active_function ,functionActivator.get_active_function(position));

    functionActivator.work(jButton);
    active_function = false;
    assertEquals(active_function, functionActivator.get_active_function(position));
}
```

```java
@Test
void confirmActiveTest() {
    Queue<Integer> modeQ = new LinkedList<~>();
    modeQ.offer(watch_Type.ALARM.ordinal());
    modeQ.offer(watch_Type.WORLDTIME.ordinal());
    modeQ.offer(watch_Type.STOPWATCH.ordinal());
    FunctionActivator functionActivator = new FunctionActivator(modeQ);
    JButton jButton = new JButton( text: "");

    //nextActivateFunction()
    //position 0 -> 1
    int position = 1;
    jButton.setText("Button1");
    functionActivator.work(jButton);
    assertEquals(position, functionActivator.get_position());

    //onOffFunction()
    //active_count 3 -> 2
    int active_count = 2;
    jButton.setText("Button2");
    functionActivator.work(jButton);
    assertEquals(active_count, functionActivator.get_active_count());

    //confirmActive - not confirm
    //position 1 -> 1
    position = 1;
    jButton.setText("Button4");
    functionActivator.work(jButton);
    assertEquals(active_count, functionActivator.get_active_count());
```

```java
        assertEquals(position, functionActivator.get_position());

        //onOffFunction()
        //active_count 2 -> 3
        active_count = 3;
        jButton.setText("Button2");
        functionActivator.work(jButton);
        assertEquals(active_count, functionActivator.get_active_count());

        //confirmActive - confirm
        //position 1 -> 0
        position = 0;
        jButton.setText("Button4");
        functionActivator.work(jButton);
        assertEquals(active_count, functionActivator.get_active_count());
        assertEquals(position, functionActivator.get_position());
    }

    @Test
    void get_activeTest() {
        Queue<Integer> modeQ = new LinkedList<~>();
        modeQ.offer(watch_Type.ALARM.ordinal());
        modeQ.offer(watch_Type.WORLDTIME.ordinal());
        modeQ.offer(watch_Type.STOPWATCH.ordinal());
        FunctionActivator functionActivator = new FunctionActivator(modeQ);

        boolean active_function = true;
        int position = 0;
        assertEquals(active_function, functionActivator.get_active(position));
    }

    @Test
    void get_active_nameTest() {
        Queue<Integer> modeQ = new LinkedList<~>();
        modeQ.offer(watch_Type.ALARM.ordinal());
        modeQ.offer(watch_Type.WORLDTIME.ordinal());
        modeQ.offer(watch_Type.STOPWATCH.ordinal());
        FunctionActivator functionActivator = new FunctionActivator(modeQ);

        String active_function_name = "arm";
        int position = 0;
        assertEquals(active_function_name, functionActivator.get_active_name(position));
    }

    @Test
    void get_active_countTest() {
        Queue<Integer> modeQ = new LinkedList<~>();
        modeQ.offer(watch_Type.ALARM.ordinal());
        modeQ.offer(watch_Type.WORLDTIME.ordinal());
        modeQ.offer(watch_Type.STOPWATCH.ordinal());
        FunctionActivator functionActivator = new FunctionActivator(modeQ);

        int active_count = 3;
        assertEquals(active_count, functionActivator.get_active_count());
    }
```

```java
@Test
void get_modeQTest() {
    Queue<Integer> modeQ = new LinkedList<~>();
    modeQ.offer(watch_Type.ALARM.ordinal());
    modeQ.offer(watch_Type.WORLDTIME.ordinal());
    modeQ.offer(watch_Type.STOPWATCH.ordinal());
    FunctionActivator functionActivator = new FunctionActivator(modeQ);

    assertEquals(modeQ, functionActivator.get_modeQ());
}
@Test
void get_positionTest() {
    Queue<Integer> modeQ = new LinkedList<~>();
    modeQ.offer(watch_Type.ALARM.ordinal());
    modeQ.offer(watch_Type.WORLDTIME.ordinal());
    modeQ.offer(watch_Type.STOPWATCH.ordinal());
    FunctionActivator functionActivator = new FunctionActivator(modeQ);

    int position = 0;
    assertEquals(position, functionActivator.get_position());
}
```

Unit Test <Buzzer> - 4개

```java
@Test
void testgetbuzzer() {
    Buzzer bz = new Buzzer();
    int result = bz.getbuzzer();
    assertEquals( expected: 0, result);
}
@Test
void testonBuzzer(){
    Buzzer bz = new Buzzer();
    bz.onBuzzer( type: 1);
    assertEquals( expected: 1,bz.getbuzzer());
    assertEquals( expected: 15,bz.getLeftTime());

    bz.onBuzzer( type: 2);
    assertEquals( expected: 2,bz.getbuzzer());
    assertEquals( expected: 15,bz.getLeftTime());
}


@Test
void testsubTimeBuzzer(){
    Buzzer bz = new Buzzer();
    bz.onBuzzer( type: 1);
    assertEquals( expected: 15,bz.getLeftTime());

    bz.subTimeBuzzer();
    assertEquals( expected: 14,bz.getLeftTime());
}
```
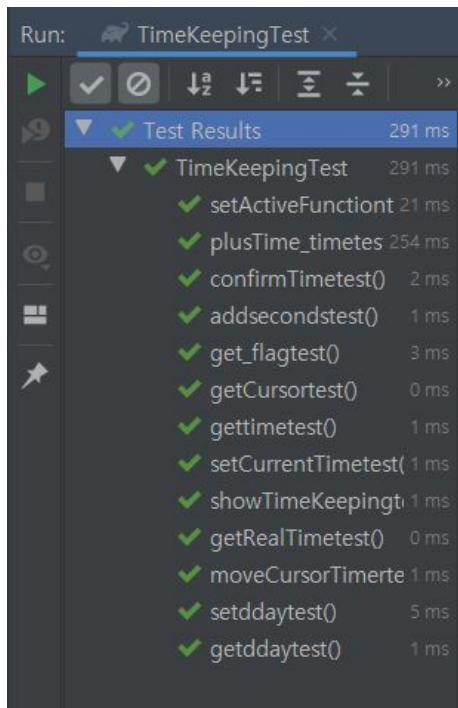
```java
@Test
void turnOffBuzzer(){
    Buzzer bz = new Buzzer();
    bz.onBuzzer( type: 1);
    assertEquals( expected: 1,bz.getbuzzer());

    bz.turnOffBuzzer();
    assertEquals( expected: 0,bz.getbuzzer());
}
```

2061. Unit Testing

TimeKeep

Alarm

WorldTime

```
WorldTimeTest > changeModeTest() PASSED




WorldTimeTest > get_keyTest() PASSED




WorldTimeTest > confirmCountryTest() PASSED




WorldTimeTest > get_valueTest() PASSED




WorldTimeTest > getCursorTest() PASSED




WorldTimeTest > changeCountryTest() PASSED




WorldTimeTest > nextCountryTest() PASSED
```
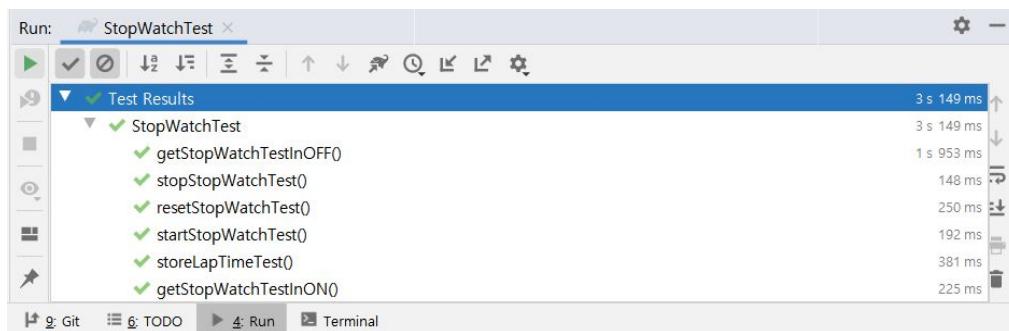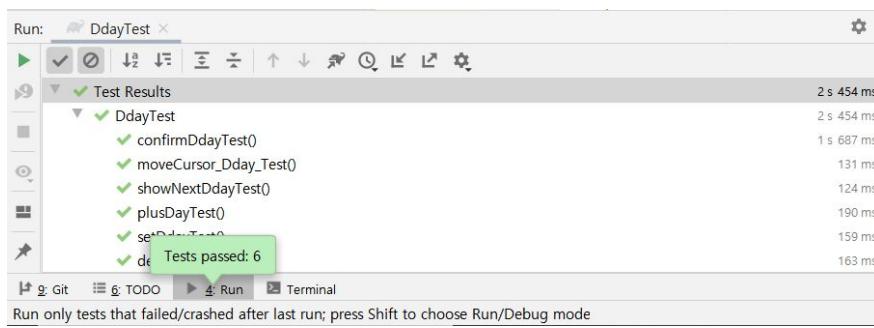
StopWatch

Dday

mTimer

Run: TimerTest ×

▼ ✔ Test Results   233 ms
  ▼ ✔ TimerTest   233 ms
      ✔ stopTimertest()   221 ms
      ✔ getPauseFlagtest()   2 ms
      ✔ getFlagtest()   1 ms
      ✔ plusTimertest()   1 ms
      ✔ startTimertest()   1 ms
      ✔ getCursortest()   1 ms
      ✔ moveCursorTimerte   1 ms
      ✔ confirmTimertest()   1 ms
      ✔ getTimerTimetest()   1 ms
      ✔ setTimertest()   1 ms
      ✔ pauseTimertest()   2 ms
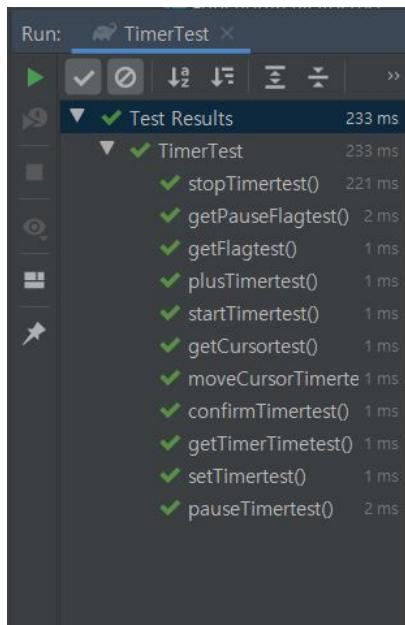
FunctionActivator

```
FunctionActivatorTest > onOffFunctionTest() PASSED


FunctionActivatorTest > get_active_nameTest() PASSED


FunctionActivatorTest > get_modeQTest() PASSED


FunctionActivatorTest > get_active_countTest() PASSED


FunctionActivatorTest > get_positionTest() PASSED


FunctionActivatorTest > get_activeTest() PASSED
```

```
FunctionActivatorTest > confirmActiveTest() PASSED




FunctionActivatorTest > nextActivateFunctionTest() PASSED
```

Buzzer

```
BuzzerTest > turnOffBuzzer() PASSED


BuzzerTest > testgetbuzzer() PASSED




BuzzerTest > testonBuzzer() PASSED




BuzzerTest > testsubTimeBuzzer() PASSED
```

2062 System Testing

Test No.	Test 항목	Description	Function
1	Show Time Test	현재 시간을 월, 일, 요일(디데이 라벨), 시, 분, 초, 연도 순서대로 화면에 표시하는지 test	showTimeKeeping
2	Show Time Test	사용자가 설정한 시간을 월, 일, 요일(디데이 라벨), 시, 분, 초, 연도 순서대로 화면에 표시하는지 test	showTimeKeeping
3	Show Time Test	현재 시간이 1초마다 1초씩 증가하는지 test	showTimeKeeping
4	Set Time Test	현재 시간 모드에서 setTime버튼을 눌렀을 때 setTime으로 진입되는지 test	setCurrentTime
5	Set Time Test	현재 시간 모드의 setTime에서 (+1)버튼을 눌렀을 때 월, 일, 요일(디데이 라벨), 시, 분, 초, 연도 중 커서가 위치한 값이 1씩 증가하는지 test	setCurrentTime
6	Set Time Test	현재 시간 모드의 setTime에서 next버튼을 눌렀을 때 커서가 다음 textview로 이동하는지 test	setCurrentTime
7	Set Alarm Test	알람 모드에서 setAlarm 버튼을 눌렀을때 setAlarm으로 진입되는지 test	setAlarm
8	Set Alarm Test	setAlarm 모드에서 버튼1를 누르면서 커서가 잘 이동하는지 test	setAlarm
9	Set Alarm Test	setAlarm 모드에서 버튼 2를 누르면 1씩 잘 증가되는지 test	setAlarm
10	Set Alarm test	setAlarm 모드에서 버튼4를 눌렀을때 데이터가 잘 저장 되었는지 테스트	setAlarm
11	Delete Alarm test	알람 모드에서 버튼4를 눌렀을때 기존 알람 데이터가 잘 지워지고 OFF로 표시 되는지 확인	DeleteAlarm
12	Show Alarm Test	알람 화면이 올바르게 표시	showAlarm
13	Sound	부저가 잘 울리고 시간이 설정 되는지 테스트	OnBuzzer

	Buzzer Test		
14	Turn Off Buzzer Test	함수를 호출하면 부저가 잘 종료되는지 테스트	TurnOffBuzzer
15	Buzzer Timeout	부저의 남은 시간이 잘 감소되고 0초에 부저가 종료되는지 테스트	getLeftTime
16	Watch World TIme Test	나라를 변경하고자 버튼을 눌렀을 때, 나라를 출력해주는 화면이 1초마다 깜빡거리는지 Test	WatchWorldTime
17	Watch World TIme Test	세계 시간을 월, 일, 지역, 시, 분, 초, 연도 순서대로 화면에 보여주는지 Test 나라를 설정하던 도중 버튼 3을 눌러 모드를 변경 하였을 때, 저장한 나라와 시간을 화면에 보여주는지 Test	WatchWorldTime, ChangeMode
18	Watch World TIme Test	나라를 설정하던 도중 버튼 4를 눌러 저장하였을 때, 저장한 나라와 시간을 화면에 보여주는지 Test	confirmCountry
19	Change Country Test	큐에 저장된 나라 순서대로 월, 일, 지역, 시, 분, 초, 연도 순서대로 화면에 보여주는지 Test	changeCountry
20	Reset StopWatch test	스톱 워치와 관련된 변수들이 초기화 되는지 test	resetStopWatch
21	Start Stopwatch test	스톱 워치가 시작되는지 test	startStopWatch
22	Stop StopWatch test	스톱 워치가 멈추는지 test	stopStopWatch
23	StoreLaptime test	랩 타임이 저장되는지, 랩 타임이 제대로 계산되는지 test	storeLapTime
24	Show stopWatch test	스톱 워치의 경과시간을 제대로 계산하여 화면에 표시 하는지 test	getStopwatch
25	Set Dday test	디데이와 디데이 메모가 설정되는지 test	setDday
26	Set Dday	디데이 설정 시 커서가 작동하는지 test	setDday

	test		
27	Set Dday test	디데이 설정 시 시간이 증가하는지 test	setDday
28	Set Dday test	디데이가 설정 완료되는지 test	setDday
29	Show Next Dday test	다음 디데이가 출력되는지 test	showNextDday
30	Delete Dday test	디데이가 삭제되는지 test	deleteDday
31	Show Dday	타임 키핑 모드의 요일 textView에 현재 날짜로 설정된 디데이가 표시되는지 test	showDday
32	Show Timer Test	타이머의 시, 분, 초가 초기값대로 화면에 표시되는지 test	getTimer
33	Show Timer Test	타이머의 시, 분, 초가 설정한 시간대로 화면에 표시되는지 test	getTimer
34	Set Timer Test	타이머 모드에서 setTimer버튼을 눌렀을 때 setTimer로 진입되는지 test	setTimer
35	Set Timer Test	타이머 모드의 setTimer에서 (+1)버튼을 눌렀을 때 시, 분, 초 중 커서가 위치한 값이 1씩 증가하는지 test	setTimer
36	Set Timer Test	타이머 모드의 setTimer에서 next버튼을 눌렀을 때 커서가 다음 textview로 이동하는지 test	setTimer
37	Start Timer Test	타이머가 지정된 시간으로부터 1초씩 감소하는지 test	startTimer
38	Pause Timer Test	타이머가 시작 된 상태에서 Pause에 해당하는 버튼을 눌렀을 때 시, 분, 초가 일시정지하는지 test	pauseTimer
39	Stop Timer Test	타이머가 시작, 혹은 Pause된 상태에서 Stop에 해당하는 버튼을 눌렀을 때 0시0분0초로 초기화되는지 test	stopTimer
40	Set Active Function	on/off, 기능 이름을 순서대로 화면에 보여주는지 Test 버튼1을 눌러 다음 기능으로 넘어가면 다음 기능에 대한 on/off와 기능 이름을 순서대로 화면에 보여주는지 Test	setActiveFunction

	Test		
41	Set Active Function Test	현재 기능에서 버튼 2를 눌러 on -> off 혹은 off -> on으로 전환할 때, 전환이 되는지 test 전환하고 나서 화면에 보여주는지 test	setActiveFunction
42	Set Active Function Test	3개의 기능을 활성화, 2개의 기능이 비활성화 되었을 때, 버튼 4를 누르면 저장이 되고 TimeKeeping 모드로 넘어가는지 Test 활성화된 기능이 3개가 아닐 때, 버튼 4를 누르면TimeKeeping 모드로 넘어가는지 Test	setActiveFunction
43	Change Mode Test	mode에 해당하는 버튼을 눌렀을 때 다음 모드로 변경되는지 test	changeMode
44	ShowNextAlarm	알람이 하나 이상 설정되어 있을 때 next에 해당하는 버튼을 누르면 다음 알람을 보여준다.	showNextAlarm

2063 Testing Traceability Analysis

System Testing No.	System Function		Essential Use Case	Operation in sequence diagram
1,2,3	showTimeKeeping		Show Current Time	O1, O41, O42
4,5,6	setCurrentTime		Set Current Time	O1, O2, O3, O4
7,8,9,10	setAlarm		Set Alarm When I Want	O9, O10, O11,O12
13	OnBuzzer		Sound Buzzer	O6
14	TurnOffBuzzer		Turn Off Buzzer	O5
11	Delete Alarm		Delete Alarm	O14
12	Show Alarm		Show Alarm	O44
15	getLeftTime		Buzzer Timeout	O7, O8

16,17	WatchWorldTime		Watch World Time	O46, O47
19	changeCountry		Change Country	O34, O35
21	StartStopWatch		Start StopWatch	O30
22	stopStopWatch		Pause StopWatch	O31
20	resetStopWatch		Reset StopWatch	O32
24	getStopWatch		ShowStopWatch	O33
	getLaptime		WatchLapTime	O28
23	storeLapTime		StoreLapTime	O29
25,26,27,28	setDday		Set D-day	O22, O23, O25, O27
31	showDday		Show D-day	O45
30	deleteDday		Delete D-day	O26
29	showNextDday		Show Next D-day Calendar	O24
37	startTimer		Start Timer	O15, O40
34,35,36	setTimer		Set Timer	O16, O19, O20, O21
38	pauseTimer		Pause Timer	O17
39	stopTimer		Stop Timer	O18
32,33	getTimer		Show Timer	O40
40,41,42	setActiveFunction		Set Active Function	O48, O49
43	Change Mode		Change Mode	O43

44	showNextAlarm		show next Alarm	O49

O - number	Operation in sequence diagram
O1	setCurrentTime()
O2	plusTime_time()
O3	moveCursor_time()
O4	confirmTime()
O5	turnOffBuzzer()
O6	onBuzzer()
O7	getleftTime()
O8	subTimeBuzzer()
O9	moveCursor_alarm()
O10	plusTime_alarm()
O11	confirmAlarm()
O12	setAlarm()
O13	getAlarm()
O14	deleteAlarm()
O15	startTimer()
O16	setTimer()
O17	pauseTimer()
O18	stopTimer()
O19	confirmTimer()
O20	moveCursor_timer()
O21	plusTimer()
O22	setDday()
O23	moveCursor_Dday()
O24	showNextDday()

O25	plusDay()
O26	deleteDday()
O27	confirmDday()
O28	getLapTime()
O29	storeLapTime()
O30	startStopWatch()
O31	stopStopWatch()
O32	resetStopWatch()
O33	getStopWatch()
O34	changeCountry()
O35	confirmCountry()
O36	nextActivateFunction()
O37	confirmActive()
O38	onOffFunction()
O39	get_active()
O40	getTimer()
O41	showTimerKeeping()
O42	gettime()
O43	changeMode()
O44	showAlarm()
O45	showDday()
O46	get_key()
O47	get_value()
O48	setActivateFunction()
O49	showNextAlarm()

Operation in sequence diagram	M-Link

setCurrentTime()	M2, M69, M72
plusTime_time()	M8, M69, M72
moveCursor_time()	M6, M69, M72
confirmTime()	M9, M69, M72
turnOffBuzzer()	M20, M25, M69
onBuzzer()	M21
getleftTime()	M20
subTimeBuzzer()	M22
moveCursor_alarm()	M17, M18, M69, M72
plusTime_alarm()	M15, M69, M72
confirmAlarm()	M14, M69, M72
setAlarm()	M13, M69, M72
getAlarm()	M19
deleteAlarm()	M16, M69, M72
startTimer()	M51, M69, M72
setTimer()	M56, M69, M72
pauseTimer()	M55, M69, M72
stopTimer()	M54, M69, M72
confirmTimer()	M59, M69, M72
moveCursor_timer()	M57, M69, M72
plusTimer()	M58, M69, M72
setDday()	M11, M69, M72
moveCursor_Dday()	M46, M69, M72
showNextDday()	M45, M69, M72
plusDday()	M47, M69, M72
deleteDday()	M44, M69, M72
confirmDday()	M48, M69, M72
getLapTime()	M38, M69, M72

storeLapTime()	M37, M69, M72
startStopWatch()	M31,M34
stopStopWatch()	M32, M69, M72
resetStopWatch()	M36, M69, M72
getStopWatch()	M33,M34,M35
changeCountry()	M26, M69, M72
confirmCountry()	M30, M69, M72
nextActivateFunction()	M61, M69, M72
confirmActive()	M68, M69, M72
onOffFunction	M65, M69, M72
get_active()	M65
getTimer()	M52
showTimerKeeping()	M1, M5
gettime()	M5
changeMode()	M71, M69
showAlarm()	M12
showDday()	M43
get_key()	M29
get_value()	M27
setActivateFunction()	M60, M69, M72
showNextAlarm()	M19, M72, M11

MID	Method	Unit Test	Class
M1	showTimeKeeping()	showTimeKeepingtest	TimeKeeping
M2	setCurrentTime()	setCurrentTimetest	
M3	addseconds()	addsecondstest	
M4	getdday()	getddaytest	
M5	gettime()	gettimetest	
M6	moveCursortime()	moveCursorTimeTest	
M7	getCursor()	getCursortest	
M8	plusTimetime()	plustimetest	
M9	confirmTime()	confirmTimeTest	
M10	setdday()	setddaytest	
M11	showNextAlarm()		Alarm
M12	showAlarm()		
M13	setAlarm()	testsetAlarm	
M14	confirmAlarm()	testcomfirmAlarm	
M15	plusTimealarm()	testplusTime_alarm	
M16	deleteAlarm()	testdeleteAlarm	
M17	moveCursorAlarm()	testCursor	
M18	getCursor()	testCursor	
M19	getAlarm()	testplusTime_alarm	
M20	getLeftTime()	testonBuzzer	Buzzer
M21	onBuzzer()	testonBuzzer	
M22	subTimeBuzzer()	testsubTimeBuzzer	
M23	soundBuzzer()		

M24	turnoffBuzzer()	turnOffBuzzer	
M25	watchWorldTime()		WorldTime
M26	changeCountry()	changeCountrytest	
M27	getValue()	get_valuetest	
M28	nextCountry()	nextCountryTest	
M29	getKey()	get_keytest	
M30	confirmCountry()	confirmCountryTest	
M31	startStopWatch()	startStopWatchtest	StopWatch
M32	stopStopWatch()	stopStopWatchtest	
M33	getStopWatch()	getStopWatchtestinoff getStopWatchtestinon	
M34	calculatorTime()	startStopWatchtest	
M35	splitElapse()	getStopWatchtestinon	
M36	resetStopWatch()	resetStopWatchTest	
M37	storeLapTime()	storeLaptimeTest	
M38	getLapTime()	storeLaptimeTest	
M39	showStopWatch()		
M40	setDday()	setDdaytest	Dday
M41	getDday()		
M42	getCursor()	moveCursor_DdayTest	
M43	ShowDday()		
M44	deleteDday()	deleteDdayTest	
M45	showNextDday()	showNextDdayTest	
M46	moveCursor_Dday()	moveCursor_DdayTest	

M47	plusDday()	plusDayTest	
M48	confirmDday()	confirmDdayTest	
M49	cmpday()		
M50	showTimer()		Timer
M51	startTimer()	startTimertest	
M52	getTimer()	getTimertest	
M53	get_flag()	getFlagTest	
M54	stopTimer()	stopTimertest	
M55	pauseTimer()	pauseTimertest	
M56	setTimer()	setTimertest	
M57	moveCursor_timer()	moveCursor_timertest	
M58	plusTimer()	plusTimertest	
M59	confirmTimer()	confirmTimertest	
M60	setActivateFunction()	setActiveFunctiontest	FunctionActivator
M61	nextActivateFunction()	nextActivateFunctiontest	
M62	get_active_count()	get_active_counttest	
M63	get_modeQ()	get_modeQtest	
M64	get_position()	get_position_test	
M65	get_active()	get_activetest	
M66	get_active_name()	get_active_name_test	
M67	onOffFunction()	onOffFuctionTest	
M68	confirmActive()	confirmActivateTest	
M69	pressButton()		Watch
M70	display()		
M71	changeMode()	changeModetest	

| M72 | work() | | Mode |