

(INTERMEDIATE) JAVA PROGRAMMING

Sound / JAR
Chapter 14

JAVA: SOUND

자바의 오디오 API

- 자바의 오디오 API로 다룰 수 있는 오디오 종류
 - 디지털 오디오(Digital Audio)
 - 음악, 목소리 등의 아날로그 데이터를 디지털로 만든 오디오
 - WAV, AU, AIFF, AIFC
 - 미디(MIDI: Music Instrument Digital Interface) 데이터
 - 피아노, 바이올린 등 악기 소리와 높이 등을 표현하는 오디오
 - 미디 장치에서 연주

디지털 오디오 재생 과정

1. 오디오 클립 만들기 : Clip 인터페이스 이용

```
Clip clip = AudioSystem.getClip();
```

2. 오디오 클립에 오디오 스트림 연결.
오디오 파일로부터 오디오를 읽을
스트림 객체 생성

```
File audioFile = new File("애국가.wav");  
AudioInputStream audioStream =  
    AudioSystem.getAudioInputStream(audioFile);
```

오디오 클립에 오디오 스트림 연결

```
clip.open(audioStream);
```

3. 오디오 재생
Clip 클래스의 start() 메소드 호출

```
clip.start(); // 오디오 재생 시작
```

void open(AudioInputStream stream)

오디오 클립은 stream에 존재하는 오디오의 형식과 데이터를 인식하고, 재생할 수 있는 준비를 갖춘다.

void start()

현재 프레임 위치에서 오디오 클립을 재생한다. 처음 실행될 때 프레임 위치는 0

void stop()

재생중인 오디오 클립의 재생을 중단한다.

void setFramePosition(int frames)

오디오 샘플 내에 재생할 프레임의 위치를 지정한다. 프레임의 시작 번호는 0이다.

void loop(int count)

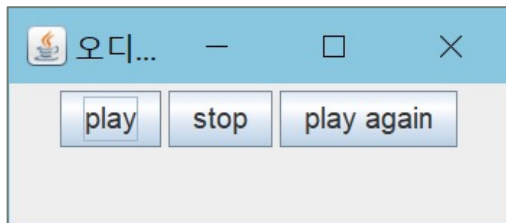
현재 프레임의 위치에서 시작하여 count만큼 반복 재생한다. count 값이 LOOP_CONTINUOUSLY 이면 무한 반복한다.

void close()

오디오 클립의 모든 자원을 반환한다.

예제 14-12 : 오디오 재생/중지

play, stop, play again의 3 버튼을 가진 응용프로그램을 작성하고, 애국가.wav 파일을 연주하라.
애국가.wav는 프로젝트 폴더 밑의 audio 폴더에 있어야 한다.



```
import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class AudioEx extends JFrame {
    private JButton btns[] = { new JButton("play"),
                               new JButton("stop"), new JButton("play again")};
    private Clip clip;
    public AudioEx() {
        setTitle("오디오 제어");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,200);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        MyActionListener al = new MyActionListener();
        for(int i=0; i<btns.length; i++) {
            c.add(btns[i]);
            btns[i].addActionListener(al);
        }
        setVisible(true);
        loadAudio("audio/애국가1절.wav");
    }
}
```

```
private void loadAudio(String pathName) {
    try {
        clip = AudioSystem.getClip(); // 비어있는 오디오 클립 만들기
        File audioFile = new File(pathName); // 오디오 파일의 경로명
        AudioInputStream audioStream =
            AudioSystem.getAudioInputStream(audioFile);
        clip.open(audioStream); // 재생할 오디오 스트림 열기
    }
    catch (LineUnavailableException e) { e.printStackTrace(); }
    catch (UnsupportedAudioFileException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

class MyActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        switch(e.getActionCommand()) {
            case "play": clip.start(); // 오디오 재생 시작
                        break;
            case "stop": clip.stop(); // 오디오 재생 중단
                        break;
            case "play again": clip.setFramePosition(0);
                             clip.start();
                             break;
        }
    }
}

public static void main(String [] args) {
    new AudioEx();
}
```

오디오 클립에 Line 이벤트 처리

- Line 이벤트
 - 오디오 재생 과정에서 오디오 클립(Clip 객체)에 발생하는 이벤트
 - 오디오가 재생을 시작할 때
 - 재생이 중단되었을 때
 - 오디오 클립이 닫혔을 때
- LineListener 인터페이스

```
public void update(LineEvent e)
```

오디오가 재생을 시작할 때, 재생이 중단되었을 때, 오디오 클립이 닫혔을 때 호출된다.

- LineEvent 객체

```
Line getLine()
```

이벤트가 발생하는 오디오 클립 객체(Clip은 Line 인터페이스 상속받음)

```
Long getFramePosition()
```

오디오 내에 이벤트가 발생한 프레임 위치(0부터 시작)

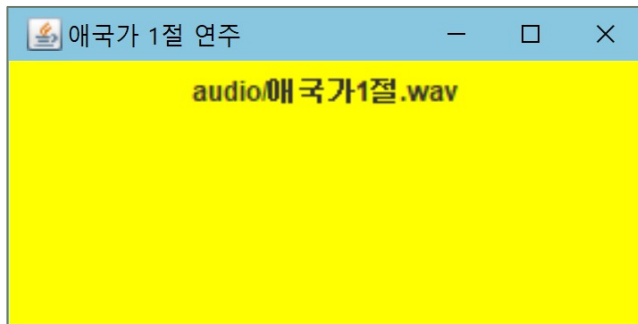
```
LineEvent.Type getType()
```

이벤트 종류로 다음 4가지 값 중 하나 리턴

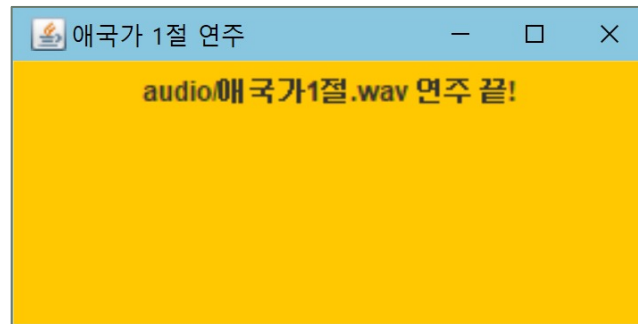
- LineEvent.Type.OPEN: 오디오 클립이 열릴 때(Clip의 open() 호출시)
- LineEvent.Type.START: 재생이 시작될 때(Clip의 start() 호출시)
- LineEvent.Type.STOP: 재생이 중단될 때(Clip의 stop()이나 끝까지 재생되었을 때)
- LineEvent.Type.CLOSE: 오디오 클립이 닫히고 모든 자원이 반환되었을 때(Clip의 close() 호출시)

예제 14-13 : 오디오 연주가 끝나면 콘텐츠팬의 배경색을 ORANGE 색으로 만들기

애국가1절.wav 파일을 재생시키고 연주가 끝나면 콘텐츠팬의 바탕색이 ORANGE 색으로 바뀌도록 하라.



애국가의 연주 중



애국가의 연주가 완전히 끝났을 때

예제 14-13 : 정답 코드

```
import javax.sound.sampled.*;
import javax.swing.*.*;
import java.awt.*.*;
import java.io.*.*;

public class LineEventEx extends JFrame {
    private Clip clip;
    private String song="audio/애국가1절.wav";
    private JLabel label=new JLabel(song);

    public LineEventEx() {
        setTitle("애국가 1절 연주");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();

        c.setLayout(new FlowLayout());
        c.setBackground(Color.YELLOW);
        c.add(label);

        setSize(300,150);
        setVisible(true);

        loadAudio(song);
    }
}
```

```
private void loadAudio(String pathName) {
    try {
        File audioFile = new File(pathName);
        final AudioInputStream audioStream =
            AudioSystem.getAudioInputStream(audioFile);
        clip = AudioSystem.getClip();

        clip.addLineListener(new LineListener() {
            public void update(LineEvent e) {
                if (e.getType() == LineEvent.Type.STOP) {
                    try {
                        getContentPane().setBackground(Color.ORANGE);
                        label.setText(song + " 연주 끝!");
                        audioStream.close();
                    } catch (IOException e1) {
                        e1.printStackTrace();
                    }
                }
            }
        });
        clip.open(audioStream);
        clip.start();
    }
    catch (LineUnavailableException e) { e.printStackTrace(); }
    catch (UnsupportedAudioFileException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

public static void main(String [] args) {
    new LineEventEx();
}
}
```


JAVA:

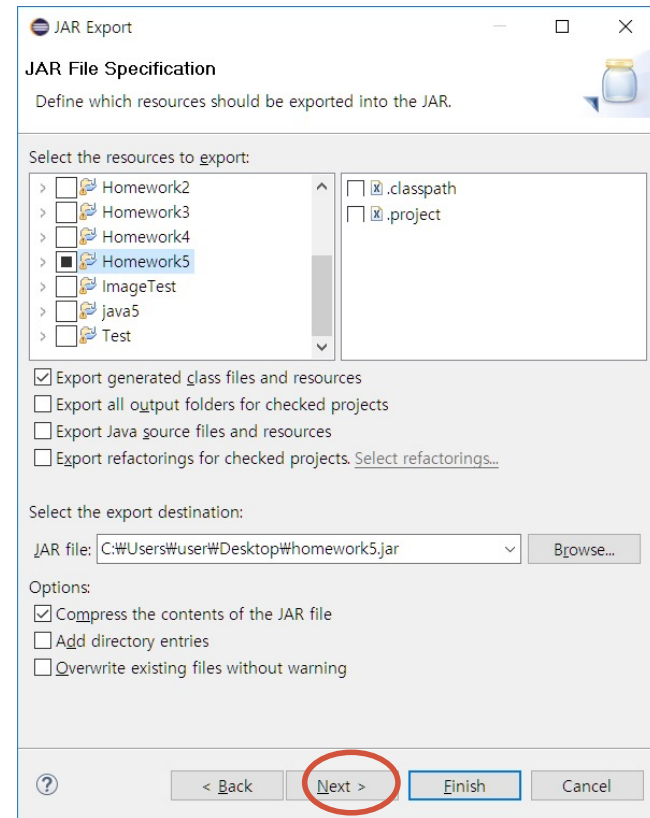
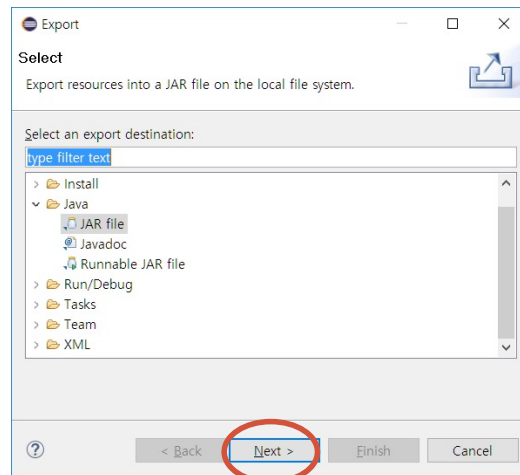
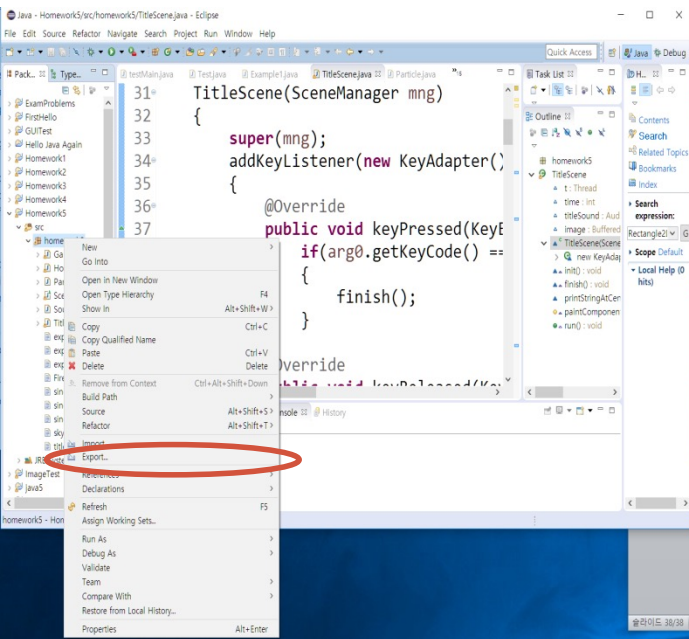
JAVA 파일 배포 (JAR)

JAR 파일

- Java **AR**chive File (JAR) :
 - 다양한 소스코드를 하나로 묶어서 압축하여 배포하는 용도
 - Byte Code 뿐만 아니라, 추가적인 파일(resource)도 포함 가능
 - 예) 사운드 파일, 그림 파일 등
- 만드는 법:
 - Eclipse 없이 만드는 법:
<https://docs.oracle.com/javase/tutorial/deployment/jar/>
 - Eclipse로 만드는 법:

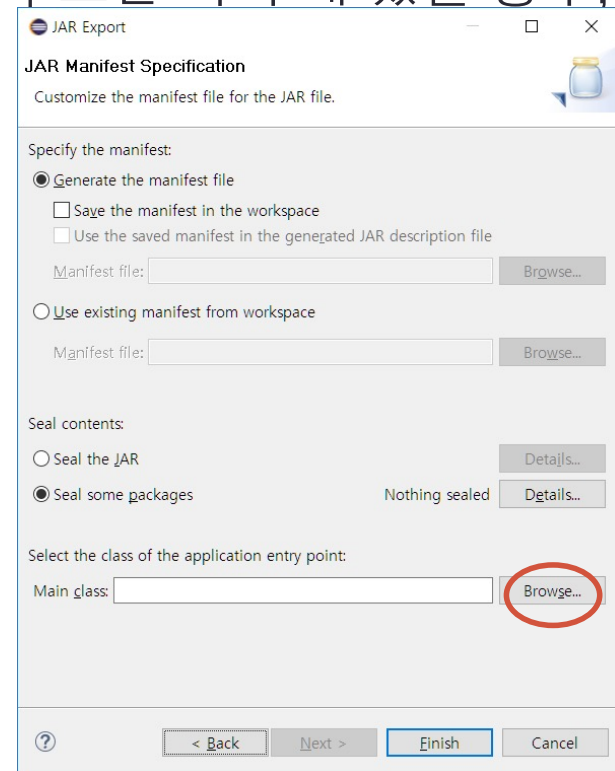
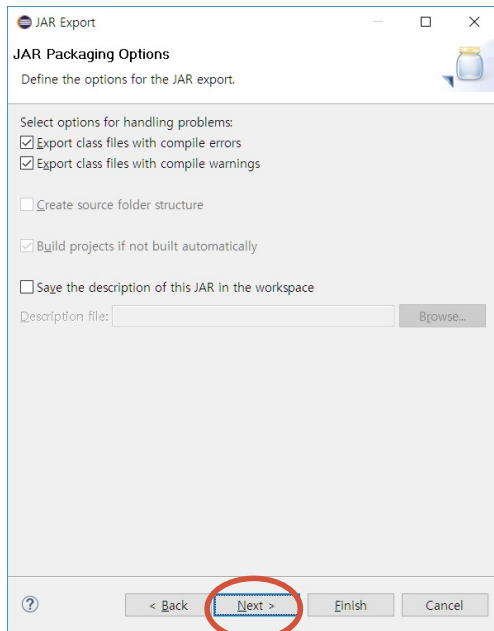
Eclipse를 통해 JAR 파일 만들기

- Package 이름에서 마우스 오른쪽 클릭 → export 선택
- JAR 선택 확인 후 next 클릭
- 내용 확인 후 next 클릭



Eclipse를 통해 JAR 파일 만들기(계속)

- 옵션 확인 후 next 클릭
- **(중요)** Manifest Specification 창에서 Main Class를 Browse 버튼을 눌러 선택
 - main 함수를 갖고 있는 class를 선택 또는 여러 개 있을 경우, 프로그램이 우선 실행할 class선택



참고1 : Image file 또는 Audio file

소스코드의 상대경로로 읽기

- src 폴더아래 java 파일이 위치한 곳에 image파일을 가져다 놓고, 다음과 같은 방법으로 이미지를 로드 할 수 있다.

```
URL url = getClass().getResource("Image.jpg");  
BufferedImage image = ImageIO.read(url);
```

<경로의 위치는 source code 위치를 기준으로 함>

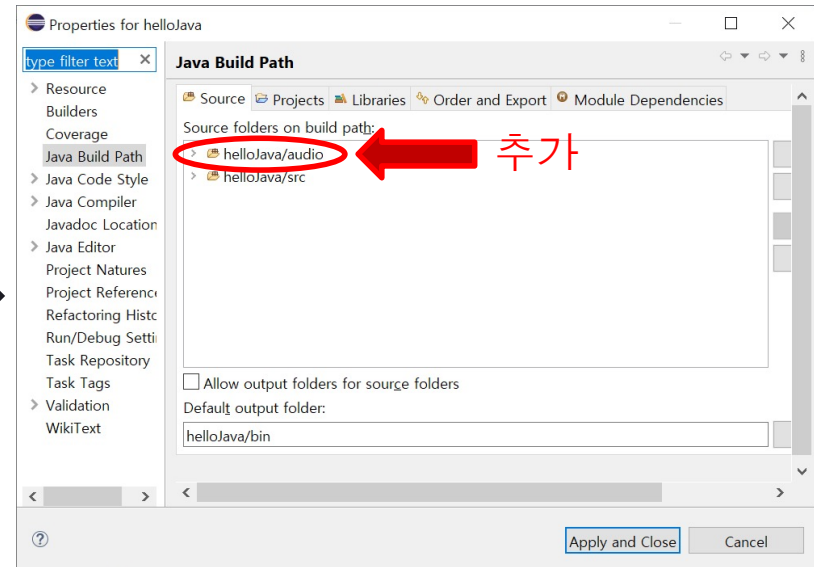
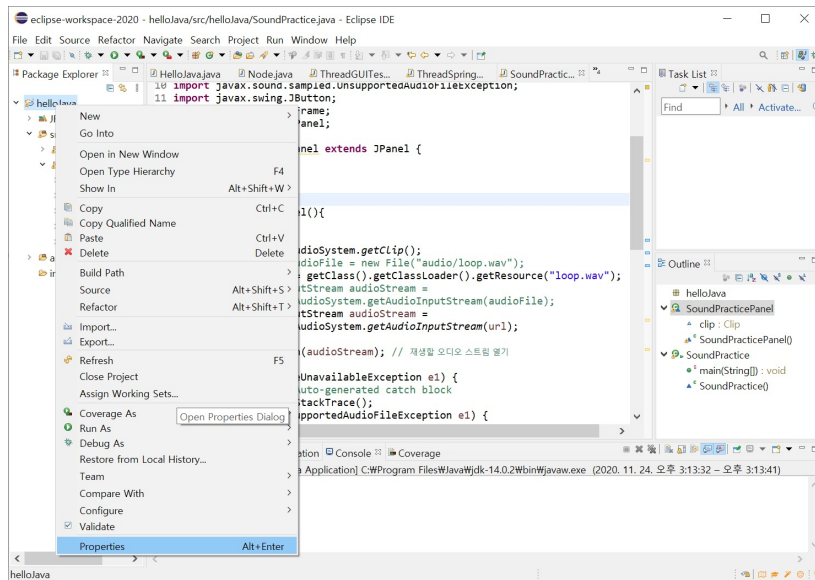


```
URL url = getClass().getResource("wave.wav");  
AudioInputStream audio = AudioSystem.getAudioInputStream(url);
```

숙제 제출시 반드시 url을 써서 소스코드 상대 경로로
입력할 것 (JAR 파일 제출 시 필수!!!)

참고2-1: Image file 또는 Audio file 세팅에서 리소스 경로 추가하기

- 프로젝트의 세팅에서 Build Path의 source에서 폴더를 추가



참고2-2: Image file 또는 Audio file 세팅에서 리소스 경로 추가하기

- URL을 사용하여 추가된 폴더의 경로로 이미지/사운드 사용

```
URL url = getClass().getClassLoader().getResource("Image.jpg");  
BufferedImage image = ImageIO.read(url);
```

<경로의 위치는 추가된 폴더의 위치를 기준으로 함>



```
URL url = getClass().getClassLoader().getResource("wave.wav");  
AudioInputStream audio = AudioSystem.getAudioInputStream(url);
```

숙제 제출시 반드시 url을 써서 상대 경로로 입력할 것
(JAR 파일 제출 시 필수!!!)