

Applet-metodien käyttö sovelluksissa

- staattinen metodi `newAudioClip` palauttaa `AudioClip`-oliion
 - ◆ huom. parametriksi annetaan URL-olio eikä esim. tiedostokalva
 - ◆ vinkki: sovelluksen oletushakemiston polku saa metodikutsulla `System.getProperty("user.dir")`
- saatua `AudioClip`-oliota voidaan käyttää normaalisti

AudioApplication.java 1(3)

```
public class AudioApplication {
    public static void
        main(String[] args) {
        AudioClip[] sounds =
            new AudioClip[args.length];
        String base = "file:" +
            System.getProperty("user.dir") + "/";
```

AudioApplication.java 2(3)

```
for (int i = 0; i < args.length; i++) {
    try {
        sounds[i] = Applet.newAudioClip(
            new URL(base + args[i]));
    } catch (MalformedURLException e) {
        throw new RuntimeException(
            "Cannot load audio file "
            + args[i] + ".");
    } // try
} // for
```

AudioApplication.java 3(3)

```
for (int i = 0;
     i < sounds.length; i++) {
    sounds[i].loop();
} // for
} // main()
} // class
```

3.1. javax.sound.sampled

- mukana JDK-versiosta 1.3 alkaen
- tarjoaa matolan tason liittymän alustan äänilaitteistoon (myös havainnointi)
- pyrkii sihti olemaan alustariippumaton ja yhteistyvässä oleva rakennelma
- mahdollistaa äänisignaalin
 - ◆ vastantottamisen (esim. äänitys)
 - ◆ käsittelyn (esim. vahvistus tai kailunta)
 - ◆ toistamisen
- yhä kehitysvaiheessa? ←outoja bugeja ja puutteita

Pakkaukset

- `javax.sound.sampled`
 - ◆ rajapintoja ja luokkia samplattun äänisignaalin tallennukseen, muokkaamiseen ja toistoon
- `javax.sound.midi`
 - ◆ rajapintoja ja luokkia MIDI-käyttöön
- `javax.sound.sampled.spi`
- `javax.sound.midi.spi`
 - ◆ ulkopuolisille palveluntarjoajille (service providers) tarkoitettuja apuluokkia

Piirteitä

- pähkinäkuoressa: äänidataa sisältävien tavujen lukua, kirjoitusta ja operointia
- liittyvät syömo- (esim. mikrofoni tai tiedosto) ja tulostinlaitteisiin (esim. kaiutin tai tiedosto)
- äänidatan puskurointi (esim. reaaliaikainen äänivirta)
- äänisignaalien yhdistäminen
- käyttäjän komennot: aloita, pysäytä, jatka, lopeta

Äänidatan käsittelytavat

- puskuroitu (*buffered*)
 - ◆ virta (*streaming*): reaaliaikaisen äänidatan käsittelyä
 - ◆ operoitava (esim. äänitettävä tai käsiteltävä) tavuja liikimain samassa tahdissa kuin missä niitä lähetetään
- puskuroimaton (*unbuffered*)
 - ◆ äänidata sijaitsee (kokonaisuudessaan) muistissa
 - ◆ monipuolisempi toisto: silmäkohtai, aloituspaikan valinta

Äänidatan formaatit 1(2)

- dataformaatti
 - ◆ kertoo kuinka sarja tavuja eli "raaka" samplattu äänidata pitää tulkita
- **AudioFormat**-luokka
- tiedostoformaatti
 - ◆ määrittelee äänitiedoston rakenteen
- **AudioFileFormat**-luokka

Äänidatan formaatit 2(2)

- vaikka tarjolla on metodeja
 - ◆ erilaisten ääniformaattien muuttamiseen
 - ◆ yleisten tiedostoformaattien lukemiseen ja tallentamiseen
- kyse ei silti ole kaikenkattavasta äänityökalusta
 - ◆ palveluntarjoajilta tukea ja täydennystä valikoimann

AudioFormat-luokka

- koodusteknikka (esim. PCM, a-law tai µ-law)
- kanavien määrä (1 = mono, 2 = stereo jnc.)
- samplustaajaus
- kvantisointitaso (so. käytettyjen bittien määrä)
- kehystaajaus (*frame rate*)
 - ◆ kehys (*frame*) = kaikki tiettyyn hetkeen kuuluvu data; (esim. kanavien nykyiset näytearvot)
- kehyyksen koko tavuina
- tavujärjestys: *big-endian* tai *little-endian*

AudioFileFormat-luokka

- tiedostotyyppi (esim. WAVE, AIFF)
- tiedoston pituus tavuina
- äänidatan pituus kehyyksiä
- **AudioFormat**-olio, joka määrittelee tiedoston sisältämän äänidatan muodon