

Activity 2052. Implement Windows

1. WorldTimeMode

Name	initWorldTimeMode()
Responsibilities	세계시간모드를 초기화한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	20개 국가들의 이름, 시간 정보로 world 배열 초기화, 시간 계속 업데이트한다.

Name	nextWorldTime
Responsibilities	다음 세계시간을 보여준다.
Type	System
Cross References	R8.1
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	세계시간 배열의 다음 인덱스로 넘어가 해당 인덱스의 세계시간이 화면에 출력된다.

Name	prevWorldTime()
Responsibilities	이전 세계시간을 보여준다.
Type	System
Cross References	R8.2
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	세계시간 배열의 이전 인덱스로 넘어가 해당 인덱스의 세계시간이 화면에 출력된다.

Name	increaseWorldTimeIndex()
Responsibilities	세계시간을 담은 배열의 인덱스를 가리키는 변수를 1씩 증가시킨다.
Type	System
Cross References	R8.1
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	인덱스를 가리키는 변수가 1 증가한다.

Name	decreaseWorldTimeIndex()
Responsibilities	세계시간을 담은 배열의 인덱스를 가리키는 변수를 1씩 감소시킨다.
Type	System
Cross References	R8.2
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	인덱스를 가리키는 변수가 1 감소한다.

Name	holdCurrentWorldTime()
Responsibilities	20개의 세계시간 중 현재 화면에 출력되어지는 세계시간으로 화면에 고정시킨다.
Type	System
Cross References	R8.3
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	prex, next 버튼을 눌러도 작동하지 않는다.

Name	releaseCurrentWorldTimeRock()
Responsibilities	화면에 고정시킨 세계시간을 풀어준다.
Type	System
Cross References	R8.4
Exceptions	N/A
Output	N/A

Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	prev, next 버튼을 누르면 작동한다.

Name	syncWorldTime()
Responsibilities	매개변수로 넘겨진 세계시간으로 동기화한다.
Type	System
Cross References	R8.5
Exceptions	N/A
Output	N/A
Pre-Conditions	WorldTimeMode에 진입한 상태이어야한다.
Post-Conditions	화면에 출력되어질 세계시간을 현재시간에 가중치를 더해 동기화한 후, 해당 세계시간이 출력된다.

2. ThemeMode

Name	initThemeMode()
Responsibilities	테마모드를 초기화한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	ThemeMode에 진입한 상태이어야한다.
Post-Conditions	8개의 테마로 배열 초기화, 디폴트 테마로 적용한다.

Name	nextTheme()
Responsibilities	다음 테마를 보여준다.
Type	System
Cross References	R9.1
Exceptions	N/A
Output	N/A
Pre-Conditions	ThemeMode에 진입한 상태이어야한다.
Post-Conditions	테마의 인덱스를 가리키는 변수가 1 증가해 다음 테마가 화면에 출력된다.

Name	prevTheme()
Responsibilities	이전 테마를 보여준다.
Type	System
Cross References	R9.2
Exceptions	N/A
Output	N/A
Pre-Conditions	ThemeMode에 진입한 상태이어야한다.
Post-Conditions	테마를 가리키는 변수가 1 감소해 이전 테마가 화면에 출력된다.

Name	decideTheme()
Responsibilities	8개의 테마 중 현재 화면에 출력되는 테마로 결정한다.
Type	System
Cross References	R9.3
Exceptions	N/A
Output	N/A
Pre-Conditions	ThemeMode에 진입한 상태이어야한다.
Post-Conditions	현재 화면에 출력중인 테마로 테마가 반영된다.

Name	syncUiWithTheme()
Responsibilities	결정된 테마를 시계에 적용시킨다.
Type	System
Cross References	R9.3
Exceptions	N/A
Output	N/A
Pre-Conditions	ThemeMode에 진입한 상태이어야한다.
Post-Conditions	시계의 모든 부분이 해당 테마로 적용된다.

3. TimerMode

Name	initTimerMode()
Responsibilities	타이머를 초기화해준다.
Type	System
Cross References	R1.1
Exceptions	N/A

Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다.
Post-Conditions	타이머에 관한 스레드를 만들어 주고, 0이 되었을 때 버저를 울리는 스레드를 생성해준다.

Name	startTimer()
Responsibilities	타이머를 시작한다.
Type	System
Cross References	R6.1
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 사용자가 타이머를 설정한 이후이어야 한다.
Post-Conditions	사용자가 설정한 시간을 초단위로 바꾸고 타이머를 시작한다. A, B, C 의 버튼 매핑을 바꿔준다.

Name	pauseTimer()
Responsibilities	타이머를 중지해준다.
Type	System
Cross References	R6.4
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 타이머가 시작된 이후여야 한다.
Post-Conditions	타이머에 대한 스레드를 중지시키고 그에 맞게 버튼 매핑을 바꿔준다.

Name	continueTimer()
Responsibilities	중지한 타이머를 다시 시작한다.
Type	System
Cross References	R6.5
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 타이머가 중지한 이후여야 한다.
Post-Conditions	타이머에 대한 스레드를 실행시키고

	그에 맞게 버튼 매핑을 바꿔준다.
--	--------------------

Name	cancelTimer()
Responsibilities	타이머를 취소한다.
Type	System
Cross References	R6.6
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 타이머가 중지한 이후여야 한다.
Post-Conditions	타이머에 대한 스레드를 취소시키고 그에 맞게 버튼 매핑을 바꿔준다.

Name	increaseTimerMinute()
Responsibilities	타이머 설정시간을 1분 증가한다.
Type	System
Cross References	R6.2
Exceptions	59분이 되었을 때 이 함수를 호출하게 되면 0 분으로 설정해준다.
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다.
Post-Conditions	타이머 설정시간이 1분 증가된다. 증가된 설정시간을 출력해준다.

Name	increaseTimerSeconds()
Responsibilities	타이머 설정시간을 1초 증가한다.
Type	System
Cross References	R6.3
Exceptions	59초 되었을 때 이 함수를 호출하게 되면 0초 로 설정해준다.
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다.
Post-Conditions	타이머 설정시간이 1초 증가된다. 증가된 설정시간을 출력해준다.

Name	checkTimerZero()
------	------------------

Responsibilities	타이머 설정 시간이 00:00 인지 알아본다.
Type	System
Cross References	R6.1
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다.
Post-Conditions	설정시간이 00 : 00 이라면 true를 반환하고 아니라면 false를 반환한다.

Name	checkTimerEnded()
Responsibilities	타이머 시간이 0초가 되었는 지 알아본다.
Type	System
Cross References	R6.7
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 타이머 스레드가 호출된 이후이어야한다.
Post-Conditions	타이머 스레드가 0초가 되었다면 true, 아니면 false를 반환한다.

Name	decreaseTimer()
Responsibilities	타이머 시간이 1초씩 줄어든다.
Type	System
Cross References	R6.1
Exceptions	N/A
Output	N/A
Pre-Conditions	TimerMode에 진입한 상태이어야한다. 타이머가 시작된 이후이어야 한다.
Post-Conditions	타이머 시간이 초당 1씩 줄어들고, 그것을 출력해준다.

Name	mappingTimerState()
Responsibilities	버튼 매핑을 바꿔준다.
Type	System
Cross References	R6.6
Exceptions	N/A
Output	N/A

Pre-Conditions	N/A
Post-Conditions	buttonA에는 IncreaseTimerMinute()을 호출하는 기능을 저장하고, buttonB에는 IncreaseTimerSeconds()을 호출하는 기능을 buttonC에는 StartTimer()을 호출하는 기능을 저장한다.

Name	mappingTimerRunning()
Responsibilities	Run 상태일 때, 버튼 매핑을 바꿔준다.
Type	System
Cross References	R6.1, R6.5
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	buttonA에는 아무 기능도 호출되지 않게 하고 buttonB에는 PauseTimer() buttonC에는 아무 기능도 호출되지 않게 한다.

Name	mappingTimerPause()
Responsibilities	중지 상태일 때, 버튼 매핑을 바꿔준다.
Type	System
Cross References	R6.4
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	buttonA에는 아무 기능도 호출되지 않게 하고 buttonB에는 ContinueTimer() buttonC에는 CancelTimer()을 호출하는 기능을 저장한다.

Name	displayTimer()
Responsibilities	타이머의 작동에 맞는 화면을 출력해준다.
Type	System
Cross References	R6.1, R6.2, R6.3, R6.4, R6.5, R6.6
Exceptions	N/A
Output	N/A

Pre-Conditions	TimerMode에 진입한 상태이어야한다.
Post-Conditions	타이머의 상태에 따라 변하는 화면을 세그먼트에 출력해준다.

4. StopwatchMode

Name	initStopWatchMode()
Responsibilities	스톱워치를 초기화해준다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다.
Post-Conditions	스톱워치에 관한 스레드를 만들어 준다. 스톱워치 시간설정은 00:00으로 초기화해준다.

Name	startStopWatch()
Responsibilities	스톱워치를 시작한다.
Type	System
Cross References	R5.1
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다.
Post-Conditions	스톱워치 스레드를 작동시키고 A, B, C 의 버튼 매핑을 바꿔준다. 스톱워치가 시작되고 있는 화면을 출력해준다.

Name	pauseStopWatch()
Responsibilities	스톱워치를 중지한다.
Type	System
Cross References	R5.2
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다. 스톱워치가 시작된 이후여야 한다.
Post-Conditions	스톱워치 스레드를 중지시키고 A, B, C 의 버튼 매핑을 바꿔준다. 타이머가 중지한 것을 출력해준다.

Name	continueStopWatch()
Responsibilities	정지한 스톱워치를 다시 시작한다.
Type	System
Cross References	R5.3
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다. 스톱워치가 정지된 이후여야 한다.
Post-Conditions	스톱워치 스레드를 중지시키고 A, B, C 의 버튼 매핑을 바꿔준다. 타이머가 다시 작동하는 것을 출력해준다.

Name	increaseStopWatchSeconds()
Responsibilities	스톱워치를 초당 1씩 증가해준다.
Type	System
Cross References	R5.1
Exceptions	60초가 되었을 때, 초는 0으로 바꿔주고 분은 1만큼 증가해준다.
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다. 스톱워치가 시작된 이후여야 한다.
Post-Conditions	스톱워치가 초당 1씩 증가하고, 그에 맞는 시 간을 출력해준다.

Name	testStopWatchMax()
Responsibilities	스톱워치가 한계치에 도달했는지 확인해준다.
Type	System
Cross References	R5.1
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다. 스톱워치가 시작된 이후여야 한다.
Post-Conditions	스톱워치가 한계치인 59:59가 되었을 때 true 를 return 해준다.

Name	resetStopWatch()
Responsibilities	스톱워치를 reset 해준다.
Type	System
Cross References	R5.4
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다. 스톱워치가 시작된 이후여야 한다.
Post-Conditions	스톱워치의 스레드를 정지시키고, 00:00 으로 초기화해준다. 이후 이에 관한 내용을 출력해준다.

Name	mappingStopWatchState()
Responsibilities	버튼 매핑을 바꿔준다.
Type	System
Cross References	R5.1, R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	buttonA에는 StartStopWatch()를 호출하는 기 능을 buttonB, buttonC에는 기능이 없는 상태로 할당한다

Name	mappingStopWatchRunning()
Responsibilities	버튼 매핑을 바꿔준다.
Type	System
Cross References	R5.1, R5.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	buttonA 에는 기능이 없는 상태로 할당한다. buttonB, 에는 PauseStopWatch()를 호출하는 기능을 할당한다. buttonC에는 기능이 없는 상태로 할당한다

Name	mappingStopWatchPause()
Responsibilities	버튼 매핑을 바꿔준다.
Type	System
Cross References	R5.3
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	buttonA 에는 기능이 없는 상태로 할당한다. buttonB, 에는 ContinueStopWatch()를 호출하는 기능을 할당한다. buttonC에는 ResetStopWatch()를 호출하는 기능을 할당한다.

Name	displayStopWatchState()
Responsibilities	스톱워치의 작동에 맞는 화면을 출력해준다.
Type	System
Cross References	R5.1, R5.2, R5.3, R5.4
Exceptions	N/A
Output	N/A
Pre-Conditions	StopWatchMode에 진입한 상태이어야한다.
Post-Conditions	스톱워치의 상태에 따라 변하는 화면을 세그먼트에 출력해준다.

5. CurrentTimeMode

Name	initCurrentTimeMode()
Responsibilities	현재시간을 초기화해준다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	CurrentTimeMode에 진입한 상태이어야한다.
Post-Conditions	현재시간에 관한 스레드를 만들어준다.

Name	SyncWithCurrentTime()
Responsibilities	현재시간을 동기화해준다.
Type	System
Cross References	R4.1
Exceptions	N/A
Output	N/A
Pre-Conditions	CurrentTimeMode에 진입한 상태이어야한다.
Post-Conditions	현재시간을 얻어온다. 얻어온 현재시간과 날짜를 기준으로 세그먼트에 표시해준다.

6. AlarmMode

Name	initAlarmMode()
Responsibilities	알람 모드의 변수를 초기화한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	시간을 0으로 초기화 해주고 actived를 false로 초기화해준다.

Name	enableAlarm()
Responsibilities	알람을 활성화시킨다.

Type	System
Cross References	R7.6
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	현재 알람의 상태를 활성화시킨다..

Name	disableAlarm()
Resposibilities	알람을 비활성화시킨다.
Type	System
Cross References	R7.7
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	현재 알람의 상태를 비활성화시킨다.

Name	checkRingAlarmExist()
Resposibilities	알람을 확인해서 버저를 올린다.
Type	System
Cross References	R7.5
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이고 actived가 true 여야한다.
Post-Conditions	현재시간과 알람시간을 비교해서 버저가 올리 게 한다.

Name	increaseAlarmIndex()
Resposibilities	알람 인덱스를 증가시킨다.
Type	System
Cross References	R7.1
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	알람 인덱스를 증가시켜 다음 알람에 접근할 수 있도록 한다.

Name	addAlarm()
Responsibilities	알람을 추가한다.
Type	System
Cross References	R7.3.1
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	AlarmTimeSettingMode를 호출하여 버튼 매핑을 해주도록한다.

Name	increaseOneHour()
Responsibilities	알람 설정시간을 한 시간 증가시킨다.
Type	System
Cross References	R7.3.2
Exceptions	23시가 되었을 때 이 함수를 호출하게 되면 0시으로 설정해준다.
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	알람 설정시간이 1시간 증가된다. 증가된 설정시간을 출력해준다.

Name	increasefiveMinute()
Responsibilities	알람 설정시간을 5분 증가한다.
Type	System
Cross References	R7.3.3
Exceptions	55분이 되었을 때 이 함수를 호출하게 되면 0분으로 설정해준다.
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	알람 설정시간이 5분 증가된다. 증가된 설정시간을 출력해준다.

Name	alarmInitTime()
Responsibilities	알람 설정 시간을 초기화해준다.
Type	System

Cross References	R7.2
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	설정시간을 0으로 초기해주고 actived 또한 none 상태로 만든다.

Name	mappingAlarmTimeSettingMode()
Responsibilities	알람 설정 모드로 진입한다.
Type	System
Cross References	R7.3.1
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이고 addAlarm을 누른 상태여야한다..
Post-Conditions	새로운 알람을 세팅할 수 있도록 버튼 매핑을 해준다.

Name	mappingAlarmMode()
Responsibilities	알람 확인 모드로 진입한다.
Type	System
Cross References	R7.4
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	알람이 없다면 buttonA에 addAlarm 기능만 활성화하고, 알람이 있다면 buttonA에 enable/disableAlarm buttonB에 nextAlarm, buttonC에 deleteAlarm 을 매핑해준다.

Name	nextAlarm()
Responsibilities	다음 알람으로 넘어간다.
Type	System
Cross References	R7.1
Exceptions	N/A
Output	N/A

Pre-Conditions	AlarmMode에 진입한 상태이어야하며 알람이 존재해야한다..
Post-Conditions	다음 알람을 화면에 출력한다.

Name	decideAlarm()
Responsibilities	설정된 알람 시간을 추가한다.
Type	System
Cross References	R 7.4
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야하며 알람 설정 상태여야한다.
Post-Conditions	설정된 알람시간을 알람리스트에 추가한다..

Name	deleteAlarm()
Responsibilities	알람을 제거한다.
Type	System
Cross References	R7.2
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야하며 알람이 존재해야한다..
Post-Conditions	현재 알람을 알람리스트에서 제거한다.

Name	displayCurrentAlarm()
Responsibilities	알람의 작동에 맞는 화면을 출력해준다.
Type	System
Cross References	Rdi, R7.2, R7.3, R7.4, R7.6, R7.7
Exceptions	N/A
Output	N/A
Pre-Conditions	AlarmMode에 진입한 상태이어야한다.
Post-Conditions	알람의 상태에 따라 변하는 화면을 세그먼트에 출력해준다.

7. System

Name	Press A Button
Responsibilities	사용자가 A 버튼을 누른다.
Type	GUI
Cross Reference	R7.6, R7.7, R7.3.1, R6.2, R1.4, R5.1, R8.2, R9.2
Notes	<ul style="list-style-type: none"> -메인화면 <ul style="list-style-type: none"> - 이전 모드 전환 -ModeConfig <ul style="list-style-type: none"> - change -현재시간 <ul style="list-style-type: none"> - 비활성화 -알람 <ul style="list-style-type: none"> - 활성화/비활성화 - 알람 추가 - 1시간 추가 -타이머 <ul style="list-style-type: none"> - 분 추가 - 비활성화 / 실행중 - 비활성화 / pause 중 -스톱위치 <ul style="list-style-type: none"> - 시작 - 비활성화 -세계시간 <ul style="list-style-type: none"> - 이전 시간 - 비활성화 / hold 상태 -테마 <ul style="list-style-type: none"> - 이전 테마
Pre-Conditions	N/A
Post-Conditions	현재 모드에서, A버튼에 매핑 된 기능이 호출된다.

Name	Press B Button
Responsibilities	사용자가 B 버튼을 누른다.
Type	GUI
Cross Reference	R1.3, R7.1, R7.3.3, R6.3, R6.4, R6.5, R5.2, R5.3, R9.1, R8.1
Notes	<ul style="list-style-type: none"> -메인화면 <ul style="list-style-type: none"> - 다음 모드 전환 -ModeConfig <ul style="list-style-type: none"> - change

	<ul style="list-style-type: none"> -현재시간 <ul style="list-style-type: none"> - 비활성화 -알람 <ul style="list-style-type: none"> - 다음 알람 - 5분 추가 -타이머 <ul style="list-style-type: none"> - 초 추가 - 일시정지 / 실행중 - 재시작 / pause 중 -스톱위치 <ul style="list-style-type: none"> - 비활성화 - pause - continue -세계시간 <ul style="list-style-type: none"> - 다음 시간 - 비활성화 / hold 상태 -테마 <ul style="list-style-type: none"> - 다음 테마
Pre-Conditions	N/A
Post-Conditions	현재 모드에서, B버튼에 매핑 된 기능이 호출된다.

Name	Press C Button
Responsibilities	사용자가 C 버튼을 누른다.
Type	GUI
Cross Reference	R1.5, R7.2, R7.4, R6.1, R6.6, R5.4, R8.3, R8.4
Notes	<ul style="list-style-type: none"> -메인화면 <ul style="list-style-type: none"> - 현재 모드 사용 -현재시간 <ul style="list-style-type: none"> - 비활성화 -알람 <ul style="list-style-type: none"> - 알람 삭제 - 비활성화 - 알람 설정 -타이머 <ul style="list-style-type: none"> - 타이머 시작 - 타이머 취소 -스톱위치

	<ul style="list-style-type: none"> - 초기화 - 비활성화 -세계시간 <ul style="list-style-type: none"> - hold/release -테마 <ul style="list-style-type: none"> - hold/release
Pre-Conditions	N/A
Post-Conditions	현재 모드에서, C버튼에 매핑 된 기능이 호출된다.

Name	Press D Button
Responsibilities	사용자가 D 버튼을 누른다.
Type	GUI
Cross Reference	R1.6, R1.2
Notes	<ul style="list-style-type: none"> - 메인화면 <ul style="list-style-type: none"> - ModeConfig - 메인화면 제외 모두 <ul style="list-style-type: none"> - BackToMainScreen
Pre-Conditions	N/A
Post-Conditions	현재 모드에서, D버튼에 매핑 된 기능이 호출된다.

Name	getSegmentContentLower()
Responsibilities	하단 세그먼트에 출력되고 있는 내용(8)자리를 얻어온다, 세그먼트가 비활성화 상태가 아니라면 출력은 항상 8자리여야 한다.
Type	GUI
Cross References	N/A
Exceptions	N/A
Output	하단 세그먼트에 표시되고 있는 내용이며, 8자리의 문자열을 반환한다. 만약, 하단 세그먼트가 비활성화 상태라면, null을 반환한다.
Pre-Conditions	N/A
Post-Conditions	N/A

Name	getSegmentContentUpper()
Responsibilities	상단 세그먼트에 출력되고 있는 내용(12)자리를 얻어온다, 세그먼트가 비활성화 상태가 아

	나라면 출력은 항상 12자리여야 한다.
Type	GUI
Cross References	N/A
Exceptions	N/A
Output	상단 세그먼트에 표시되고 있는 내용이며, 12 자리의 문자열을 반환한다. 만약, 상단 세그먼트가 비활성화 상태라면, null 을 반환한다.
Pre-Conditions	N/A
Post-Conditions	N/A

Name	getTextColor()
Responsibilities	세그먼트에 표시중인 글자의 색상을 얻어온다. 반환 값은 절대로 null일 수 없다.
Type	GUI
Cross References	N/A
Exceptions	N/A
Output	글자의 색상
Pre-Conditions	N/A
Post-Conditions	N/A

Name	getBackgroundColor()
Responsibilities	세그먼트에 표시중인 배경의 색상을 얻어온다. 반환 값은 절대로 null일 수 없다.
Type	GUI
Cross References	N/A
Exceptions	N/A
Output	배경의 색상
Pre-Conditions	N/A
Post-Conditions	N/A

Name	initWatch()
Responsibilities	시계의 상태를 초기화 시킨다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A

Pre-Conditions	N/A
Post-Conditions	N/A

8. ModeManager

Name	initWatch()
Responsibilities	각 모드들을 생성하고 초기화한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	시계에 설정된 모드들이 생성되고, 각 모드들이 초기화된다

Name	requestButtonA()
Responsibilities	현재 사용중인 모드의 버튼 A에 매핑 된 기능을 호출한다. 강제된 동작이 있다면, 무조건 해당 동작을 실행하고, 사용중인 모드가 없다면, 기본 동작인, decreaseModelIndex()로 동작한다.
Type	System
Cross References	R7.6, R7.7, R7.3.1, R6.2, R1.4, R5.1, R8.2, R9.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	버튼 A에 매핑 된 기능을 호출한다.

Name	requestButtonB()
Responsibilities	현재 사용중인 모드의 버튼 B에 매핑 된 기능을 호출한다. 강제된 동작이 있다면, 무조건 해당 동작을 실행하고, 사용중인 모드가 없다면, 기본 동작인, increaseModelIndex()로 동작한다.
Type	System
Cross References	R1.3, R7.1, R7.3.3, R6.3, R6.4, R6.5, R5.2, R5.3, R9.1, R8.1
Exceptions	N/A
Output	N/A

Pre-Conditions	N/A
Post-Conditions	버튼 B에 매핑 된 기능을 호출한다.

Name	requestButtonC()
Responsibilities	현재 사용중인 모드의 버튼 C에 매핑 된 기능을 호출한다. 강제된 동작이 있다면, 무조건 해당 동작을 실행하고, 사용중인 모드가 없다면, 기본 동작인, UseCurrentMode()로 동작한다.
Type	System
Cross References	R1.5, R7.2, R7.4, R6.1, R6.6, R5.4, R8.3, R8.4
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	버튼 C에 매핑 된 기능을 호출한다.

Name	OnButtonD()
Responsibilities	버튼 D는 시스템 전용 버튼으로, 사용중인 모드가 없다면, useConfigMode, 아니라면, mainScreen상태로 되돌리는 기능을 한다. 강제된 기능이 있다면, 이 기능들 보다 높은 우선순위를 부여하여, 해당 기능이 먼저 실행된다.
Type	System
Cross References	R1.6, R1.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	버튼 D에 매핑 된 기능을 호출한다.

Name	registerCallback()
Responsibilities	ModeManagerInteracter에 이 클래스의 기능 중, 다른 클래스에서 사용해야 하는 기능들을 CallBack으로 전달한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A

Pre-Conditions	N/A
Post-Conditions	ModeManagerInteracter에 이 클래스의 기능들이 등록된다.

Name	increaseModelIndex()
Responsibilities	mode_index를 1만큼 증가 시키고, 3보다 크다면, 0으로 설정한다. 이후, mode_index에 해당하는 모드를 화면에 출력한다.
Type	System
Cross References	R1.3
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	모드의 인덱스가 증가하고, 해당 모드로 화면이 갱신된다.

Name	decreaseModelIndex()
Responsibilities	mode_index를 1만큼 감소 시키고, 0보다 작다면, 3으로 설정한다. 이후, mode_index에 해당하는 모드를 화면에 출력한다.
Type	System
Cross References	R1.4
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	모드의 인덱스가 감소하고, 해당 모드로 화면이 갱신된다.

Name	useConfigMode()
Responsibilities	현재 사용중인 모드를 이 모드로 설정한다.
Type	System
Cross References	R1.2
Exceptions	N/A
Output	N/A
Pre-Conditions	ls_main_screen이 true여야 한다.

Post-Conditions	현재 사용중인 모드가 이 모드로 변경된다.
-----------------	-------------------------

Name	useCurrentMode()
Responsibilities	현재 인덱스에 해당하는 모드를 사용중인 모드로 설정한다.
Type	System
Cross References	R1.5
Exceptions	N/A
Output	N/A
Pre-Conditions	is_main_screen이 true여야 한다.
Post-Conditions	현재 인덱스에 해당하는 모드가 적용된다.

Name	useThisMode()
Responsibilities	인자로 들어온 모드를 사용할 모드로 설정해 준다. 인자로 들어온 모드를 초기화 시키고, is_main_screen을 false로 설정한다.
Type	System
Cross References	R1.2, R1.5, R1.6
Exceptions	N/A
Output	N/A
Pre-Conditions	is_main_screen이 true여야 한다.
Post-Conditions	인자로 들어온 모드로, 사용중인 모드가 변경된다.

Name	reserveForcedAction()
Responsibilities	모든 버튼의 기능을 강제로 인자로 들어온 버튼의 기능으로 매핑시킨다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	모든 버튼의 기능이 인자로 들어온 기능으로 매핑된다.

Name	cancelForcedAction()
Responsibilities	모든 버튼에 기능 매핑이 강제되었다면, 강제된 매핑을 해제한다.
Type	System
Cross References	R2.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	모든 버튼에 기능 매핑이 강제되었다면, 강제된 매핑이 해제된다.

9. ModeManagerInteracter

Name	registerReserveForcedAction()
Responsibilities	ModeManager가 자신의 기능 중, reserveForcedAction액션을 등록할 수 있게 한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	ModeManager의 reserveForcedAction()이 등록된다.

Name	registerCancelForcedAction()
Responsibilities	ModeManager가 자신의 기능 중, cancelForcedAction액션을 등록할 수 있게 한다.
Type	System
Cross References	R1.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	ModeManager의 cancelForcedAction()이 등록된다.

Name	activeReserveForcedAction()
Responsibilities	등록된

	reserveForcedAction 액션을 실행한다.
Type	System
Cross References	R2.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	ModeManager의 reserveForcedAction()이 실행된다.

Name	activeCancelForcedAction()
Responsibilities	등록된, cancelForcedAction 액션을 실행한다.
Type	System
Cross References	R2.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	ModeManager의 cancelForcedAction()이 실행된다.

10. Buzzer

Name	OnBuzzer()
Responsibilities	버저가 울리고 있었다면, OffBuzzer를 한 뒤, 버저를 울리고, 모든 버튼의 기능을 일시적으로, OffBuzzer로 설정한다.
Type	System
Cross References	R2.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	버저가 울린다.

Name	OffBuzzer()
Responsibilities	버저를 끄고, 모든 버튼에 매핑된 OffBuzzer를 해제한다.
Type	System

Cross References	R2.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	버저가 꺼진다.

11. Segment

Name	setSegmentUpper()
Responsibilities	상단 세그먼트에 출력되는 내용을 인자를 trimElement한 결과값으로, 변경한다.
Type	System
Cross References	R3.1
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	상단 세그먼트에 출력되는 내용이 변경된다.

Name	setSegmentLower()
Responsibilities	하단 세그먼트에 출력되는 내용을 인자를 trimElement한 결과값으로, 변경한다.
Type	System
Cross References	R3.2
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	하단 세그먼트에 출력되는 내용이 변경된다.

Name	setBackgroundColor()
Responsibilities	인자로 들어온 값으로, 세그먼트 뒤 배경으로 변경한다.
Type	System
Cross References	R9.3
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	세그먼트 뒤 배경이 변경된다.

Name	setTextColor()
Responsibilities	인자로 들어온 값으로, 세그먼트의 글자 색상이 변경된다.
Type	System
Cross References	R9.3
Exceptions	N/A
Output	N/A
Pre-Conditions	N/A
Post-Conditions	세그먼트의 글자 색상이 변경된다.

Name	getTextColor()
Responsibilities	세그먼트의 글자 색을 반환한다.
Type	System
Cross References	N/A
Exceptions	N/A
Output	세그먼트의 글자 색
Pre-Conditions	N/A
Post-Conditions	N/A

Name	getBackgroundColor()
Responsibilities	세그먼트의 배경 색을 반환한다.
Type	System
Cross References	N/A
Exceptions	N/A
Output	세그먼트의 배경 색
Pre-Conditions	N/A
Post-Conditions	N/A

Name	getSegmentUpper()
Responsibilities	세그먼트 상단에 표시되는 내용을 반환한다.
Type	System
Cross References	N/A
Exceptions	N/A
Output	세그먼트 상단에 표시되는 내용(12자리)
Pre-Conditions	N/A

Post-Conditions	N/A
-----------------	-----

Name	getSegmentLower()
Responsibilities	세그먼트 하단에 표시되는 내용을 반환한다.
Type	System
Cross References	N/A
Exceptions	N/A
Output	세그먼트 하단에 표시되는 내용(8자리)
Pre-Conditions	N/A
Post-Conditions	N/A

Name	trimElement()
Responsibilities	들어온 문자열의 길이를, 인자로 들어온 정수 값이 되도록 조정한다.
Type	System
Cross References	R3.1, R3.2
Exceptions	N/A
Output	8 또는 12글자로 조절된 문자열
Pre-Conditions	N/A
Post-Conditions	N/A

12. TimeManager

Name	getCurrentTime()
Responsibilities	현재 시간 및 날짜를 반환한다.
Type	System
Cross References	R4.1, R8.5
Exceptions	N/A
Output	현재 시간과 날짜
Pre-Conditions	N/A
Post-Conditions	N/A

Activity 2055. Write Test Code

1. SystemUseCases

```
@Test
public void initWithTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    assertNotNull(mm.used_modes[0]);
    assertNotNull(mm.used_modes[1]);
    assertNotNull(mm.used_modes[2]);
    assertNotNull(mm.used_modes[3]);
    assertNotNull(mm.unused_modes[0]);
    assertNotNull(mm.unused_modes[1]);
}

@Test
public void modeConfigTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    Mode comp = mm.unused_modes[0];
    mm.useConfigMode();
    mm.requestButtonA();
    assertEquals(comp,mm.used_modes[mm.mode_index]);
    comp = mm.unused_modes[1];
    mm.useConfigMode();
    mm.requestButtonB();
    assertEquals(comp,mm.used_modes[mm.mode_index]);
}
```

```
@Test
public void indexIncreaseTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    mm.mode_index = 0;
    for(int i = 0; i < 4; i ++){
        mm.increaseModeIndex();
    }
    assertEquals( expected: 0,mm.mode_index);
}

@Test
public void indexDecreaseTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    mm.mode_index = 0;
    for(int i = 0; i < 4; i ++){
        mm.decreaseModeIndex();
    }
    assertEquals( expected: 0,mm.mode_index);
}
```

```
@Test
public void useCurrentModeTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    Mode current = mm.used_modes[mm.mode_index];
    mm.useCurrentMode();
    assertEquals(current,mm.current_mode);
}

@Test
public void backToMainTest(){
    Segment segment = new Segment();
    ModeManager mm = new ModeManager(segment);
    mm.initWatch();
    mm.useCurrentMode();
    mm.onButtonD();
    assertEquals( expected: true,mm.is_main_screen);
}
```

```
@Test
public void buzzerMultifunRingTest(){
    Buzzer buzzer = new Buzzer();
    buzzer.OnBuzzer();
    try {
        Thread.sleep( millis: 15000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    buzzer.OnBuzzer();
    try {
        Thread.sleep( millis: 15000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    assertEquals( expected: true,buzzer.is_buzzer_running);
    try {
        Thread.sleep( millis: 16000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    assertEquals( expected: false,buzzer.is_buzzer_running);
}
```

```
@Test
public void segmentLowerTest(){
    Segment segment = new Segment();
    segment.setSegmentLower( content: "short", enable: true);
    assertEquals( expected: 8, segment.getSegmentLower().length());
    segment.setSegmentLower( content: "looooooooooong", enable: true);
    assertEquals( expected: 8, segment.getSegmentLower().length());
}
```

```
@Test
public void segmentUpperTest(){
    Segment segment = new Segment();
    segment.setSegmentUpper( content: "short", enable: true);
    assertEquals( expected: 12, segment.getSegmentUpper().length());
    segment.setSegmentUpper( content: "loooooooooooooooooooooooooong", enable: true);
    assertEquals( expected: 12, segment.getSegmentUpper().length());
}
```

2. TimerMode

```
@Test
public void checkZeroNotStart() {
    timerMode.startTimer();
    assertTrue(timerMode.checkTimerZero());
    assertFalse(timerMode.is_timer_running);
}

@Test
public void startTimer() throws InterruptedException {
    for(int i = 0; i < 3; i++) {
        timerMode.increaseTimerSeconds();
    }
    timerMode.startTimer();
    assertTrue(timerMode.is_timer_running);
    assertTrue(timerMode.is_timer_started);
    Thread.sleep(1000);
    assertEquals( expected: 2, timerMode.total_seconds);
}

@Test
public void increaseTimerMinute() {
    timerMode.increaseTimerMinute();
    assertEquals( expected: 1, timerMode.setted_time.getMinute());
}

@Test
public void maxIncreaseMinute() {
    for(int i = 0; i < 59; i++) {
        timerMode.increaseTimerMinute();
    }
    assertEquals( expected: 59, timerMode.setted_time.getMinute());
    timerMode.increaseTimerMinute();
    assertEquals( expected: 0, timerMode.setted_time.getMinute());
}
```

```
@Test
public void increaseTimerSeconds() {
    timerMode.increaseTimerSeconds();
    assertEquals( expected: 1, timerMode.setted_time.getSeconds());
}

@Test
public void maxIncreaseSeconds() {
    for(int i = 0; i < 59; i++) {
        timerMode.increaseTimerSeconds();
    }
    assertEquals( expected: 59, timerMode.setted_time.getSeconds());
    timerMode.increaseTimerSeconds();
    assertEquals( expected: 0, timerMode.setted_time.getSeconds());
}

@Test
public void pauseTimer() {
    timerMode.increaseTimerSeconds();
    timerMode.increaseTimerSeconds();
    timerMode.startTimer();
    timerMode.pauseTimer();
    assertFalse(timerMode.is_timer_running);
}

@Test
public void continueTimer() {
    timerMode.increaseTimerSeconds();
    timerMode.increaseTimerSeconds();
    timerMode.startTimer();
    timerMode.pauseTimer();
    timerMode.continueTimer();
    assertTrue(timerMode.is_timer_running);
}

@Test
public void cancelTimer() {
    timerMode.increaseTimerSeconds();
    timerMode.increaseTimerSeconds();
    timerMode.startTimer();
    assertTrue(timerMode.is_timer_running);
    timerMode.cancelTimer();
    assertFalse(timerMode.is_timer_running);
}
```

3. StopwatchMode

```
@Test
public void startStopWatch() throws InterruptedException {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.startStopWatch();
    assertTrue(stopWatchMode.is_stop_watch_running);
    assertTrue(stopWatchMode.is_stop_watch_started);
    Thread.sleep(1L * 1010);
    assertEquals( expected: 1, stopWatchMode.elapsed_time.getSeconds());
}
}
```

```
@Test
public void maxStopWatch() throws InterruptedException {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.elapsed_time.setMinute(59);
    stopWatchMode.elapsed_time.setSeconds(58);
    stopWatchMode.startStopWatch();
    Thread.sleep(1L * 1010);
    assertTrue(stopWatchMode.testStopWatchMax());
    assertFalse(stopWatchMode.is_stop_watch_running);
}
}
```

```
@Test
public void pauseStopWatch() throws InterruptedException {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.startStopWatch();
    assertTrue(stopWatchMode.is_stop_watch_running);
    stopWatchMode.pauseStopWatch();
    assertFalse(stopWatchMode.is_stop_watch_running);
}
}
```

```
@Test
public void continueStopWatch() {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.startStopWatch();
    stopWatchMode.pauseStopWatch();
    stopWatchMode.continueStopWatch();
    assertTrue(stopWatchMode.is_stop_watch_running);
}
}
```

```
@Test
public void resetTimer() throws InterruptedException {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.startStopWatch();
    stopWatchMode.pauseStopWatch();
    stopWatchMode.resetStopWatch();
    assertFalse(stopWatchMode.is_stop_watch_running);
    assertEquals( expected: 0, stopWatchMode.elapsed_time.getMinute());
    assertEquals( expected: 0, stopWatchMode.elapsed_time.getSeconds());
}
}
```

```
@Test
public void isResetRun() {
    StopwatchMode stopWatchMode = new StopwatchMode(segment);
    stopWatchMode.initStopWatchMode();
    stopWatchMode.startStopWatch();
    stopWatchMode.OnButtonC(); // reset에 해당하는 버튼
    assertTrue(stopWatchMode.is_stop_watch_running);
}
}
```

4. AlarmMode

```
@Test
public void initAlarmMode() {
    alarm.initAlarmMode();

    for(int i=0;i<4;i++){
        assertEquals( expected: 0,alarm.alarms[i].getHour());
        assertEquals( expected: 0,alarm.alarms[i].getMinute());
        assertEquals( expected: -1,alarm.alarms[i].getActive());
    }
}

@Test
public void enableAlarm() {
    alarm.initAlarmMode();
    alarm.enableAlarm();
    assertEquals( expected: 1, alarm.alarms[alarm.alarm_index].active);
}

@Test
public void disableAlarm() {
    alarm.initAlarmMode();
    alarm.disableAlarm();
    assertEquals( expected: 0, alarm.alarms[alarm.alarm_index].active);
}

@Test
public void checkRingAlarmExist() {
    alarm.initAlarmMode();
    assertEquals( expected: false, alarm.checkRingAlarmExist());
}

@Test
public void increaseAlarmindex() {
    alarm.increaseAlarmindex();
    assertEquals( expected: 1, alarm.alarm_index);
    alarm.increaseAlarmindex();
    assertEquals( expected: 2, alarm.alarm_index);
}
```

```
@Test
public void increaseOneHour() {
    alarm.initAlarmMode();
    alarm.alarm_index = 0;
    alarm.increaseOneHour();
    assertEquals( expected: 1,alarm.alarms[0].hour);
    alarm.alarms[0].hour = 23;
    alarm.increaseOneHour();
    assertEquals( expected: 0,alarm.alarms[0].hour);
}

@Test
public void increasefiveMinute() {
    alarm.initAlarmMode();
    alarm.alarm_index = 0;
    alarm.increasefiveMinute();
    assertEquals( expected: 5,alarm.alarms[0].minute);
    alarm.alarms[0].minute = 55;
    alarm.increasefiveMinute();
    assertEquals( expected: 0,alarm.alarms[0].minute);
}

@Test
public void alarmInitTime() {
    alarm.initAlarmMode();
    alarm.alarmInitTime();
    assertEquals( expected: 0, alarm.alarms[alarm.alarm_index].hour);
    assertEquals( expected: 0, alarm.alarms[alarm.alarm_index].minute);
    assertEquals( expected: 0, alarm.alarms[alarm.alarm_index].active);
}
```

```

@Test
public void nextAlarm() {
    alarm.initAlarmMode();
    alarm.nextAlarm();
    assertEquals( expected: 1, alarm.alarm_index);

    alarm.nextAlarm();
    assertEquals( expected: 2, alarm.alarm_index);

    alarm.nextAlarm();
    assertEquals( expected: 3, alarm.alarm_index);

    alarm.nextAlarm();
    assertEquals( expected: 0, alarm.alarm_index);
}

@Test
public void decideAlarm() {
    alarm.initAlarmMode();
    alarm.decideAlarm();
    assertEquals( expected: 1, alarm.alarms[alarm.alarm_index].activated);
}

@Test
public void deleteAlarm() {
    alarm.initAlarmMode();
    alarm.deleteAlarm();
    assertEquals( expected: -1, alarm.alarms[alarm.alarm_index].activated);
}

```

5. WorldTimeMode

```

@Test
public void initWorldTimeMode() {
    // initWorldTimeMode() 실행 후, 각 인덱스에 도시 별 정보가 알맞게 들어갔는지 테스트
    testMode.initWorldTimeMode();
    assertEquals( expected: -17, testMode.worlds[0].weight.getHour());
    assertEquals( expected: "WASHINGTON", testMode.worlds[0].name);
}

```

```

@Test
public void increaseWorldTimeIndex() {
    // 0 -> 1
    testMode.world_index = 0;
    testMode.increaseWorldTimeIndex();
    assertEquals( expected: 1, testMode.world_index);

    //마지막 인덱스(19) -> 처음 인덱스(0)
    testMode.world_index = testMode.worlds.length - 1;
    testMode.increaseWorldTimeIndex();
    assertEquals( expected: 0, testMode.world_index);
}

```

```

@Test
public void decreaseWorldTimeIndex() {
    // 1 -> 0
    testMode.world_index = 1;
    testMode.decreaseWorldTimeIndex();
    assertEquals( expected: 0, testMode.world_index);

    //처음 인덱스(0) -> 마지막 인덱스(19)
    testMode.world_index = 0;
    testMode.decreaseWorldTimeIndex();
    assertEquals( expected: testMode.worlds.length - 1, testMode.world_index);
}

```

```

@Test
public void holdCurrentWorldTime() {
    testMode.locked = false;
    testMode.holdCurrentWorldTime();
    assertEquals( expected: true, testMode.locked);
}

```

```

@Test
public void releaseCurrentWorldTimeRock() {
    testMode.locked = true;
    testMode.releaseCurrentWorldTimeRock();
    assertEquals( expected: false, testMode.locked);
}

```

6. ThemeMode

```
@Test
public void initThemeMode() {
    // initThemeMode() 실행 후, 각 인덱스에 테마 정보가 알맞게 들어갔는지 테스트
    testTheme.initThemeMode();
    assertEquals(Color.WHITE, testTheme.themes[0].getText());
    assertEquals(Color.BLACK, testTheme.themes[0].getBackground());
    assertEquals( expected: "WINTER1", testTheme.themes[0].getName());
}
}
```

```
@Test
public void nextTheme() {
    testTheme.initThemeMode();

    // 인덱스 0 -> 1
    testTheme.theme_index = 0;
    testTheme.nextTheme();
    assertEquals( expected: 1, testTheme.theme_index);

    // 마지막 인덱스(7) -> 처음 인덱스(0)
    testTheme.theme_index = testTheme.themes.length - 1; //마지막 인덱스(7)
    testTheme.nextTheme();
    assertEquals( expected: 0, testTheme.theme_index);
}
}
```

```
@Test
public void prevTheme() {
    testTheme.initThemeMode();

    // 인덱스 1 -> 0
    testTheme.theme_index = 1;
    testTheme.prevTheme();
    assertEquals( expected: 0, testTheme.theme_index);

    // 처음 인덱스(0) -> 마지막 인덱스(7)
    testTheme.theme_index = 0; //마지막 인덱스(7)
    testTheme.prevTheme();
    assertEquals( expected: testTheme.themes.length-1, testTheme.theme_index);
}
}
```

Activity 2061. Unit Testing

✓ SystemTest	46 s 47 ms
✓ backToMainTest	40 ms
✓ indexIncreaseTest	1 ms
✓ buzzerMultifurRingTest	46 s 3 ms
✓ modeConfigTest	2 ms
✓ indexDecreaseTest	0 ms
✓ useCurrentModeTest	0 ms
✓ segmentLowerTest	0 ms
✓ initWatchTest	1 ms
✓ segmentUpperTest	0 ms

✓ AlarmModeTest	40 ms
✓ increaseOneH	16 ms
✓ checkRingAlar	21 ms
✓ increasefiveMir	0 ms
✓ enableAlarm	1 ms
✓ deleteAlarm	0 ms
✓ disableAlarm	1 ms
✓ increasealarm_	0 ms
✓ decideAlarm	1 ms
✓ alarmInitTime	0 ms
✓ initAlarmMode	0 ms
✓ nextAlarm	0 ms
✓ TimerModeTest	1 s 12 ms
✓ continueTimer	8 ms
✓ startTimer	1 s 1 ms
✓ cancelTimer	3 ms
✓ maxIncreaseMinut	0 ms
✓ increaseTimerMini	0 ms
✓ pauseTimer	0 ms
✓ maxIncreaseSecon	0 ms
✓ increaseTimerSeco	0 ms
✓ checkZeroNotStar	0 ms
✓ StopWatchMo	2 s 154 ms
✓ resetTimer	130 ms
✓ isResetRun	1 ms
✓ pauseStopWatch	0 ms
✓ maxStopWa	1 s 12 ms
✓ startStopWa	1 s 11 ms
✓ continueStopWa	0 ms

✓ WorldTimeModeTest
✓ decreaseWorldTimeIndex
✓ increaseWorldTimeIndex
✓ holdCurrentWorldTime
✓ releaseCurrentWorldTimeRock
✓ initWorldTimeMode

✓ ThemeModeTest
✓ prevTheme
✓ nextTheme
✓ initThemeMode

Activity 2063. System Testing

	Test항목	Discription	Use Case	System Function	Pass
1	initWatchTest	시계가 초기화 되고, 모드 6개가 잘 생성 되는지, Test	1. InitWatchState	1.1	P
2	indexIncreaseTest	increaseIndex에서, 인덱스가 3일 때 0에 도달하는지 Test	3. NextMode	1.3	P
3	indexDecreaseTest	decreaseIndex에서, 인덱스가 0일 때 3에 도달하는지 Test	4. PrevMode	1.4	P
4	useCurrentModeTest	useCurrentMode동작 후, 모드가 타겟 모드로 잘 변경되었는가 Test	5. UseCurrentMode	1/5	P
5	backToMainTest	backToMainScreen동작 후 모드가 알맞게 변했는지 Test	6. BackToMainScreen	1.6	P
6	buzzerMultifulRingTest	OnBuzzer가 0초, 15초에 불렸을 때, 버저가 약 45초에 근사하게 울리는지 Test	7. OnBuzzer 8. OffBuzzer	2.1	P
7	segmentLowerTest	이상한 길이의 입력이 들어와도 잘 처리하는지 Test	10. SetSegmentLower	2.2	P
8	segmentUpperTest	이상한 길이의 입력이 들어와도 잘 처리하는지 Test	9. SetSegmentUpper	3.1	P
9-1	startTimer	Timer를 실행했을 때, Timer가 지정된 시간으로부터 주기적으로 1초씩 Discount되는지 test	16. Start Timer	6.1	P
9-2	checkZeroTimerNotStart	Timer 설정시간이 00 : 00 일 땐 작동하지 않는지 test	16. Start Timer	6.1	P
10-1	increaseTimerMinute	Timer에서 버튼을 눌렀을 때 Timer의 설정시간이 1	17. SetTimerMinute	6.2	P

		분씩 증가하는 지 test			
10-2	maxIncreaseMinute	59분 되었을 때, 한 번 더 누르게 되면 0분으로 바뀌는 지 test	17. SetTimerMinute	6.2	P
11-1	increaseTimerSeconds	Timer에서 버튼을 눌렀을 때 Timer의 설정시간이 1 초씩 증가하는 지 test	18. SetTimerSeconds	6.3	P
11-2	maxIncreaseSeconds	59초 되었을 때, 한 번 더 누르게 되면 0초으로 바뀌는 지 test	18. SetTimerSeconds	6.3	P
12	pauseTimer	Timer에서 버튼을 눌렀을 때 Timer의 기능이 일시 정지 되는 지 test	19. Pause Timer	6.4	P
13	continueTimer	Timer에서 버튼을 눌렀을 때 Timer의 기능이 다시 실행되는 지 test	20. Continue Timer	6.5	P
14	cancelTimer	Timer에서 버튼을 눌렀을 때 Timer의 기능이 취소 되는 지 test	21. Cancel Timer	6.6	P
15	InitAlarmMode	알람 리스트 생성시 4개가 만들어 졌는지 확인	1.InitWatchState	1.1	P
16	enableAlarm	Alarm이 활성화 되었는지 확인	30.EnableAlarm	7.6	P
17	disableAlarm	Alarm이 비활성화 되었는지 확인	31.DisableAlarm	7.7	P
18	checkRingAlarmExist	설정된 알람 시간에 알람이 울리는지 확인	29.RingAlarm	7.5	P
19	Increasealarm_index	몇 번째 알람인지 확인하는 index 값이 증가하는 지 확인	23.NextAlarm	7.1	P
20	AddAlarm	mappingAlarmTimeSettingMode 함수가 호출되는 지 확인	25.AddAlarm	7.3.1	P
21	increaseOneHour	알람 설정 모드에서 시간이 1 증가하는지 확인/23	26.SetAlarmHour	7.3.2	P

		시에서 호출 시 00시로 출력되는지 확인			
22	increasefiveMinute	알람 설정 모드에서 분이 5 증가하는지 확인/55분에서 호출 시 0분으로 출력되는지 확인	27.SetAlarmMinute	7.3.3	P
23	alarmInitTime	시간, 분 모두 0으로 바뀌는지 확인	25.AddAlarm	7.3.1	P
24	initWorldTimeMode	스레드가 잘 돌아가는지/ 각 도시에 대한 정보가 배열에 잘 들어갔는지 Test	1.initWatchState	1.1	P
25	increaseWorldTimeIndex	인덱스 값이 범위 내에 잘 들어가 있는지/ 인덱스가 잘 증가하는지/ 처음 인덱스일 때 마지막 인덱스로 잘 순환하는지	32. NextWorldTime	8.1	P
26	decreaseWorldTimeIndex	인덱스 값이 범위 내에 잘 들어가 있는지/ 인덱스가 잘 감소하는지/ 마지막 인덱스일 때 처음 인덱스로 잘 순환하는지	33. PrevWorldTime	8.2	P
27	holdCurrentWorldTime	현재 상태가 true(lock)상태 인지 Test	34. HoldCurrentWorldTime	8.3	P
28	releaseCurrentWorldTimeRock	현재 상태가 false(unlock)/상태 인지 Test	35. ReleaseCurrentWorldTimeRock	9.4	P
29	initThemeMode	각각의 테마가 배열에 잘 들어갔는지 Test	1. InitWatchState	1.1	P
30	nextTheme	인덱스가 잘 증가 하는지 / 마지막 인덱스에서 다음으로 넘어갈 때 처음으로 잘 순환하는지 Test	37. NextTheme	9.1	P
31	prevTheme	인덱스가 잘 감소 하는지 /처음 인덱스에서 이전으로 넘어갈 때 마지막으로 잘 순환하는지 Test	38. PrevTheme	9.2	P

Activity 2067. Traceability Analysis

System	pressButtonA
	pressButtonB
	pressButtonC
	pressButtonD
	getSegmentContentLower
	getSegmentContentUpper
	getTextColor
	getBackgroundColor
ModeManager	InitWatch
	requestButtonA
	requestButtonB
	requestButtonC
	OnButtonD
	registerCallback
	increaseModeIndex
	decreaseModeIndex
	useConfigMode
	useCurrentMode
	useThisMode
	reserveForcedAction
cancelForcedAction	
ModeManagerInteracter	registerReserveForcedAction
	registerCancelForcedAction
	activeReserveForcedAction
	activeCancelForcedAction
Buzzer	OnBuzzer
	OffBuzzer
Segment	setSegmentUpper
	setSegmentLower
	setBackgroundColor
	setTextColor
	getTextColor
	getBackgroundColor
	getSegmentUpper
	getSegmentLower
trimElement	
TimeManager	getCurrentTime
WorldTimeMode	initWorldTimeMode
	nextTimeMode
	prevTimeMode
	increaseWorldTimeIndex
	decreaseWorldTimeIndex
	holdCurrentTimeMode
	releaseCurrentWorldTimeLock
	syncWorldTime
ThemeMode	inirThemeMode
	nextTheme
	prevTheme
	decideTheme
	syncUIWithTheme
TimerMode	inirTimerMode
	startTimer
	pauseTimer
	continueTimer
	cancelTimer
	increaseTimerMinute
	increaseTimerSeconds
	checkTimerZero
	checkTimerEnded

	decreaseTimer
	mappingTimerState
	mappingTimerRunning
	mappingTimerPause
	displayTimer
StopWatchMode	initStopWatchMode
	startStopWatch
	pauseStopWatch
	continueStopWatch
	increaseStopWatchSeconds
	testStopWatchMax
	resetStopWatch
	mappingStopWatchState
	mappingStopWatchRunning
	mappingStopWatchPause
	displayStopWatchState
CurrentTimeMode	initCurrentTime
	SyncWithCurrentTime
AlarmMode	initAlarmMode
	enableAlarm
	disableAlarm
	checkRingAlarmExist
	increaseAlarmIndex
	addAlarm
	increaseOneHour
	increaseFiveMinute
	alarmInitTime
	mappingAlarmSettingMode
	mappingAlarmMode
	nextAlarm
	decideAlarm
	deleteAlarm
displayCurrentAlarm	

SystemTest
initWatchTest
indexIncreaseTest
indexDecreaseTest
useCurrentModeTest
backToMainTest
buzzerMultifunctionTest
segmentLowerTest
segmentUpperTest
startTimer

checkZeroTimer
NotStart
increaseTimerMinute
maxIncreaseMinute
increaseTimerSeconds
maxIncreaseSeconds
pauseTimer
continueTimer
cancelTimer
InitAlarmMode
enableAlarm
disableAlarm
checkRingAlarmExist
Increasealarm_index
AddAlarm
increaseOneHour
increasefiveMinute
alarmInitTime
initWorldTimeMode
increaseWorldTimeIndex
decreaseWorldTimeIndex
holdCurrentWorldTime
releaseCurrentWorldTimeRock
initThemeMode
nextTheme
prevTheme

