

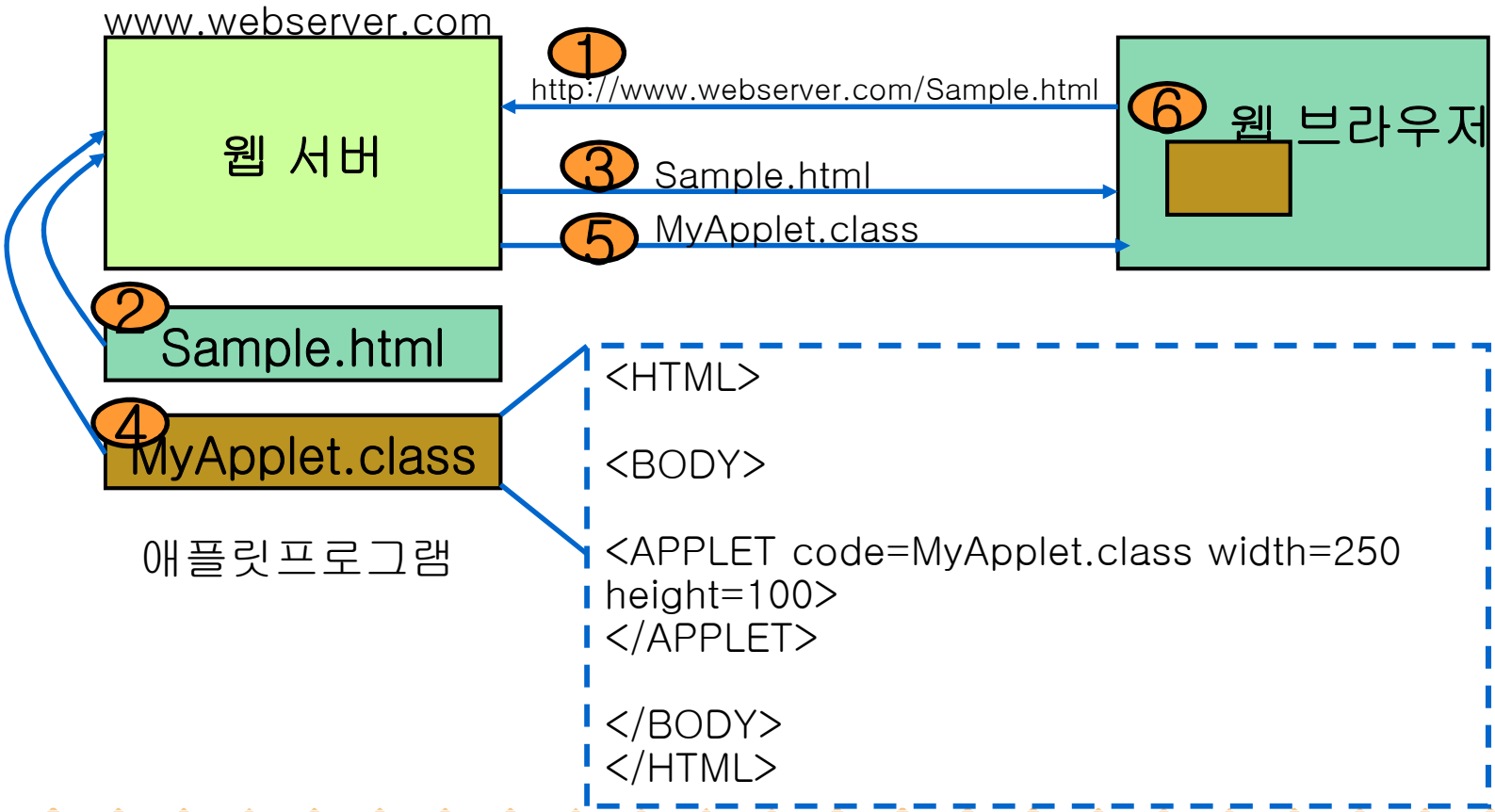

자바 애플릿

학습목표

- 애플릿 동작 환경
- 브라우저에서 애플릿이 동작하는 방법
- 애플릿 프로그램밍
- Applet 클래스
- 폰트 및 칼라
- 이미지 로드 및 출력
- 애니메이션
- 더블 버퍼링을 이용한 이미지 로드 및 출력
- 사운드 로드 및 출력
- 애플릿에서 이벤트 처리

애플릿 프로그램이란?

- 클라이언트 시스템이 서버로부터 웹페이지를 다운로드 받았을 때, 웹 페이지의 특정한 윈도우에서 동작하는 프로그램
- 동작 과정

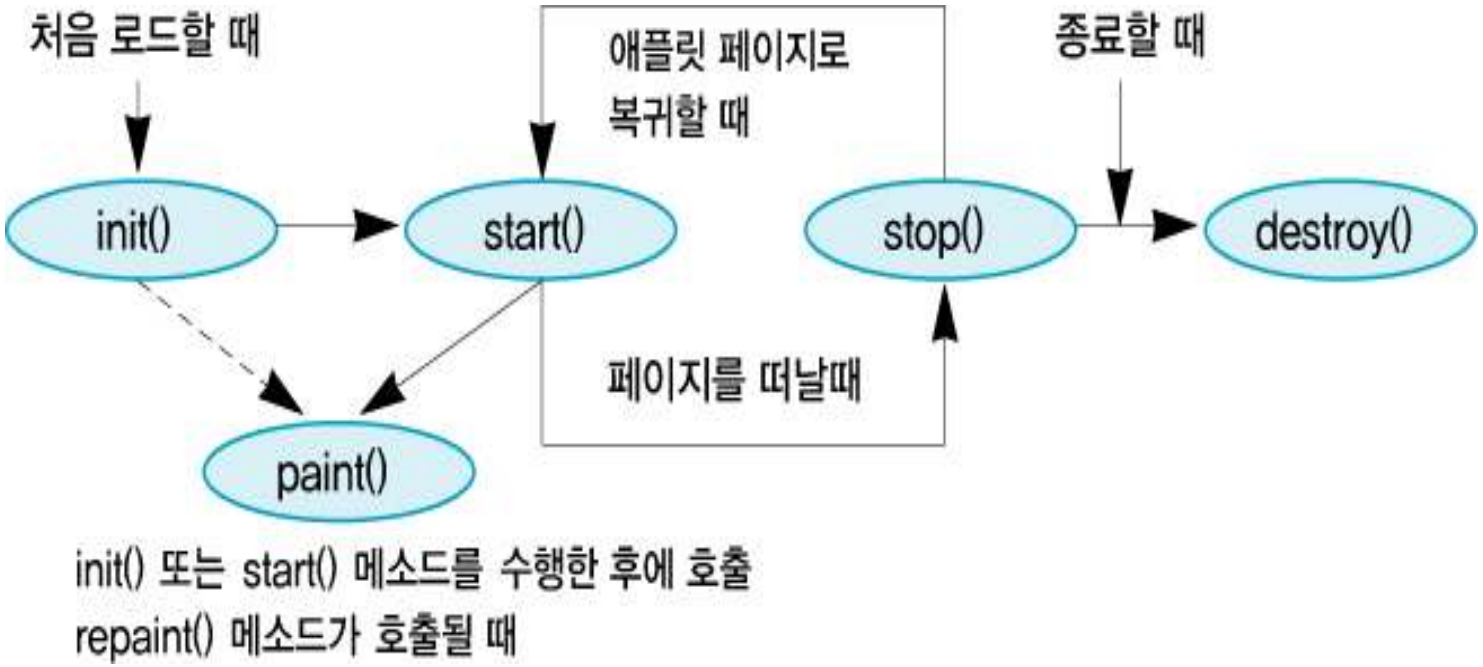


애플릿 클래스

- 애플릿은 java.awt.Applet 클래스로부터 상속받아 작성해야 함.
- Applet 클래스에서 지원하는 주요 메소드
  - void init(): 애플릿이 실행을 시작할 때 호출됨. 애플릿을 위해 호출되는 첫 번째 메서드
  - void start(): 애플릿이 실행을 시작해야 할때 브라우저에 의해 호출됨. init() 이후에 자동으로 호출
  - void stop(): 애플릿 중지를 위해 브라우저에 의해 호출
  - void destroy(): 애플릿이 종결되기전에 브라우저에 의해 호출
  - void paint(Graphics g): 애플릿 창에 텍스트 출력 및 그림을 그리는 기능등을 수행

애플릿 생명 주기

- 애플릿의 메소드를 호출을 통해 애플릿은 생성되어 소멸됨.
- 생성에서 소멸까지의 **생명주기**



애플릿 프로그램 예

MyApplet.java

```
import java.applet.Applet;
import java.awt.*;
public class MyApplet extends Applet
   public void init(){} // init() 함수 생략 가능
   public void paint(Graphics g){
      g.drawString("안녕하세요!! 애플릿 예제입니다.", 50, 50);
   }
}
```

MyApplet.html

```
<HTML>
   <HEAD>
      <TITLE> Hello </TITLE>
   </HEAD>
<BODY>
   <APPLET CODE="MyApplet.class", WIDTH=250 HEIGHT=100>
   </APPLET>
</BODY>
</HTML>
```

애플릿 골격

```
import java.awt.*;
import java.applet.Applet;
public class MyApplet extends Applet {
   // 먼저 호출된
   public void init(){
       // 초기화
   }
   public void start(){ //init() 이후에 두번째로 호출됨. 애플릿이 다시 시작될때마다 호출
     // 실행을 시작하거나 다시 계속한다
   }

   public void stop() { // 애플릿이 중단될 때 호출
       // 실행을 중지
   }
   public void destroy() { // 애플릿이 종결될 때 호출. 마지막으로 실행되는 메서드
     // 중단 활동을 수행
   }

   public void paint(Graphics g) {
     // 윈도우에 내용을 다시 출력
   }
}
```

APPLET 태그

- CODE, WIDTH 및 HEIGHT 필수항목을 제외한 선택항목의 APPLET 태그를 제공

```
<APPLET [CODEBASE = 애플릿의URL] CODE=애플릿파일이름
    [ALT=대체텍스트] [NAME=애플릿객체이름]
    WIDTH=애플릿창의가로픽셀수 HEIGHT=애플릿창의높이픽셀수 [ALIGN=정렬값]
    [HSPACE=픽셀수] [VSPACE=픽셀수]
    [<PARAM NAME="파라미터1" VALUE="값">]
    [<PARAM NAME="파라미터2" VALUE="값">]
    .....
</APPLET>
```

애플릿에 매개변수 전달 예(1/2)

- 애플릿은 명령어에 의하여 실행되지 않으므로 APPLET 태그내에 PARAM 태그를 사용하여 애플릿에 필요한 매개변수를 전달한다
- 예: 애플릿에서 출력되는 색깔, 글자의 크기 및 애니메이션 속도

appletparam.html

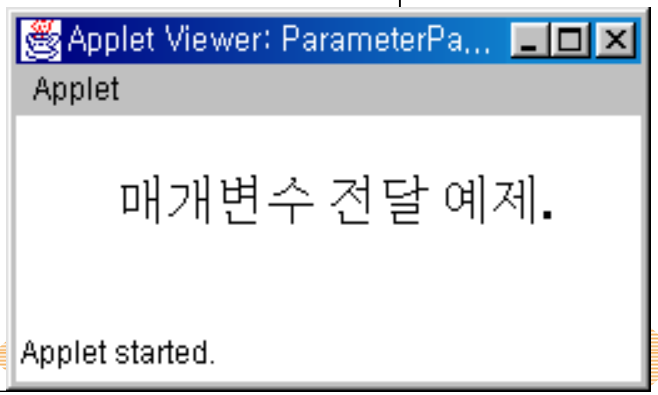
```
<HTML>
  <HEAD>
    <TITLE> Hello </TITLE>
  </HEAD>
<BODY>
<APPLET CODE="ParameterPass.class" WIDTH=250 HEIGHT=80>
<PARAM NAME="msg" VALUE="매개변수 전달 예제.">
<PARAM NAME="font" VALUE="Serif">
<PARAM NAME="size" VALUE="20">
</APPLET>
</BODY>
</HTML>
```

애플릿에 매개변수 전달 예(1/2)

```
import java.awt.*;
import java.applet.Applet;
public class ParameterPass extends Applet {
   String f_name, message, size;
   int f_size;
   Font font;
   public void init(){
      message = getParameter("msg");
      f_name = getParameter("font");
      if(f_name==null) f_name="Courier";
      size = getParameter("size"); // "size"에 해당하는 값을 전달받는다.
      try{
         if(size != null) f_size = Integer.parseInt(size);
         else f_size=10;
      }
      catch(NumberFormatException e){}
   }
   public void paint(Graphics g) {
      font = new Font(f_name, Font.BOLD, f_size);
      g.setFont(font);
      g.drawString(message, 5, 40);
   }
}
```

"msg" 해당하는 "매개변수 전달 예제." 문자열, "font"에 해당하는 문자열 "Serif"를 전달받아 각각 message 및 f_name에 할당한다.

실행결과



애플릿 클래스의 주요 메서드

메소드	설명
void init()	애플릿을 초기화 한다.
void start()	애플릿을 시작시킨다.
void stop()	애플릿을 중단한다.
void destroy()	애플릿을 종료한다.
URL getCodeBase()	애플릿 코드가 있는 URL 객체를 반환한다.
URL getDocumentBase()	HTML 문서가 있는 URL 객체를 반환한다.
Image getImage(URL url)	파일이름을 포함한 url이 지정하는 곳에서 이미지를 가져와 Image 객체로 반환한다.
Image getImage(URL url , String imgName)	파일이름을 포함하지 않는 url이 지정하는 곳에서 imgName 으로 지정된 이미지 파일을 가져와 Image 객체로 반환한다.
String getParameter (String pName)	애플릿 호출시 사용한 매개변수 중에서 pName 매개변수의 값을 반환한다.
void showStatus(String str)	검색기의 상태라인에 str을 나타낸다.
void resize(int width, int height)	

애플릿에 출력하기

- 윈도우 배경 컬러 설정(init() 메서드내에서)
  - setBackground(Color newColor), getBackground()
  - setForeground(Color newColor), getForeground()
  - Color (Color.black, Color.blue, Color.cyan, Color.green, Color.red, Color.yellow 등등)
- Graphics 클래스로 제공(paint() 메서드내에서)
  - Graphics 메서드를 사용하여 그리고자 하는 내용을 화면에 출력

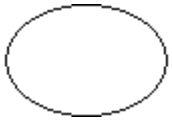
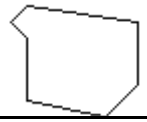
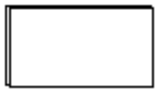
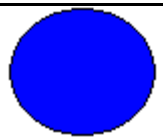
애플릿에 출력하기

```
import java.awt.*;
import java.applet.Applet;
public class SampleDraw extends Applet {
   String msg;
   public void init(){
     setBackground(Color.cyan);
     setForeground(Color.red);
     msg = “Inside init( ) –”;
   }
   public void start() {
     msg += “Inside start( ) –”;
   }
   public void paint(Graphics g) {
     msg += “Inside paint( ) –”;
     g.drawString(msg, 10, 30);
   }
}
```

다시그리기(repaint) 요청하기

- 애플릿이 윈도우에 출력되는 정보를 갱신할 필요가 있을 때마다 repaint() 호출
- 메서드
  - void repaint(): 전체 다시그리기
  - void repaint(int left, int top, int width, int height)
  - void repaint(long maxDelay)
  - void repaint(long maxDelay, int left, int top, int width, int height)

Graphics 클래스 메서드

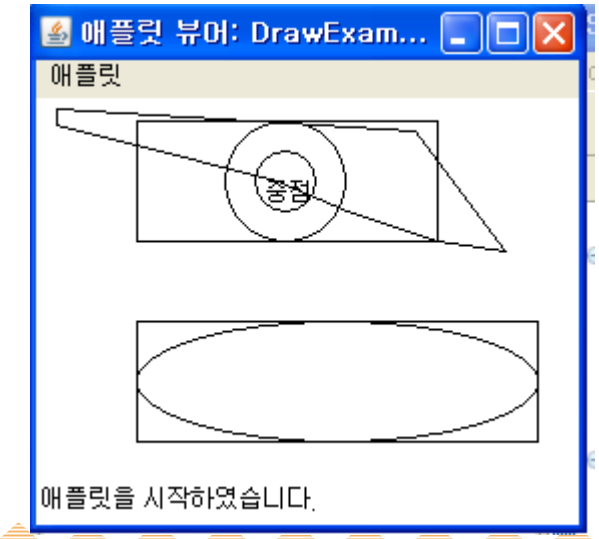
메소드	모양	형식
drawArc		public void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle);
drawOval		public void drawOval(int x, int y, int width, int height)
drawPolygon		public void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)
drawRect		public void drawRect(int x, int y, int width, int height)
drawRoundRect		public void drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)
draw3DRect		public void draw3DRect(int x, int y, int width, int height, boolean raised)
fillArc		public void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle);
fillOval		public void fillOval(int x, int y, int width, int height)

Graphics 클래스 메서드

메소드	모양	형식
fillPolygon		public void fillPolygon(int[] xPoints, int[] yPoints, int nPoints)
fillRect		public void fillRect(int x, int y, int width, int height)
fillRoundRect		public void fillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)
fill3DRect		public void fill3DRect(int x, int y, int width, int height, boolean raised)
drawLine		public void drawLine(int x1, int y1, int x2, int y2);
drawPloyline		public void drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
drawString	“ 애플릿 예제”	public void drawstring(String str, int x, int y);
drawImage		public void drawImage(Image img, int x, int y , Color color, ImageObserver imgobsr);

```
import java.applet.Applet;
import java.awt.*;
public class DrawExamples extends Apple
{
   int x=294, y=210, r=260;
   public void init(){} // 생략가능
   public void paint(Graphics g){
      g.drawLine(50, 10, 200, 10);
      g.drawLine(200, 10, 200, 70);
      g.drawLine(200, 70, 50, 70);
      g.drawLine(50, 70, 50, 10);
      g.drawRect(150, 110, 200, 60);
      g.drawOval(150, 110, 200, 60);
      circle(x, y, r, g);
      polygon(g);
     }
      public void circle(int x, int y, int r, Graphics g){
        int w=r, h=r;
        int hr=r/2, hx=x+15, hy=y+15; // 반원을 위한 변수
        g.drawOval(x, y, w, h);
        g.drawString("중점", x+20, y+40);
        halfcircle(hx, hy, hr, g);
      }
```

```
  public void halfcircle(int hx, int hy, int hr,
Graphics g){
      int hw=hr, hh=hr;
      g.drawOval(hx, hy, hw, hh);
   }
}
public void polygon(Graphics g) {
    int x[]={10, 189, 234, 200, 170, 155,
120, 81, 10};
    int y[]={4, 15, 75, 70, 60, 55, 40, 30,
12};
    int n=x.length;
    g.drawPolygon(x, y, n);
}
```



애플릿의 제한점

- 자바 애플릿 제한 사항(보안을 위해)
  - 다른 프로그램을 실행시키지 못한다.
  - 지역 파일의 입출력이 불가능하다.
  - 어떠한 네이티브 메소드도 호출하지 못한다.
  - 내려받은 서버 이외의 어떠한 서버에도 네트워크 통신이 불가능하다.

Font 클래스

- 자바에서 폰트를 처리하는 클래스
- java.awt 패키지에 포함된 클래스
- drawString() 메소드에 의하여 출력되는 문자열에 대한 폰트의 이름, 스타일 및 크기를 결정한다
- public Font(String name, int style, int size)
  - name은 "Arial", "Courier", "Serif" 및 “SansSerif"등과 같은 폰트의 이름을 의미
  - style는 Font 클래스에서 클래스 종단변수(상수값)로 정의된 PLAIN, BOLD 및 Italic 중에서 하나를 선택할 수 있으며, 각각은 일반형, 강조체 및 이탤릭체를 의미
  - size는 폰트의 크기를 결정하는 정수값

FontMetrics 클래스

- 보여지는 문자의 폭, 높이 및 현재 폰트에 관한 상세한 정보를 알아내기 위해 사용되는 클래스
- 메서드

메소드	설명
int stringWidth(String str)	주어진 문자열의 전체 폭의 길이를 픽셀단위로 반환한다.
int charWidth(char ch)	주어진 문자의 폭의 길이를 픽셀단위로 반환한다.
int getHeight()	폰트의 높이를 반환한다.


```
import java.awt.*;
import java.applet.Applet;
public class GetFontInfo extends Applet
{
   public void paint(Graphics g){
      Font ft1 = new Font("SansSerif", Font.PLAIN, 16);
      g.setFont(ft1); // ft1 폰트 설정
      g.drawString("SansSerif 폰트, 일반형 및 크기가 16", 1, 30);
      Font ft2 = new Font("Serif", Font.BOLD, 15);
      g.setFont(ft2); // ft2 폰트 설정
      g.drawString("Serif 폰트, 강조체 및 크기가 15", 25, 50);

      Font ft = new Font("SansSerif", Font.BOLD, 15);
      FontMetrics fm = getFontMetrics(ft); // fm FontMetrics 객체 생성
      g.setFont(ft);
      String s = "문자열을 애플릿의 중앙에 출력한다";
      int x = (getSize().width-fm.stringWidth(s))/2;
      int y = getSize().height/2;
      g.drawString(s, x, y);
   }
}
```

Color 클래스

- 그리기 작업에 사용되는 현재의 색 및 애플릿이나 기타 창들의 배경색을 설정하는 클래스
- 생성자
  - public Color(int c):정수변수는 32비트 중 24비트만을 사용하여 색을 표현 24-31:0xff, 16-23:red, 8-15:green, 0-7: blue
  - public Color(int r, int g, int b): r,g,b(0-255)
- 관련메서드
  - setColor(), setBackground(), setForeground()

Color 클래스 사용 예제

```
import java.awt.*;
import java.applet.Applet;
public class SetByColor extends Applet
{
   public void paint(Graphics g){
      Color c1 = Color.red;
      g.setColor(c1); // 빨간색으로 설정
      g.drawOval(50, 20, 20, 20);
      g.drawLine(50, 20, 200, 70);
      Color c2 = new Color(0, 255, 0);
      g.setColor(c2); // 녹색으로 설정
      g.fillRect(150, 50, 20, 20);
   }
}
```

Image 로드 및 출력

- 이미지를 로드해서 애플릿에 보여주는 과정
  - getImage() 메소드를 이용하여 파일 시스템의 이미지 파일을 로드

    Image img; // img Image 객체선언
    img=getImage(getDocumentBase(), “khyoo.jpg");
  - 화면에 이미지를 보여주기 위해서는 Graphics 클래스의 drawImage() 메서드를 이용

    public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer)

이미지 로드 및 출력 예

```
import java.awt.*;
import java.applet.Applet;
import java.awt.image.*;
public class ImageLoad extends Applet
{
    Image image;
    public void init(){
      image=getImage(getDocumentBase(), “khyoo.jpg");
    }
    public void paint(Graphics g){
        g.drawImage(image, 0, 0, this);
     }
   }
```

여러 이미지 로드하여 출력(1/2)

```
import java.awt.*;
import java.applet.*;
import java.io.*;
public class Animation extends Applet implements Runnable
{
   int i;
   Thread engine=null;
   MediaTracker tracker;
   final int imageNum=10;
   Image im[]=new Image[imageNum];
   public void init(){
      tracker=new MediaTracker(this);
      for(i=2; i<=10; i++){
         im[i-1]=getImage(getCodeBase(), "images/"+“FMAG00"+i+".gif");
         tracker.addImage(im[i-1], i-1);
      }
   }
   public void start(){
      if(engine==null){
         engine=new Thread(this); // 또는 this 대신에 Thread(new Animation())
         engine.start();
      }
   }
```

여러 이미지 로드하여 출력(2/2)

```
public void stop(){  engine=null;}
   public void run(){
      Thread me=Thread.currentThread(); // 현재 실행중인 스레드를 반환한다.
      me.setPriority(Thread.MIN_PRIORITY);
      while(true){
         try{
            for(i=0; i<10; i++){
               repaint();
              //Thread.sleep(500);
            }
         }
         catch(InterruptedException e){}
    }
   }
   public void paint(Graphics g){
      if(tracker.checkAll(true)) // 이미지 로드를 체크한다.
           g.drawImage(im[i],0,0,this);
        else
           g.drawString("이미지를 읽고 있습니다.", 20,20);
   }
}
```

오디오 로드 및 출력

- 처리과정
  - getAudioClip() 메소드를 이용하여 AudioClip 클래스 객체를 생성한다. getAudioClip() 메소드는 Applet 클래스에서 지원하며, 사용형식은 다음과 같다
    AudioClip getAudioClip(URL, 오디오 파일이름)
  - 생성된 AudioClip 객체 및 AudioClip 클래스에서 지원하는 3개의 메소드를 이용하여 해당 오디오 파일을 플레이 한다.

메소드	설명
void loop()	호출한 객체가 지정하는 오디오를 반복하여 플레이 한다.
void play()	호출한 객체가 지정하는 오디오를 플레이 한다.
void stop()	호출한 객체가 지정하는 오디오의 플레이를 중단한다.

오디오 로드 및 출력 예

```
import java.awt.*; import java.applet.*; import java.io.*;
public class AudioPlay extends Applet implements Runnable
{
   Thread engine=null;
   AudioClip music;
   public void init(){
         music = getAudioClip(getCodeBase(), "audios/"+"spacemusic.au");
   }
   public void start(){
      if(engine==null){ engine=new Thread(this);  engine.start();
      }
   }
   public void stop(){
      engine=null; music.stop();
   }
   public void run(){
      music.loop();
   }
   public void paint(Graphics g){
         g.drawString("오디오를 플레이중 입니다.", 20, 20);
   }
}
```