Check It Out - Digital Watch

2019 Software Modeling & Analysis

OOPT Stage 2050 & Stage 2060

T6

201613856 소아이린

201711381 김소현

201711401 염혜지

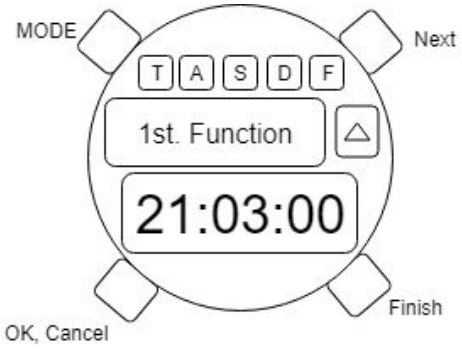
201711420 임수연

201711428 조은지

Contents

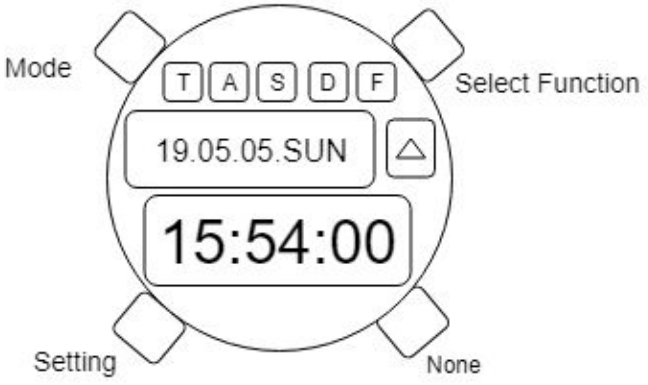
1. Digital Watch Manual

Clock-1. Select Function화면



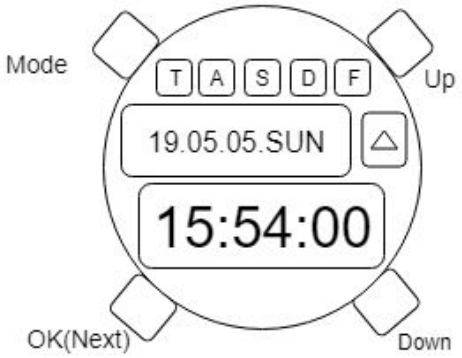
MODE를 클릭하면 시간보기 화면으로 돌아간다.
Next 버튼을 누르면 다음 기능을 도트화면에 보여준다.
OK 버튼을 누르면 현재 기능이 조합될 기능으로 선택되고 LCD 화면에 표시된다. 한번 더 누르면 취소된다.
3개의 기능을 선택하고 Finish 버튼을 누르면 3개라면 선택된 기능들을 활성화시키고 시간보기 화면으로 전환된다.

Clock-2. TimeKeeping 화면(초기화면)



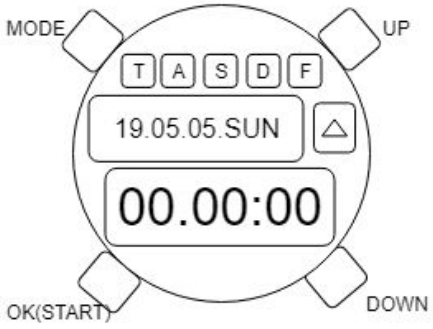
Mode버튼을 통해 다음 기능을 사용 할 수 있다.
Setting버튼을 통해서 시간을 원하는 시간으로 설정할 수 있다.
None버튼은 기능이 없는 버튼으로써 버튼을 눌러도 아무 기능이 작동하지 않는다.
SelectFunction버튼을 통해서 6가지 기능 중 4가지 기능을 선택할 수 있다.

Clock-3. TimeKeeping Setting(시간 설정)화면



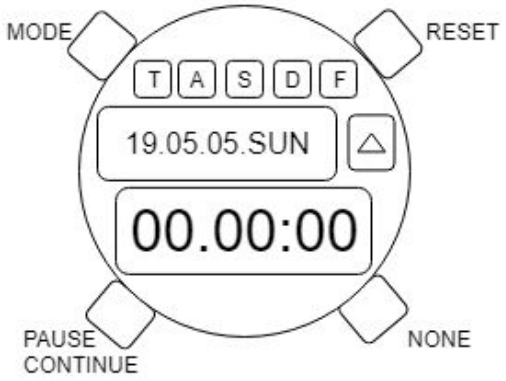
Mode 버튼을 통해서 다음 기능을 사용할 수 있다.
그 후 다시 TimeKeeping으로 돌아온다면 설정화면으로 돌아온다.
Up과 Down버튼을 통해서 시간을 선택할 수 있다.
OK버튼을 통해서 연,월,일,시,분,초 의 설정이 가능하며 다음 시간선택으로 넘어간다.
마지막 초 설정을 끝낸 뒤 OK를 누르면 현재 시간 설정이 완료된다.

Clock-4. Timer 시간 설정 화면



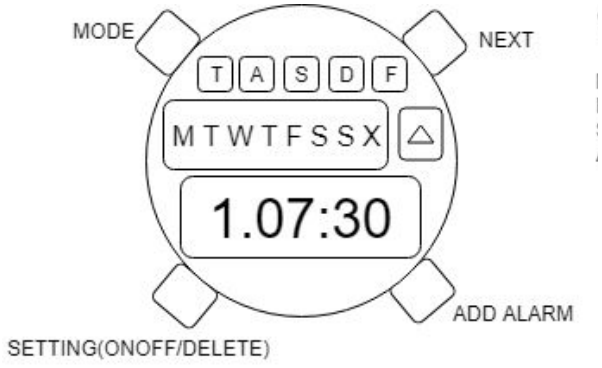
MODE를 클릭하면 다른 기능으로 전환이 가능하다.
UP,DOWN을 클릭해서 타이머의 시/분/초를 선택한다.
OK를 클릭해서 해당 단위의 시간을 저장하고 다음 단위를 설정한다.
초 단위까지의 설정을 끝마치고 OK를 클릭하면 타이머가 실행된다.

Clock-5. Timer 실행 화면



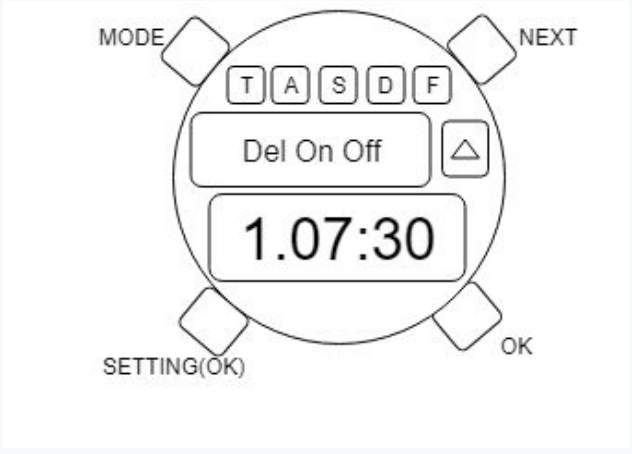
MODE를 클릭하면 다른 기능으로 전환이 가능하다.
PAUSE를 클릭하면 실행중이던 타이머를 일시정지한다.
PAUSE상태에서 PAUSE를 한 번 더 누르면 타이머가 재실행된다.
RESET을 클릭해서 타이머의 실행을 취소하고 값을 초기화한다.

Clock-6. Alarm 목록 화면



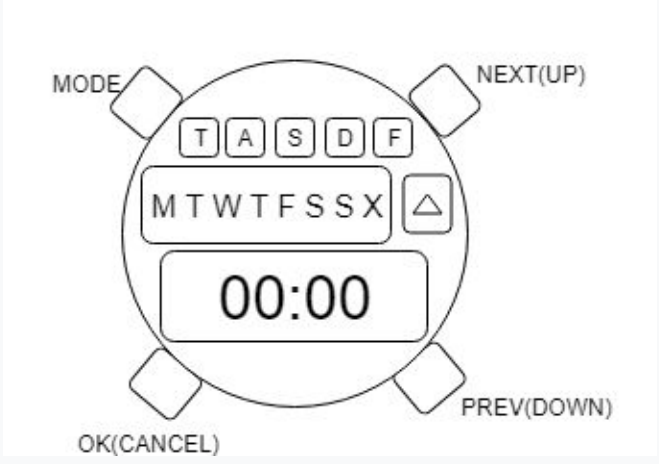
MODE를 클릭하면 다른 기능으로 전환이 가능하다.
NEXT를 클릭해서 현재 저장된 알람을 순서대로 볼 수 있다.
SETTING을 클릭하면 알람 ON-OFF 및 DELETE화면으로 전환된다.
ADD ALARM을 클릭하면 현재 알람을 수정하거나 새로운 알람을 추가할 수 있다.

Clock-7. Alarm 설정화면(ONOFF/DELETE)



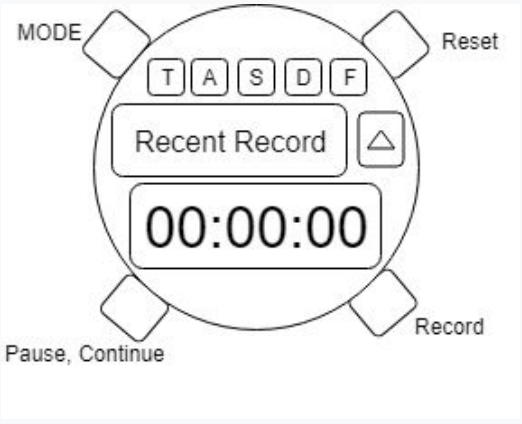
MODE를 클릭해서 다른 기능으로 전환이 가능하다.
NEXT를 클릭해서 알람을 ON 또는 OFF 또는삭제를 지정 할 수 있다.
OK를 클릭하면 현재 상태를 저장하고 알람 목록화면으로 돌아간다.

Clock-8. Alarm 추가화면



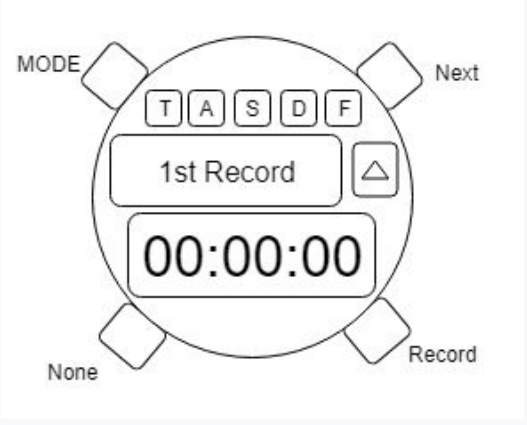
MODE를 클릭하면 다른 기능으로 전환이 가능하다.
요일의 경우, NEXT와 PREV를 통해서 원하는 요일로 이동할 수 있고,
OK버튼을 통해서 알람을 설정할 수 있다.
이 때, OK을 눌렀던 요일을 한 번 더 누르게 되면 CANCEL을 할 수 있다.
마찬가지로 NEXT,PREV를 클릭해서 주기, 알람의 시/분 을 각각 선택할 수 있다.
OK를 클릭하면 알람이 저장된다.

Clock-9. StopWatch 실행 화면



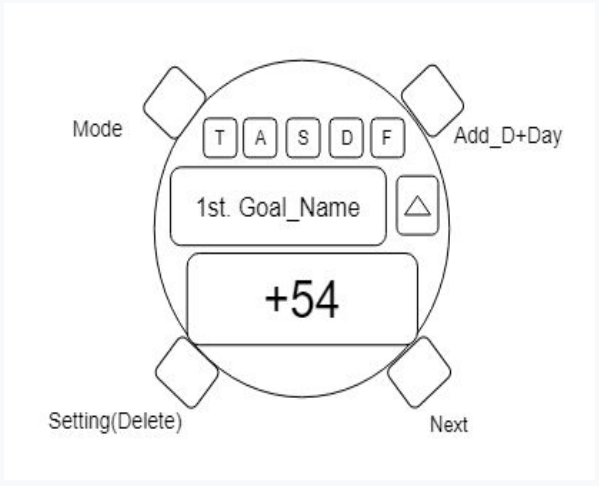
MODE를 클릭하면 다음 기능으로 전환이 가능하다.
Pause 버튼을 통해 일시정지를 할 수 있고, 다시 한 번 누르면 Continue 기능이 실행된다.
Reset 버튼을 누르면 현재 측정중이던 시간이 00:00:00으로 초기화된다.
Record 버튼을 누르면 현재 시간을 기록한다. 가장 최근에 측정한 시간이 화면에 나타난다.
일시정지 상태일 때 Record 버튼을 누르면 기록 목록화면으로 넘어간다.

Clock-10. StopWatch 기록 보기 화면



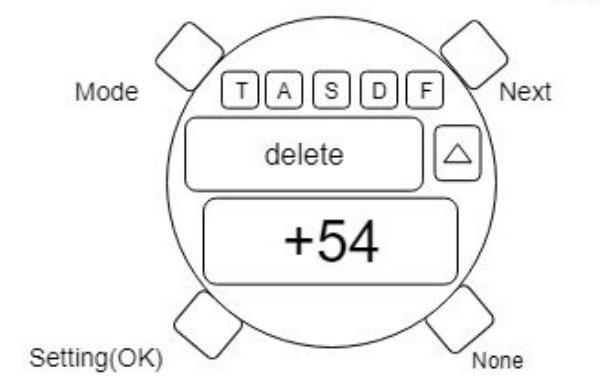
MODE를 클릭하면 다른 기능으로 전환이 가능하다.
Next를 누르면 다음 기록을 보여준다. 마지막 기록을 보여주고 나면 처음 기록을 보여준다.
Record 버튼을 누르면 일시정지 상태로 되돌아간다.

Clock-11. D+Day 목록 보기 화면



D+day기능의 초기화면은 목록화면이다.
목록이 없다면 빈 화면을 보여준다.
목록이 있다면 목표내용과 경과날짜를 보여준다.
Next버튼을 통해서 다음 목록들을 볼 수 있다.
Setting버튼을 통해 목록을 지울 수 있다.

Clock-12. D+Day Setting(삭제) 화면

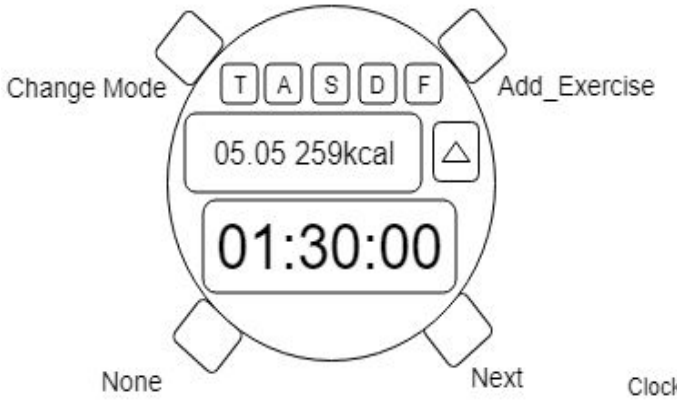


Next버튼을 통해서 삭제를 결정할 수 있다.
OK버튼을 통해서 삭제 또는 변경없음을 실행한다.
그 후 목록보기 화면으로 돌아간다.
삭제를 했다면 그 다음 목록을 삭제를 안했다면 해당목록부터 보여준다.

Clock-13. D+Day 목록 추가 화면

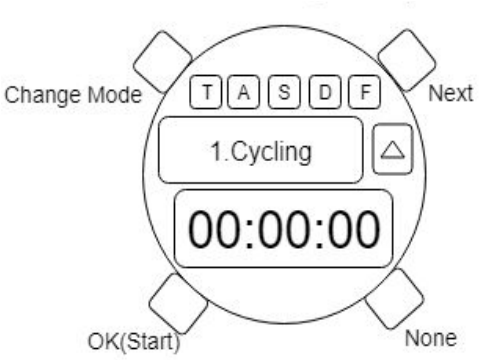
Up,Down버튼을 통해서 시스템이 제공하는 목표목록을 볼 수 있고,
날짜를 변경할 수 있다.
연도선택으로 넘어가고, 날짜까지의 설정이 끝나 OK버튼을 누르면 목표를 선택할 수 있고
OK버튼을 눌러서 설정을 끝낸다.
해당 D+Day가 추가가되고 목록화면으로 돌아간다.

Clock-14. Fitness 목록 화면



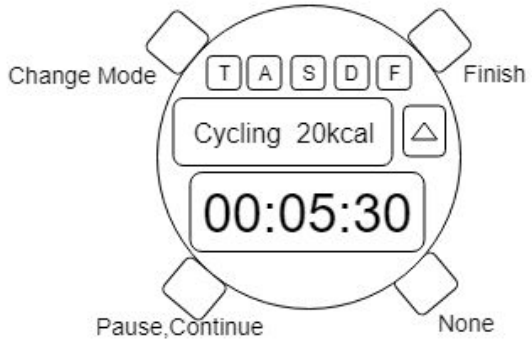
오늘의 운동량을 보여준다.
이전에 운동기록이 있다면 Next버튼을 통해서 이전 날짜의 운동량과 운동시간을 볼 수 있다.
Add_Exercise 버튼을 통해서 오늘의 운동을 추가 할 수 있다.

Clock-15. Fitness 운동 추가 화면(종목 선택)



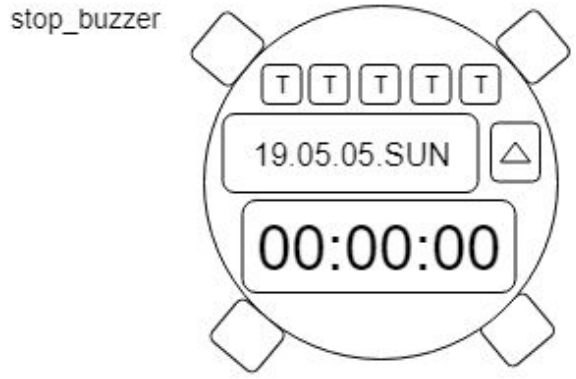
Next를 통해서 시스템이 제공하는 3가지 운동종목 중 하나를 선택할 수 있다.
OK버튼을 통해 결정을 하면 바로 운동시간 측정이 실행된다.

Clock-16. Fitness 운동 실행 화면



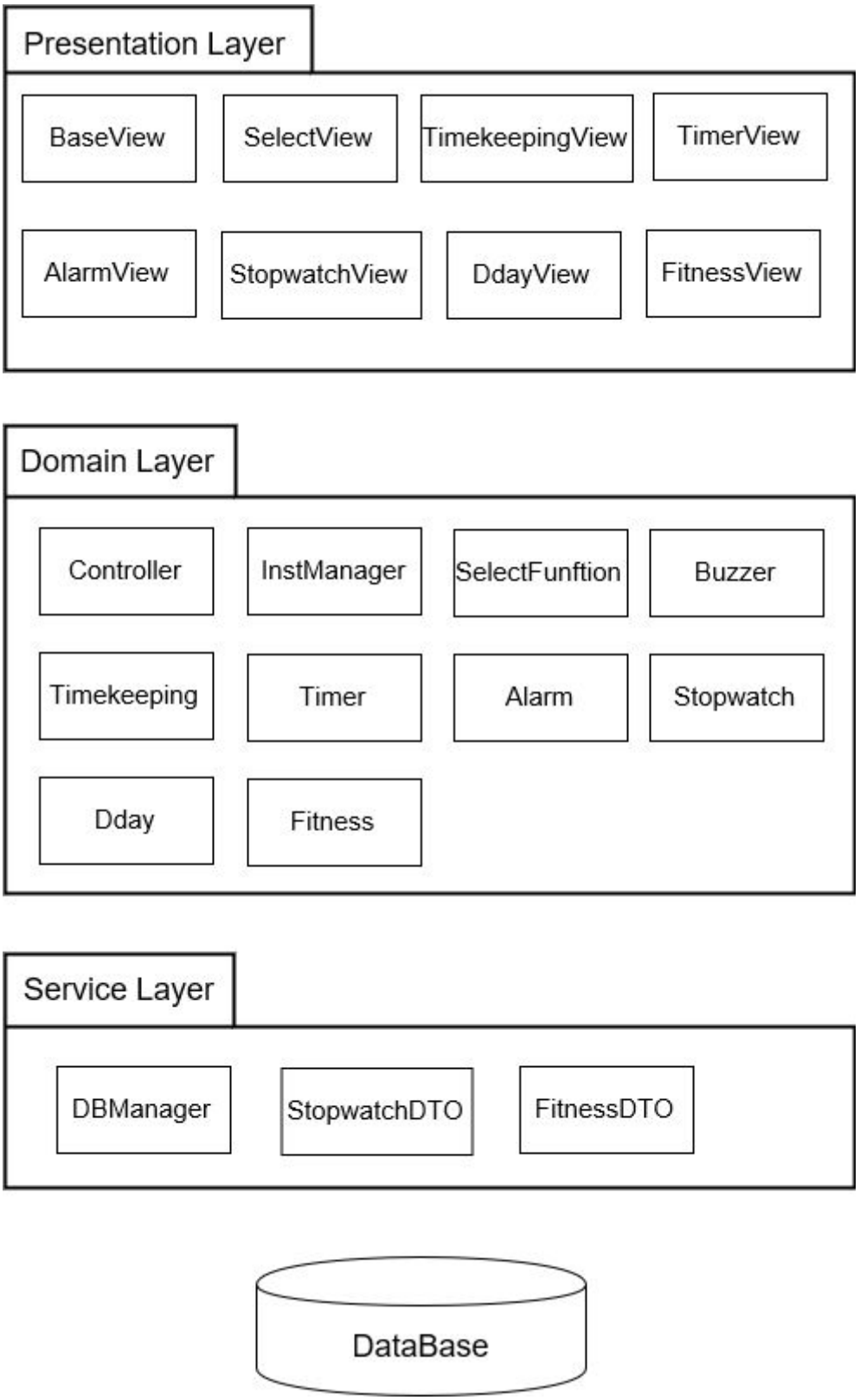
운동시간 측정이 되면서 소모되는 칼로리량을 화면에 함께 실시간으로 보여준다.
Pause버튼을 통해 일시정지를 할 수 있고, Pause버튼을 한 번 더 누르게 되면 Continue기능이 실행된다.
Finish버튼을 통해서 운동을 종료할 수 있다.

Clock-17. Stop Buzzer



4개의 버튼 중 임의의 것을 눌러서 버저를 멈출 수 있다.
화면은 전환되지 않고 현재 화면을 그대로 보여준다.

2. Layered Architecture


Presentation Layer
BaseView
SelectView
TimekeepingView
TimerView
AlarmView
StopwatchView
DdayView
FitnessView
Domain Layer
Controller
InstManager
SelectFunftion
Buzzer
Timekeeping
Timer
Alarm
Stopwatch
Dday
Fitness
Service Layer
DBManager
StopwatchDTO
FitnessDTO
DataBase

3. Methods Description

- Timekeeping

Type	Class
Name	Timekeeping
Purpose	시간보기기능을 제공한다.
Overview	현재시간, 요일 등을 가지고 있다. 시간 설정이 가능하다.
Cross Reference	System function : R.4.1, R.4.2 User cases : look_time, set_time
Exceptional Course of Event	N/A

Type	Method
Name	countUp
Purpose	1초마다 시간, 분, 초, 월, 일을 갱신한다.
Cross Reference	System function : R.1.2 User cases : count_up
Input	
Output	void
Abstract operation	초단위로 초 정보를 갱신하고 분, 시, 월, 일을 바꾸어준다.
Exceptional Course of Event	N/A

Type	Method
Name	setTime
Purpose	시간, 분, 초를 입력받아 시간을 바꾼다.
Cross Reference	System function : R.4.2 User cases : set_time

Input	hour, minute, second : int
Output	void
Abstract operation	입력받은 시간, 분, 초 정보를 현재 시간, 분, 초에 대입해준다.
Exceptional Course of Event	N/A

- Timer

Type	Class
Name	Timer
Purpose	타이머 기능을 제공한다.
Overview	시간, 분, 초의 정보를 가지고 카운트다운 기능을 가진다.
Cross Reference	System function : R.5.1, R.5.2, R.5.3, R.5.4 User cases : set_timer, dount_down, pause_timer, cancel_timer
Exceptional Course of Event	N/A

Type	Method
Name	countDown
Purpose	1초마다 시간을 카운트 다운하여 갱신하고 설정한 시간이 지나면 버저를 울린다.
Cross Reference	System function : R.5.2 User cases : count_down
Input	
Output	void
Abstract operation	1초마다 초를 카운트다운 하고 0초가 되면 다음 단위의 수를 카운트다운한다. 모든 단위의 수가 0이 되면 버저를 울린다.
Exceptional Course of Event	N/A

- Alarm

Type	Class
Name	Alarm
Purpose	알람기능을 제공한다.
Overview	시간, 분, 초의 정보와 알람기능을 가진다.
Cross Reference	System function : R.6.1, R.6.2, R.6.3, R.6.4, R.6.5 User cases : look_alarm, set_alarm, check_alarm, onoff_alarm, delete_alarm
Exceptional Course of Event	N/A

Type	Method
Name	chec Alarm
Purpose	현재시간이 알람시간과 일치하는지 체크하고 일치한다면 버저를 울린다.
Cross Reference	System function : 6.3 User cases : check_alarm
Input	
Output	
Abstract operation	설정된 요일, 시간, 분을 현재 요일, 시간 분과 비교하고 값이 같다면 설정된 주기 간격으로 버저를 3번 울린다.
Exceptional Course of Event	N/A

- Stopwatch

Type	Class
Name	Stopwatch
Purpose	스탑워치기능을 제공한다.

Overview	시간, 분, 초의 정보를 가지고 카운트업, 일시정지, 시간기록의 기능을 가진다.
Cross Reference	System function : R.7.1.1, R.7.2.1, R.7.2.2, R.7.2.3 User cases : record_stopwatch, pause_stopwatch, show_stopwatch, reset_stopwatch
Exceptional Course of Event	N/A

Type	Method
Name	record
Purpose	스탑워치의 현재 시간을 기록하여 저장한다.
Overview	N/A
Cross Reference	System function : R.7.1.1 User cases : record_stopwatch
Input	
Output	void
Abstract operation	저장된 기록이 10개보다 적으면 현재시간을 기록목록에 추가해준다.
Exceptional Course of Event	N/A

- Dday

Type	Class
Name	Dday
Purpose	디데이기능을 제공한다.
Overview	디데이 날짜와 목표등의 정보를 가지고 있고 설정할 수 있다.
Cross Reference	System function : R.8.1, R.8.2, R.8.3, R.8.4, R.8.5 User cases : select_date, select_goal, update_Dday, show_Dday, delete_Dday
Exceptional Course of Event	N/A

Type	Method
Name	setDday
Purpose	날짜 설정을 하고 당일의 디데이를 계산한다.
Cross Reference	System function : R.8.1 User cases : select_date
Input	
Output	void
Abstract operation	디데이 날짜를 설장하고 당일의 디데이를 계산한다.
Exceptional Course of Event	N/A

- Fitness

Type	Class
Name	Fitenss
Purpose	운동 정보와 정보를 열람하고 설정하는 기능을 제공한다.
Overview	운동시간과 기록에 대한 정보를 담고 있다.
Cross Reference	System function : R.9.1, R.9.2, R.9.3, R.9.4, R.9.5.1, R.9.5.2 User cases : show_exercise, select_exercise, calculate_calories, update_calories, pause_exercise, finish_exercise
Exceptional Course of Event	N/A

Type	Method
Name	finish
Purpose	운동을 마쳤을 때 운동기록을 저장한다..
Cross Reference	System function : R.9.5.2 User cases : finish_exercise
Input	

Output	void
Abstract operation	운동이 끝나면 당일의 운동 기록을 저장한다. 운동기록목록이 30개 이상인 경우 가장 오래된 데이터를 삭제하고 저장한다.
Exceptional Course of Event	N/A

- InstManager

Type	Class
Name	InstManager
Purpose	기능별 객체들을 생성하고 관리한다.
Overview	객체를 생성하고 관리하는 기능을 제공한다.
Cross Reference	
Exceptional Course of Event	N/A

Type	Method
Name	ceateInst
Purpose	Alarm, Dday 기능에서 목록에 객체를 추가한다.
Overview	N/A
Cross Reference	System function : R.6.2, R.8.1, R.8.2 User cases : set_alarm, select_date, selcet_goal
Input	status : String
Output	Object
Abstract operation	Alarm, Dday 기능에서 객체 생성요청이 들어오면 목록의 개수를 확인하고 정해진 수를 넘지 않았다면 목록에 추가한다.
Exceptional Course of Event	N/A

Type	Method

Name	deleteInst
Purpose	Alarm, Dday 기능에서 목록에서 객체를 삭제한다.
Overview	N/A
Cross Reference	System function : R.6.5, R.8.5 User cases : delete_alarm, delete_dDay
Input	object : String
Output	void
Abstract operation	Alarm, Dday 기능의 목록에서 해당 인덱스의 객체를 삭제해준다.
Exceptional Course of Event	N/A

- SelectFunction

Type	Class
Name	SelectFunction
Purpose	기능 간 전환과 선택하는 역할을 담당한다.
Overview	기능 전환, 선택 등의 정보를 담고 있고 설정 기능을 제공한다.
Cross Reference	System function : R.2.1, R.3.1 R.3.2,, R.3.3.1, R.3.3.2 Use Cases : change_function, check_first_display, check_default_display, look_function, select_function
Exceptional Course of Event	N/A

Type	Method
Name	setFunctionList
Purpose	활성화시키기를 원하는 기능들을 선택하여 사용가능 기능목록에 넣는다.
Overview	N/A

Cross Reference	System function : R.3.3.2 User cases : select function
Input	index : int
Output	boolean
Abstract operation	인덱스를 인자로 받아 기능목록에서 해당 인덱스의 값이 선택되어 있다면 선택을 취소하고 선택되어 있지 않다면 선택한다. 선택된 목록의 개수는 3개가 넘지 않도록 한다.
Exceptional Course of Event	N/A

- Buzzer

Type	Class
Name	Buzzer
Purpose	버저기능을 담당한다.
Overview	버저를 울리고 멈추는 기능을 제공한다.
Cross Reference	System function : R.1.1.1, R.1.1.2 Use cases : ring_buzzer, stop_buzzer
Exceptional Course of Event	N/A

Type	Method
Name	ringBuzzer
Purpose	버저를 울린다.
Overview	N/A
Cross Reference	System function : R.1.1.1 User cases : ring_buzzer
Input	

Output	void
Abstract operation	버저를 멈추는 입력이 들어오지 않으면 30초 동안 버저를 울린다.
Exceptional Course of Event	N/A

- DBmanager

Type	Class
Name	DBManager
Purpose	스탑워치 기록과 운동기록을 저장하고 관리한다.
Overview	기록을 조회하고 저장하고 삭제하고 갱신하는 기능을 제공한다.
Cross Reference	System function : R.7.1.1, R.7.2.2, R.9.1, R.9.4, R.9.5.2 Use cases : record_stopwatch, show_record, show_exercise, update_calories, finish_exercise
Exceptional Course of Event	N/A

Type	Method
Name	insertFitness
Purpose	운동결과를 DB에 입력하여 저장한다.
Cross Reference	System function : R.9.5.2 User cases : finish_exercise
Input	month, date, hour, minute, second, totalCalories : int
Output	void
Abstract operation	날짜와 시간과 총 칼로리를 입력받아 DB에 저장한다.
Exceptional Course of Event	N/A

4. JUnit Testing Plan

Test	Test항목	Description	Use Case Number & Names	Ref.#
1	버저작동시험	버저가 울려야 할 때 작동이 잘 되는지 확인	1. ring_buzzer	R.1.1.1
2	버저중지시험	버저가 울린 뒤 입력을 받았을 때 중지가 잘 되는지 확인	2. stop_buzzer	R.1.1.2
3	시간측정확인시험	초 단위로 시간이 잘 측정되고 있는지 확인	3. count_up	R.1.2
4	기능전환시험	알맞은 기능으로 전환되었는지 확인	4. change_function	R.2.1
5	첫화면확인시험	첫번째 화면에서만 조합변경기능이 가능한지 확인	5. check_first_display	R.3.1
6	초기기능설정확인시험	모든 기능이 기본상태일때만 조합변경기능이 가능한지 확인	6. check_default_display	R.3.2
7	목록화면확인시험	기능 목록이 화면에 나오는지 확인	7. look_function	R.3.3.1
8	목록선택시험	기능을 선택했을 때 개수가 4개 이하 인지 확인하고, 실행조합목록에 넣어졌는지 확인	8. select_function	R.3.3.2
9	시간 화면 확인 시험	화면에 날짜와 시간이 나오는지 확인	9. look_time	R.4.1
10	시간 설정 확인 시험	날짜와 시간이 설정한 대로 변경되는지 확인	10. set_time	R.4.2

11	타이머설정시험	타이머가 설정되는지 확인	11. set_timer	R.5.1
12	카운트다운시험	타이머 시작 시에 카운트다운이 되는지 확인	12. count_down	R.5.2
13	타이머일시정지시험	입력을 받았을 때 카운트다운이 멈추는지 확인	13. pause_timer	R.5.3
14	타이머 취소기능 시험	카운트다운이 멈추고, 시간이 00시00분00초로 잘 초기화 되었는지 확인	14. cancel_timer	R.5.4
15	알람화면확인시험	알람 목록이 화면에 잘 보이는지 확인	15. look_alarm	R.6.1
16	알람설정확인시험	알람의 요일/시간/반복/주기 설정이 저장되었는지 확인	16. set_alarm	R.6.2
17	알람체크확인시험	현재 요일, 시각과 설정한 알람 요일, 시각을 잘 비교하고 있는지 확인	17. check_alarm	R.6.3
18	알람실행여부시험	알람이 제대로 끄고 켜지는지 확인	18. onoff_alarm	R.6.4
19	알람삭제확인시험	알람 목록이 제대로 삭제되는지 확인	19. delete_alarm	R.6.5
20	시간기록확인시험	기록입력이 들어왔을 때의 시각이 목록에 잘 저장되었는지 확인	20. record_stopwatch	R.7.1.1
21	스톱워치 일시중지 확인 시험	스톱워치의 카운트업이 멈추는지 확인	21. pause_stopwatch	R.7.2.1
22	기록화면확인시험	목록에서 기록한 시간들을 잘 보여주는지 확인	22. look_record	R.7.2.2
23	스톱워치 초기화 확인 시험	카운트업을 멈추고 0초로 초기화한 후, 목록의 시간 기록들을 모두 지웠는지 확인	23. reset_stopswatch	R.7.2.3
24	날짜선택시험	D+day의 날짜를 화면에 띄워서 선택할 수 있는지	24. select_date	R.8.1

		확인		
25	목표 선택화면 확인 시험	화면에 6가지 목표목록을 띄워서 선택할 수 있는지 확인	25. select_goal	R.8.2
26	디데이 갱신 시험	하루가 지날 때마다 D+day값을 갱신하는지 확인	26. update_Dday	R.8.3
27	디데이 화면 확인 시험	목표의 종류와 D+day 값을 화면에 보여주는지 확인, D+day 값을 오름차순으로 정렬하여 화면에 보여주는지 확인	27. look_Dday	R.8.4
28	디데이목록제거시험	D+day 목록에서 하나씩 목표를 삭제할 수 있는지 확인	28. delete_Dday	R.8.5
29	운동량 화면 확인 시험	화면에 하루치 총 소모 칼로리량과 운동 시간 을 보여주는지 확인	29. look_exercise	R.9.1
30	운동선택시험	3개의 유산소 운동 종목 중 하나를 선택할 수 있는지 확인	30. select_exercise	R.9.2
31	칼로리계산시험	총 칼로리 소모량(해당 운동의 1분당 소모 칼로리량 * 운동시간)을 알맞게 계산하는지 확인, 화면에 실시간으로 보여주는 지 확인	31.calculate_calories	R.9.3
32	소모 칼로리 갱신 시험	하루 총 소모 칼로리를 갱신하는지 확인	32. update_calories	R.9.4
33	운동 일시정지 시험	운동 시간 측정을 일시정지하는지 확인	33. pause_exercise	R.9.5.1
34	운동 완료 확인 시험	운동 시간 측정을 완료하는지 확인	34. finish_exercise	R.9.5.2

5. Write Unit Test Code

1. AlarmTest

```
class AlarmTest {
    public static Alarm junitTest;
    AlarmTest() {
    }

    @BeforeAll
    public static void makeInstance() {
        junitTest = new Alarm();
    }

    @Test
    void getBuzzer() throws Exception {
        try {
            junitTest.getBuzzer();
        } catch (Exception var2) {
            System.out.println("error");
        }
    }

    @Test
    void getDayList() throws Exception {
        int[] arr = new int[]{1, 2, 3, 4, 5, 6, 7};

        try {
            Assertions.assertArrayEquals(arr, junitTest.getDayList());
        } catch (Exception var3) {
            System.out.println("error");
        }
    }
```

```java
@Test
void getCheckDayList() throws Exception{
    try {
        junitTest.getCheckDayList( index: 1);
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void getDayListNum() throws Exception{
    try{
        junitTest.getDayListNum();
    }catch (Exception var2){
        System.out.println("error");
    }
}

@Test
void getCycle()throws Exception {
    try {
        junitTest.getCycle();
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void setDay() throws Exception {
    try {
        junitTest.setDay(1);
    } catch (Exception var3) {
        System.out.println("error");
    }
}

@Test
void setCycle() throws Exception {
    try {
        junitTest.setCycle(5);
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void onOffAlarm() throws Exception {
    try {
        junitTest.OnOffAlarm();
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void checkAlarm()throws Exception {
    try {
        junitTest.checkAlarm();
    } catch (Exception var2) {
        System.out.println("error");
    }
}
}
```

2. BuzzerTest

```
class BuzzerTest {
    public static Buzzer junitTest;

    BuzzerTest() {
    }

    @BeforeAll
    public static void makeInstance() { junitTest = new Buzzer(); }

    @Test
    void getInstance() throws Exception {
        try {
            Buzzer var10000 = junitTest;
            Buzzer.getInstance();
        } catch (Exception var2) {
            System.out.println("error");
        }
    }

    @Test
    void beep() throws Exception {
        try {
            junitTest.beep();
        } catch (Exception var2) {
            System.out.println("error");
        }
    }
```

```java
@Test
void ringBuzzer()throws Exception  {
    try {
        junitTest.ringBuzzer();
        System.out.println("30초동안 ring");
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void stopBuzzer() throws Exception {
    try {
        junitTest.ringBuzzer();
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void getIs_stop() throws Exception {
    try {
        assertEquals( expected: false, junitTest.getIs_stop());
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
    @Test
    void setIs_stop() throws Exception {
        try {
            junitTest.setIs_stop(true);
        } catch (Exception var2) {
            System.out.println("error");
        }
    }
}
```

3. ControllerTest

```
class ControllerTest {
    public static Controller junitTest;

    ControllerTest(){
    }

    @BeforeAll
     public static void makeInstance(){
        BaseView bv = new BaseView();
        junitTest = new Controller(bv);
    }

    @Test
    void req_countUp() throws Exception {
        try{
            junitTest.req_countUp( status: "stopwatch");
        }catch(Exception e){
            System.out.println("req_countUp failed");
        }
    }

    @Test
    void req_continue()  throws Exception {
        try{
            junitTest.req_continue( status: "stopwatch");
        }catch(Exception e){
            System.out.println("req_continue failed");
        }
    }
```

```java
@Test
void req_stopBuzzer()  throws Exception {
    try{
        junitTest.req_stopBuzzer( status: "alarm");
    }catch(Exception e){
        System.out.println("req_stopBuzzer failed");
    }
}

@Test
void req_changeMode()  throws Exception {
    try{
        junitTest.req_changeMode();
    }catch(Exception e){
        System.out.println("req_changeMode failed");
    }
}

@Test
void req_lookFunc() throws Exception {
    try{
        junitTest.req_lookFunc();
    }catch(Exception e){
        System.out.println("req_lookFunc failed");
    }
}
```

```java
@Test
void req_funcList() throws Exception {
    try{
        junitTest.req_funcList();
    }catch(Exception e){
        System.out.println("req_funcList failed");
    }
}

@Test
void req_selectFunc() throws Exception {
    try{
        junitTest.req_selectFunc( index: 3);
    }catch(Exception e){
        System.out.println("req_selectFunc failed");
    }
}

@Test
void req_finishSelect() throws Exception{
    try{
        assertEquals( expected: false, junitTest.req_finishSelect());
    }catch (Exception e){
        System.out.println("req_finishSelect failed");
    }
}
```

```java
@Test
void req_setDate() throws Exception {
    try{
        junitTest.req_setDate( status: "dDay",  year: 2019,  month: 5,  date: 30);
    }catch (Exception e){
        System.out.println("req_setDate failed");
    }
}

@Test
void req_setTime() throws Exception {
    try{
        junitTest.req_setTime( status: "timer",  hour: 13,  minute: 5,  second: 30);
    }catch (Exception e){
        System.out.println("req_setTime failed");
    }
}

@Test
void req_countDown() throws Exception{
    try {
        junitTest.req_countDown();
    }catch(Exception e){
        System.out.println("req_counDown failed");
    }
}
```

```java
@Test
void req_pause() throws Exception {
    try {
        junitTest.req_pause( status: "timer");
    }catch (Exception e){
        System.out.println("req_pause failed");
    }
}

@Test
void req_reset() throws Exception {
    try {
        junitTest.req_reset();
    }catch (Exception e){
        System.out.println("req_reset failed");
    }
}

@Test
void req_alarmList() throws Exception {
    try {
        junitTest.req_alarmList();
    }catch (Exception e){
        System.out.println("req_alarmList failed");
    }
}
```

```java
@Test
void req_setAlarm() throws Exception {
    try {
        junitTest.req_setAlarm();
    }catch (Exception e){
        System.out.println("req_setAlarm failed");
    }
}

@Test
void req_setDate1()throws Exception {
    try {
        junitTest.req_setDate( checkDayList: null,  dayListNum: 3,  cycle: 5,  hour: 9,  minute: 30);
    }catch (Exception e){
        System.out.println("req_setDate1 failed");
    }
}

@Test
void req_onOff() throws Exception {
    try {
        junitTest.req_onOff();
    }catch (Exception e){
        System.out.println("req_onoff failed");
    }
}
```

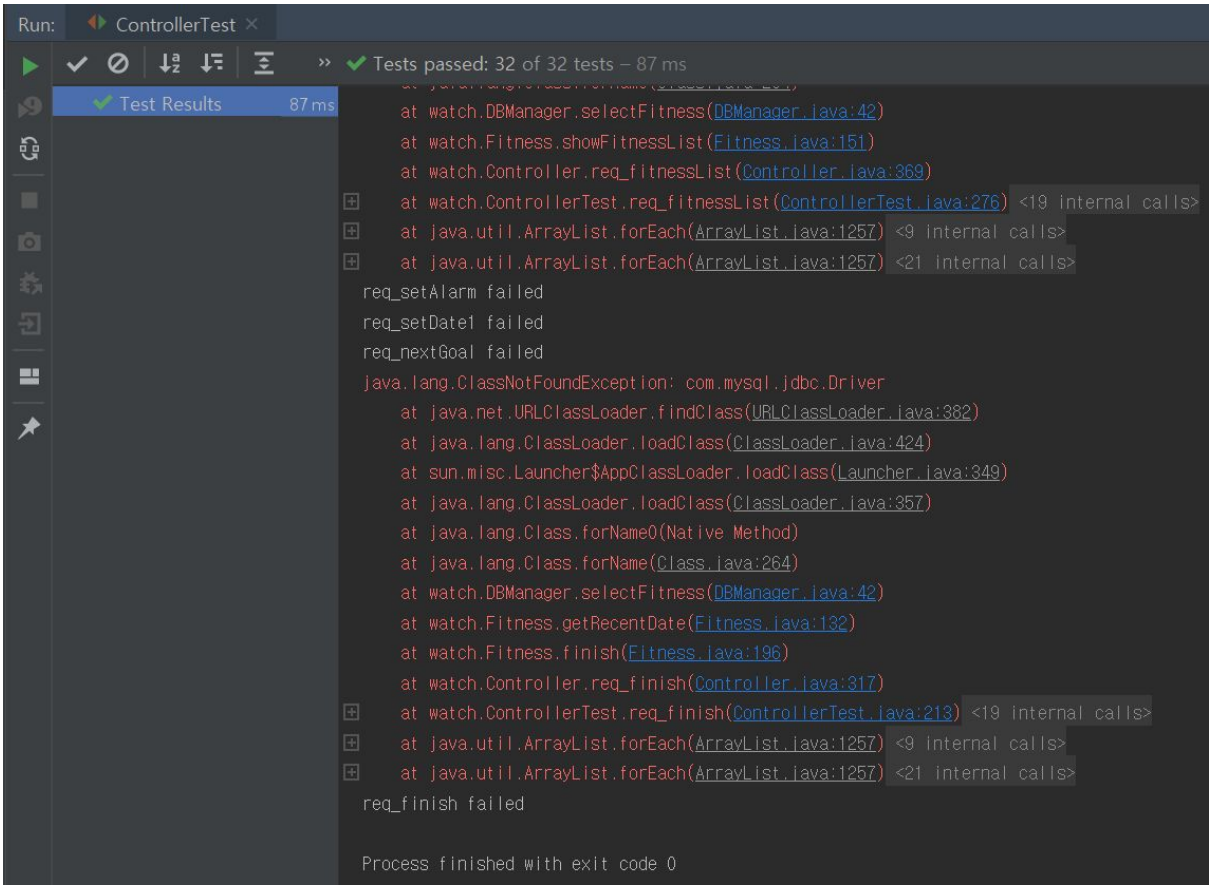
```java
@Test
void req_deleteAlarm()throws Exception {
    try {
        junitTest.req_deleteAlarm();
    }catch (Exception e){
        System.out.println("req_deleteAlarm failed");
    }
}

@Test
void req_record() throws Exception {
    try {
        junitTest.req_record();
    }catch (Exception e){
        System.out.println("req_record failed");
    }
}

@Test
void req_pause1() throws Exception {
    try {
        junitTest.req_pause();
    }catch (Exception e){
        System.out.println("req_pause1 failed");
    }
}
```

```java
@Test
void req_recordList() throws Exception {
    try {
        junitTest.req_recordList();
    }catch (Exception e){
        System.out.println("req_recordList failed");
    }
}

@Test
void req_finish() throws Exception {
    try {
        junitTest.req_finish( status: "fitness");
    }catch (Exception e){
        System.out.println("req_finish failed");
    }
}

@Test
void getBaseView() throws Exception {
    try {
        junitTest.getBaseView();
    }catch (Exception e){
        System.out.println("getBaseView failed");
    }
}
```

```java
@Test
void req_selectDate() throws Exception {
    try {
        junitTest.req_selectDate();
    }catch (Exception e){
        System.out.println("req_selectDate failed");
    }
}

@Test
void req_nextGoal() throws Exception {
    try {
        junitTest.req_nextGoal( status: "nextGoal");
    }catch (Exception e){
        System.out.println("req_nextGoal failed");
    }
}

@Test
void req_setGoal() throws Exception {
    try {
        junitTest.req_setGoal( currGoal: "stop smoking");
    }catch (Exception e){
        System.out.println("req_setGoal failed");
    }
}
```

```java
@Test
void req_DdayList()throws Exception {
    try {
        junitTest.req_DdayList();
    }catch (Exception e){
        System.out.println("req_DdayList failed");
    }
}

@Test
void req_deleteDday() throws Exception {
    try {
        junitTest.req_deleteDday();
    }catch (Exception e){
        System.out.println("req_deleteDday failed");
    }
}

@Test
void req_fitnessList() throws Exception {
    try {
        junitTest.req_fitnessList();
    }catch (Exception e){
        System.out.println("req_fitnessList failed");
    }
}
```

```
    @Test
    void req_nextExercise() throws Exception {
        try {
            junitTest.req_nextExercise();
        }catch (Exception e){
            System.out.println("req_nextExercise failed");
        }
    }

    @Test
    void req_setExercise()throws Exception {
        try {
            junitTest.req_setExercise("running");
        }catch (Exception e){
            System.out.println("req_setExercise failed");
        }
    }

    @Test
    void getInstManager() { junitTest.getInstManager(); }
}
```

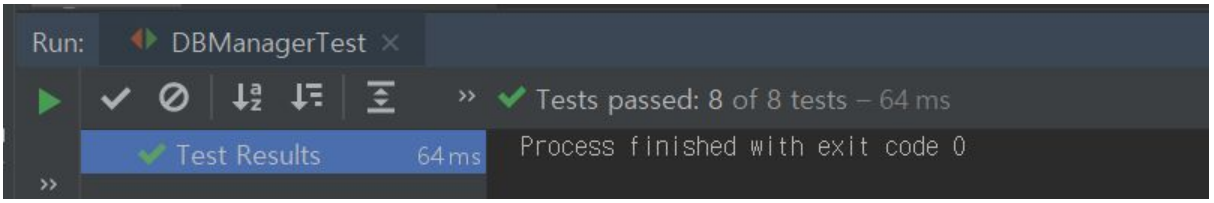

Run:
ControllerTest
Tests passed: 32 of 32 tests – 87 ms
Test Results
87 ms
at watch.DBManager.selectFitness(DBManager.java:42)
at watch.Fitness.showFitnessList(Fitness.java:151)
at watch.Controller.req_fitnessList(Controller.java:369)
at watch.ControllerTest.req_fitnessList(ControllerTest.java:276) <19 internal calls>
at java.util.ArrayList.forEach(ArrayList.java:1257) <9 internal calls>
at java.util.ArrayList.forEach(ArrayList.java:1257) <21 internal calls>
req_setAlarm failed
req_setDate1 failed
req_nextGoal failed
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)
at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:264)
at watch.DBManager.selectFitness(DBManager.java:42)
at watch.Fitness.getRecentDate(Fitness.java:132)
at watch.Fitness.finish(Fitness.java:196)
at watch.Controller.req_finish(Controller.java:317)
at watch.ControllerTest.req_finish(ControllerTest.java:213) <19 internal calls>
at java.util.ArrayList.forEach(ArrayList.java:1257) <9 internal calls>
at java.util.ArrayList.forEach(ArrayList.java:1257) <21 internal calls>
req_finish failed
Process finished with exit code 0

4. DBManagerTest

```
class DBManagerTest {
    public static DBManager junitTest;

    DBManagerTest(){
    }

    @BeforeAll
    public static void makeInstance() { junitTest = new DBManager(); }

    @Test
    void selectFitness() throws Exception{
        try{
            junitTest.selectFitness( status: "look");
        }catch (Exception e){
            System.out.println("selectFitness failed");
        }
    }

    @Test
    void insertFitness() throws Exception {
        try{
            junitTest.insertFitness( month: 3, date: 11,  hour: 9, minute: 30, second: 00,  totalCalories: 100);
        }catch (Exception e){
            System.out.println("insertFitness failed");
        }
    }
```

```java
@Test
void updateFitness() throws Exception{
    try {
        junitTest.updateFitness( hour: 9, minute: 30, second: 00,  totalCalories: 300);
    }catch (Exception e){
        System.out.println("updateFitness failed");
    }
}

@Test
void deleteFitness() throws Exception {
    try{
        junitTest.deleteFitness();
    }catch (Exception e){
        System.out.println("deleteFitness failed");
    }
}

@Test
void selectStopwatch() throws Exception {
    try{
        junitTest.selectStopwatch();
    }catch (Exception e){
        System.out.println("selectStopwatch failed");
    }

}
```

```java
@Test
void insertStopwatch() throws Exception {
    try{
        junitTest.insertStopwatch( hour: 9, minute: 30, second: 00);
    }catch (Exception e){
        System.out.println("insertStopwatch failed");
    }
}

@Test
void deleteStopwatch() throws Exception {
    try{
        junitTest.deleteStopwatch();
    }catch (Exception e){
        System.out.println("deleteStopwatch failed");
    }
}

@Test
void resetStopwatch() throws Exception {
    try{
        junitTest.resetStopwatch();
    }catch (Exception e){
        System.out.println("resetStopwatch failed");
    }
}
```



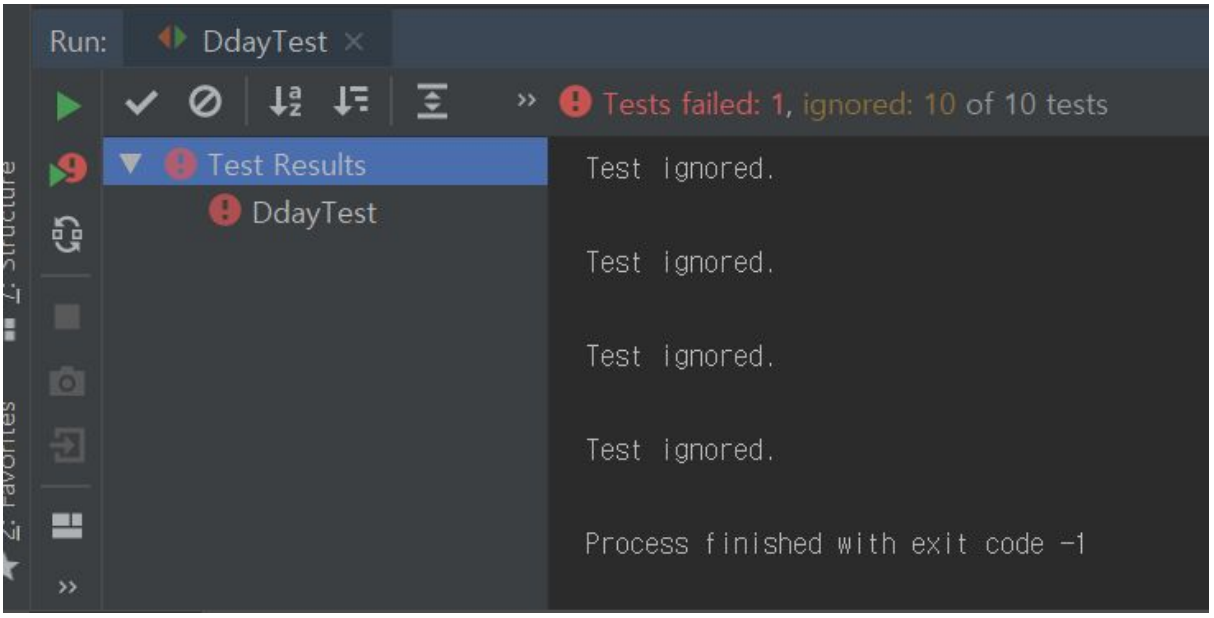
5. DdayTest

```
class DdayTest {
    public static Dday junitTest;

    DdayTest() {
    }

    @BeforeAll
    public static void makeInstance(){
        junitTest = new Dday();
    }

    @Test
    void setDate() throws Exception {
        try{
            junitTest.setDate( year: 2020,  month: 3,  date: 11);
        }catch(Exception e){
            System.out.println("error");
        }

    }

    @Test
    void showGoal() throws Exception {
        try {
            junitTest.showGoal( status: "nextGoal");
        } catch (Exception var2) {
            System.out.println("error");
        }
    }
```

```java
@Test
void calculateDday() throws Exception{
    try {
        junitTest.calculateDday();
    } catch (Exception var2) {
        System.out.println("error: 목록에 디데이가 없습니다(정상)");
    }
}

@Test
void setDday() throws Exception {
    try {
        junitTest.setDday();
    } catch (Exception var2) {
        System.out.println("error");
    }
}



@Test
void getGoal() throws Exception {
    try {
        Assertions.assertEquals((Object)null, junitTest.getGoal());
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void getDayCount() throws Exception{
    try {
        junitTest.getDayCount();
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void getGoalList() throws Exception {
    try {
        junitTest.getGoalList();
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void setDate1() throws Exception {
    try {
        junitTest.setDate(24);
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void setGoal()throws Exception {
    try {
        junitTest.setGoal("stop drinking");
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void setGoalList()throws Exception {
    String[] arr = new String[]{"happy", "smile"};

    try {
        junitTest.setGoalList(arr);
    } catch (Exception var3) {
        System.out.println("error");
    }
}
```



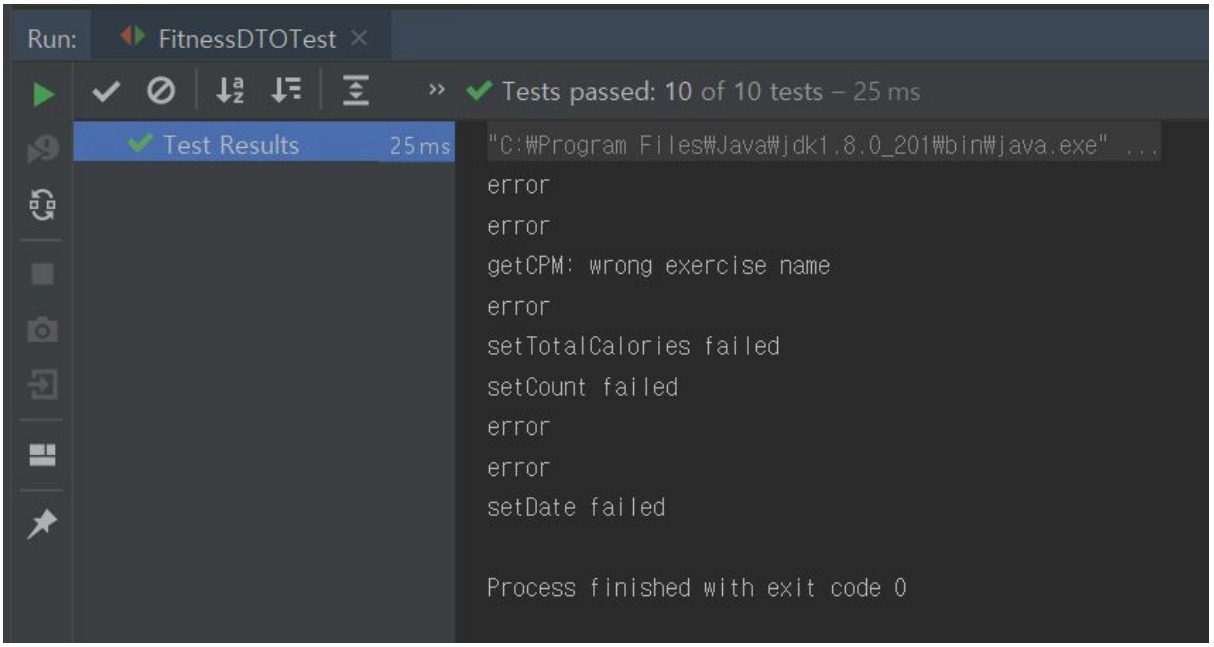
6. FitnessDTOTest

```
class FitnessDTOTest {
    public static FitnessDTO junitTest;

    FitnessDTOTest(){
    }

    @Test
    void getInstance() throws Exception{
        try {
            junitTest.getInstance();
        }catch (Exception e){
            System.out.println("getInstance failed");
        }
    }

    @Test
    void getMonth() throws Exception {
        try{
            assertEquals( expected: 5, junitTest.getMonth());
        }catch (Exception e){
            System.out.println("error");
        }
    }
```

```java
@Test
void getDate() throws Exception {
    try{
        assertEquals( expected: 0, junitTest.getDate());
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void getTotalCalories() throws Exception {
    try{
        assertEquals( expected: 0, junitTest.getTotalCalories());
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void getCount() throws Exception {
    try{
        assertEquals( expected: 0, junitTest.getCount());
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void getCPM()throws Exception {
    try{
        assertEquals( expected: 0, junitTest.getCPM( exercise: "running"));
    }catch (Exception e){
        System.out.println("getCPM: wrong exercise name");
    }
}

@Test
void getNextExercise() throws Exception {
    try{
        assertEquals( expected: "null", junitTest.getNextExercise());
    }catch(Exception e){
        System.out.println("error");
    }
}


@Test
void setDate() throws  Exception {
    try{
        junitTest.setDate(11);
    }catch (Exception e){
        System.out.println("setDate failed");
    }
}
```

```java
@Test
void setTotalCalories() throws  Exception {
    try{
        junitTest.setTotalCalories(200);
    }catch (Exception e){
        System.out.println("setTotalCalories failed");
    }
}

@Test
void setCount() throws  Exception {
    try{
        junitTest.setCount(5);
    }catch (Exception e){
        System.out.println("setCount failed");
    }
}
```



7. FitnessTest

```
@BeforeAll
public static void makeInstance() { junitTest = new Fitness(); }

@Test
void getCPM()throws Exception{
    try{
        assertEquals(0,junitTest.getCPM());
    }catch(Exception e){
        System.out.println("error");
    }
}

@Test
void getTotalCalories() throws Exception{
    try{
        assertEquals(0,junitTest.getTotalCalories());
    }catch(Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void getCount() throws Exception{
    try{
        assertEquals(0,junitTest.getCount());
    }catch(Exception e){
        System.out.println("error");
    }
}

@Test
void getExercise() throws Exception{
    try {
        assertEquals(null, junitTest.getExercise());
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void isIs_exist() throws Exception{
    try {
        junitTest.isIs_exist();
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void getIs_stop()  throws Exception{
    try {
        junitTest.getIs_stop();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void setCPM()throws Exception{
    try {
        junitTest.setCPM("bike");
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void setTotalCalories()throws Exception{
    try {
        junitTest.setTotalCalories(2100);
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void setCount() throws Exception{
    try {
        junitTest.setCount(5);
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void setExercise()throws Exception{
    try {
        junitTest.setExercise("running");
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void setIs_exist() throws Exception {
    try {
        junitTest.setIs_exist(true);
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void setIs_stop()  throws Exception {
    try {
        junitTest.setIs_stop(false);
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void getRecentDate() throws Exception {
    try {
        junitTest.getRecentDate();
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void checkDate()  throws Exception {
    try {
        junitTest.checkDate();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void showFitnessList()  throws Exception {
    try {
        junitTest.showFitnessList();
    }catch (Exception e){
        System.out.println("error");
    }
}

@Ignore
void countUp() throws Exception {
    try {
        junitTest.countUp();
    }catch (Exception e){
        System.out.println("error: 무한히 test하기 때문에 ignore처리");
    }
}
```

```java
@Test
void calcultateCalories()  throws Exception {
    try {
        junitTest.calcultateCalories();
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void finish()  throws Exception {
    try {
        junitTest.finish();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void updateFitness()  throws Exception {
    try {
        junitTest.updateFitness(1, 30, 30, 2000);
    }catch (Exception e){
        System.out.println("error");
    }
}

@Test
void deleteFitness()  throws Exception {
    try {
        junitTest.deleteFitness();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void initFitness() throws Exception {
    try {
        junitTest.initFitness();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

8. InstManagerTest

```
@Test
void getInstance() { junitTest.getInstance(); }

@Test
void getAlarmIndex() throws Exception{
    try{
        junitTest.getAlarmIndex();
    }catch (Exception e){
        System.out.println("getAlarmIndex failed");
    }
}

@Test
void getdDayIndex()  throws Exception{
    try{
        junitTest.getdDayIndex();
    }catch (Exception e){
        System.out.println("getdDayIndex failed");
    }
}
```

```java
@Test
void setAlarmIndex()  throws Exception{
    try{
        junitTest.setAlarmIndex(3);
    }catch (Exception e){
        System.out.println("setAlarmIndex failed");
    }
}

@Test
void setdDayIndex()  throws Exception{
    try{
        junitTest.setdDayIndex(2);
    }catch (Exception e){
        System.out.println("setdDayIndex failed");
    }
}
```

```java
@Test
void getAlarmInstNum()  throws Exception{
    try{
        junitTest.getAlarmInstNum();
    }catch (Exception e){
        System.out.println("getAlarmInstNum failed");
    }
}

@Test
void getdDayInstNum()  throws Exception{
    try{
        junitTest.getdDayInstNum();
    }catch (Exception e){
        System.out.println("getdDayInstNum failed");
    }
}
```

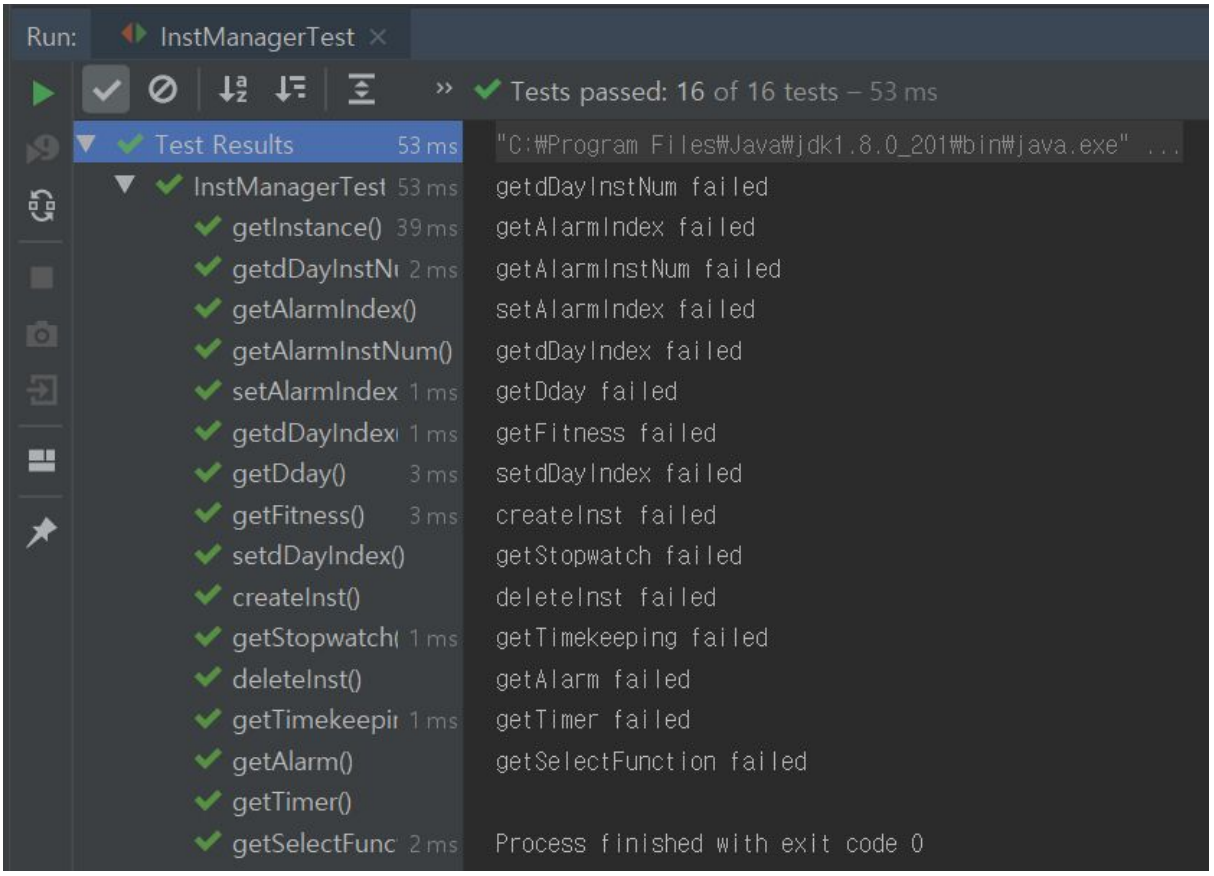
```java
@Test
void getTimekeeping()  throws Exception{
    try{
        junitTest.getTimekeeping();
    }catch (Exception e){
        System.out.println("getTimekeeping failed");
    }
}

@Test
void getTimer()  throws Exception{
    try{
        junitTest.getTimer();
    }catch (Exception e){
        System.out.println("getTimer failed");
    }
}
```

```java
@Test
void getAlarm()  throws Exception{
    try{
        junitTest.getAlarm();
    }catch (Exception e){
        System.out.println("getAlarm failed");
    }
}

@Test
void getStopwatch() throws Exception{
    try{
        junitTest.getStopwatch();
    }catch (Exception e){
        System.out.println("getStopwatch failed");
    }
}
```

```
@Test
void getDday()  throws Exception{
    try{
        junitTest.getDday();
    }catch (Exception e){
        System.out.println("getDday failed");
    }
}

@Test
void getFitness()  throws Exception{
    try{
        junitTest.getFitness();
    }catch (Exception e){
        System.out.println("getFitness failed");
    }
}
```

```java
@Test
void getSelectFunction()  throws Exception{
    try{
        junitTest.getSelectFunction();
    }catch (Exception e){
        System.out.println("getSelectFunction failed");
    }
}

@Test
void deleteInst() throws Exception{
    try{
        junitTest.deleteInst("alarm");
    }catch (Exception e){
        System.out.println("deleteInst failed");
    }
}
```

```java
@Test
void createInst() throws Exception{
    try{
        junitTest.createInst("dDay");
    }catch (Exception e){
        System.out.println("createInst failed");
    }
}
```

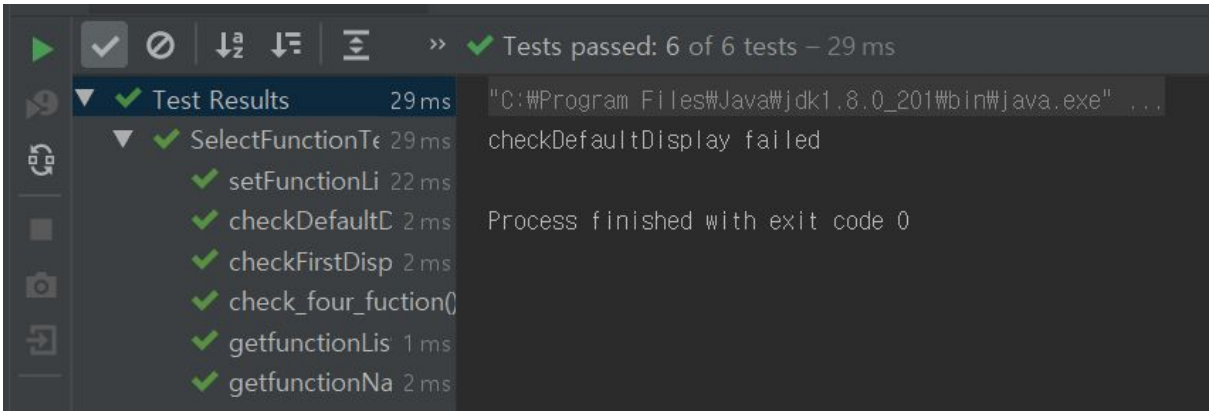

Run:
InstManagerTest
Tests passed: 16 of 16 tests – 53 ms
Test Results 53 ms
InstManagerTest 53 ms
getInstance() 39 ms
getdDayInstN 2 ms
getAlarmIndex()
getAlarmInstNum()
setAlarmIndex 1 ms
getdDayIndex 1 ms
getDday() 3 ms
getFitness() 3 ms
setdDayIndex()
createInst()
getStopwatch 1 ms
deleteInst()
getTimekeepi 1 ms
getAlarm()
getTimer()
getSelectFunc 2 ms
"C:₩Program Files₩Java₩jdk1.8.0_201₩bin₩java.exe" ...
getdDayInstNum failed
getAlarmIndex failed
getAlarmInstNum failed
setAlarmIndex failed
getdDayIndex failed
getDday failed
getFitness failed
setdDayIndex failed
createInst failed
getStopwatch failed
deleteInst failed
getTimekeeping failed
getAlarm failed
getTimer failed
getSelectFunction failed
Process finished with exit code 0

9. SelectFunctionTest

```
@BeforeAll
public static void makeInstance() { junitTest = new SelectFunction(); }

@Test
void getfunctionName() throws Exception{
    try{
        assertEquals("TimeKeeping", junitTest.getfunctionName());
    }catch (Exception e){
        System.out.println("getfunctionName failed");
    }
}

@Test
void getfunctionList() throws Exception{
    try{
        assertEquals(1, junitTest.getfunctionList());
    }catch (Exception e){
        System.out.println("getfunctionList failed");
    }
}
```

```
@Test
void checkFirstDisplay() throws Exception{
    try{
        assertEquals(true, junitTest.checkFirstDisplay("TimeKeeping"));
    }catch (Exception e){
        System.out.println("checkFirstDisplay failed");
    }
}

@Test
void checkDefaultDisplay() throws Exception{
    try{
        assertEquals(false, junitTest.checkDefaultDisplay());
    }catch (Exception e){
        System.out.println("checkDefaultDisplay failed");
    }
}
```

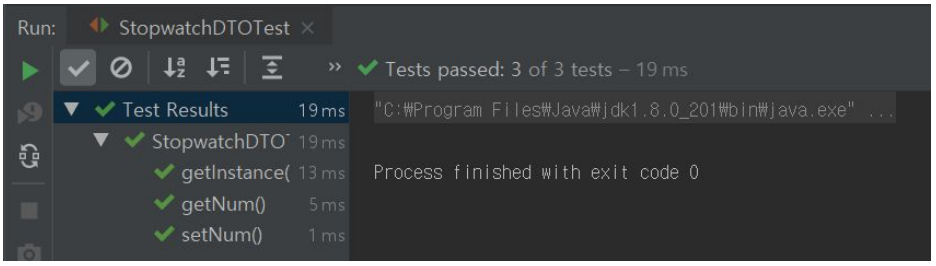
```java
@Test
void setFunctionList() throws Exception{
    try{
        assertEquals(true, junitTest.setFunctionList(2));
    }catch (Exception e){
        System.out.println("setFunctionList failed");
    }
}

@Test
void check_four_fuction() throws Exception{
    try{
        assertEquals(false, junitTest.check_four_fuction());
    }catch (Exception e){
        System.out.println("check_four_function failed");
    }
}
```



10. StopwatchDTOTest

```
@BeforeAll
public static void makeInstance() { junitTest = new StopwatchDTO(); }

@Test
void getInstance() throws Exception{
    try {
        junitTest.getInstance();
    }catch (Exception e){
        System.out.println("getInstance failed");
    }
}
```

```
@Test
void getNum()  throws Exception {
    try {
        assertEquals(0, junitTest.getNum());
    }catch (Exception e){
        System.out.println("getNum failed");
    }
}


@Test
void setNum()  throws  Exception{
    try{
        junitTest.setNum(3);
    }catch (Exception e){
        System.out.println("setNum failed");
    }
}
```



11. StopwatchTest

```
@BeforeAll
public static void makeInstance() throws Exception{
    try {
        junitTest = new Stopwatch();
    }catch (Exception e){
        System.out.println("error");
    }
}


@Test
void getIs_stop() throws Exception {
    try {
        assertEquals(false, junitTest.getIs_stop());
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```
@Test
void setIs_stop() throws Exception{
    try{
        junitTest.setIs_stop(true);
    }catch (Exception e){
        System.out.println("error");
    }
}

@Ignore
void run() {
    junitTest.run();
    System.out.println("run이 끝나지 않아 ignore처리");
}
```

```java
@Ignore
void countUp()  throws Exception{
    try{
        junitTest.countUp();
    }catch (Exception e){
        System.out.println("error:무한히 test되므로 Ignore");
    }
}

@Test
void reset() throws Exception {
    try {
        junitTest.reset();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

```java
@Test
void record() throws Exception {
    try {
        junitTest.record();
    }catch (Exception e){
        System.out.println("error");
    }
}
```

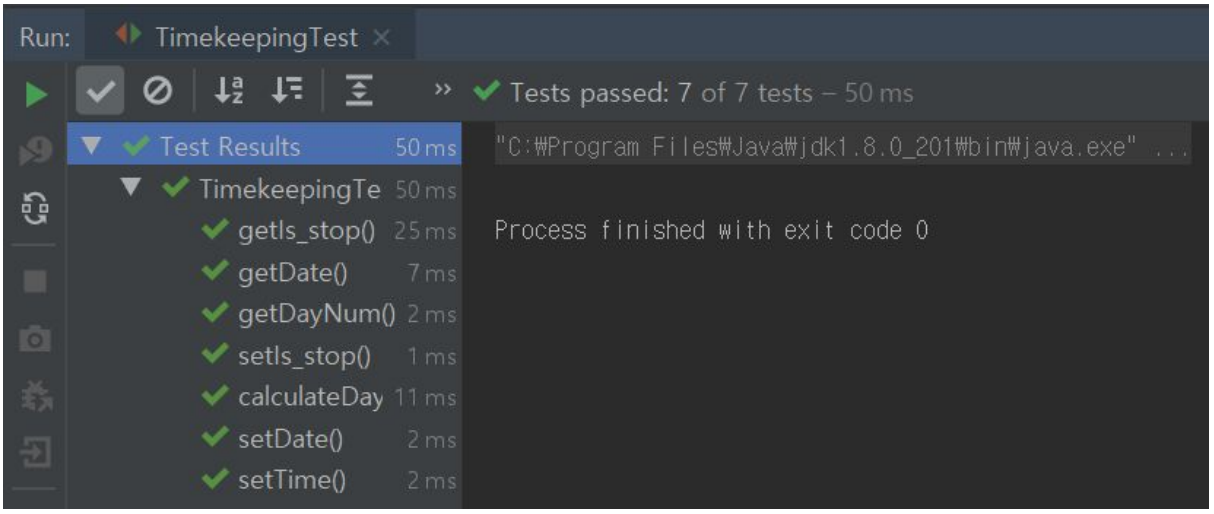
12. TimekeepingTest

```
@BeforeAll
public static void makeInstance(){
    try {
        junitTest = new Timekeeping();
    } catch (Exception var1) {
        System.out.println("error");
    }
}


@Test
void getDate() throws Exception {
    try {
        Assertions.assertEquals(26, junitTest.getDate());
    } catch (Exception var2) {
        System.out.println("error: expect값이 현재 '일'과 동일하지 않습니다");
    }
}
```

```
@Test
void getDayNum()  throws Exception {
    try {
        Assertions.assertEquals(1, junitTest.getDayNum());
    } catch (Exception var2) {
        System.out.println("error: expect값이 현재 '요일'과 동일하지 않습니다.");
    }
}

@Test
void getIs_stop() throws Exception{
    try{
        junitTest.getIs_stop();
    }catch (Exception var2){
        System.out.println("error");
    }
}
```

```java
@Test
void setDate() throws Exception {
    try {
        junitTest.setDate(2020, 3, 11);
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void setTime() throws Exception {
    try {
        junitTest.setTime(6, 30, 30);
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void setIs_stop() throws Exception {
    try{
        junitTest.setIs_stop(false);
    }catch (Exception var2){
        System.out.println("error");
    }
}

@Ignore
void countUp()  throws Exception {
    try {
        junitTest.countUp();
    } catch (Exception var2) {
        System.out.println("error: 테스트가 무한히 진행되므로 Ignore처리해줍니다");
    }
}
```


Run:
TimekeepingTest
Tests passed: 7 of 7 tests – 50 ms
Test Results 50 ms
TimekeepingTe 50 ms
getIs_stop() 25 ms
getDate() 7 ms
getDayNum() 2 ms
setIs_stop() 1 ms
calculateDay 11 ms
setDate() 2 ms
setTime() 2 ms
"C:₩Program Files₩Java₩jdk1.8.0_201₩bin₩java.exe" ...
Process finished with exit code 0

13. TimerTest

```
@BeforeAll
public static void makeInstance() { junitTest = new Timer(); }


@Test
void getBuzzer()  throws Exception {
    try {
        junitTest.getBuzzer();
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```
@Test
void setTime()  throws Exception {
    try {
        junitTest.setTime(2, 10, 30);
    } catch (Exception var2) {
        System.out.println("error");
    }
}


@Ignore
void countDown() throws Exception {
    try {
        junitTest.countDown();
    } catch (Exception var2) {
        System.out.println("error: 무한하게 test하기때문에 ignore처리");
    }
}
```

```java
@Test
void checkTimer() throws Exception {
    try {
        junitTest.checkTimer();
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void reset() throws Exception {
    try {
        junitTest.reset();
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void getIs_stop() throws Exception{
    try{
        assertEquals(false, junitTest.getIs_stop());
    }catch (Exception var2){
        System.out.println("error");
    }
}

@Test
void setIs_stop() throws Exception{
    try {
        junitTest.setIs_stop(true);
    }
    catch (Exception var2){
        System.out.println("error");
    }
}
```