

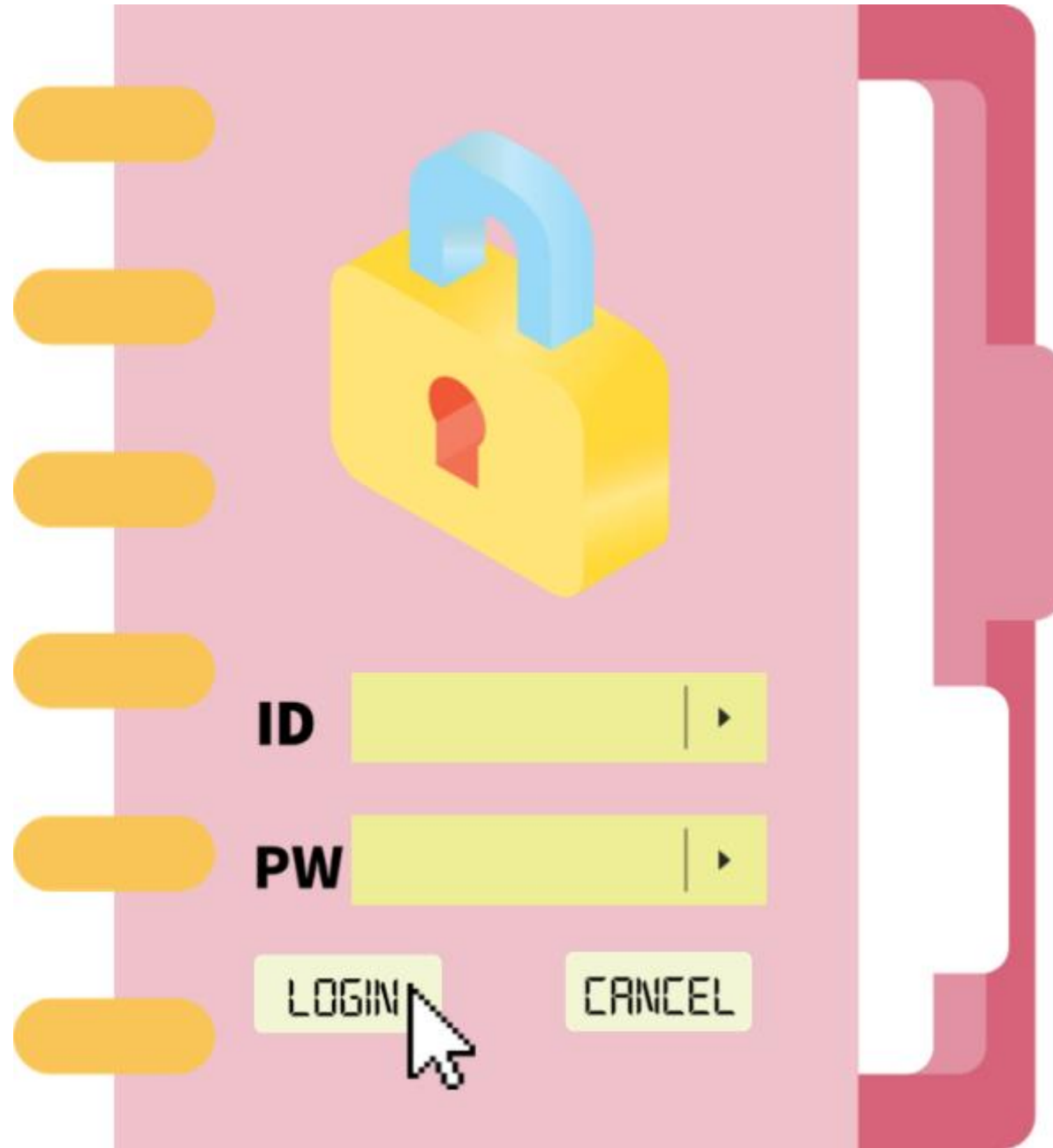
INDEX

- 01 프로그램의 목적
- 02 프로그램 개발 내용 및 구현 결과
- 03 사용한 클래스
- 04 프로그램 제작 과정
- 05 팀플 참여도와 역할

01

프로그램의 목적

프로그램의 목적 - 다이어리 프로그램



나만의 다이어리 프로그램으로 사용자가 원하는 날짜에 일기와 할 일, 기분을 기록할 수 있는 스케줄러 프로그램입니다.

로그인 -> 다이어리 프로그램 접속 -> 캘린더에서 날짜 선택
-> 오늘의 일기 or 할 일, or 기분 선택

오늘의 일기는 사진업로드와 일기를 작성하고 저장할 수 있고 오늘의 할일을 적고 저장할수 있으며 오늘의 기분은 8가지 중에 내 기분을 선택 이 가능합니다.

02

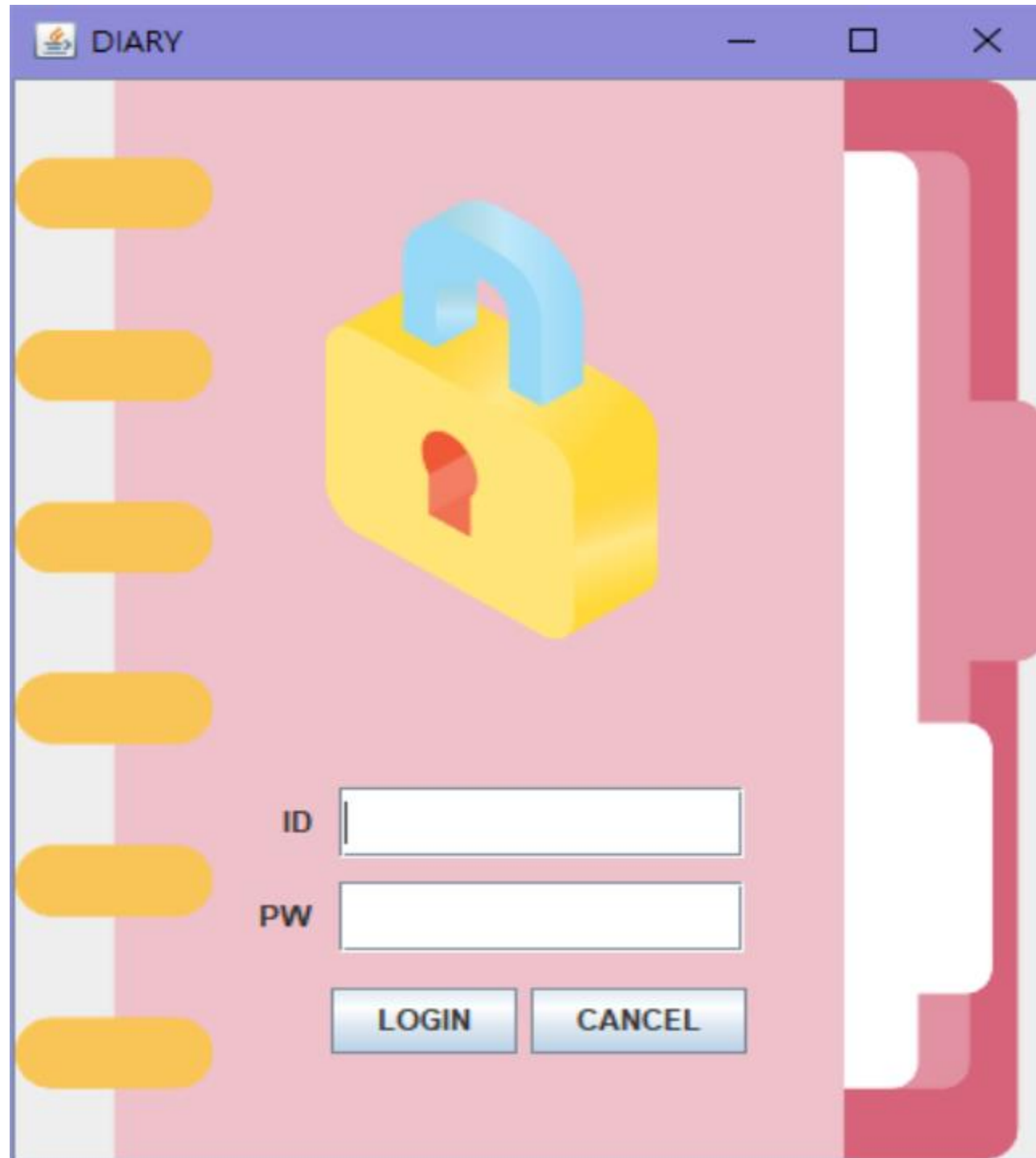
프로그램 개발 내용 및 구현 결과

1. 로그인
2. 달력 날짜 선택
3. 오늘의 일기 쓰기
4. 오늘의 할 일 쓰기
5. 오늘의 기분 체크하기

01

로그인 - # 로그인 기능

사용할 수 있는 기능



The screenshot shows a login window titled "DIARY". It features a large yellow padlock icon in the center. On the left side, there are six yellow horizontal bars. Below the padlock, there are two input fields: "ID" and "PW". At the bottom, there are two buttons: "LOGIN" and "CANCEL".

STEP 01

ID 입력하기

STEP 02

PW 입력하기

STEP 03

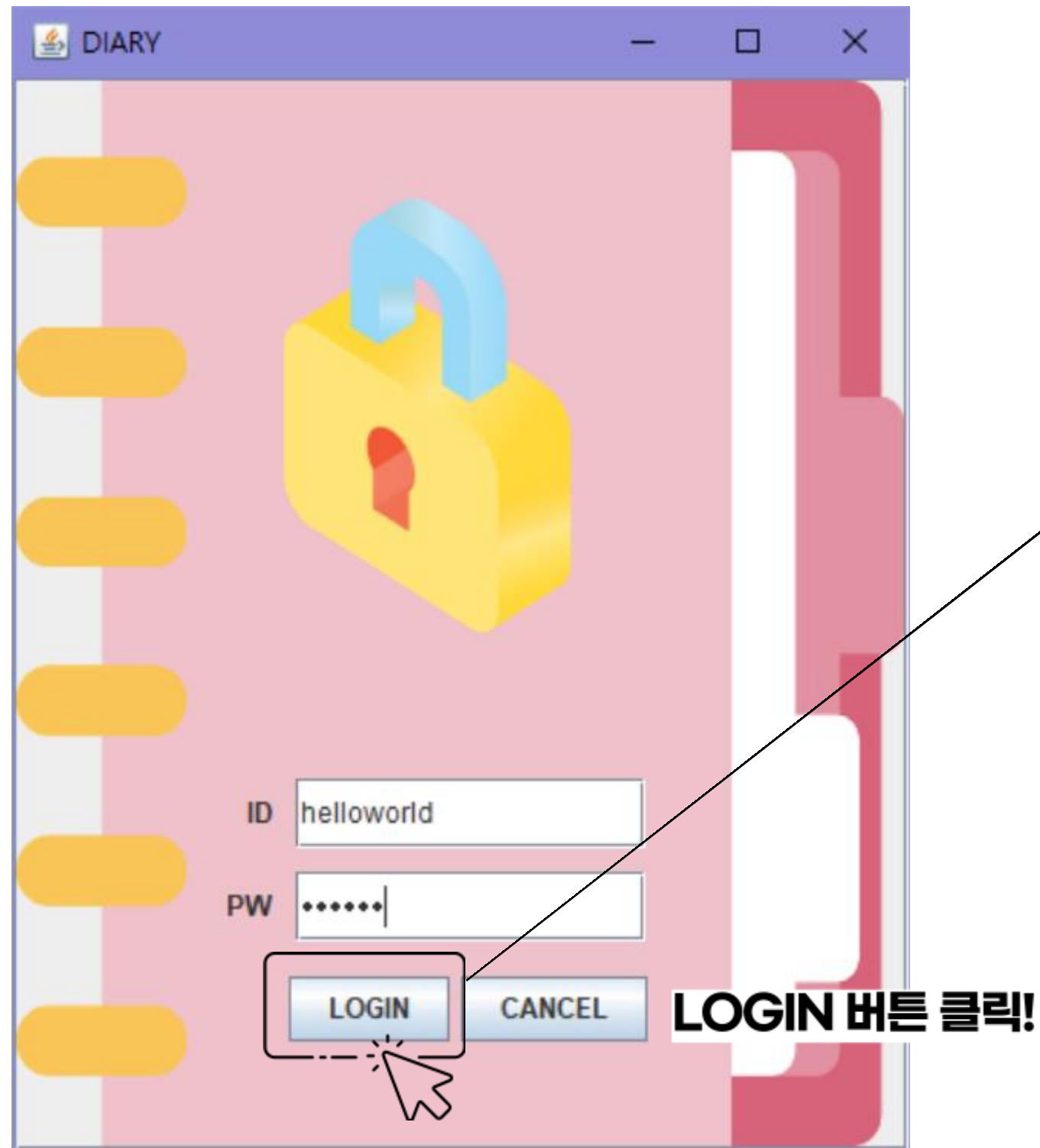
LOGIN 버튼
클릭하여 로그인

STEP 04

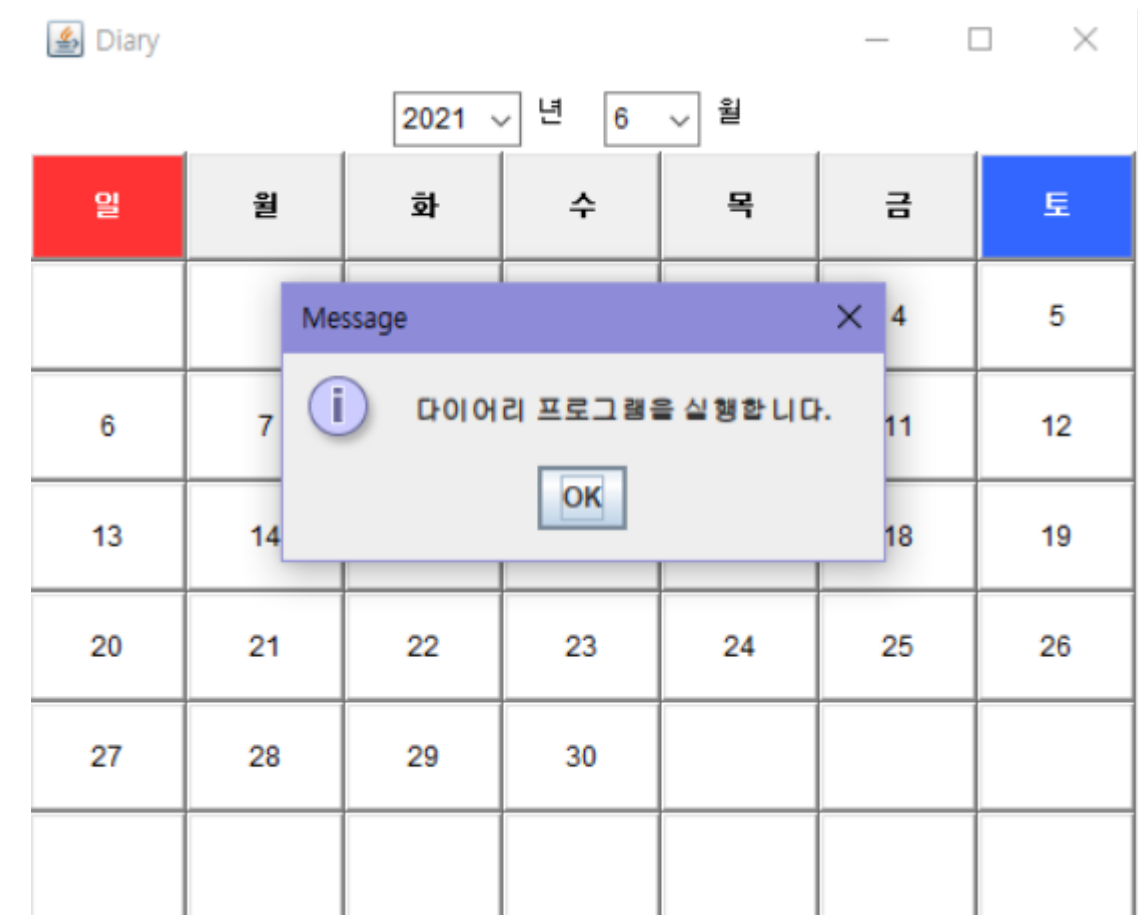
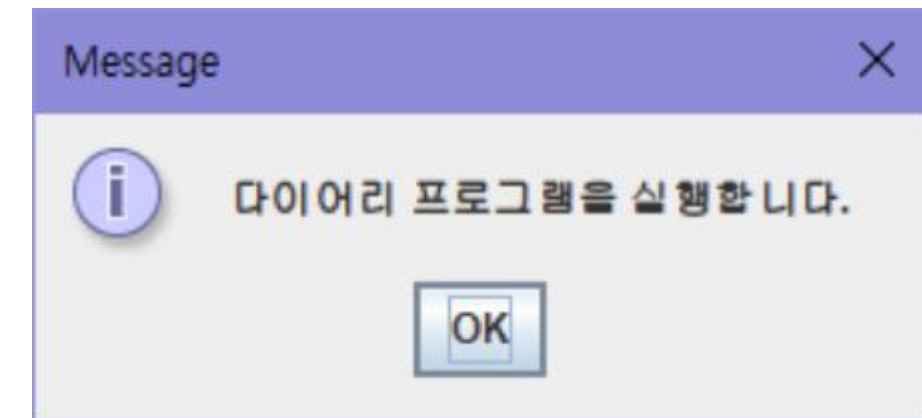
CANCEL 버튼
클릭하여 취소
및 종료

01

로그인 - # 로그인에서 달력화면으로 넘어갈 때 문구 출력

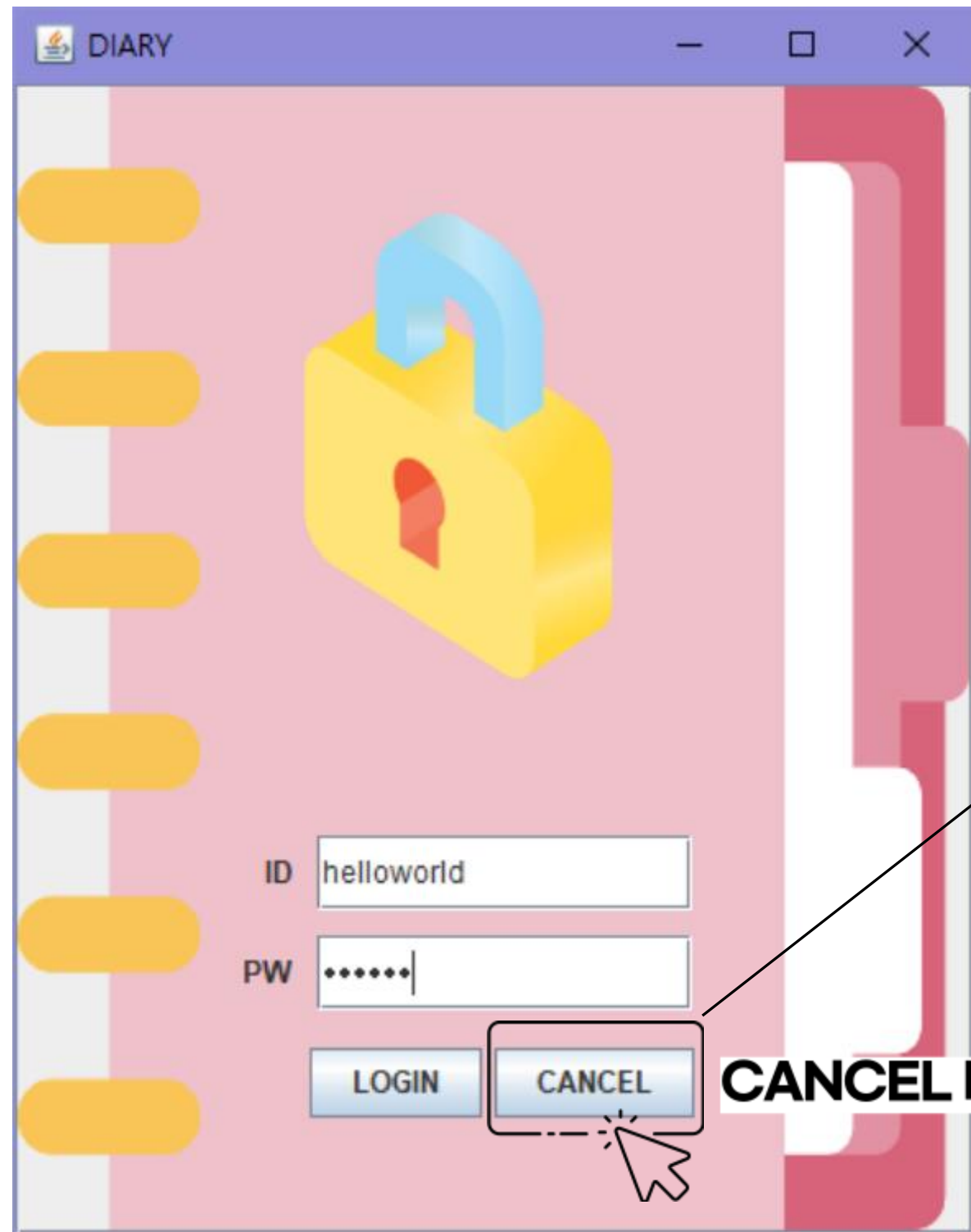


다이어리 실행 문구 출력



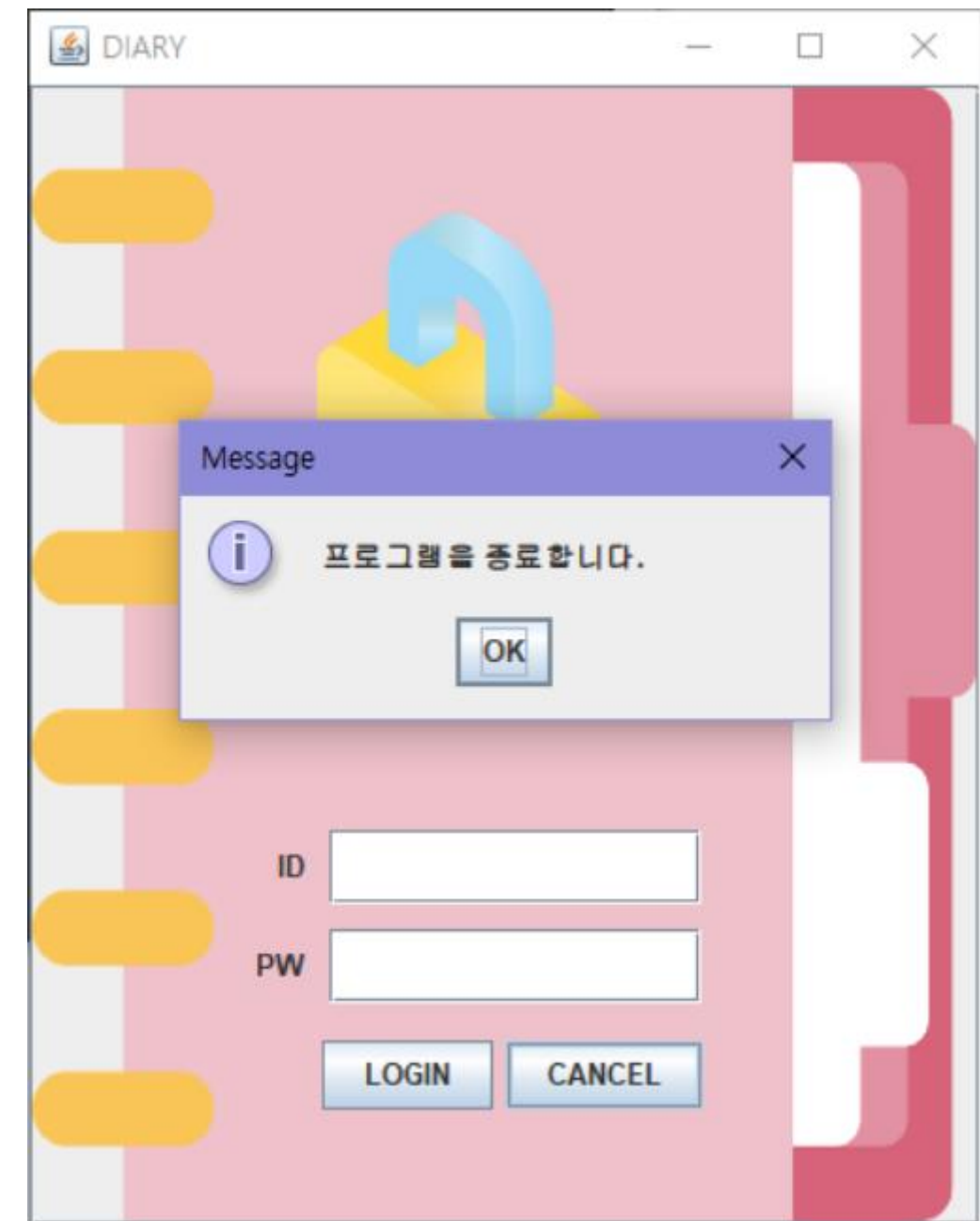
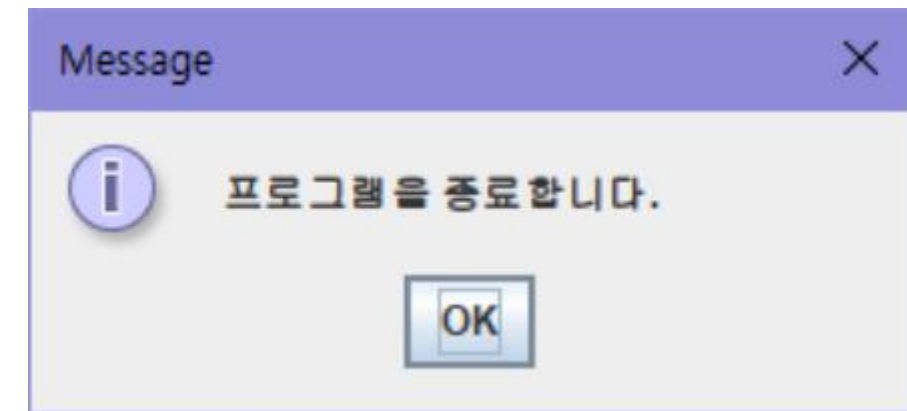
01

로그인 - # CANCEL 기능



CANCEL 버튼 클릭!

프로그램 종료 문구 출력



프로그램이 종료 된다.

달력 날짜 선택 - # 달력의 날짜 선택해서 기능 사용하기

기능 선택하기 창 화면 생성

Diary

2021 년 6 월

일	월	화	수	목	금	토
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

원하는 날짜 클릭!

화면 이동

기능 선택하기

사용하고 싶은 기능을 선택하세요 !

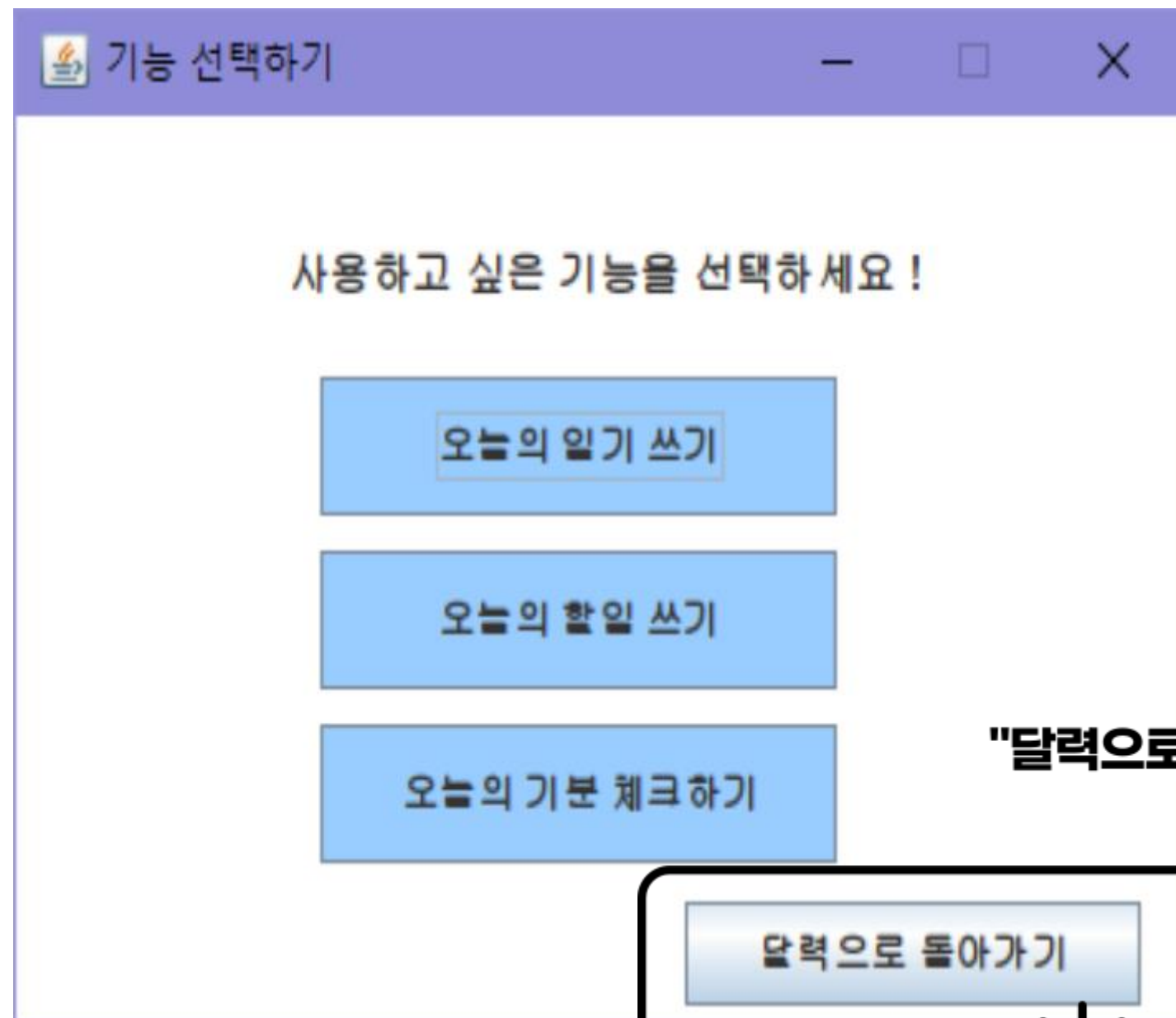
오늘의 일기 쓰기

오늘의 할일 쓰기

오늘의 기분 체크하기

달력으로 돌아가기

달력 날짜 선택 - # 기능 선택하기 버튼과 달력으로 돌아가기 버튼



"달력으로 돌아가기" 클릭!

Diary

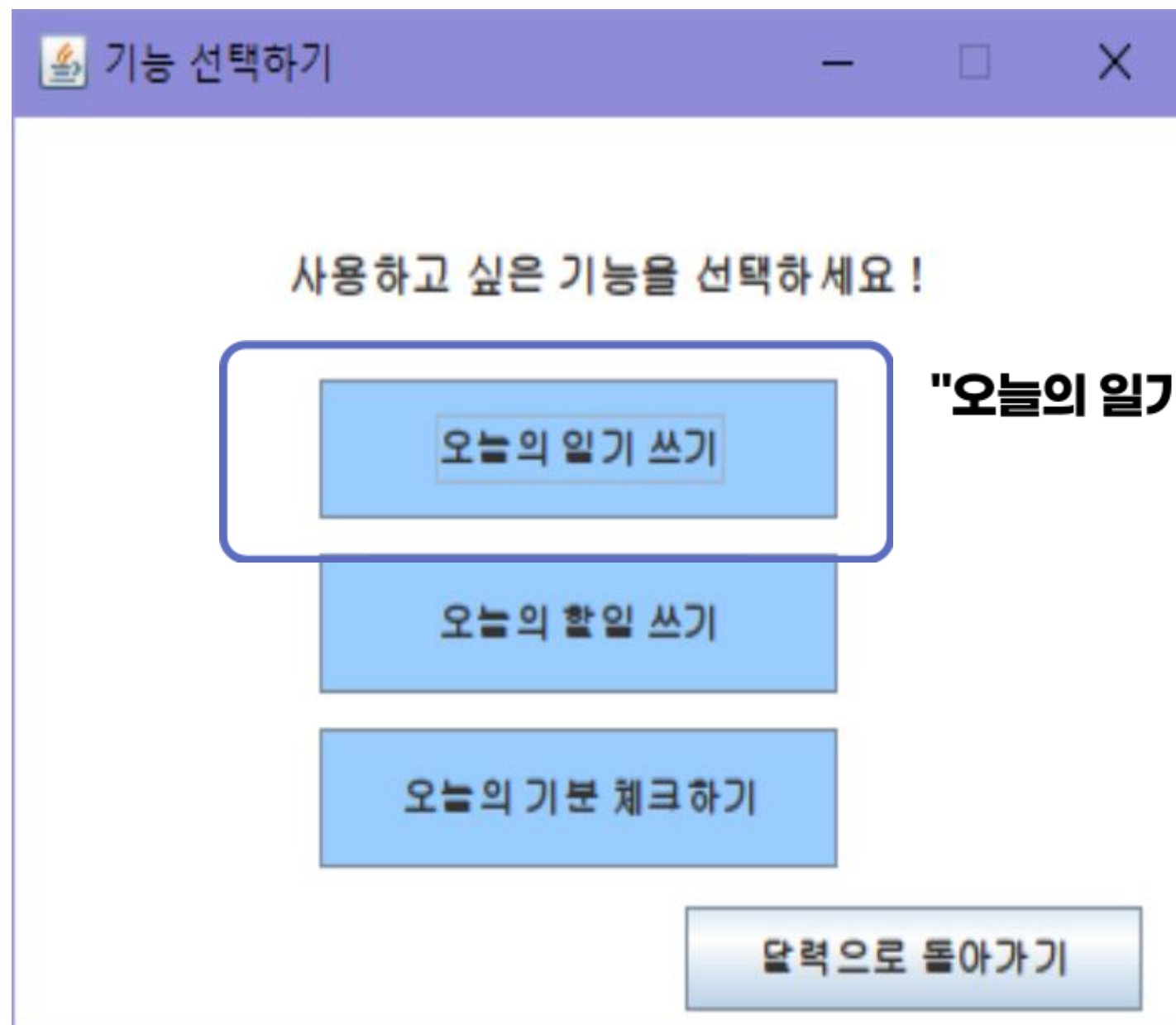
2021 년 6 월

일	월	화	수	목	금	토
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

The calendar shows the month of June 2021. The days of the week are listed in the header. The date 16 is highlighted with a dotted border.

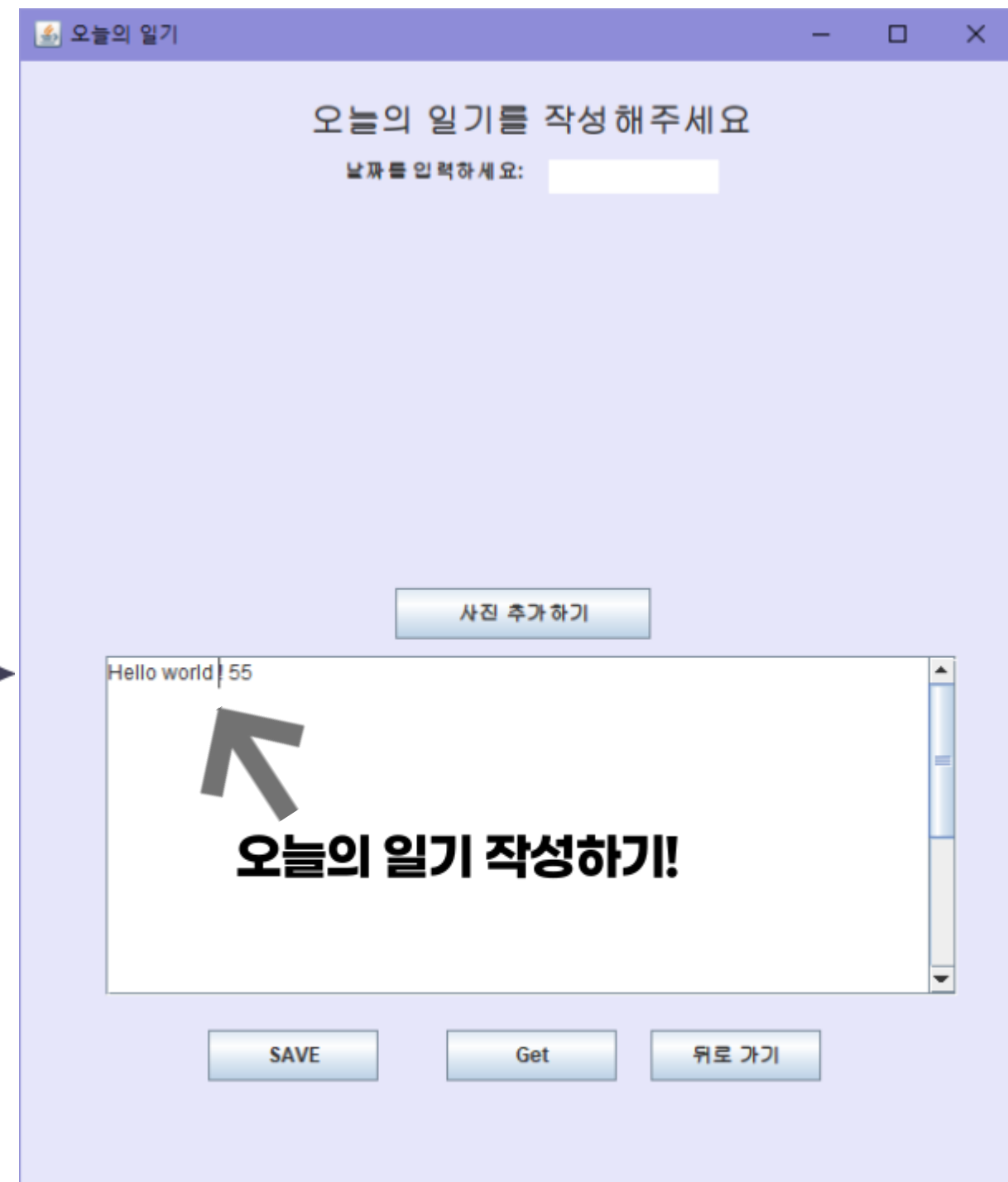
다시 달력화면으로 돌아가게 된다!

오늘의 일기 쓰기 - # 다이어리에 글 작성하기



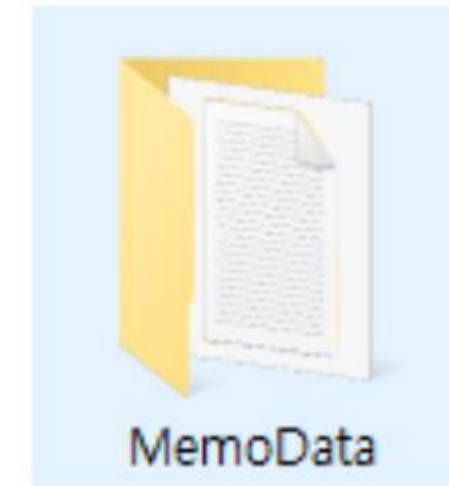
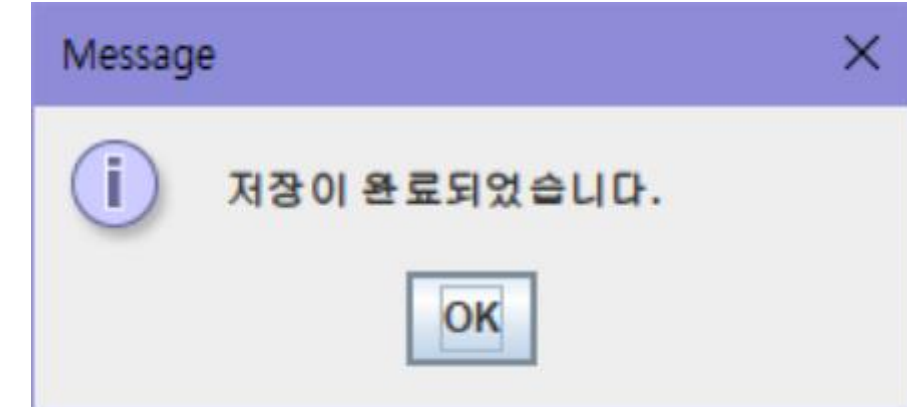
"오늘의 일기 쓰기" 버튼 클릭!

화면 이동

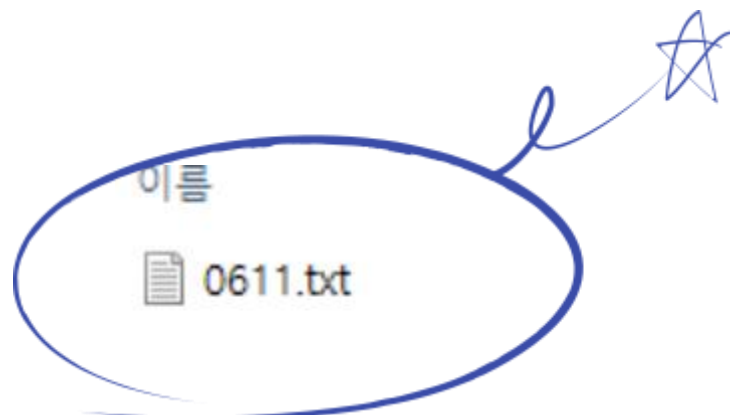


03

오늘의 일기 쓰기 - # 작성한 글 해당 날짜에 저장하기

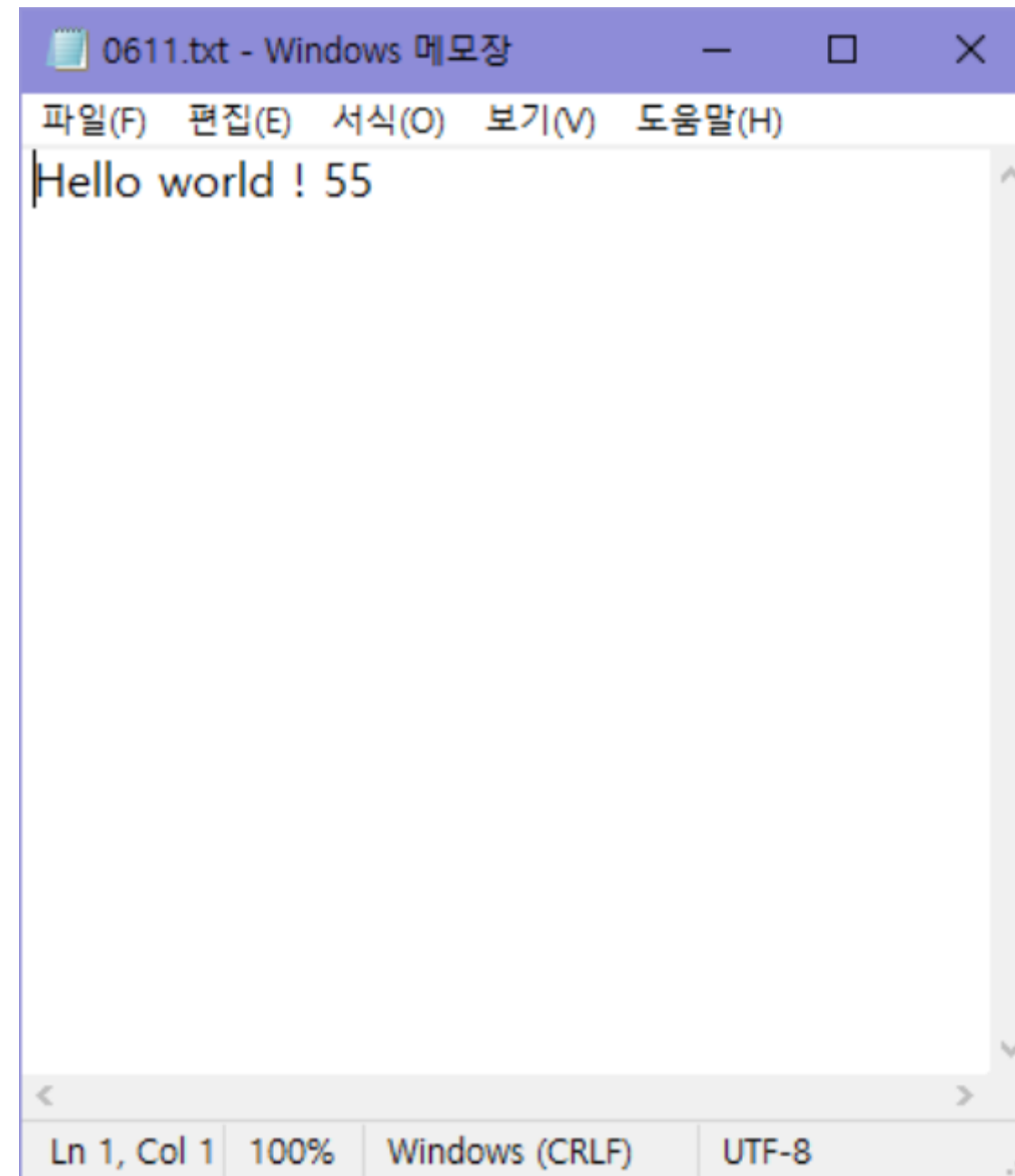


MemoData 폴더가 생성됨!



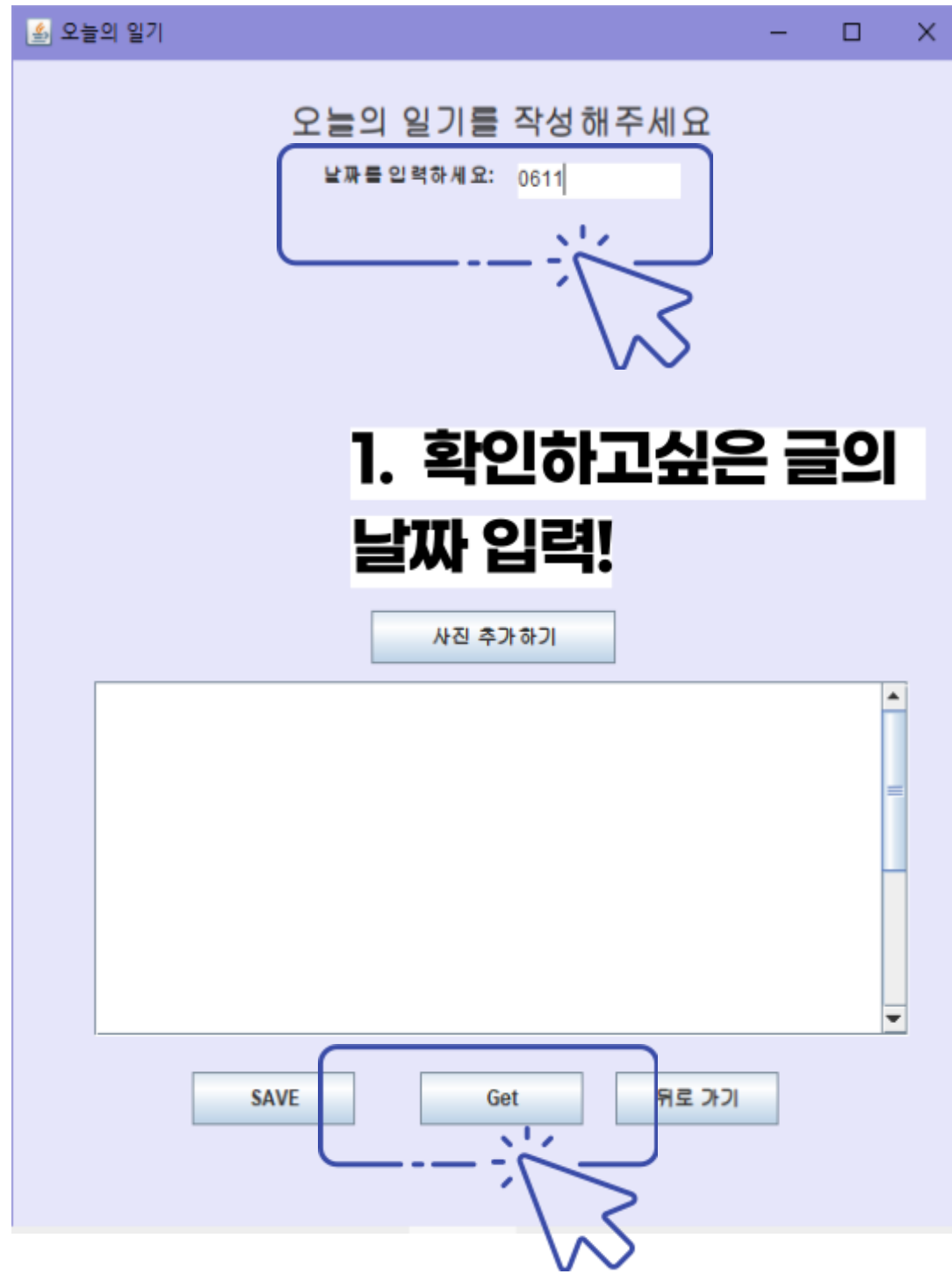
이름	수정된 날짜	유형	크기
0611.txt	2021-06-12 오후 4:41	텍스트 문서	1KB

MemoData 폴더안에 지정한 날짜명으로 텍스트 파일 생성



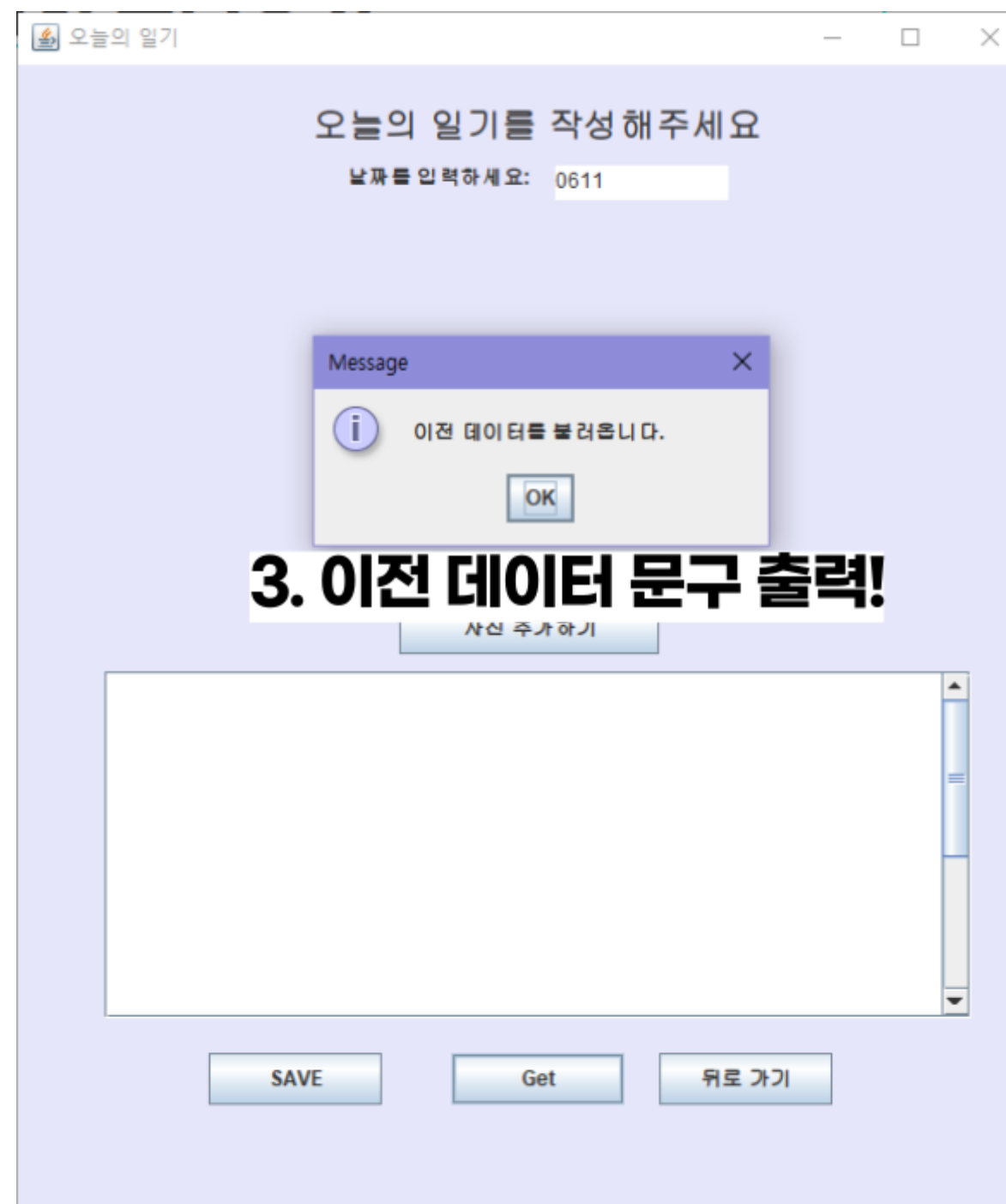
**텍스트 파일을 열면 작성한 글이
저장되어진 것을 확인 할 수 있다.**

오늘의 일기 쓰기 - # 작성한 글 다시 불러오기

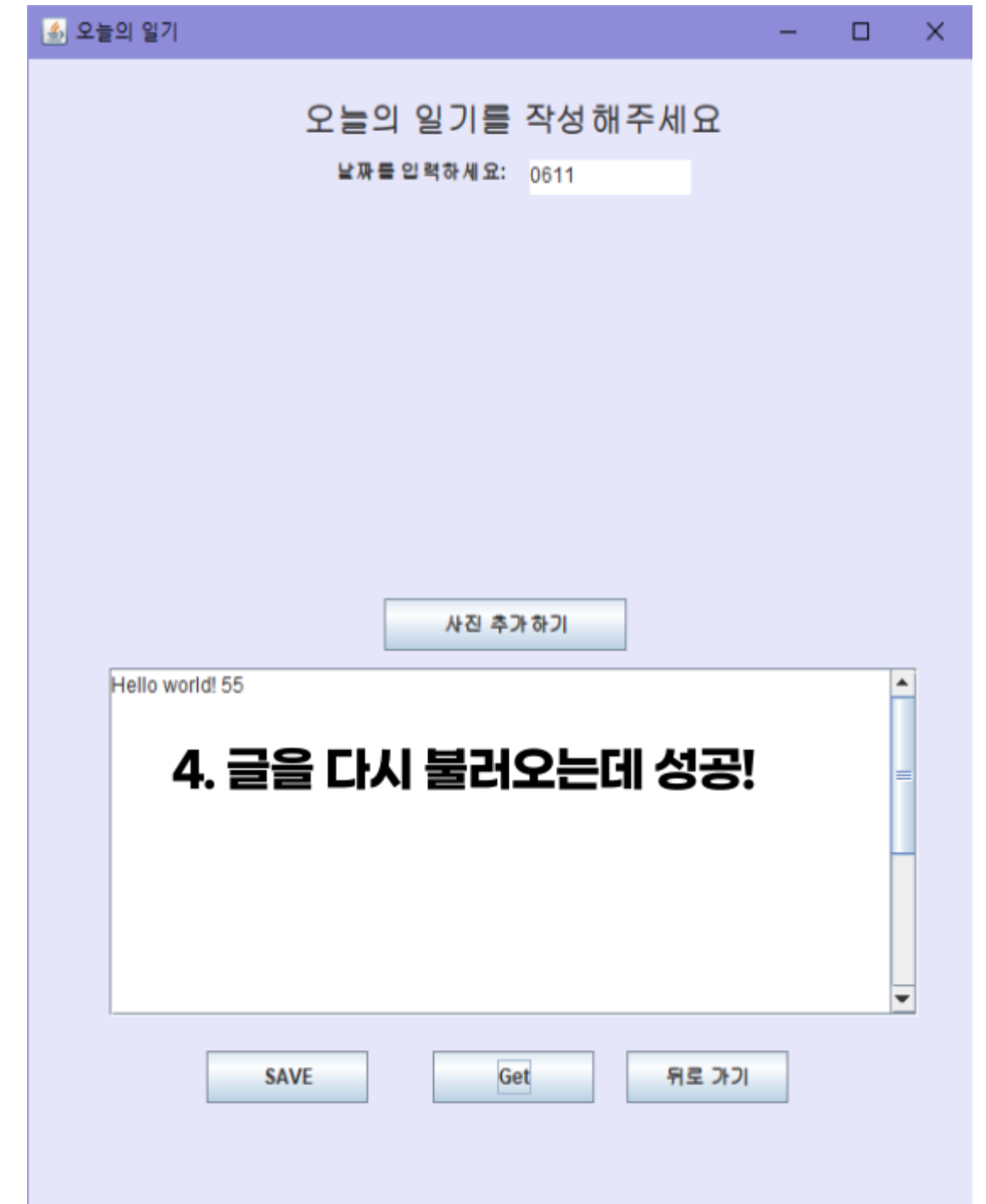


**1. 확인하고싶은 글의
날짜 입력!**

2. Get 버튼 클릭!



3. 이전 데이터 문구 출력!



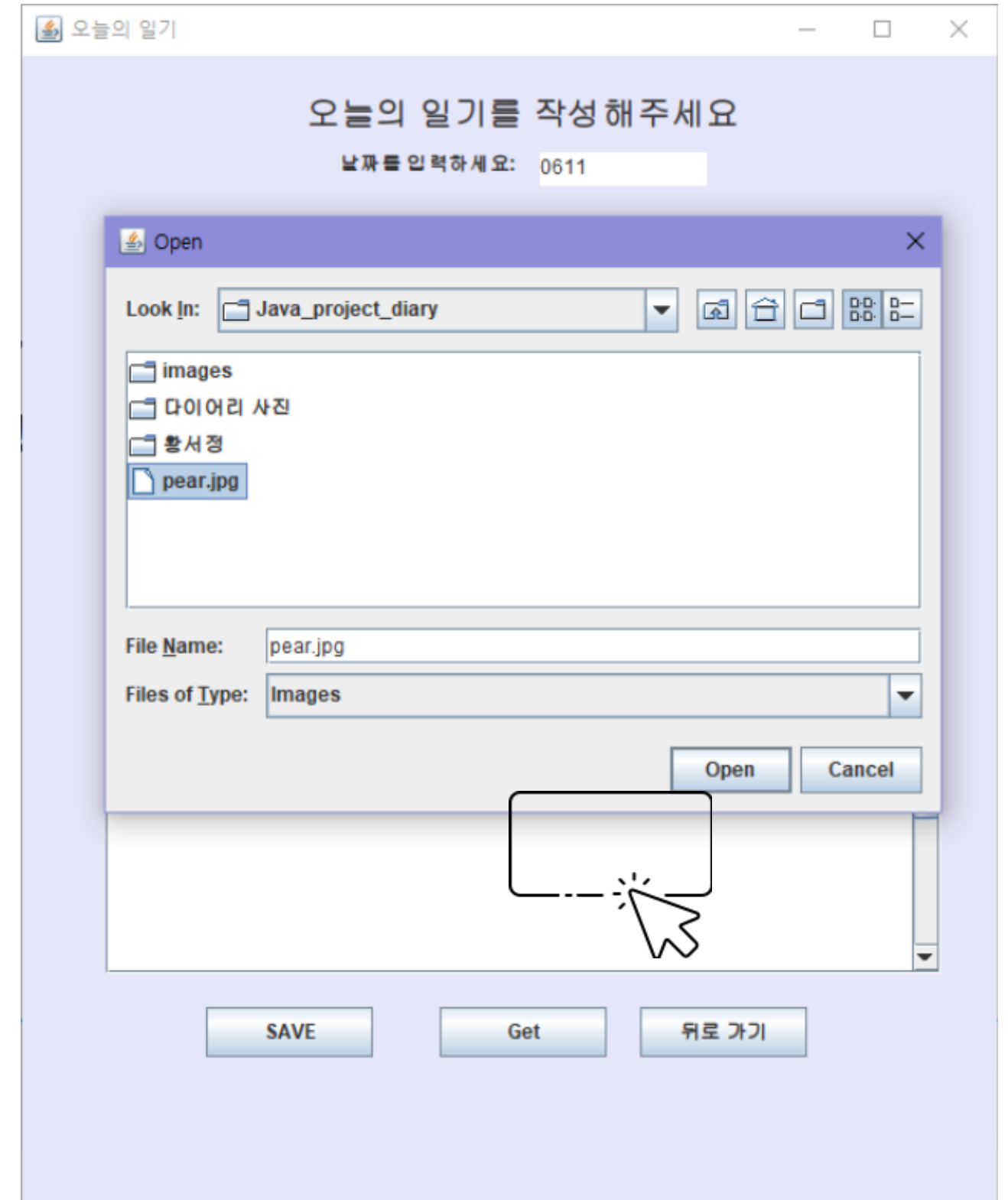
4. 글을 다시 불러오는데 성공!

03

오늘의 일기 쓰기 - # 다이어리에 사진 업로드하기



"사진 추가하기" 클릭!



원하는 사진파일 선택 후 Open

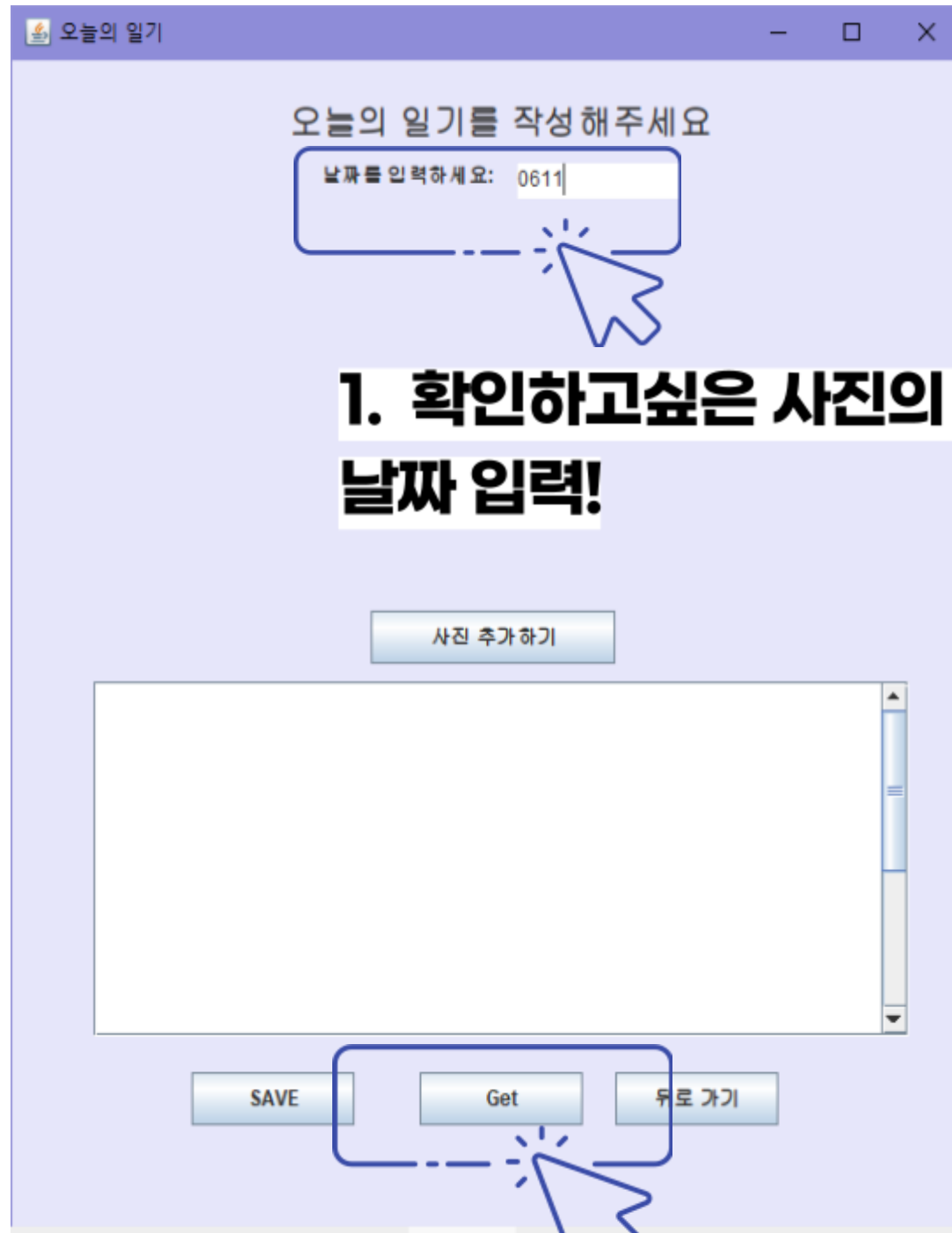
오늘의 일기 쓰기 - # 다이어리에 사진 업로드하기



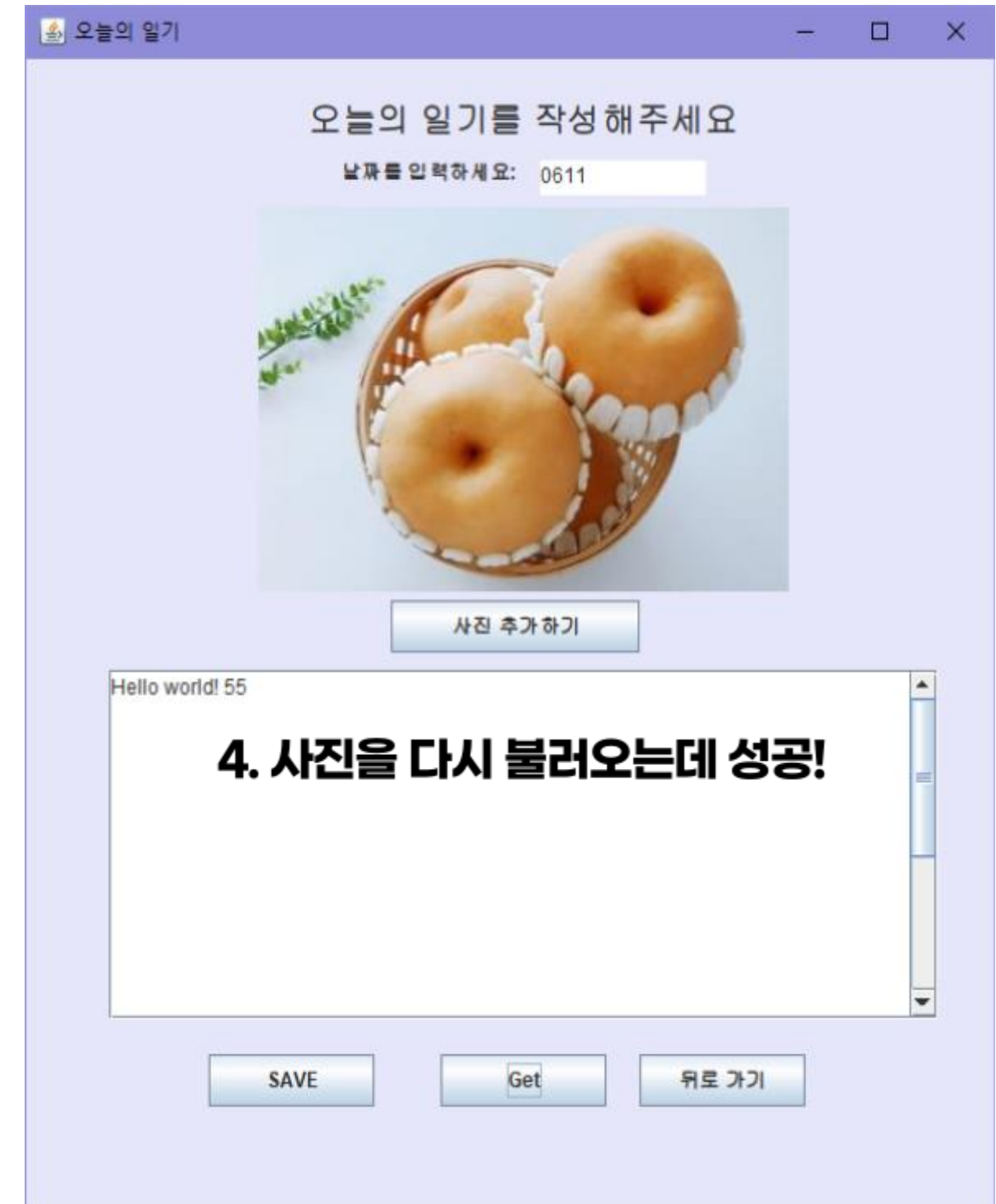
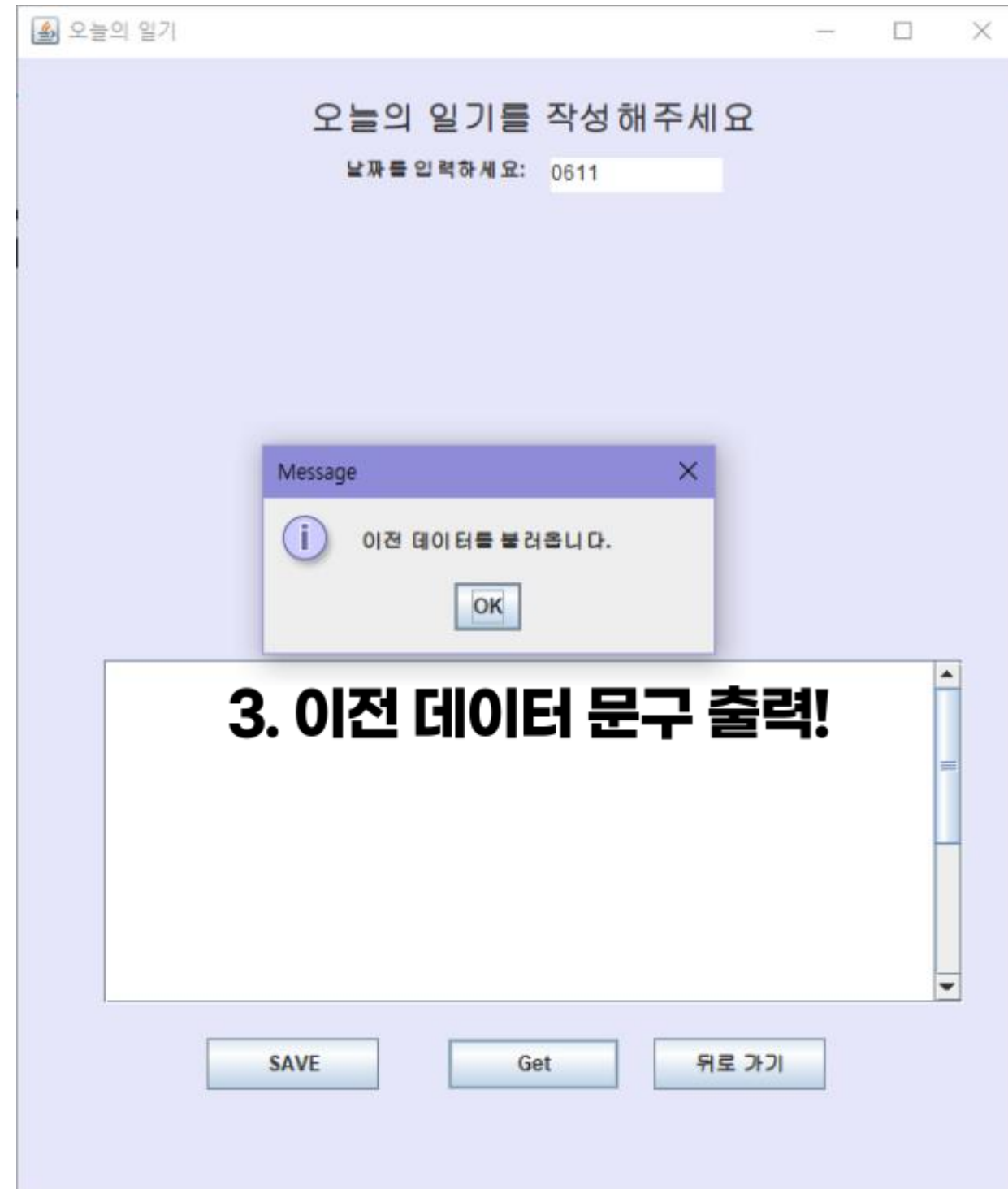
사진이 성공적으로 업로드 된 것을 확인할 수 있다.

03

오늘의 일기 쓰기 - # 업로드한 사진 불러오기



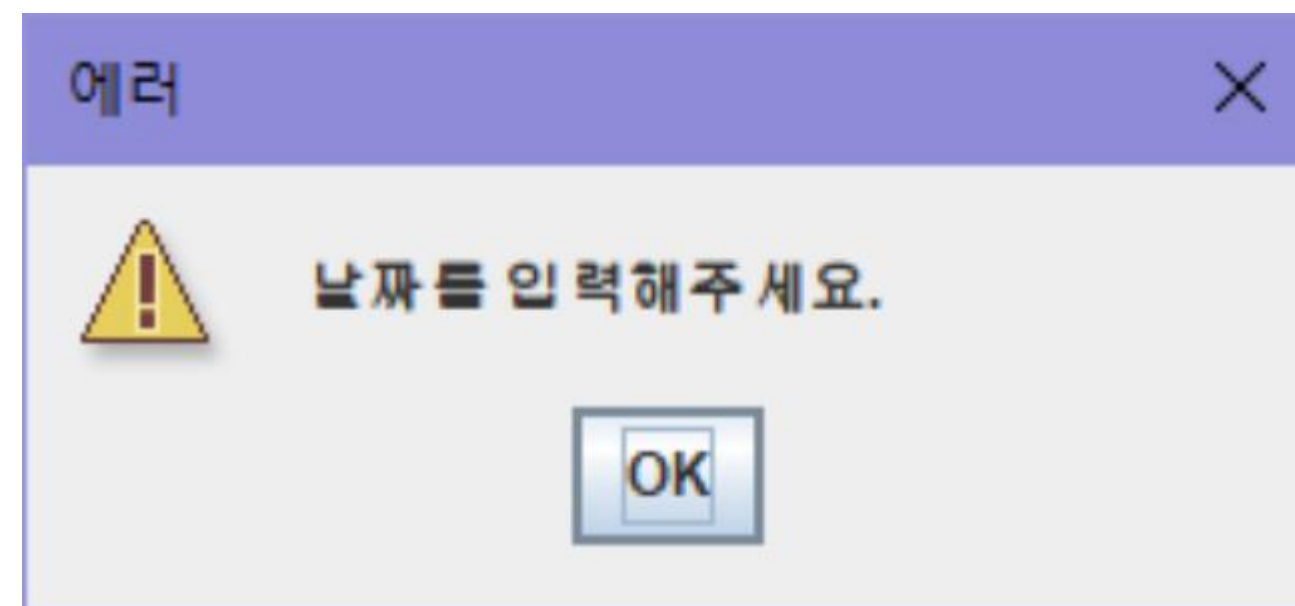
2. Get 버튼 클릭!



오늘의 일기 쓰기 - # 날짜를 입력하지 않고 SAVE 버튼 클릭하였을 때



글을 쓰고 SAVE 버튼 클릭!

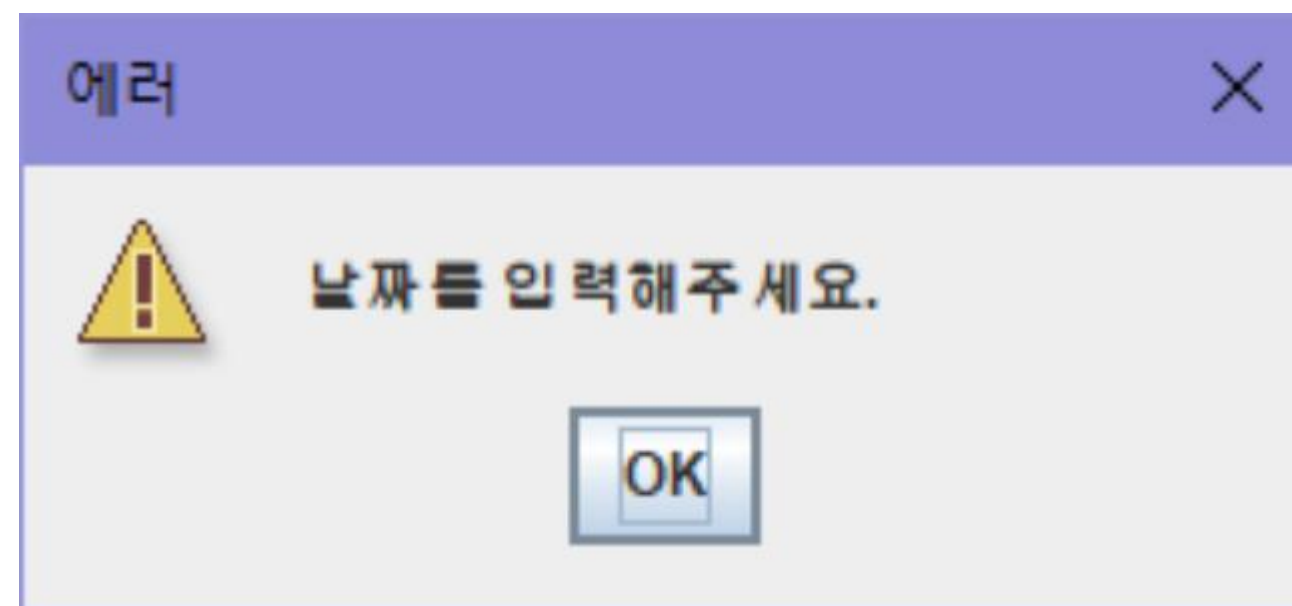


에러 경고창이 발생하면서 날짜를 입력하라는 문구 출력

오늘의 일기 쓰기 - # 날짜를 입력하지 않고 GET 버튼 클릭하였을 때



GET 버튼 클릭!

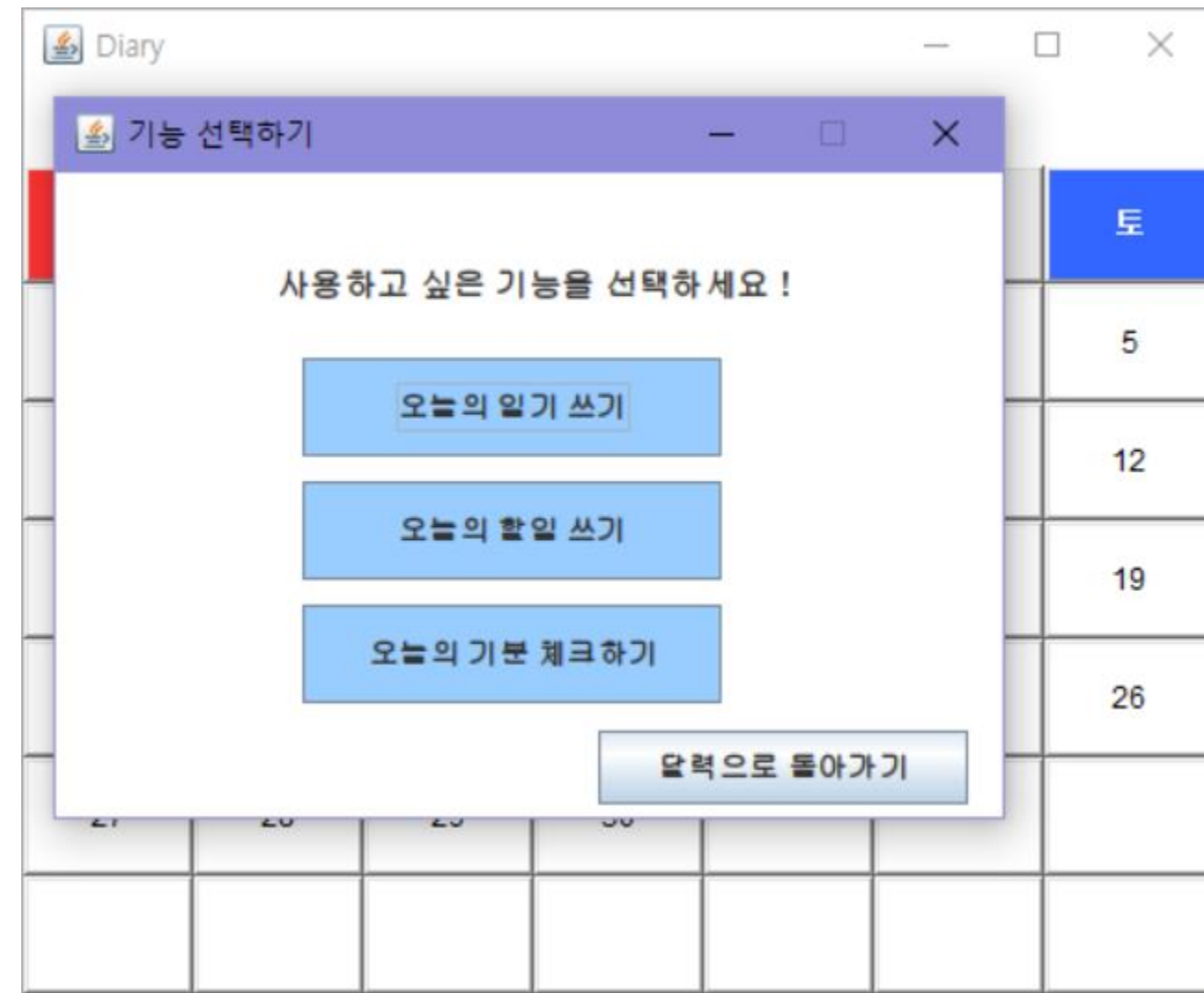


에러 경고창이 발생하면서 날짜를 입력하라는 문구 출력

오늘의 일기 쓰기 - # 뒤로가기 버튼



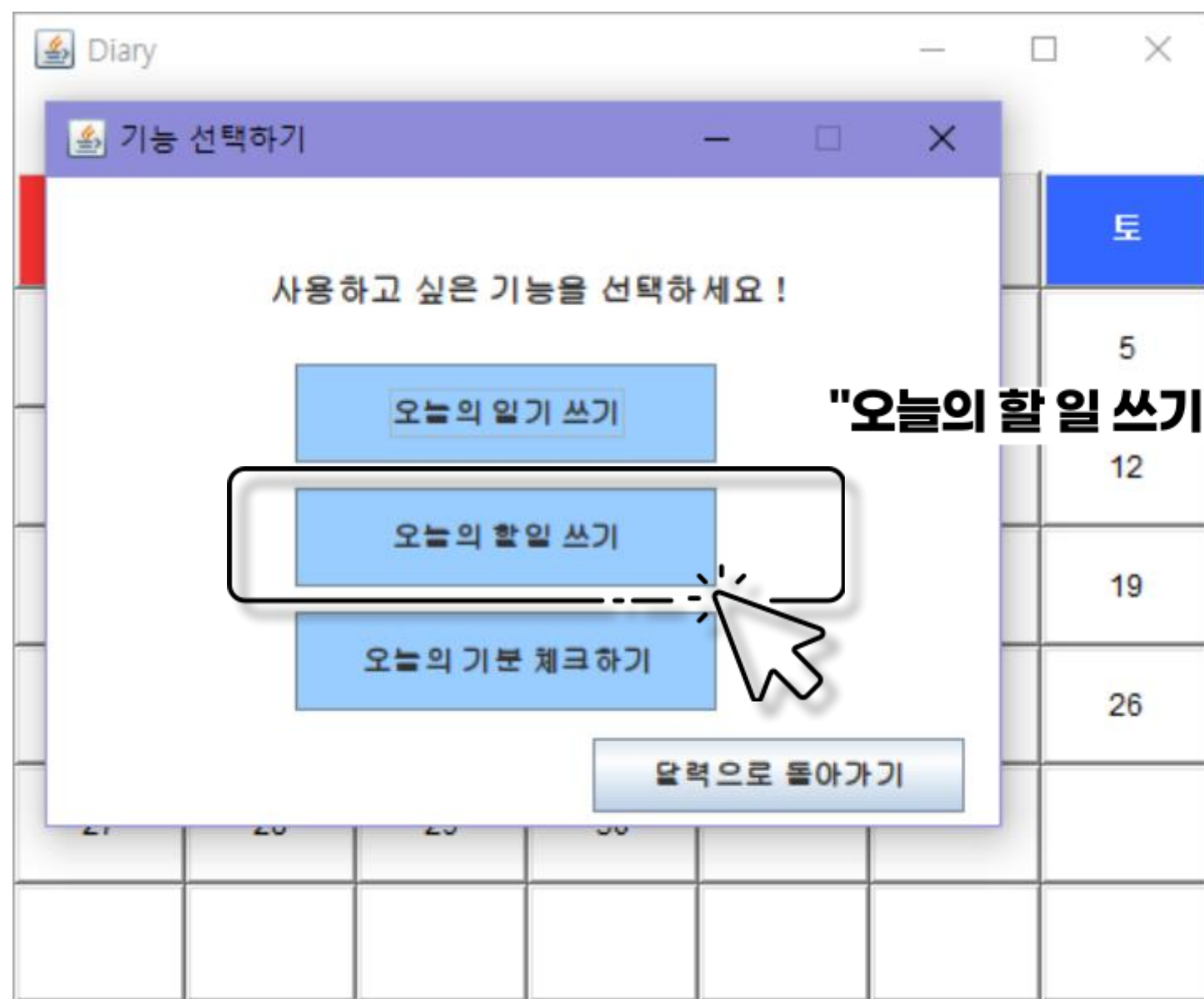
뒤로가기 버튼 클릭!



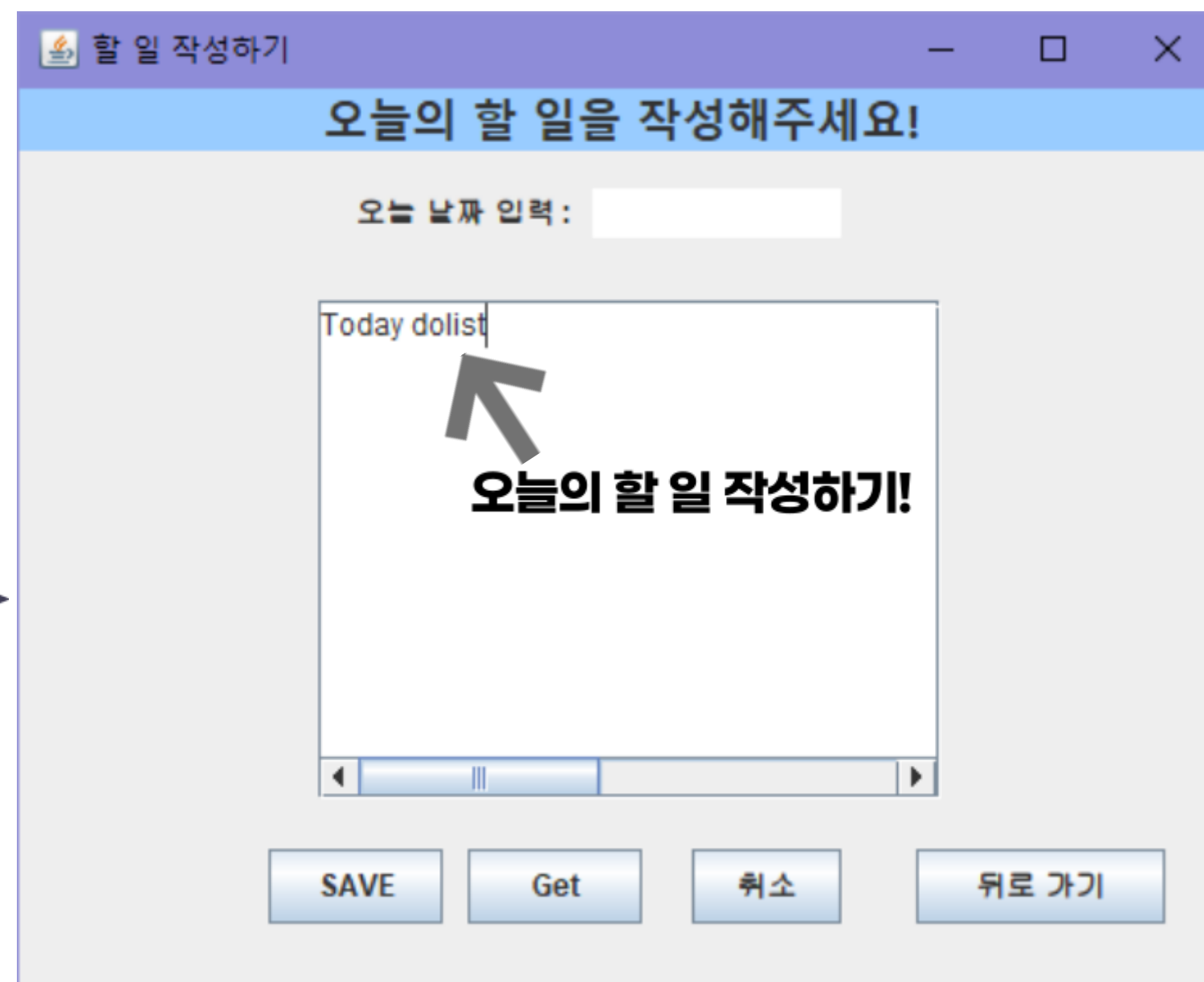
뒤로가기가 되면서 기능 선택 창이 뜨게 된다.

04

오늘의 할 일 쓰기 - # 할 일 작성하기

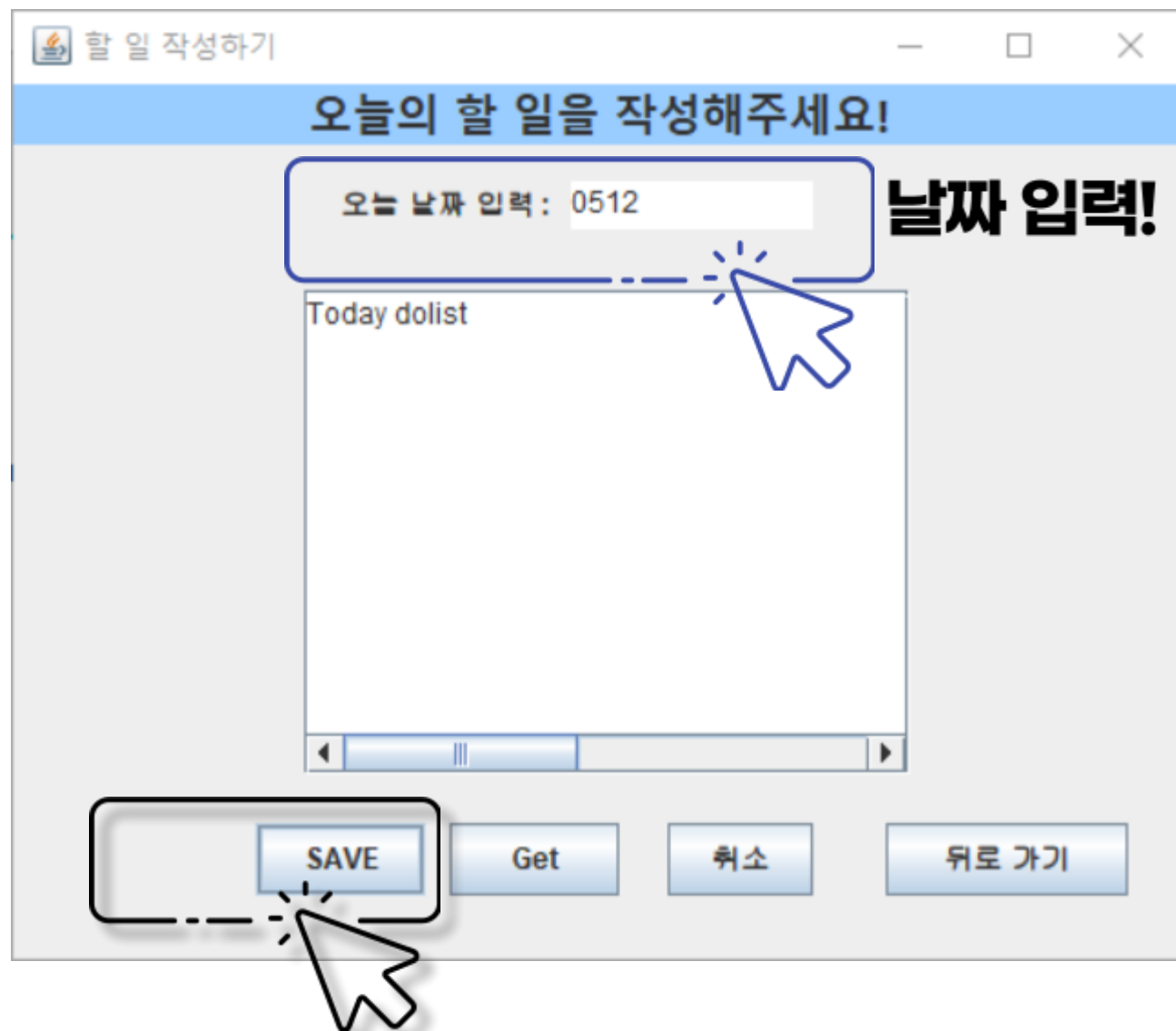


"오늘의 할 일 쓰기" 버튼 클릭!

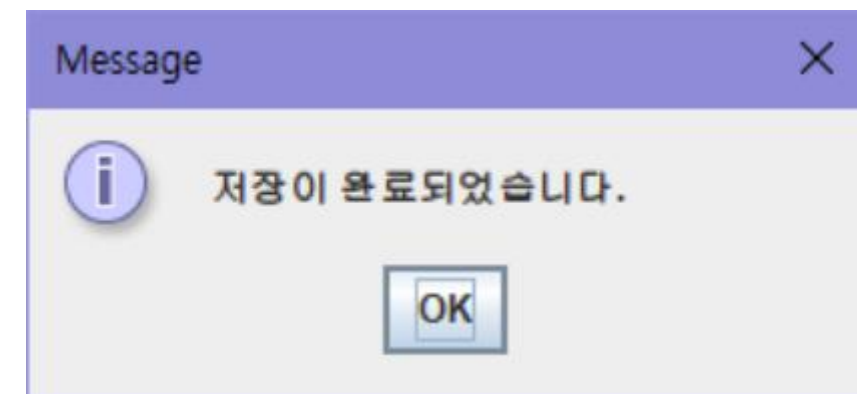


04

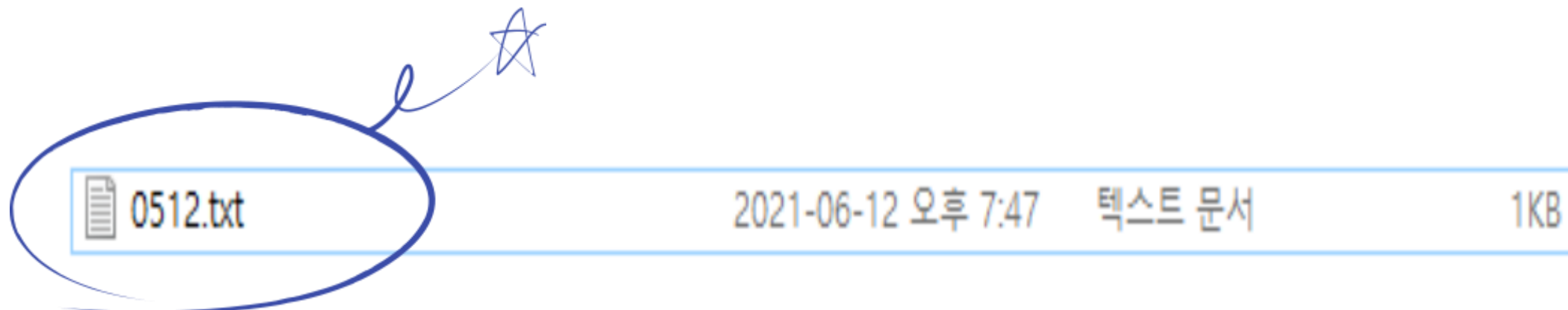
오늘의 할 일 쓰기 - # 할 일 작성한 글 해당 날짜에 저장하기



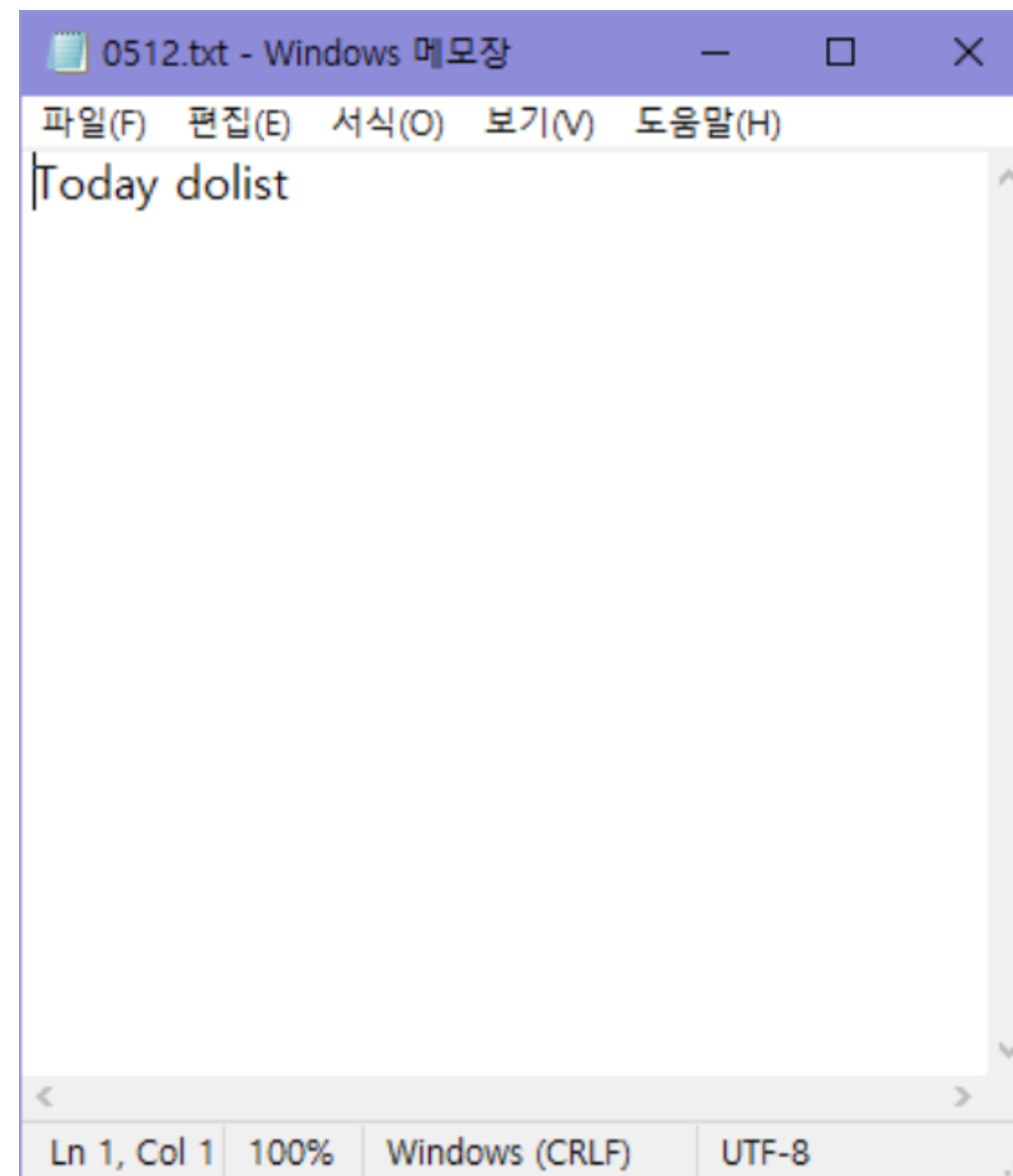
SAVE 버튼 클릭!



ListFolder 폴더가 생성됨!

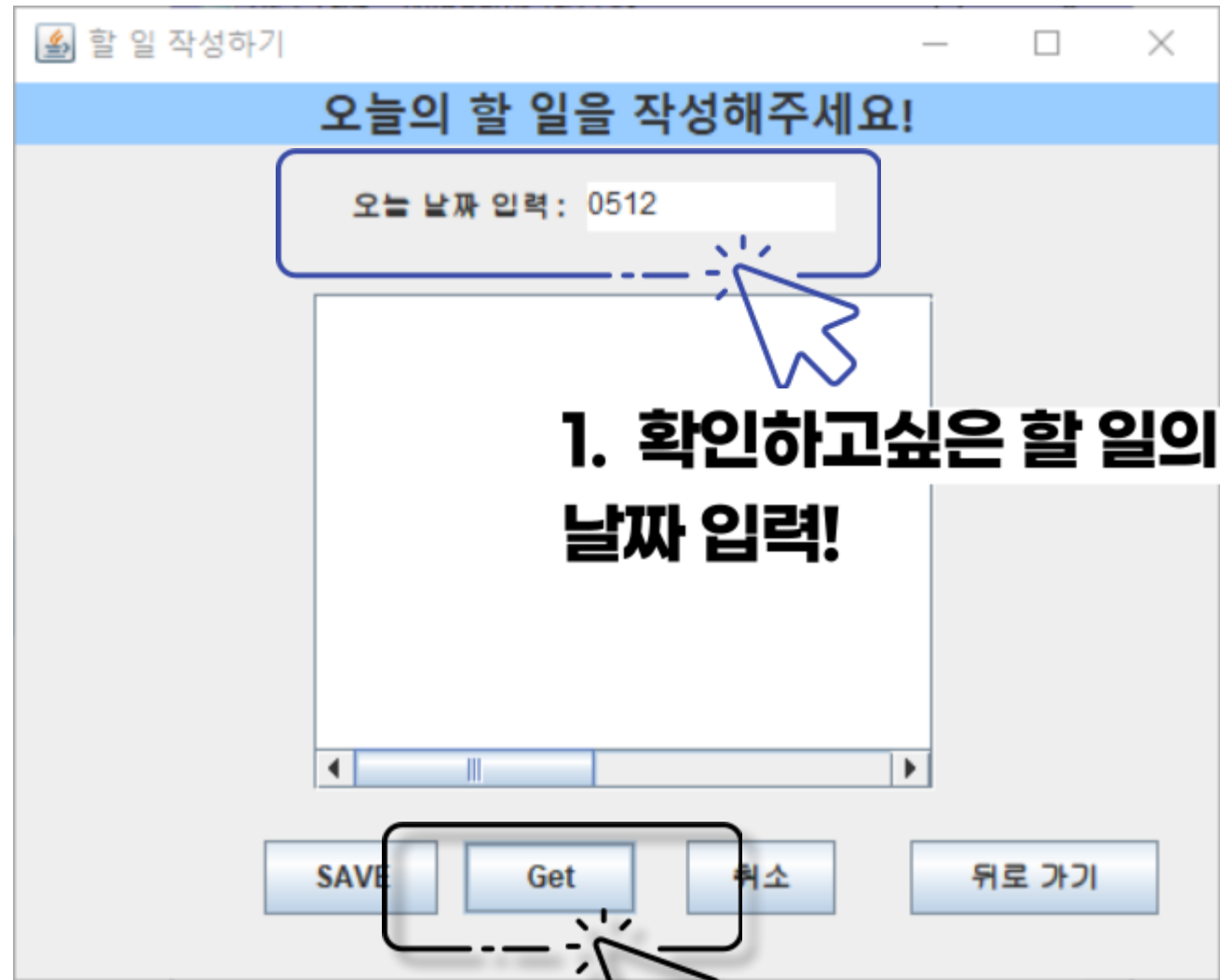


ListFolder 폴더안에 지정한 날짜명으로 텍스트 파일 생성



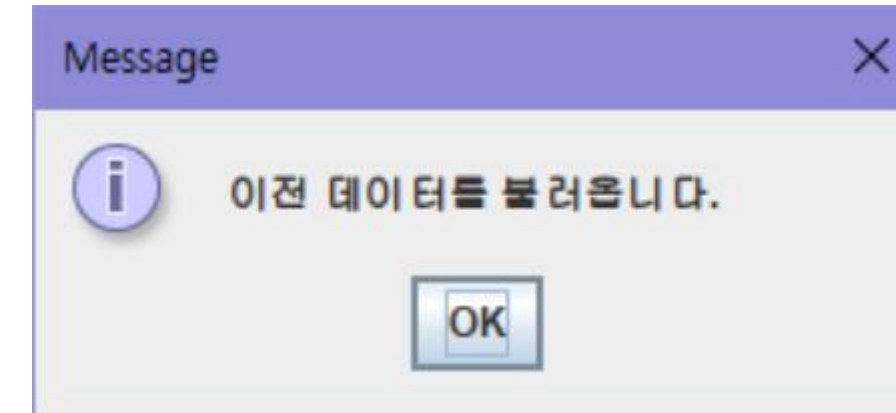
**텍스트 파일을 열면 작성한 할일이
저장되어진 것을 확인 할 수 있다.**

오늘의 할 일 쓰기 - # 작성한 할 일 다시 불러오기

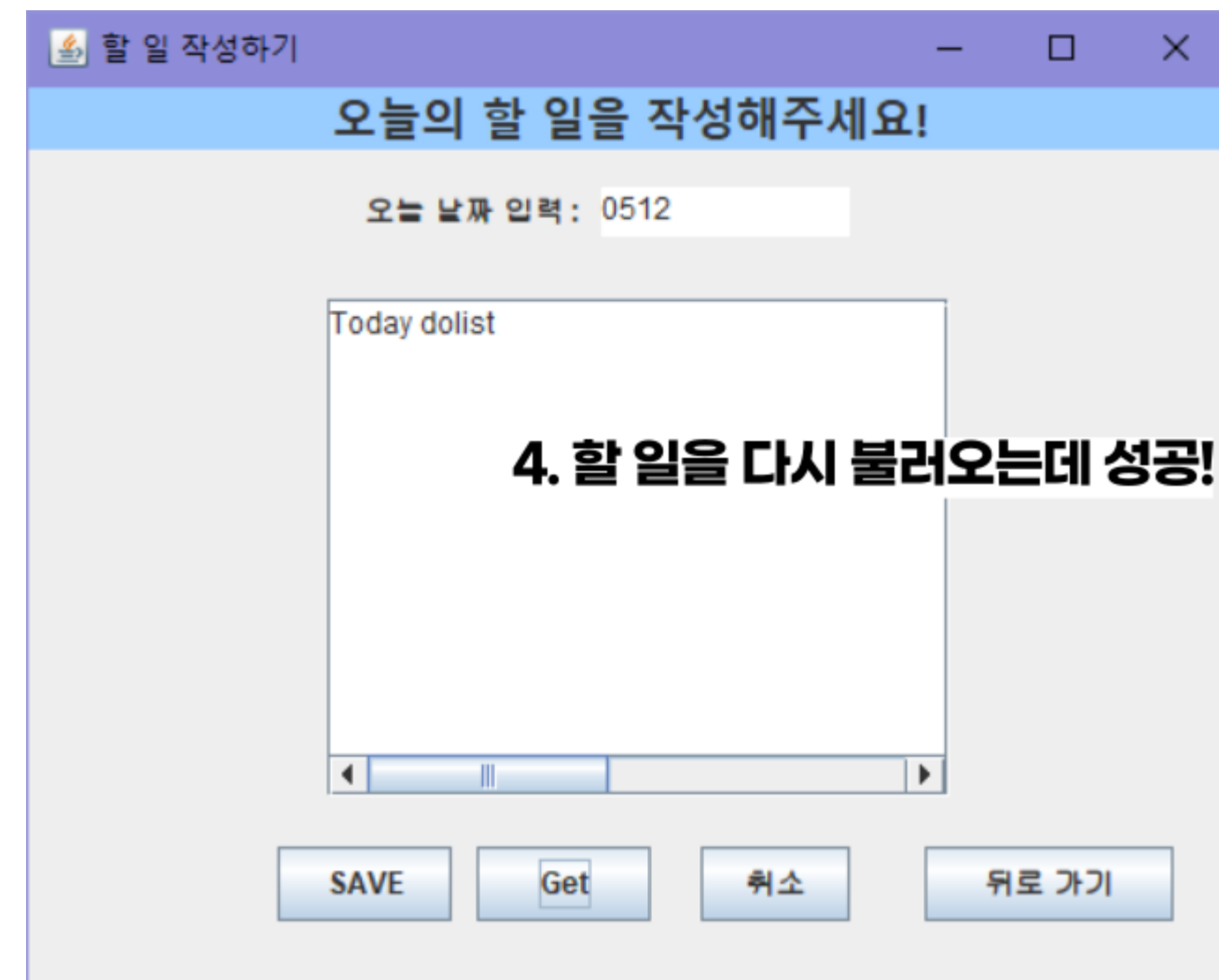


**1. 확인하고싶은 할 일의
날짜 입력!**

2. GET 버튼 클릭!



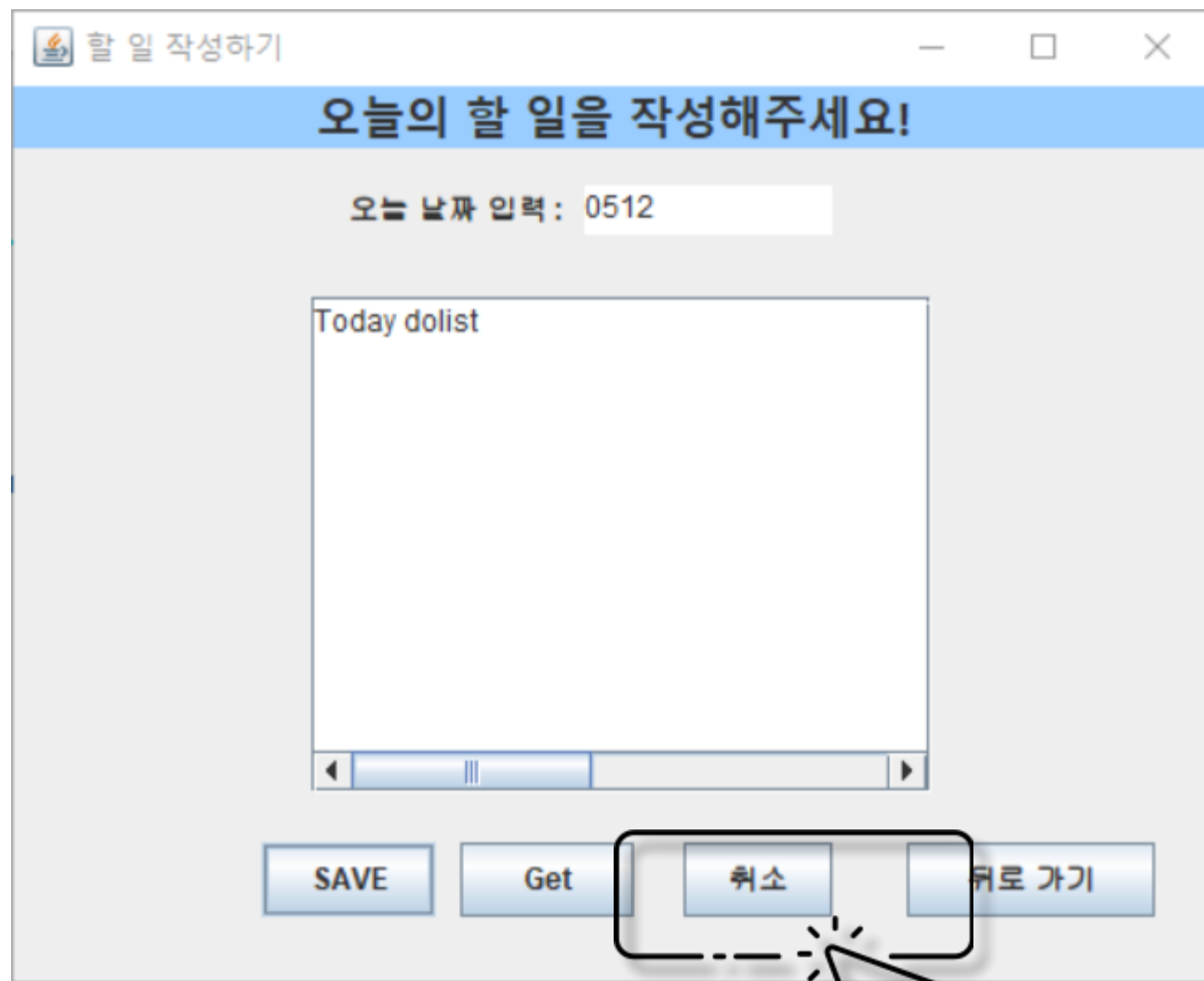
**3. 이전 데이터
문구 출력!**



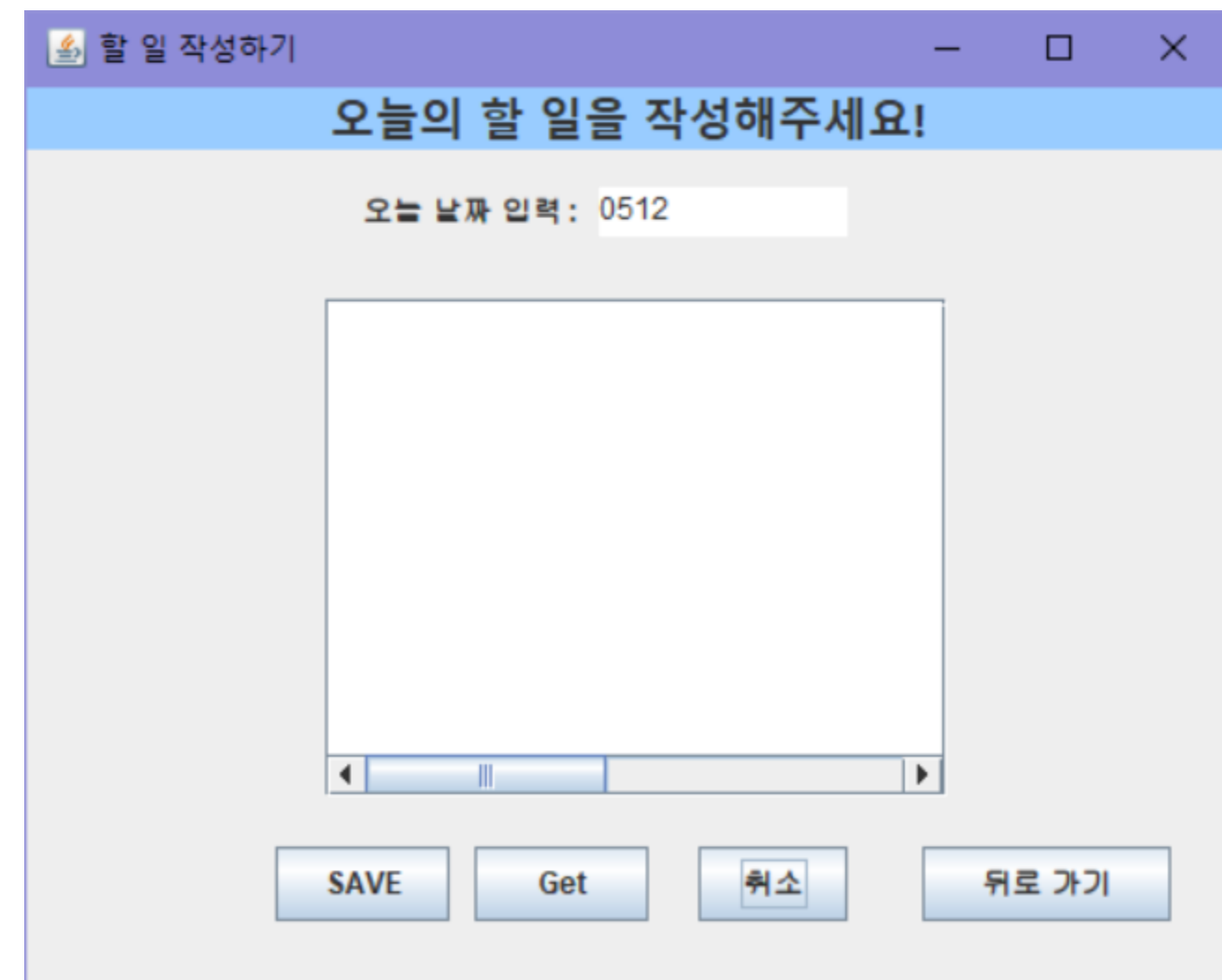
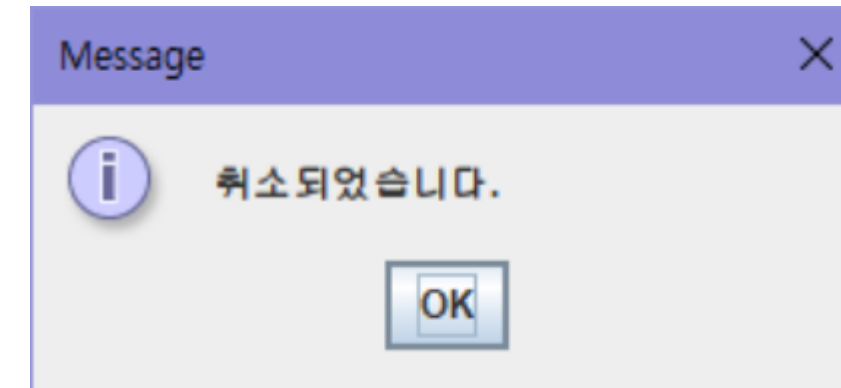
4. 할 일을 다시 불러오는데 성공!

04

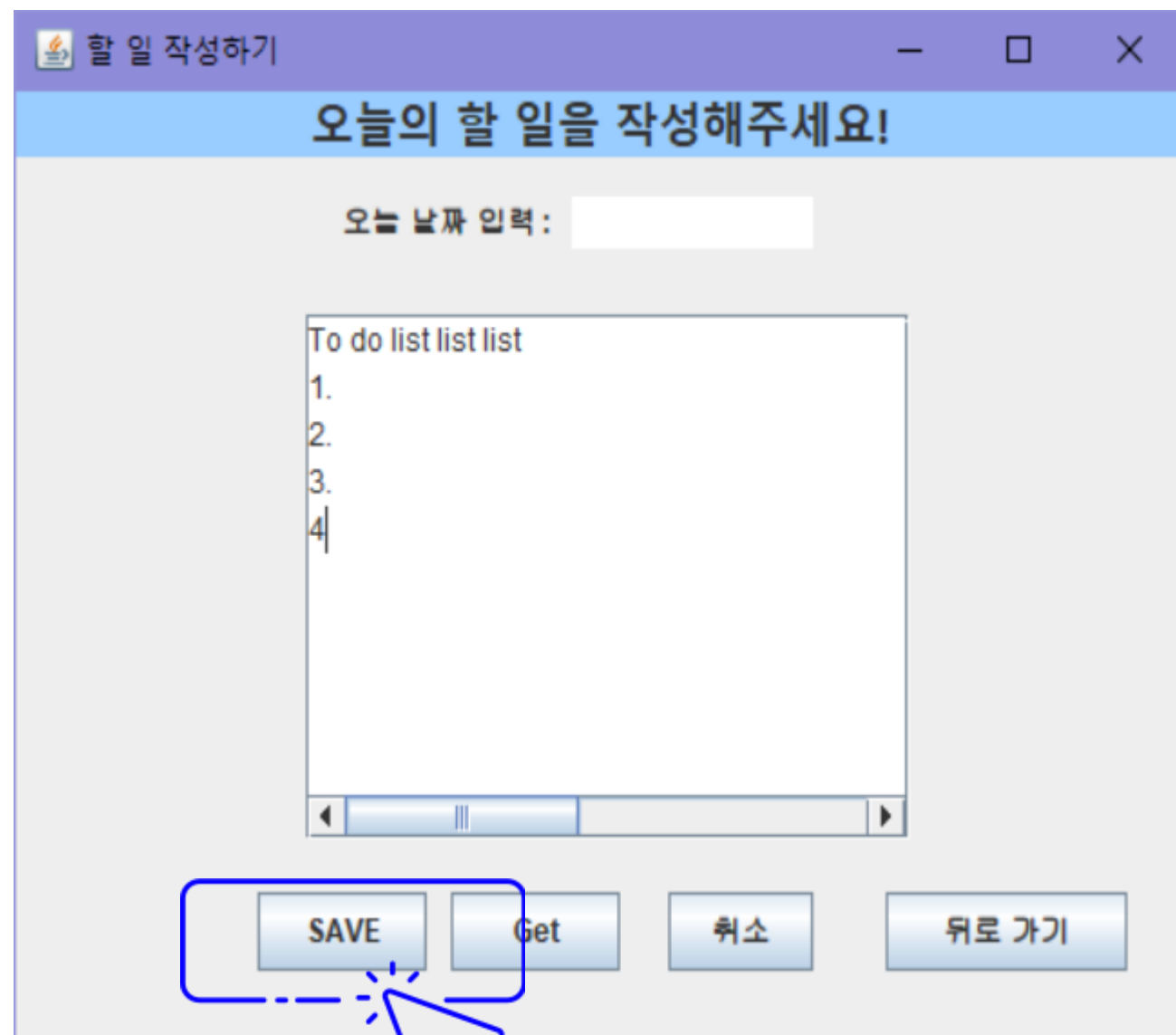
오늘의 할 일 쓰기 - # 취소버튼



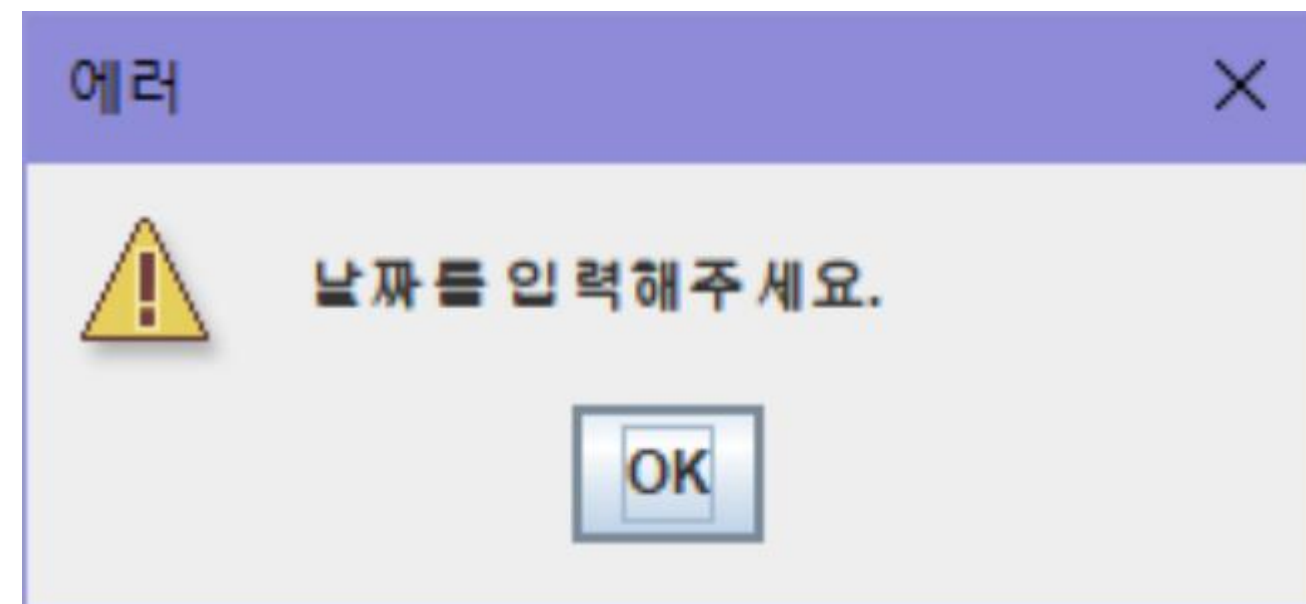
취소 버튼 클릭!



오늘의 할 일 쓰기 - # 날짜를 입력하지 않고 SAVE 버튼 클릭하였을 때

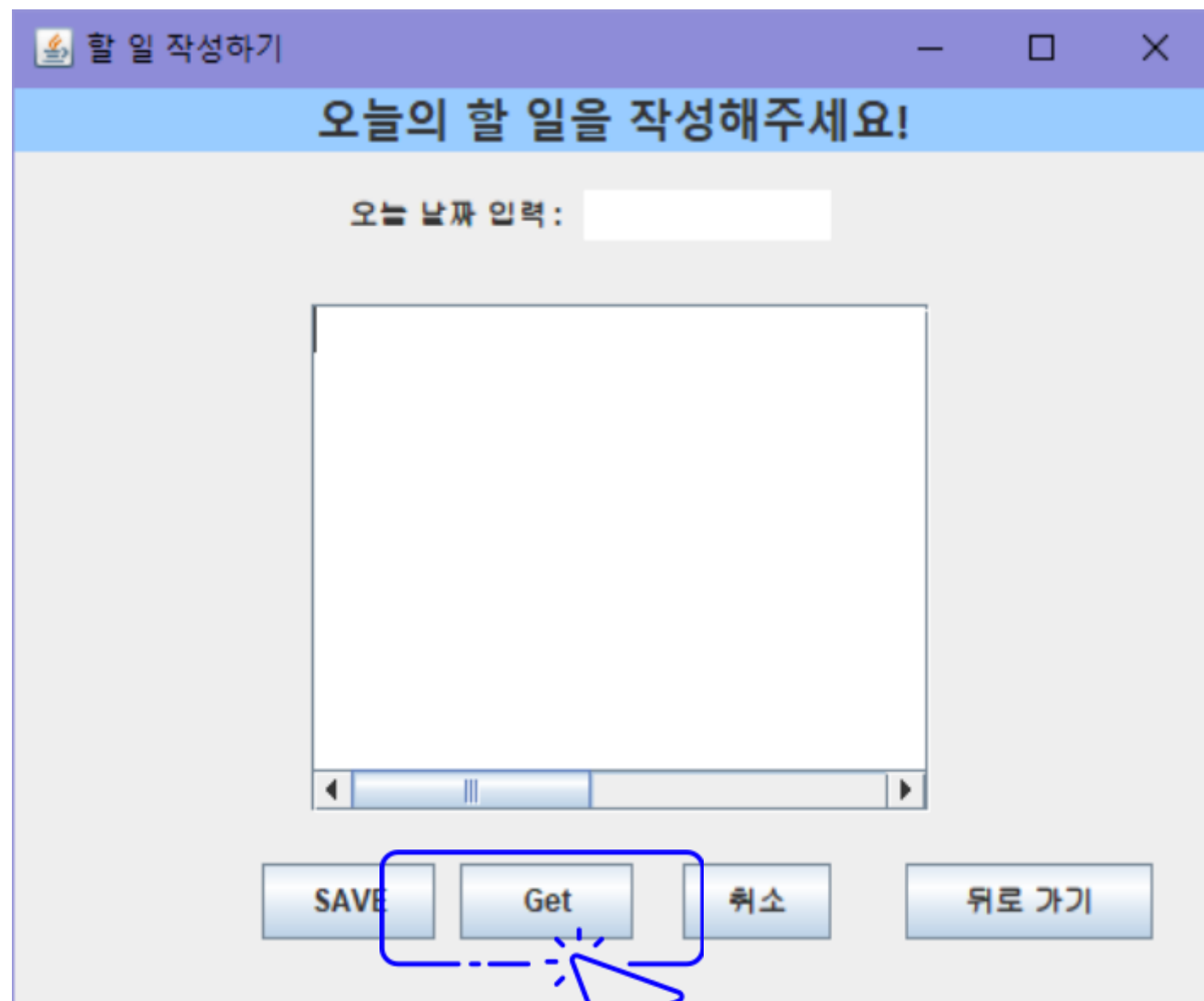


글을 쓰고 SAVE 버튼 클릭!

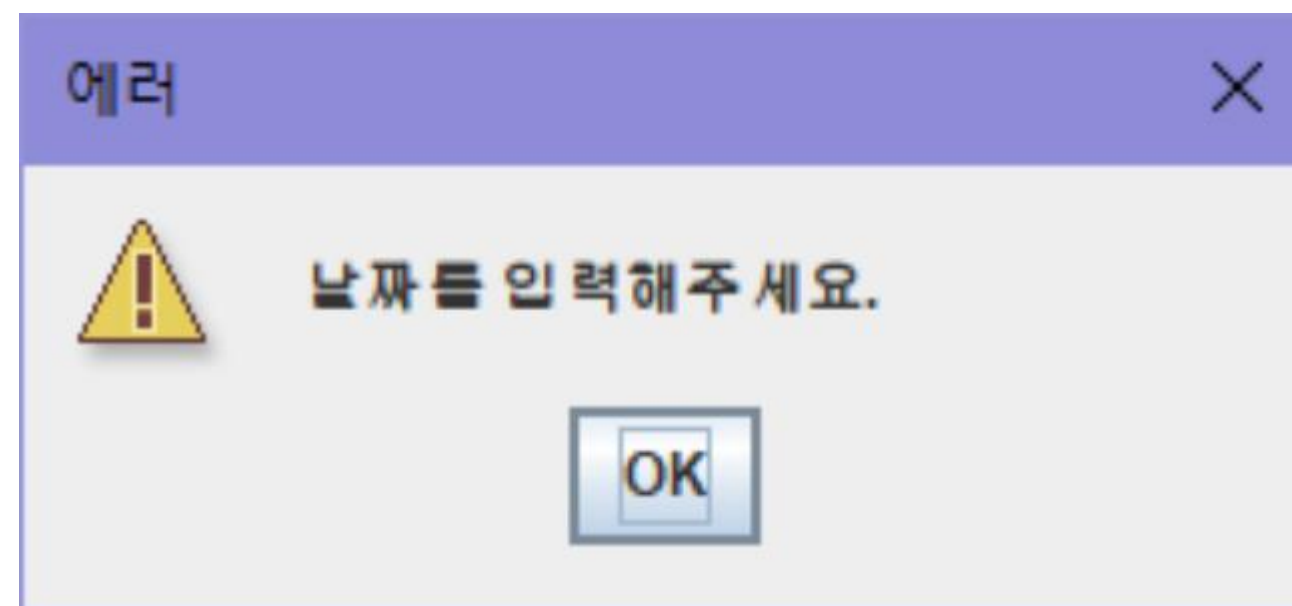


에러 경고창이 발생하면서 날짜를 입력하라는 문구 출력

오늘의 할 일 쓰기 - # 날짜를 입력하지 않고 Get 버튼 클릭하였을 때



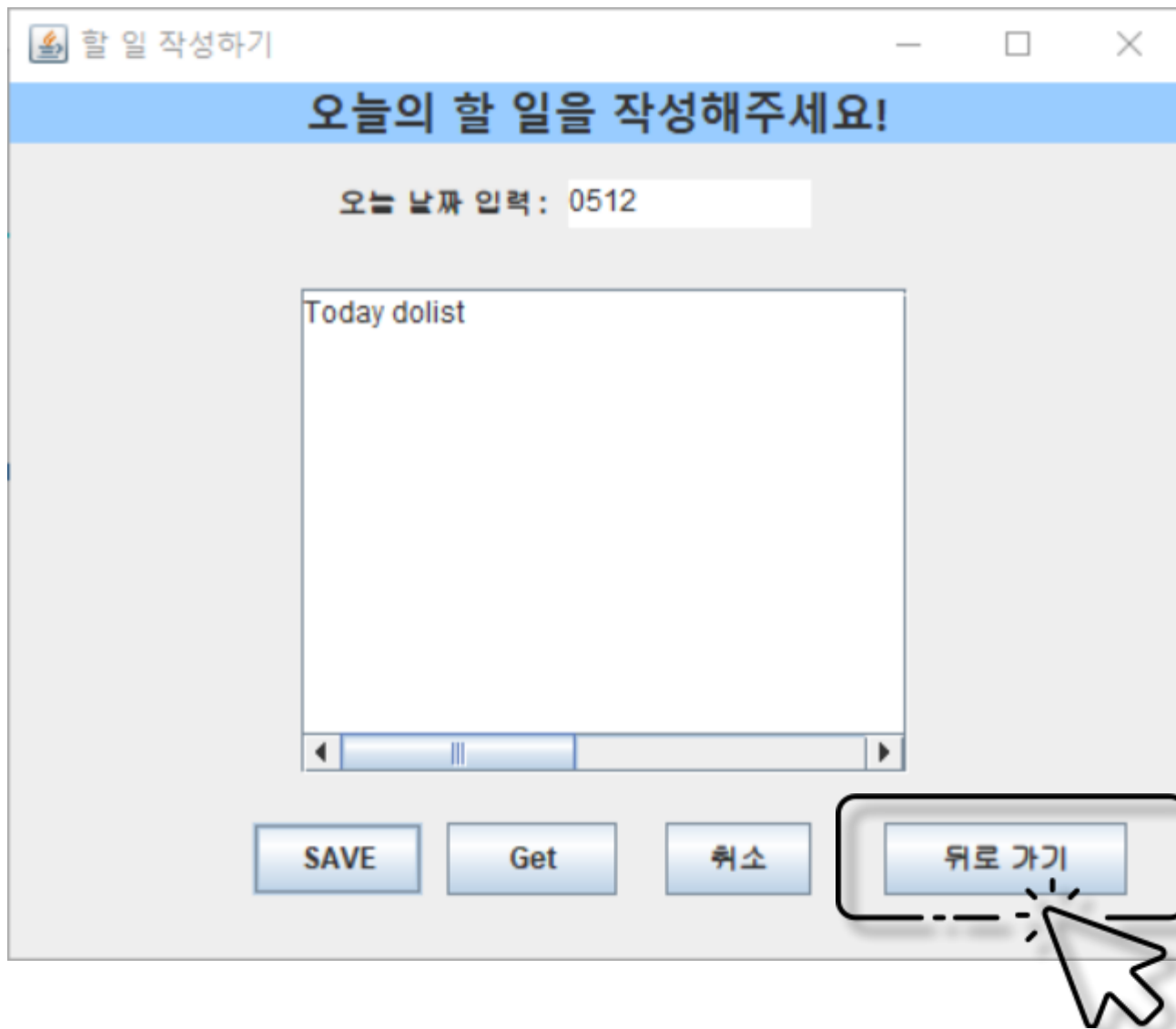
Get 버튼 클릭!



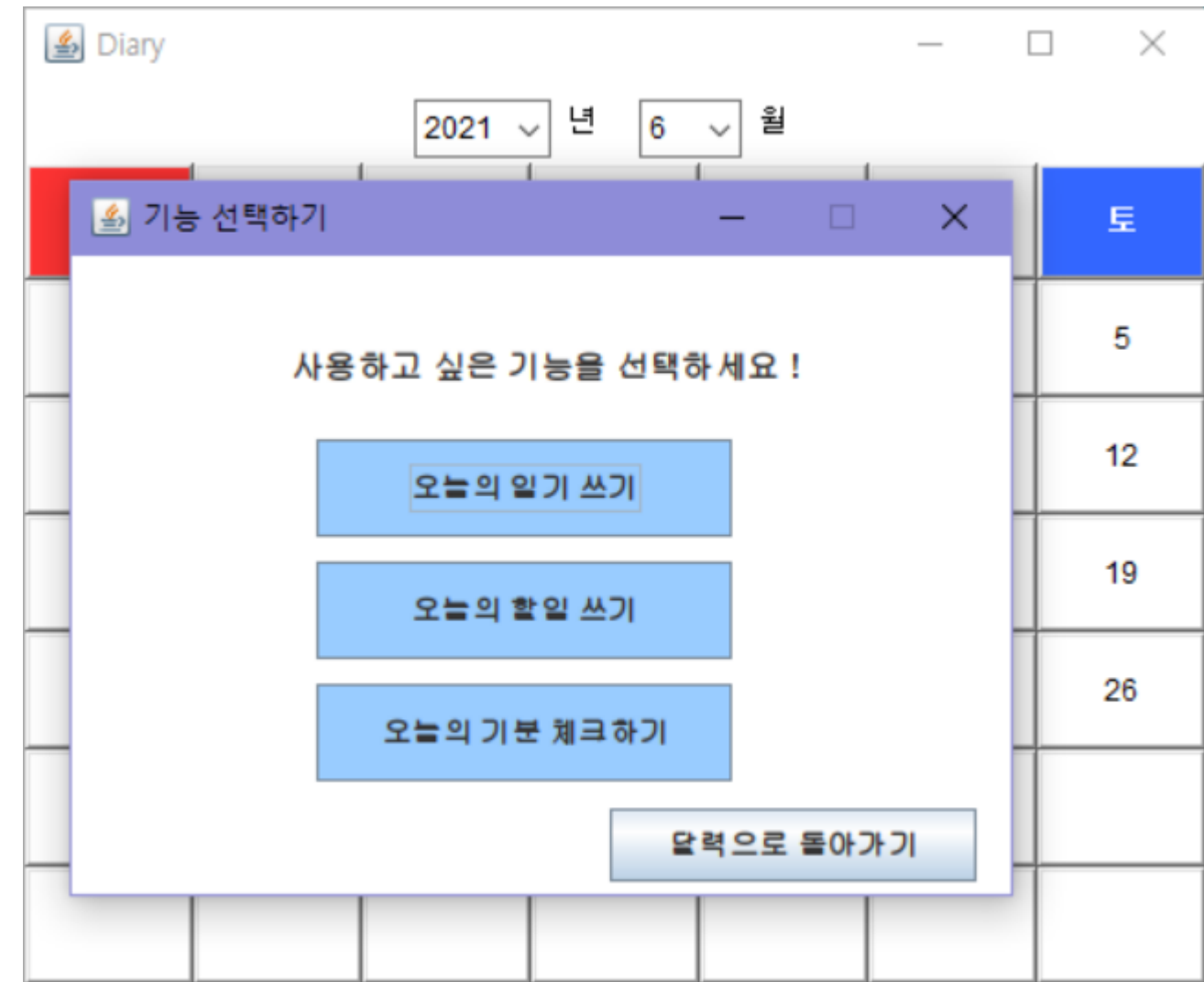
에러 경고창이 발생하면서 날짜를 입력하라는 문구 출력

04

오늘의 할 일 쓰기 - # 뒤로가기 버튼

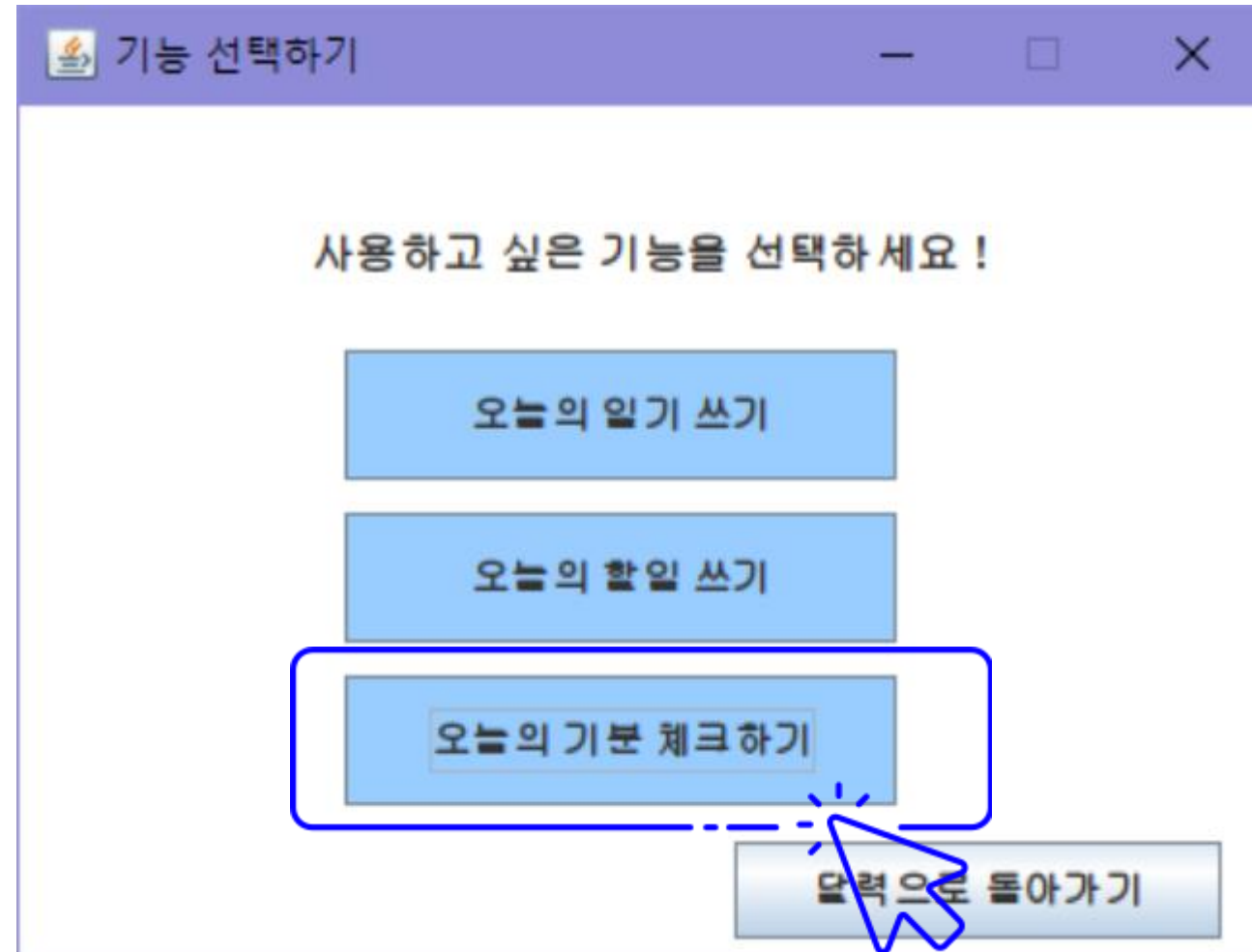


뒤로가기 버튼 클릭!

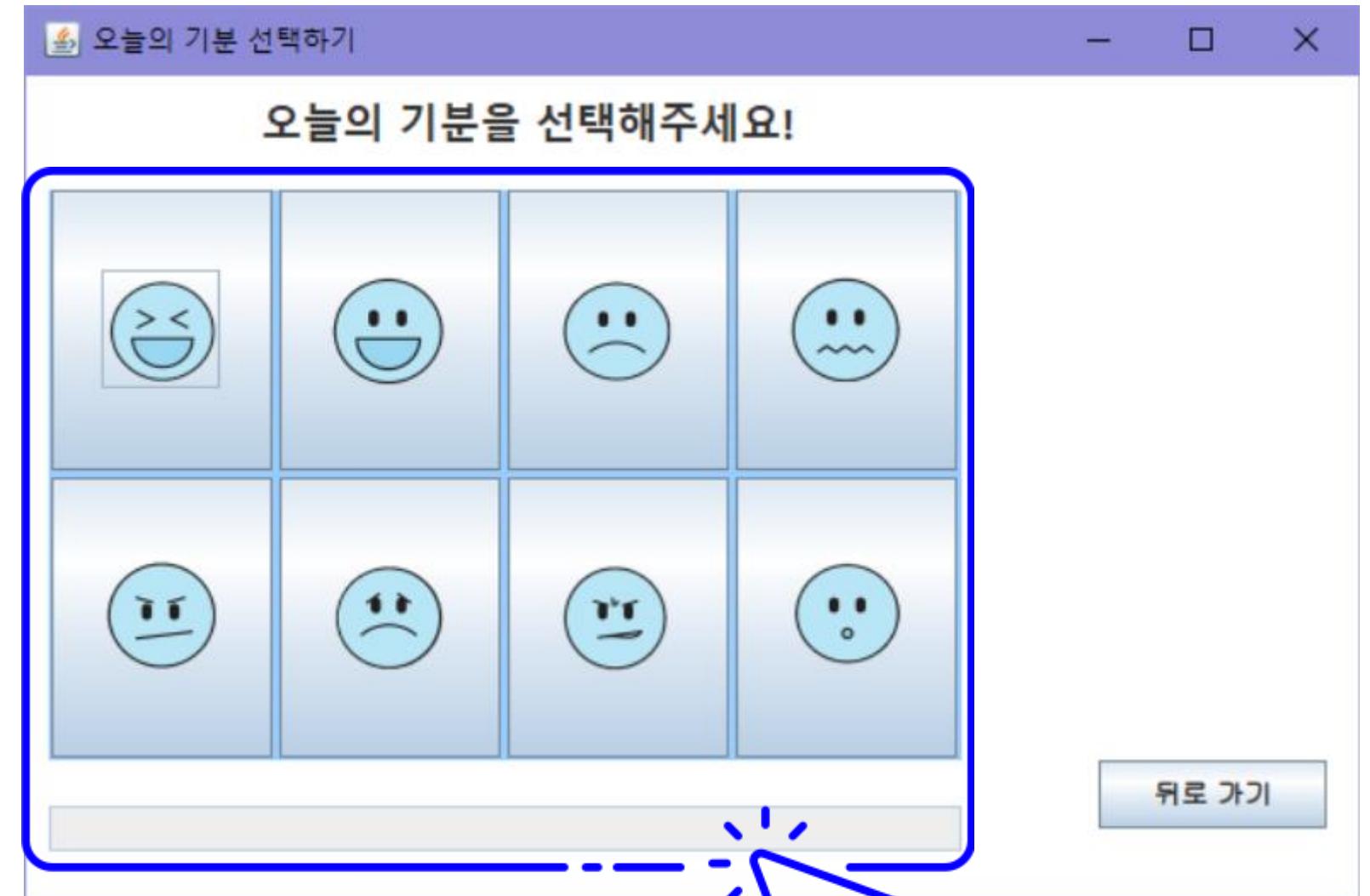


기능선택 창으로 이동

오늘의 기분 체크하기 - # 오늘의 기분 선택하기



"오늘의 기분 체크하기" 버튼 클릭!



8개의 기분 중 오늘의 기분 선택하기

오늘의 기분 체크하기 - # 오늘의 기분 선택하기



기분 버튼을 클릭하면 버튼 당 다른 기분을 출력!

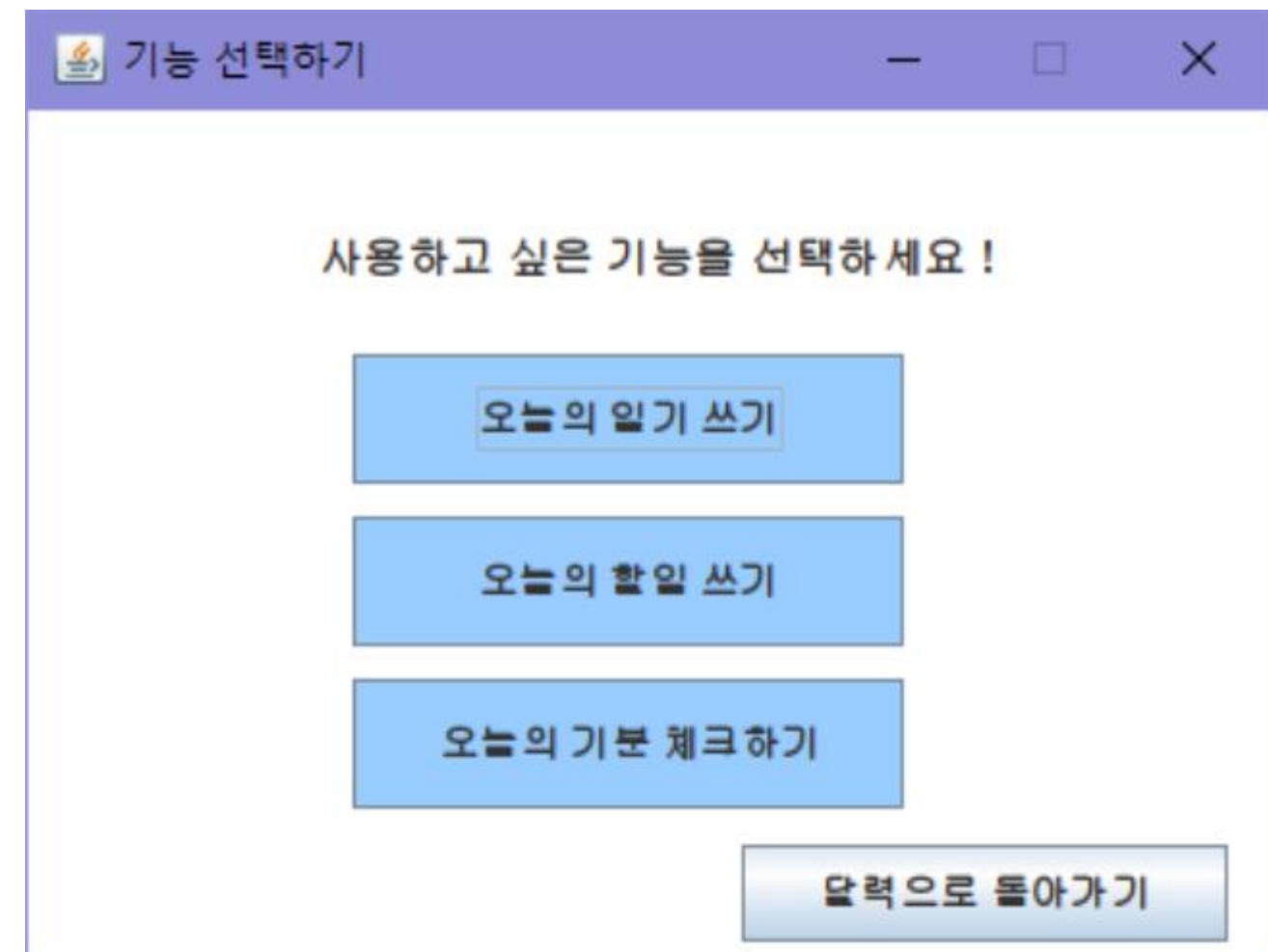


버튼을 클릭할 때마다 선택완료 아이콘이 계속해서 생성!

오늘의 기분 체크하기 - # 뒤로가기 버튼



뒤로가기 버튼 클릭!



다시 기능 선택 창으로 돌아간다.

03

사용한 클래스

Main.java

Main.java에서는 액션리스너를 활용해서 버튼누르면 다이얼로그 메시지를 띄우도록 했다. 이때, `show_MessageBoxDialog` 메소드를 사용했다.

Calendar.java

Calendar.java에서는 배열 만들어서 버튼 붙였고 `Font`와 `Color`로 색도 바꿨고 `getInstance` 메소드로 현재 날짜 및 시간 구해왔다.

PICTURE.java

PICTURE.java - `mkdir()`로 파일 폴더 생성, `-getText()`로 입력된 텍스트 얻어옴, 파일 읽고 쓸 때 `try-catch`문으로 예외 처리, `BufferedReader()` 이용하였다.

Main.java

```
46 background.add(p);
47
48 b2.addActionListener(new ActionListener() {
49     public void actionPerformed(ActionEvent e) {
50         tf.setText("");
51         pf.setText("");
52         JOptionPane.showMessageDialog(null, "프로그램을 종료합니다.");
53         setVisible(false);
54     }
55 });
56
57 b1.addActionListener(new ActionListener() {
58     public void actionPerformed(ActionEvent e) {
59         Calendar cl = new Calendar();
60
61         Dimension frameSize = cl.getSize();
62         Dimension windowSize = Toolkit.getDefaultToolkit().getScreenSize();
63         cl.setLocation((windowSize.width - frameSize.width) / 2,
64             (windowSize.height - frameSize.height) / 2);
65         setVisible(false);
66         JOptionPane.showMessageDialog(null, "다이어리 프로그램을 실행합니다.");
67     }
68 });
```

05

새로 가져온 클래스

calendar.java
todolist.java

picture.java
feeling.java

04

프로그램의 제작과정

프로그램 제작과정 - 프로그램 기획 및 제작

1. 줌 회의를 통해서 각자 프로그램 기획 하나씩 짜온 후, 가장 찬성 의견이 많은 프로그램 선택!

2. 각자 역할을 나누어 프로그램 구상 및 코드 짜서 줌 회의를 통해 다시 의견 조정!
- 사다리 타기를 통해 공정하게 진행!



3. 각자 맡은 기능을 구현하며 의견 조정하면서 서로 맞춰가며 코드 구현!

4. 프로그램의 전반적인 코드들과 기능들을 직접 같이 만나서 프로그램 코딩을 진행!

프로그램 제작과정 - # 함께한 노력의 인증



1번째 만남



2번째 만남



3번째 만남

그 외에도 정말 자주 만나서 프로그램 오류를 해결하고 기능을 구현하면서 그 결과, 성공적으로 프로그램을 제작할 수 있었습니다.

05

팀플 참여도와 역할

팀플 참여도와 역할

<팀플 참여도>

2012354 황서정 : 100%
2012187 황서경 : 100%
2010244 황영서 : 100%
1813151 정다연 : 100%

<맡은 코드>

login.java : 1813151 정다연
Calendar.java, window.java: 2012354 황서정
feeling.java, ToDoList.java: 2012187 황서경
picture.java: 2010244 황영서
디자인, 연동, 이벤트, 기획, 기획피피티: 다 같이

<맡은 역할>

디자인, 연동, 이벤트, 기획, 기획피피티: 다 같이
발표: 황영서, 황서경
발표 대본작성: 정다연
발표 피피티 제작: 황서정