



Stage 2050

Subject : Software Modeling

Professor : 유준범

T1 | 201111341 김성민

201111379 이한빈

201111397 황정아

OUTLINE

I. Revise Plan

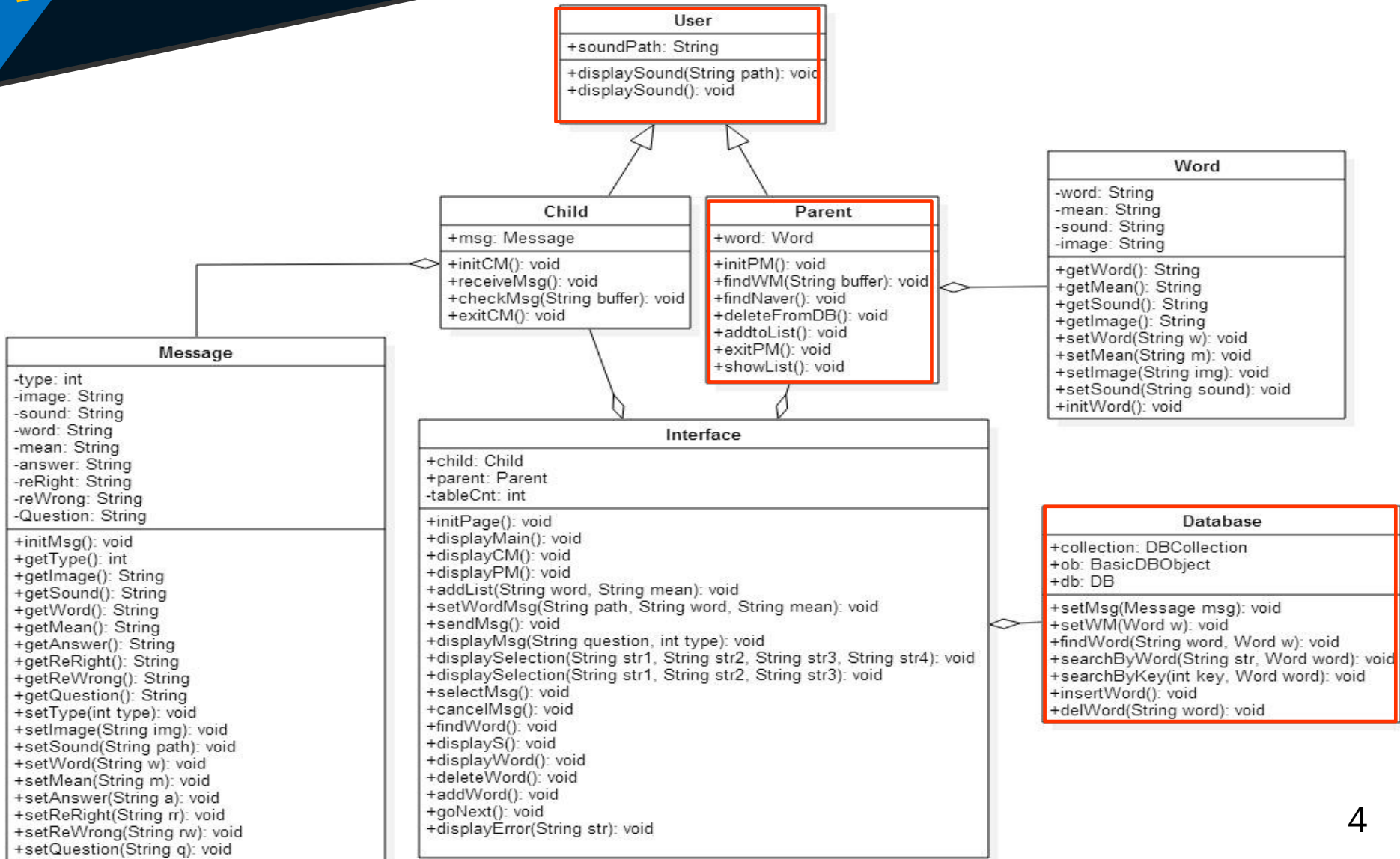
II. Program Structure

III. Unit Test

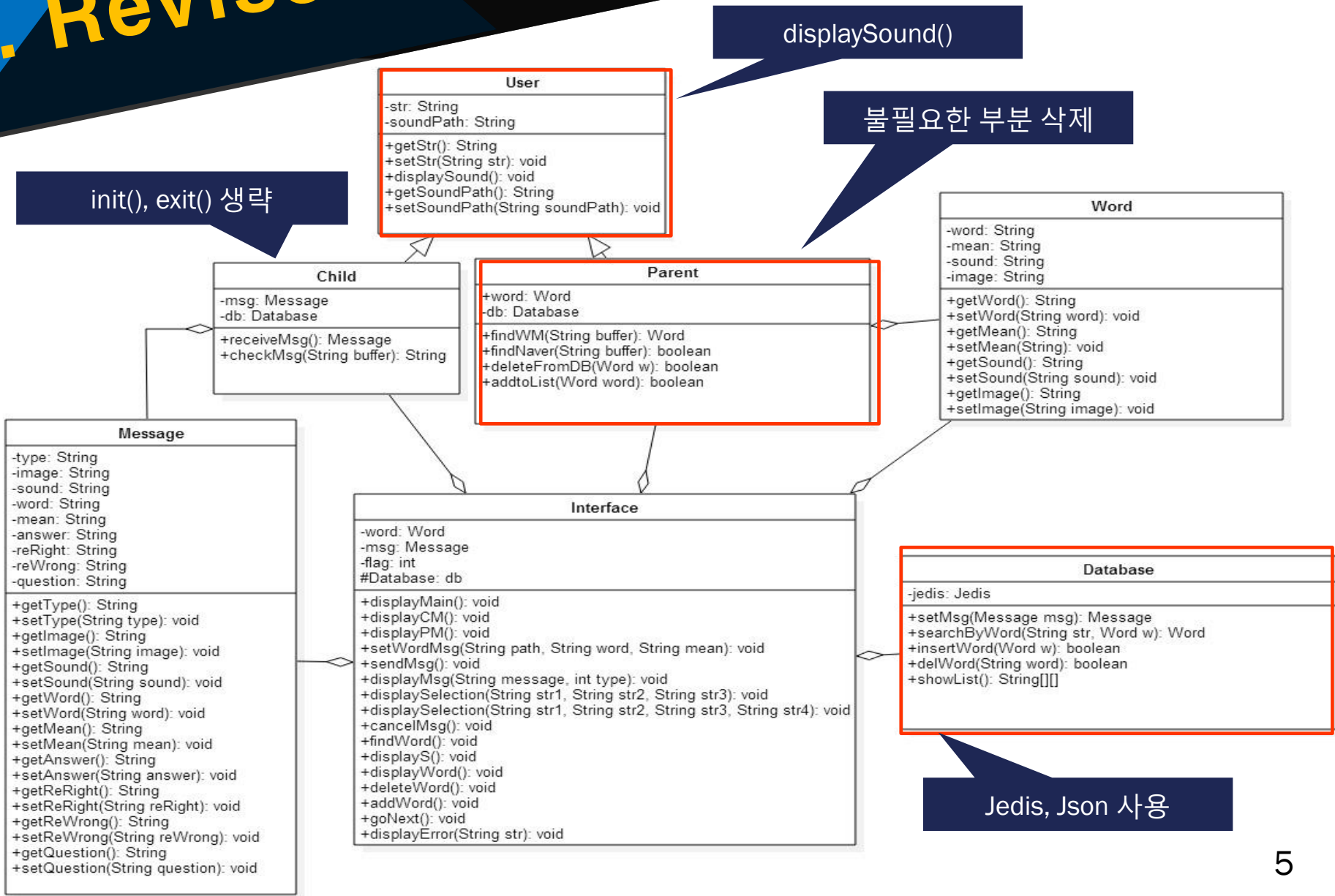
IV. Demonstration

I. Revise Plan

I. Revise Plan



I. Revise Plan



II. Program Structure

II. Program Structure

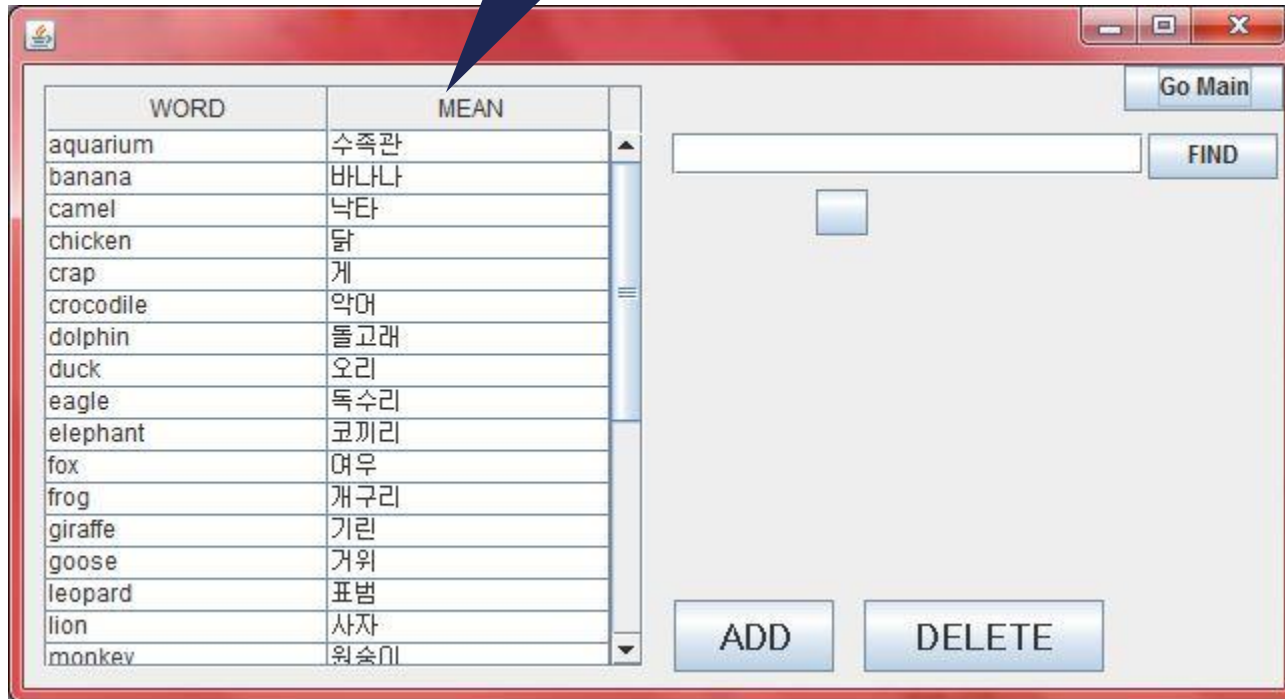
1. Display Main Menu



II. Program Structure

2. Parent Mode

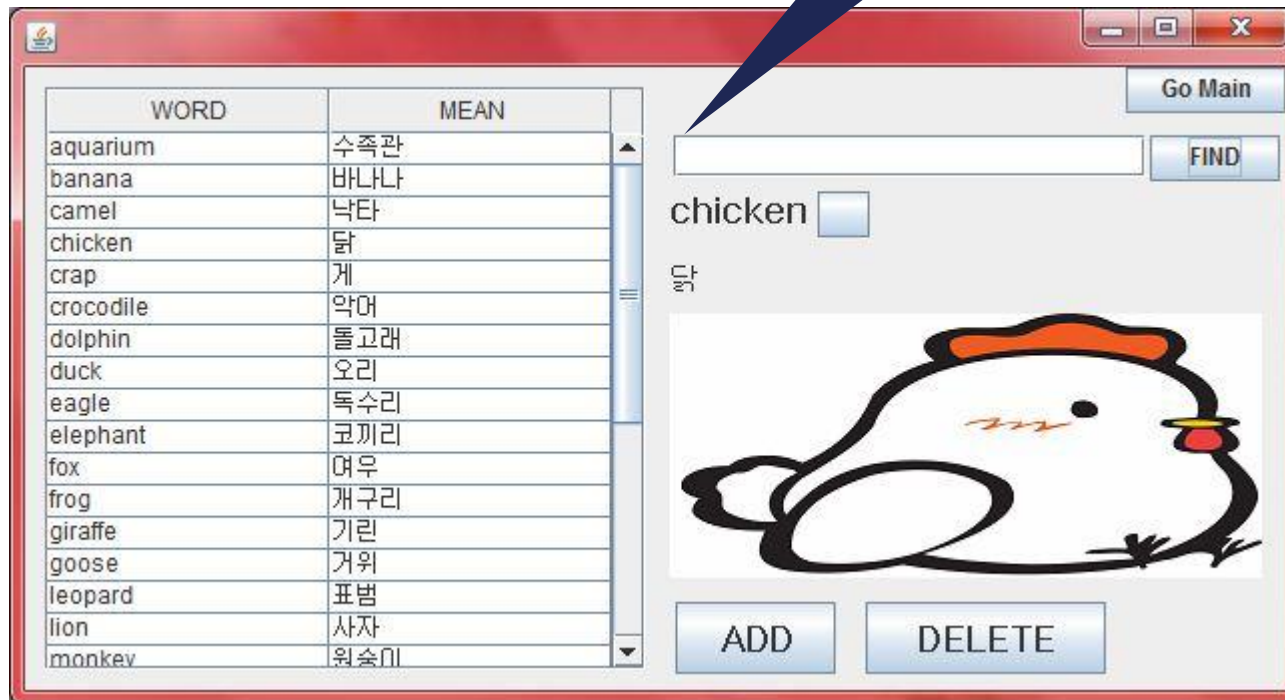
모든 단어 알파벳 순



II. Program Structure

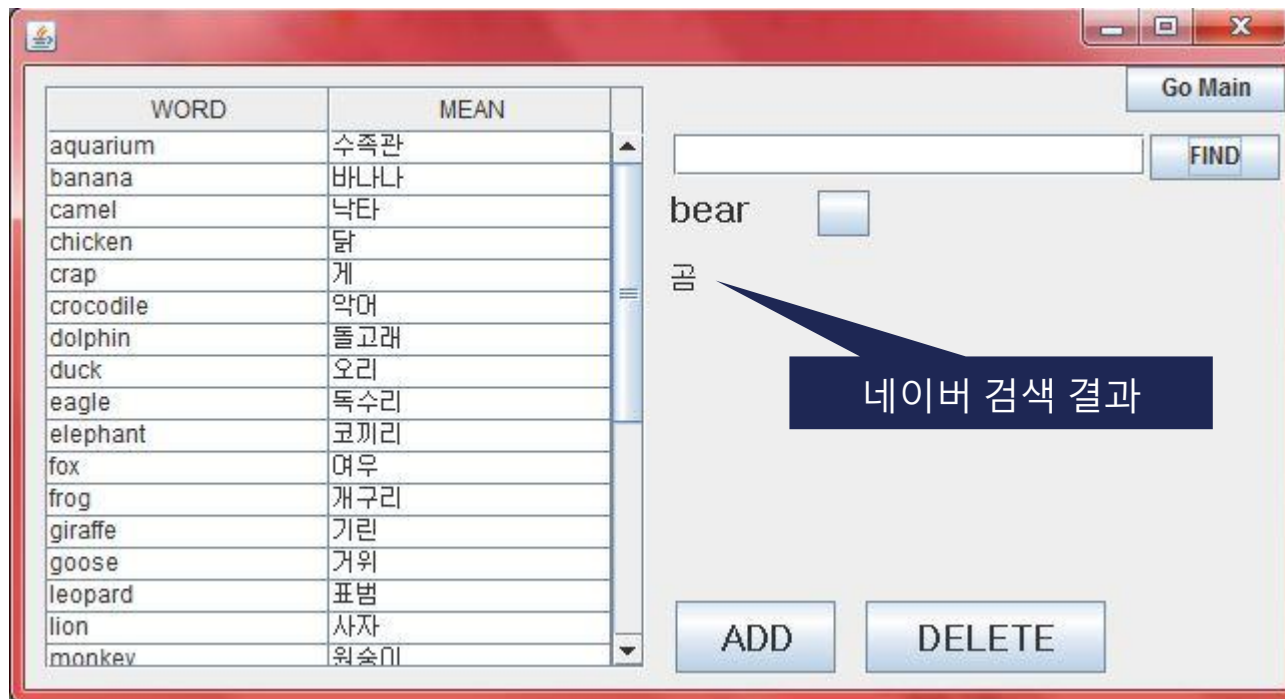
2. Parent Mode

chicken 검색 결과



II. Program Structure

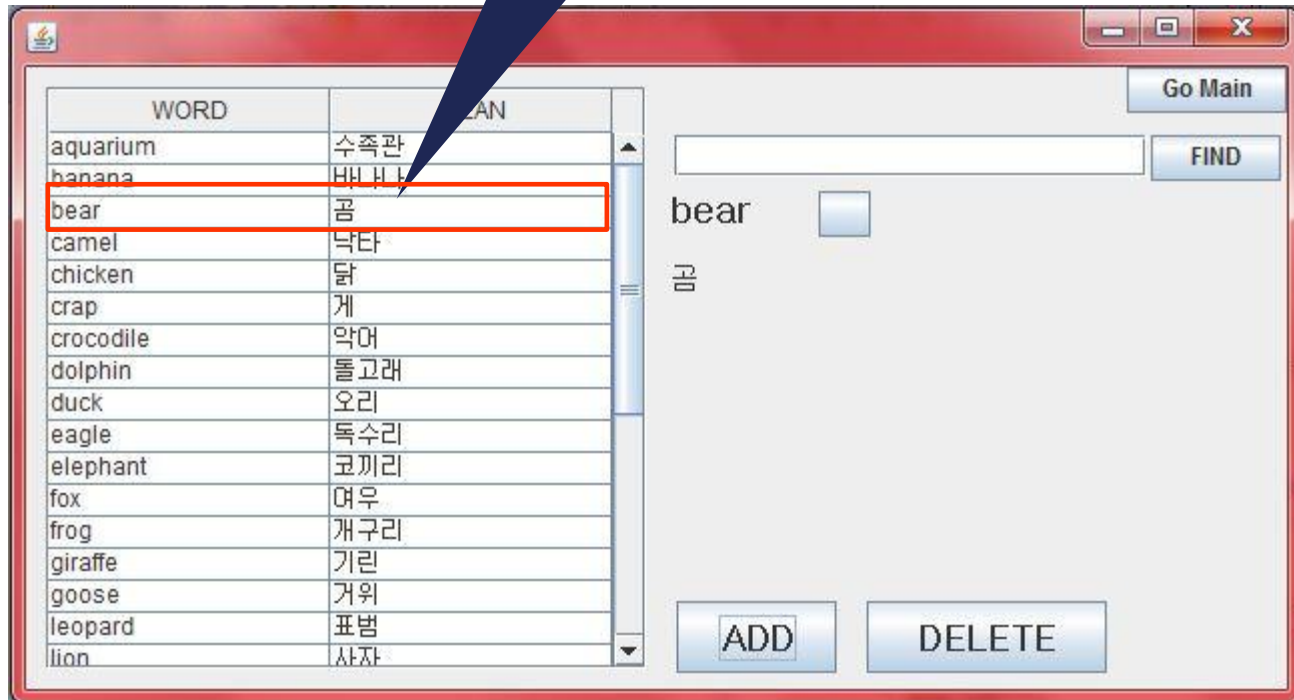
2. Parent Mode



II. Program Structure

2. Parent Mode

새로운 단어 추가



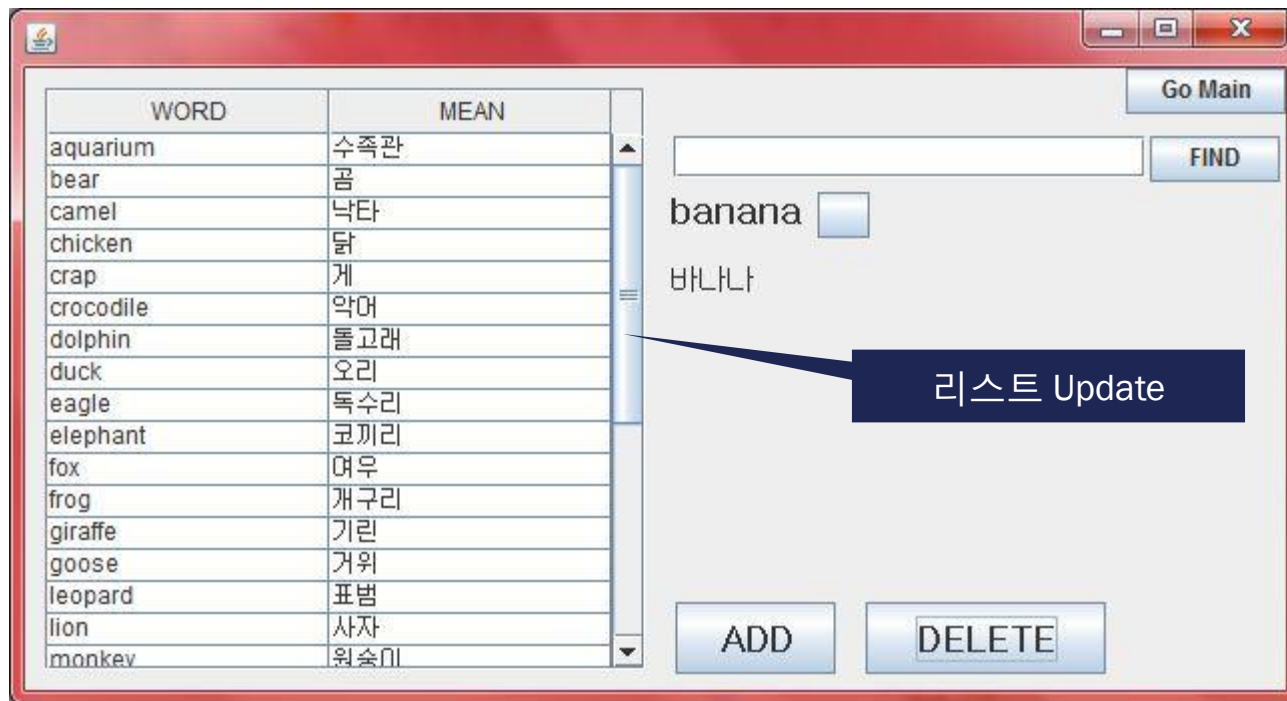
II. Program Structure

2. Parent Mode



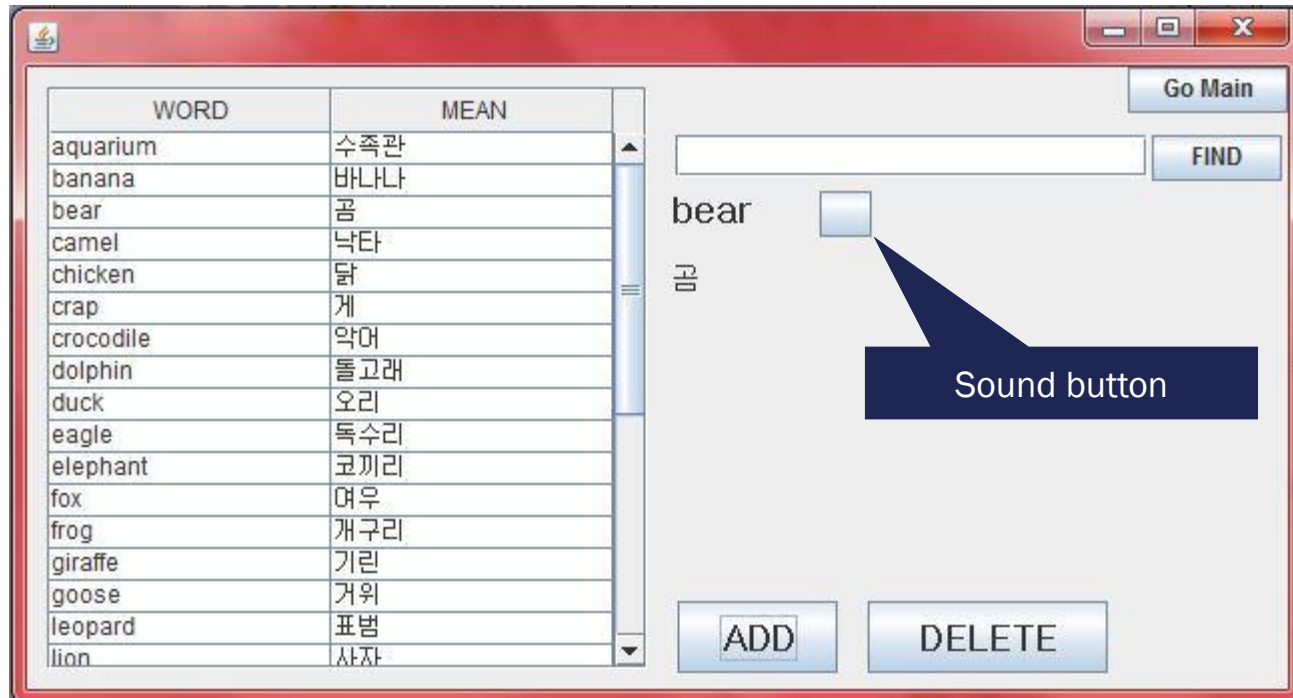
II. Program Structure

2. Parent Mode



II. Program Structure

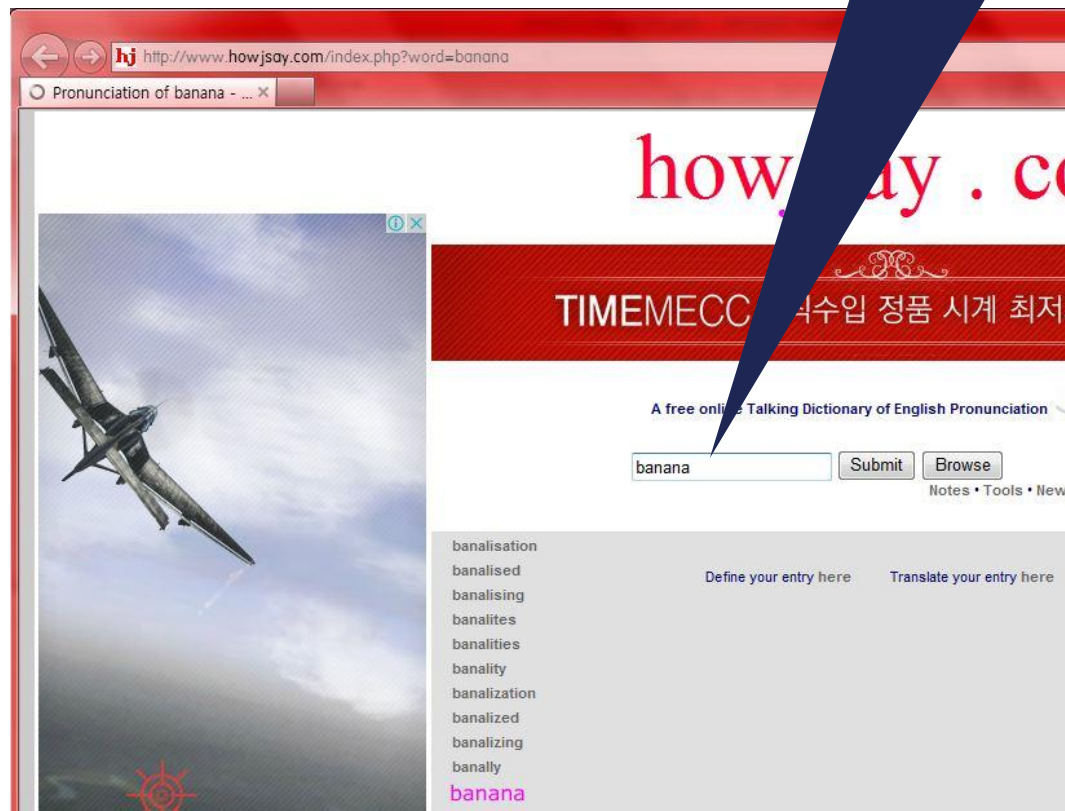
2. Parent Mode



II. Program Structure

2. Parent Mode

새로 찾은 단어는 인터넷에서
sound play 가능



II. Program Structure

3. Child Mode



II. Program Structure

3. Child Mode



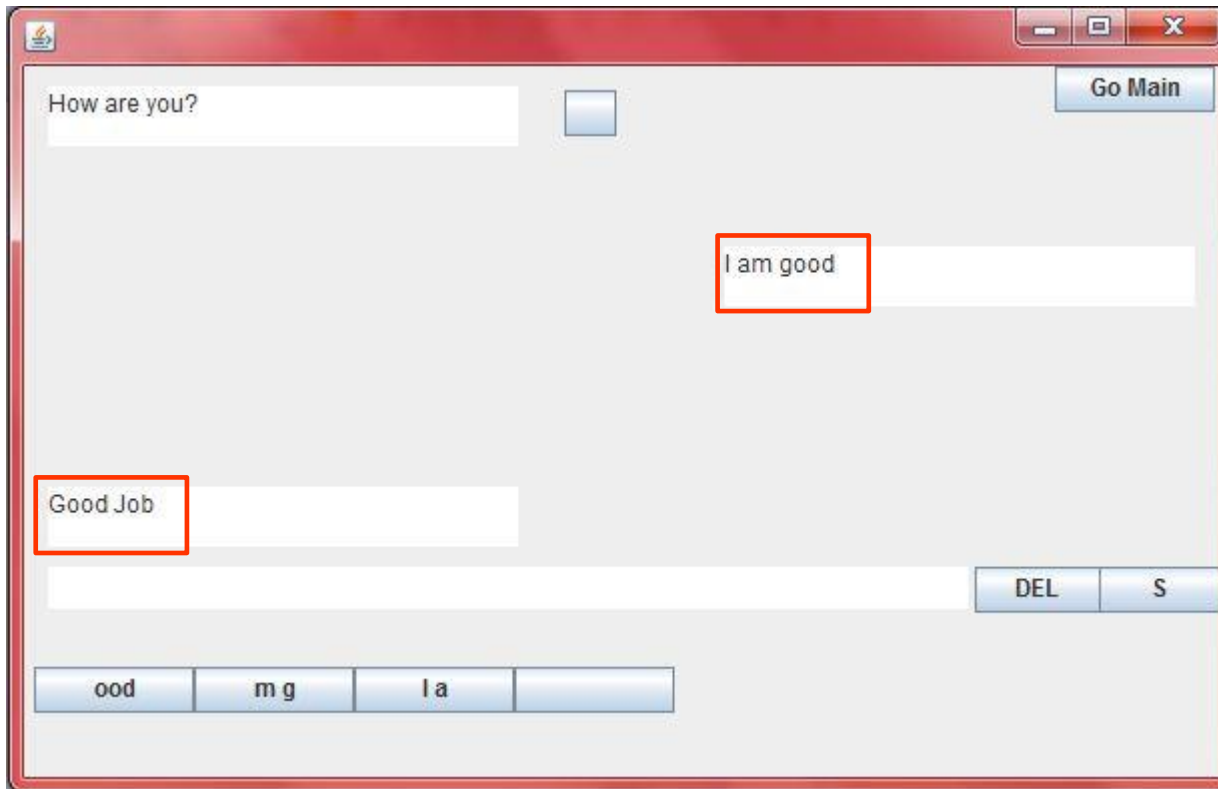
II. Program Structure

3. Child Mode



II. Program Structure

3. Child Mode



III. Unit Test

III. Unit Test

Package Explorer JUnit

Finished after 0.452 seconds

Runs: 10/10 Errors: 0 Failures: 0

- Testing [Runner: JUnit 4] (0.156 s)
 - testCheckMsg (0.031 s)
 - testReceiveMSg (0.125 s)
- UserTest [Runner: JUnit 4] (0.000 s)
 - testDisplaySound (0.000 s)
- ParentTest [Runner: JUnit 4] (0.203 s)
 - testFindNaver (0.203 s)
 - testFindWM (0.000 s)
- DatabaseTest [Runner: JUnit 4] (0.047 s)
 - testInsertWord (0.016 s)
 - testShowList (0.015 s)
 - testSearchByWord (0.016 s)
 - testDelWord (0.000 s)
 - testSetMsg (0.000 s)

```
import static org.junit.Assert.*;

public class Testing extends TestC
    Child c;
    Database db;
    @BeforeClass
    public static void setUpBefore
    }

    @AfterClass
    public static void tearDownAft
    }

    @Before
    public void setUp() throws Exc
        super.setUp();
        db = new Database();
        c = new Child(db);
    }
```

III. Unit Test

1	Name = "testCheckMsg"	Input	Output	Result
1-1	assertEquals("Good Job", result);	tiger	Good Job	PASS
	올바른 단어형 메시지 입력 시 "Good Job" 출력 확인			
1-2	assertEquals("Good Job", result);	Hello	Good Job	PASS
	올바른 대화형 메시지 입력 시 "Good Job" 출력 확인			
1-3	assertEquals("Hello is right answer", result);	ellHo	Hello is right answer	PASS
	잘못된 대화형 메시지 입력 시 "Hello is right answer" 출력 확인			
1-4	assertEquals("Hello is right answer", result);	Null	Hello is right answer	PASS
	Null 입력 시 "Hello is right answer" 출력 확인			

III. Unit Test

2	Name = "testDelWord"	Input	Output	Result
2-1	assertTrue(db.delWord("tiger"));	tiger	True	PASS
	DB에 저장되어 있는 단어 입력 시 삭제 수행, true return			
2-2	assertFalse(db.delWord("apple"));	apple	False	PASS
	DB에 저장되어 있지 않은 단어 입력 시 false return			
2-3	assertFalse(db.delWord(null));	null	False	PASS
	null 입력 시 false return			

III. Unit Test

3	Name = "testInsertWord"	Input	Output	Result
3-1	assertTrue(db.insertWord(w));	w	True	PASS
	DB에 저장되어 있지 않은 Word 입력 시 삽입 수행, true return			
3-2	assertFalse(db.insertWord(w));	w	False	PASS
	DB에 저장되어 있는 Word 입력 시 false return			
3-3	assertFalse(db.insertWord(w));	w (=null)	False	PASS
	null 입력 시 false return			

III. Unit Test

4	Name = "testSearchByWord"	Input	Output	Result
4-1	<pre>assertNotNull(db.searchByWord("tiger"));</pre> DB에 저장되어 있는 단어 입력 시 해당 단어의 Word data return	tiger	Word	PASS
4-2	<pre>assertNotNull(db.searchByWord("tiger").getMean());</pre> DB에 저장되어 있는 단어 입력 시 해당 단어의 Word Mean data 가 null이 아니다.	tiger	호랑이	PASS
4-3	<pre>assertEquals("호랑이", db.searchByWord("tiger").getMean());</pre> DB에 저장되어 있는 단어 입력 시 해당 단어의 Word Mean data 가 입력 단어의 한글 뜻이다.	tiger	호랑이	PASS
4-4	<pre>assertNotNull(db.searchByWord("apple"));</pre> DB에 저장되어 있지 않은 단어 입력 시 해당 단어의 Word data return	apple	Word	PASS

III. Unit Test

4	Name = "testSearchByWord"	Input	Output	Result
4-5	assertNull(db.searchByWord("apple").getMean());	apple	null	PASS
	DB에 저장되어 있지 않은 단어 입력 시 해당 단어의 Word Mean data 가 null이다.			
4-6	assertNotNull(db.searchByWord(null));	Null	Word	PASS
	Null 입력 시 해당 단어의 Word data return			
4-7	assertNull(db.searchByWord(null).getMean());	null	null	PASS
	Null 입력 시 해당 단어의 Word Mean data 가 null이다.			

III. Unit Test

5	Name = "testSetMsg()"	Input	Output	Result
5-1	assertNotNull(db.setMsg());	-	Message	PASS
	메시지를 랜덤으로 뽑아서 Message 형 리턴 확인			
6	Name = "testShowList()"	Input	Output	Result
6-1	assertNotNull(db.showList());	-	String[][]	PASS
	단어와 뜻 배열을 리턴			
7	Name = "testFindWM"	Input	Output	Result
7-1	assertNotNull(p.findWM("tiger"));	tiger	Word	PASS
	DB에 저장되어 있는 단어 입력 시 Word 형 리턴			
7-2	assertNull(p.findWM("*"));	*	Null	PASS
	특수문자 단어 입력 시 Word 형 리턴			

III. Unit Test

8	Name = "testFindNaver"	Input	Output	Result
8-1	assertTrue(p.findNaver("apple"));	apple	true	PASS
	영단어 입력 후, 검색 결과 존재 시, true 리턴			
8-2	assertTrue(p.findNaver("사탕"));	사탕	true	PASS
	한글 단어 입력 후, 검색 결과 존재 시, true 리턴			
8-3	assertTrue(p.findNaver("dkkfjksd"));	dkkfjk sd	false	PASS
	Random String 입력 후, 검색 결과가 없으면, false 리턴			
9	Name = "testReceiveMsg"	Input	Output	Result
9-1	assertNotNull(c.receiveMsg());	-	Message	PASS
	db에서 올바르게 msg를 읽어오면 true 리턴			

IV. Demonstration

Q&A ?

THANK YOU