



A Remind Watch

Software Modeling & Analysis

Team 3

201511251 김우진
201511263 박종엽
201511279 이상현
201511283 이정노

INDEX

목차

CONTENTS 1

**Implement Class &
Method Definitions**

CONTENTS 2

**Implement
Windows**

CONTENTS 3

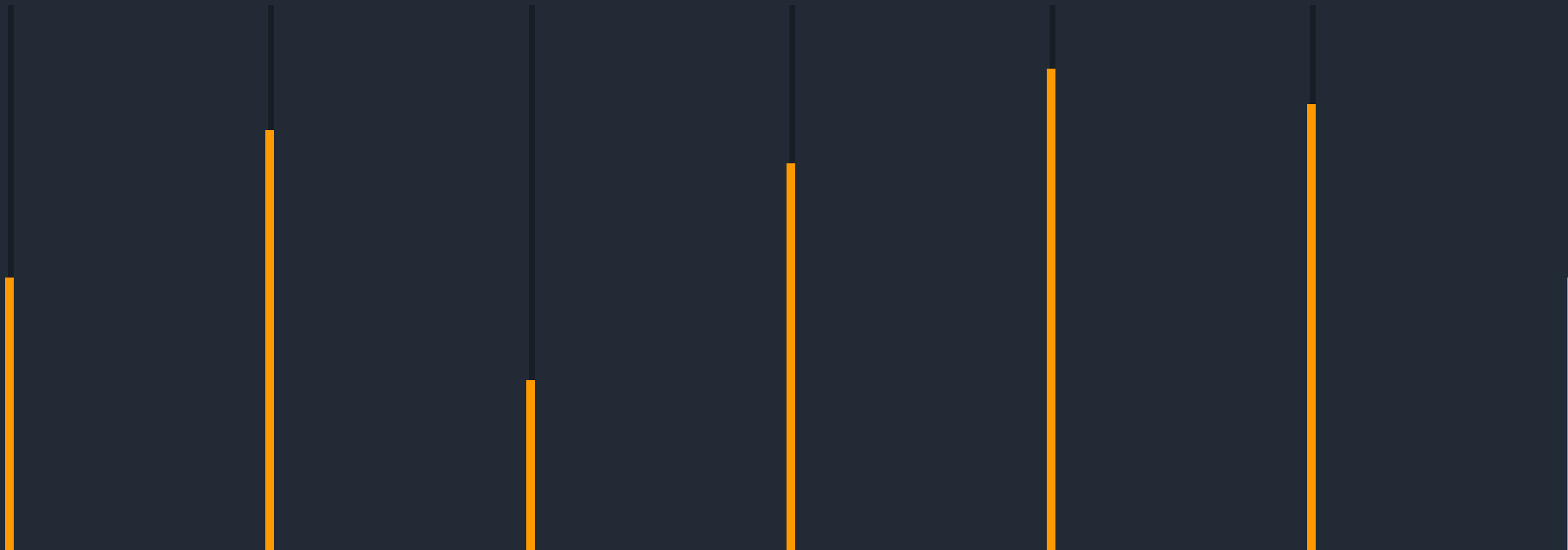
Write Unit Test Code

CONTENTS 4

Unit Testing

CONTENTS 5

System Testing



1. Implement Class & Method Definitions



Type	Method
Name	<u>moveCursor</u>
Purpose	현재 커서를 이동시키는 함수이다.
<u>CrossReference</u>	
Input	Void
Output	Void
Abstract operation	1. 커서를 이동시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	<u>increaseValue</u>
Purpose	현재 커서를 변경시킨다..
<u>CrossReference</u>	
Input	Void
Output	Void
Abstract operation	1. 커서 값을 변경하는 버튼을 누른다. 2. 현재 커서가 어떤지에 따라 값을 증가, 셋, 0 <u>으로</u> 바꿔주는 함수를 부른다.
Exceptional Courses of Events	N/A

1. Implement Class & Method Definitions



```
void moveCursor(){
    if(currentCursor+1 < cursorList.size()){
        this.currentCursor += 1;
    }
    else{
        this.currentCursor = 0;
    }
}

void increaseValue(){
    Pair<String, String> current = cursorList.get(currentCursor);
    Integer prev = data.get(current.left);
    if (prev == null) {
        prev = 0;
    }
    switch (current.right) {
        case "inc": {
            this.data.put(current.left, this.increaseElement(current.left, prev));
            break;
        }
        case "set": {
            this.data.put(current.left, this.setUnsetValue(current.left, prev));
            break;
        }
        case "zero":
            this.data.put(current.left, this.setZero(current.left, prev));
            break;
        default:
            System.out.println("[ERROR] not implemented function");
            break;
    }
}
```

1. Implement Class & Method Definitions



Type	Method
Name	<u>setZero</u>
Purpose	현재 커서의 값을 0으로 바꾸는 함수이다.
<u>CrossReference</u>	
Input	String, int
Output	Int
Abstract operation	1. 현재 커서의 값을 0으로 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	<u>SetUnsetValue</u>
Purpose	현재 커서의 값을 <u>on,off</u> 시키는 함수이다.
<u>CrossReference</u>	
Input	String, int
Output	Int
Abstract operation	1. 현재 커서의 값을 on, off로 바꿔준다.
Exceptional Courses of Events	N/A

Type	Method
Name	<u>increaseElement</u>
Purpose	현재 커서의 값을 증가시키는 함수이다.
<u>CrossReference</u>	
Input	String, int
Output	Int
Abstract operation	1. 현재 커서의 값을 1 증가 시킨다.
Exceptional Courses of Events	N/A

1. Implement Class & Method Definitions



```
private int setZero(String name, int prev) { return 0; }  
private int setUnsetValue(String name, int prev) { return prev ^ 1; }  
protected int increaseElement(String name, int prev) { return prev + 1; }
```

Class Alarm – increaseElement()

```
protected int increaseElement(String name, int prev) {  
    if (name.equals("minutes") && prev == 59) {  
        return 0;  
    } else if (name.equals("hours") && prev == 23) {  
        return 0;  
    }  
    return prev + 1;  
}
```

1. Implement Class & Method Definitions



Type	Method
Name	<u>checkFunctionCount</u>
Purpose	선택된 함수가 4개인지 확인하기 위한 함수이다.
<u>CrossReference</u>	
Input	void
Output	Boolean
Abstract operation	1. 선택된 함수가 4 개이면 true 를 반환 그 외의 경우 false 를 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	<u>saveSelectFunctionData</u>
Purpose	선택된 함수의 데이터를 데이터 객체에 저장한다.
<u>CrossReference</u>	
Input	void
Output	Boolean
Abstract operation	1. 선택된 함수의 데이터를 데이터 객체에 저장한다.
Exceptional Courses of Events	N/A

1. Implement Class & Method Definitions



```
boolean checkFunctionCount() {
    int count = 0;
    for (Pair<String, String> p : this.cursorList) {
        if( this.data.get(p.left) == 1 ) {
            count++;
        }
    }
    return count == 3;
}

boolean saveSelectFunctionData() {
    dataKeep.saveData(data, name: "SELECTFUNCTION", num: 0);
    return true;
}
```


1. Implement Class & Method Definitions



Type	Method
Name	<u>saveData</u>
Purpose	데이터를 저장해주는 함수이다.
<u>CrossReference</u>	
Input	HashMap<String, Inter>,String, int
Output	Boolean
Abstract operation	1. 현재 객체의 값을 저장한다.
Exceptional Courses of Events	N/A

1. Implement Class & Method Definitions



```
boolean saveData(HashMap<String, Integer> data, String name, int num){
    if(name.equals("SCHEDULE")){
        schedule_list.add(data);
        return TRUE;
    }
    else if(name.equals("ALARM")){
        alarm_list.get(num).putAll(data);
        return TRUE;
    }
    else if(name.equals("TIMER")){
        timer_data.putAll(data);
        return TRUE;
    }
    else if(name.equals("EXPENDITURE")){
        expenditure_data.putAll(data);
        return TRUE;
    }
    else if(name.equals("SELECTFUNCTION")){
        select_data.putAll(data);
        return TRUE;
    }

    return FALSE;
}
```

1. Implement Class & Method Definitions



Type	Method
Name	<u>loadData</u>
Purpose	데이터를 반환해주는 함수이다.
<u>CrossReference</u>	
Input	String, int
Output	HashMap<String, Integer>
Abstract operation	1. 데이터를 반환해준다.
Exceptional Courses of Events	N/A

1. Implement Class & Method Definitions

```
HashMap<String, Integer> loadData(String name, int num){
    //no exist schedule load
    if(name.equals("SCHEDULE")){
        return schedule_list.get(num);
    }
    else if(name.equals("ALARM")){
        return alarm_list.get(num);
    }
    else if(name.equals("TIMER")){
        return timer_data;
    }
    else if(name.equals("EXPENDITURE")){
        return expenditure_data;
    }
    else if(name.equals("SELECTFUNCTION")){
        return select_data;
    }
    else{
        return trash;
    }
}
```



1. Implement Class & Method Definitions

4) currentTime

Type	Method
Name	<u>pauseTimeFlow</u>
Purpose	현재 흐르고 있는 스레드를 멈추기 위한 함수이다.
<u>CrossReference</u>	
Input	void
Output	Void
Abstract operation	1. 현재 시간이 흐르는 스레드를 정지시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	<u>contTimeFlow</u>
Purpose	현재 멈춰있는 스레드를 시작하기 위한 함수이다.
<u>CrossReference</u>	
Input	Void
Output	Void
Abstract operation	1. 현재 멈춰있는 스레드를 돌린다.
Exceptional Courses of Events	N/A

```
void pauseTimeFlow() { this.timeThread.pause(); }  
void contTimeFlow() { this.timeThread.cont(); }
```

1. Implement Class & Method Definitions

9.11

Type	Method
Name	<u>initTimer</u>
Purpose	일시정지, 정지된 타이머를 원래 값으로 초기화한다.
<u>CrossReference</u>	
Input	Void
Output	Void
Abstract operation	<ol style="list-style-type: none">1. 타이머 데이터를 불러온다.2. 불러온 데이터를 보여준다.
Exceptional Courses of Events	N/A

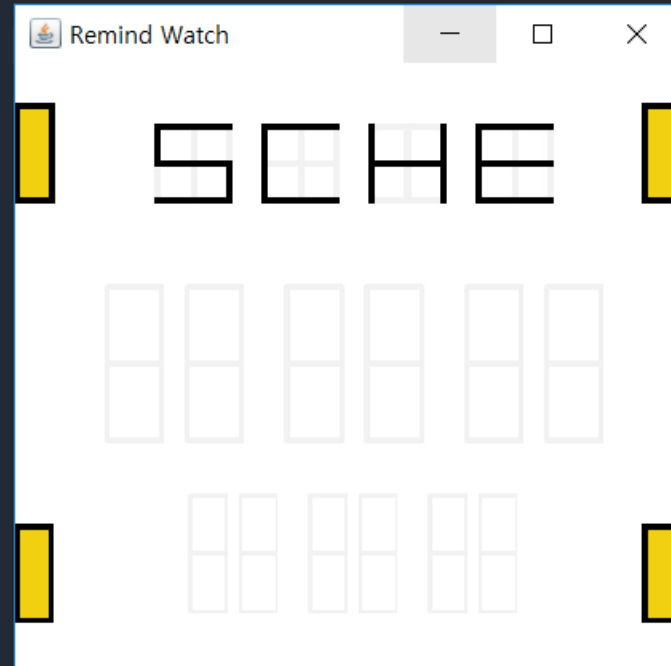
1. Implement Class & Method Definitions



```
public void initTimer(){
    if(!isRunning) {
        this.time.setData(this.data);
        this.targetTimestamp = this.time.getMsTimestamp();
        System.out.println("init stopwatch");
    }
    else{
        System.out.println("Should stopped");
    }
}
```

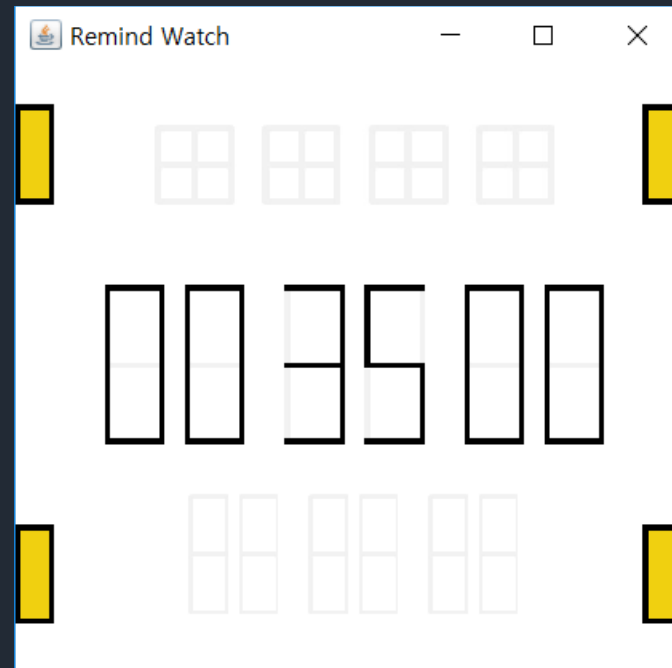
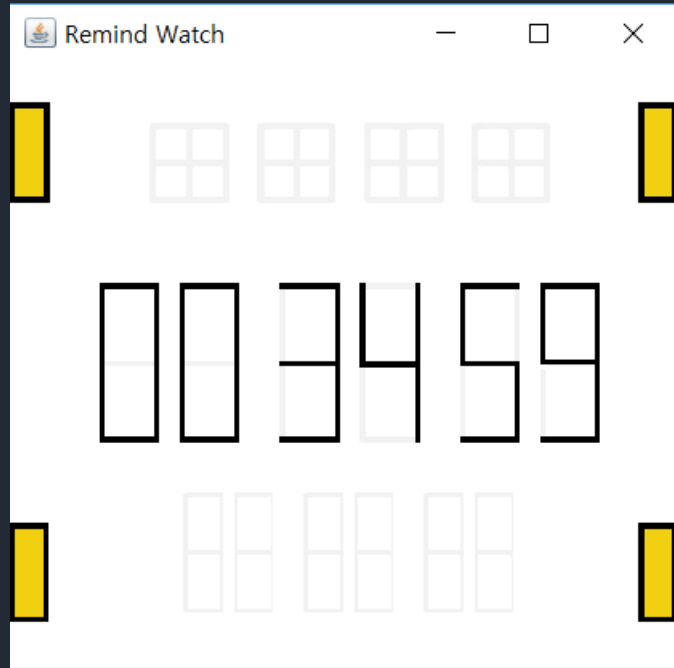
2. Implement Windows

Select Function



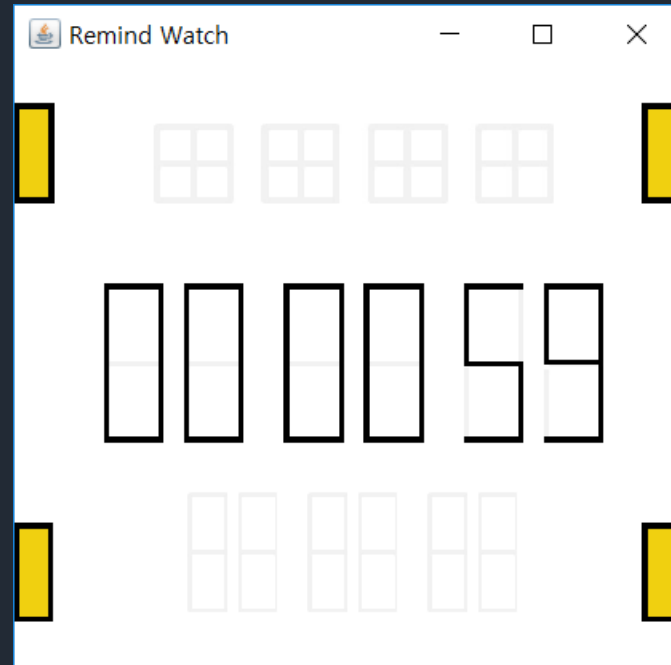
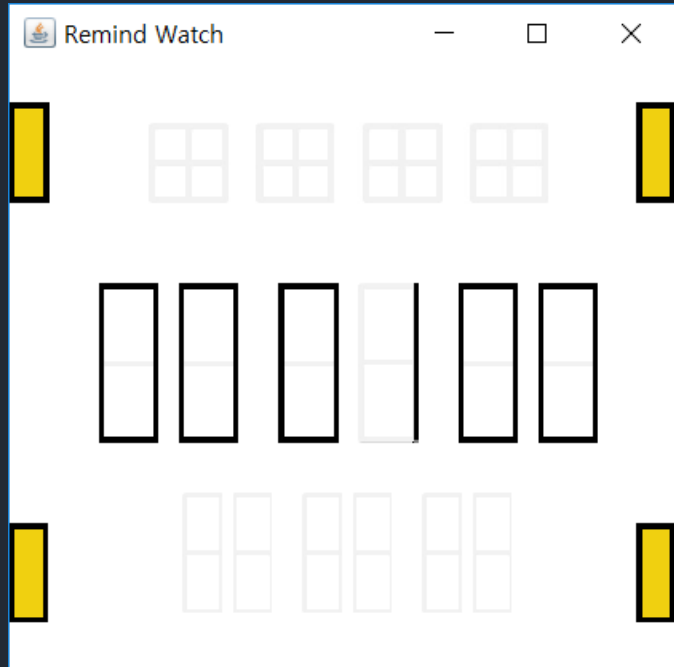
2. Implement Windows

Stopwatch



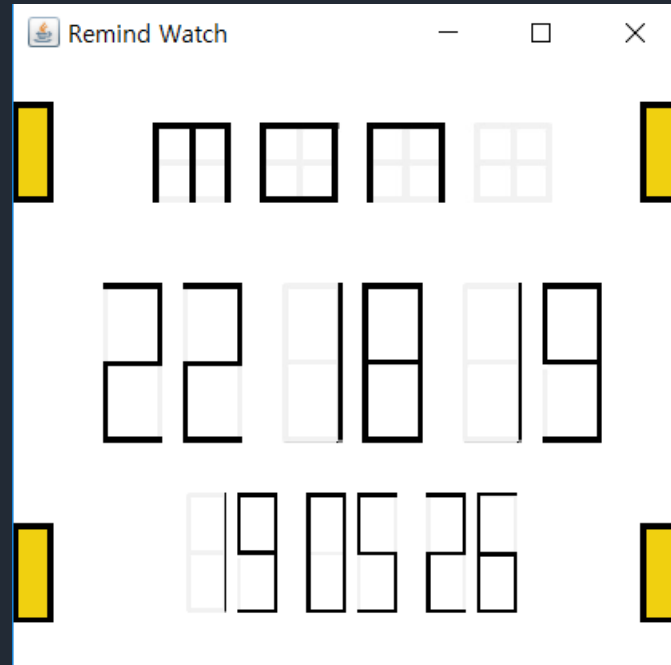
2. Implement Windows

Timer



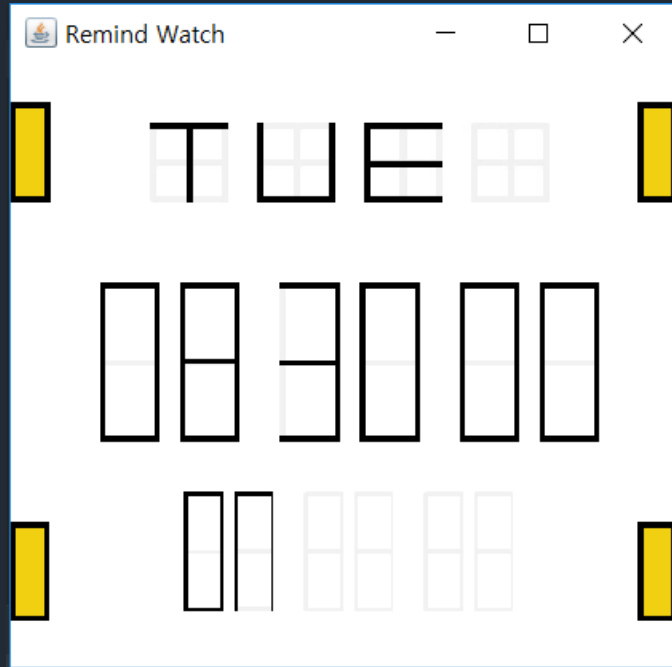
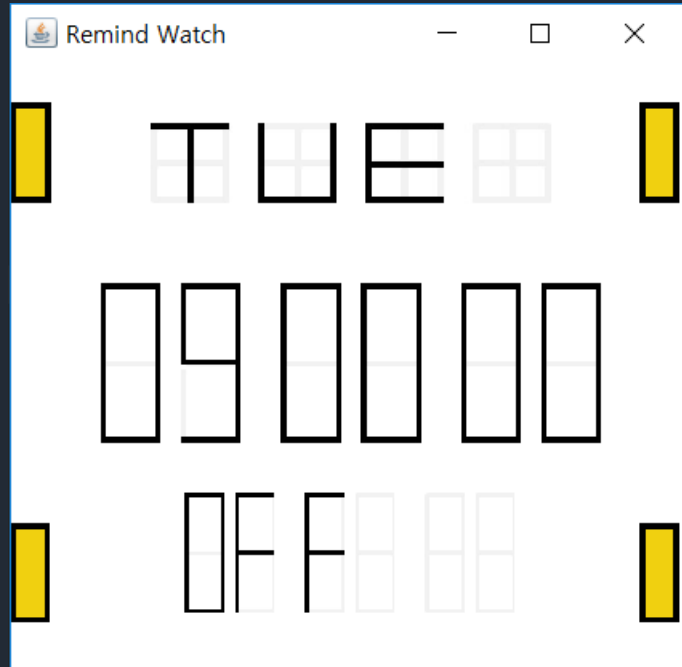
2. Implement Windows

currentTime



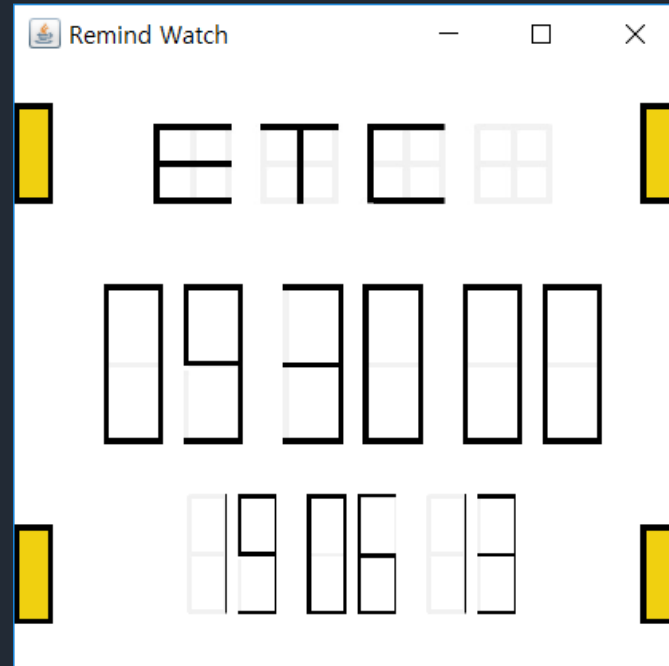
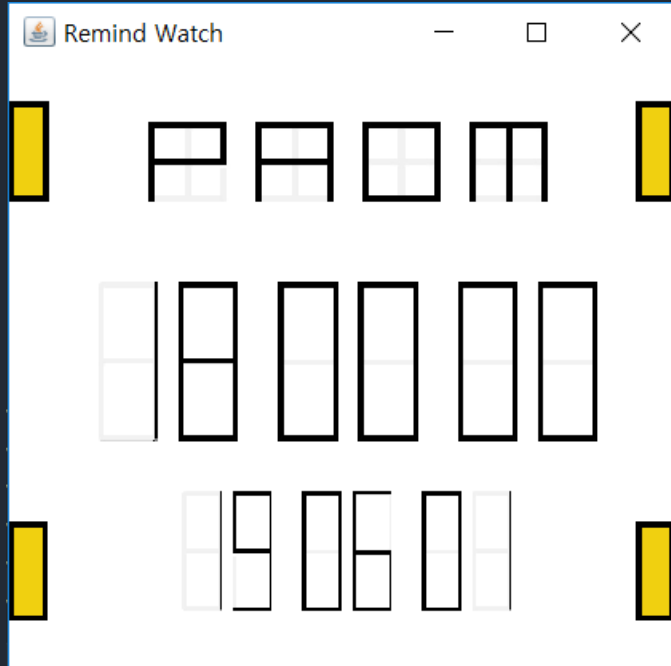
2. Implement Windows

Alarm



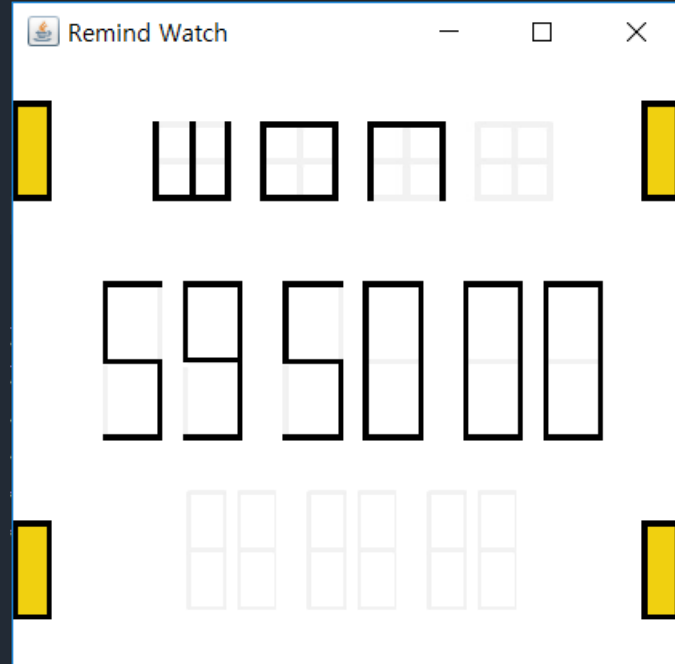
2. Implement Windows

Schedule



2. Implement Windows

Expenditure



3. Write Unit Test Code

```
@Test
public void test_increaseElement_minutes_10_times_0_to_10() {
    Alarm alarm = new Alarm();
    alarm.moveCursor();alarm.moveCursor();
    for(int i = 0; i<10; i++)
        alarm.increaseValue();
    assert(alarm.getData().get("minutes").equals(10));
}
```

3. Write Unit Test Code

```
@Test
public void test_after_deleteAlarm_three_times_getNextAlarmData_and_compare_minutes() {
    Alarm alarm = new Alarm();
    alarm.moveCursor();alarm.moveCursor();
    alarm.increaseValue();

    alarm.deleteAlarm();
    alarm.getNextAlarmData();alarm.getNextAlarmData();alarm.getNextAlarmData();

    assert(alarm.getData().get("minutes") != 1);
}
```


3. Write Unit Test Code

```
@Test
public void test_getNextAlarmData_firstAlarm_minutesValue_is_not_equal_to_secondAlarm() {
    Alarm alarm = new Alarm();
    int a, b;
    a = alarm.getData().get("minutes");
    alarm.getNextAlarmData();
    alarm.moveCursor();
    alarm.moveCursor();
    alarm.increaseValue();
    b = alarm.getData().get("minutes");

    assert(a!=b);
}
```

4. Unit Testing

▼	✘ AlarmTest (Controller)	12 ms
	✘ test_after_deleteAlarm_three_times_c	12 ms
	✓ test_increaseElement_minutes_10_tim	0 ms
	✓ test_getNextAlarmData_firstAlarm_mil	0 ms

▼	✘ ScheduleTest (Controller)	73 ms
	✓ test_increase_Schedule_value_59_plu	60 ms
	✓ test_Schedule_moveCursor	0 ms
	✘ saveScheduleValue	13 ms

▼	✓ ExpenditureTest (Controller)	5 ms
	✓ test_save_and_load_ExpenditureValue	5 ms
	✓ test_increase_Expenditure_Value	0 ms

▼	✓ TimerTest (Controller)	1 s 45 ms
	✓ startTimer	44 ms
	✓ pauseTimer	1 s 1 ms

▼	✓ StopwatchTest (Controller)	1 s 74 ms
	✓ pauseStopwatch	1 s 71 ms
	✓ startStopwatch	3 ms

5. System Testing

TEST NUMBER	REF #	TEST CASE	DESCRIPTION	USE CASE	P/F
1	r 1.1	디스플레이 전환 시험	디스플레이가 전환되는지 test	Switch Display	Fail
2	r 2.1	버튼 기능 전환 시험	각 기능이 변할 때 버튼의 기능이 변하는지 test	Change Button's function	Fail
3	r 3.1	현재 시간 표시 시험	현재 시간이 화면에 표시되는지 test	Current time	Fail
4	r 3.2	현재 시간 변경 시험	현재 시간 수정이 이루어지는지 test	Current time	Fail
5	R 3.3	현재 시간 갱신 시험	현재 시간이 1초 단위로 바뀌는지 test	Current time	Fail
6	r 4.1	타이머 설정 시험	타이머 설정이 이루어지는지 test	Timer	Pass
7	r 4.2	타이머 초기화 시험	타이머가 초기화 되는지 test	Timer	Pass
8	r 4.3	타이머 시간 시작 시험	타이머가 현재 타이머 시간으로부터 시작되는지 test	Timer	Pass
9	r 4.4	타이머 일시정지 시험	타이머가 일시정지 되는지 test	Timer	Pass
10	r 5.1	스톱워치 동작 시험	스톱워치가 현재 스톱워치 시간으로부터 동작되는지 test	stopwatch	Pass
11	r 5.2	스톱워치 일시정지 시험	스톱워치가 일시 정지 되는지 test	Stopwatch	Pass
12	r 5.3	스톱워치 초기화 시험	스톱워치가 초기화가 이루어지는지 test	Stopwatch	Pass
13	R 6.1	알람 설정 시험	알람 설정이 이루어지는지 test	Alarm	Pass
14	r 6.2	알람 설정 삭제 시험	알람 설정이 삭제되는지 test	Alarm	Fail
15	R 6.3	알람 작동 시험	알람이 작동하는지 test	Alarm	Pass
16	R 7.1	번저 시험	번저가 울리는지 test	Buzz	Fail
17	R 7.2	번저 중지 시험	번저가 중지되는지 test	Buzz	Fail
18	R 8.1	스케줄 등록 시험	스케줄이 등록되는지 test	Schedule	Fail
19	r 8.2	스케줄 삭제 시험	스케줄이 삭제되는지 test	Schedule	Pass
20	R 8.3	스케줄 초기화 시험	모든 스케줄이 삭제되는지 test	Schedule	Pass
21	R 8.4	스케줄 알람 시험	스케줄 시간에 알람이 작동하는지 test	Schedule	Pass
22	r 9.1	지출액 추가 시험	지출액이 추가 되는지 test	Expenditure	Pass
23	r 9.2	지출액 초기화 시험	이변달 지출액이 0원으로 초기화 되는지 test	Expenditure	Fail
24	r 9.3	지출액 누적 시험	누적된 지출액이 보이는지 test	Expenditure	Fail
25	r 10.1	데이터 저장 시험	데이터를 저장하는지 test	Data	Fail
26	R 10.2	데이터 로드 시험	데이터를 불러오는지 test	Data	Fail
27	R 10.3	데이터 삭제 시험	데이터가 삭제되는지 test	Data	Fail
28	R 11.1	기능 4개 선택 시험	기능 4개를 선택되는지 test	Select Function	Pass