`TestJTable.java`. Laden Sie diese herunter, testen diese und versuchen das Prinzip nachzuvollziehen.

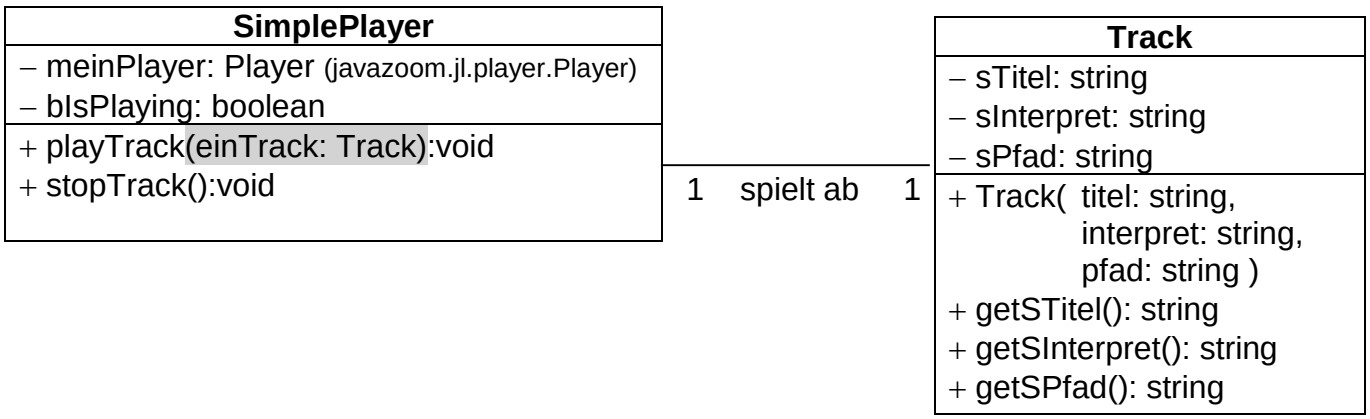
- Versuchen Sie eine Ergebnistabelle in ihr Programm einzufügen.

2.5. Einfacher mp3-Player mit kurzfristiger Assoziation

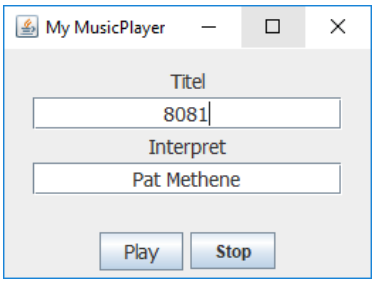
Mit einer Assoziation wird die Verbindung von Objekten einer Klasse zu einem oder mehreren anderen Objekten beschrieben (s. Skript „26_OOP_Assoziationen“).

1. Assoziationen können kurzfristig sein,
   a. wenn ein anderes Objekt an das aktuelle Objekt als Parameter übergeben wird.
   b. wenn ein Objekt in einer Methode einer anderen Klasse lokal erzeugt wird.
2. Assoziationen können langfristig sein,
   a. wenn das Objekt Referenzen auf die assoziierten Objekte speichert.

Beim folgenden Beispiel handelt es sich um eine Assoziation zwischen einem Objekt der Klasse **`MusicPlayer`** und einem Objekt der Klasse **`Track`**. (s.u. UML Klassendiagramm) Die Methode `playTrack()` eines Objekts der Klasse `MusicPlayer` wird mit einem Objekt der Klasse `Track` als Parameter aufgerufen (kurzfristige Assoziation).



Die im UML Diagramm gegebenen Klassen sollen zur Implementierung eines einfachen Abspielprogramms für .mp3 Dateien angewendet werden.



Hinweise zur Programmierung:

Implementierung der Klasse `Track`

- Das Attribut sPfad speichert den Dateipfad einer .mp3 Datei.
- Der Konstruktor initialisiert alle Attribute mit den übergebenen Parametern.

Implementierung der Klasse `SimplePlayer`

- Die Klasse verwendet die Bibliothek `jl1.0.1.jar` (aus Moodle) zum Abspielen von .mp3 Dateien (s. http://www.javazoom.net/javalayer/sources.html).
- Da die `playTrack` Methode relativ komplex ist, kann sie von Moodle kopiert werden (s. Moodle Player_PlayTrack.txt)
- Die Methode `stopTrack` prüft anhand des Attributs bIsPlaying, ob gerade ein Track gespielt wird und stoppt gegebenenfalls das Abspielen des Tracks mit Hilfe der Methode `close()` der `Player` Klasse.

Implementierung der Klasse `PlayerGUI`

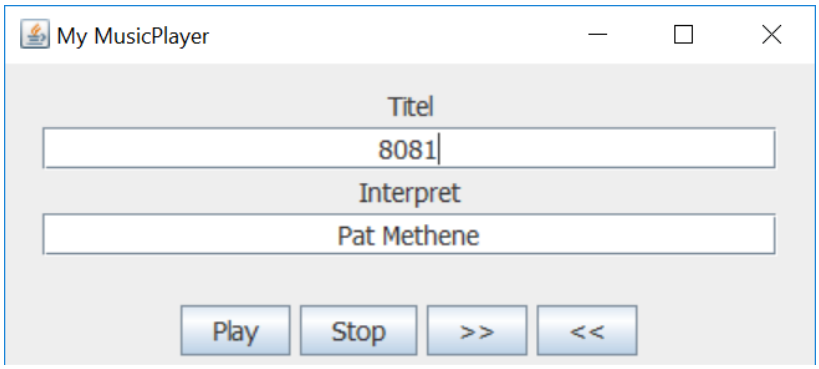
- Für den Musikplayer soll eine einfache GUI erstellt werden (s.o.) Erstellen Sie diese zunächst mit dem Window Builder.
- Fügen Sie der GUI-Klasse ein Objekt der Klasse `Track` als *Attribut* hinzu und initialisieren Sie es direkt bei der Erzeugung. Eine mp3 Datei finden Sie in Moodle oder bestimmt auch auf Ihrem Smartphone. (Achtung "\\" für einen "\" im Dateipfad angeben.)
- Fügen Sie der GUI-Klasse ein Objekt der Klasse `SimplePlayer` als *Attribut* hinzu.
- Initialisieren Sie die beiden Textfelder direkt bei ihrer Erzeugung mit dem Titel bzw. Interpret mit den entsprechenden Daten aus dem Track-Objekt.
- Programmieren Sie das Play-Button-Ereignis mit Hilfe einer mouseClicked Methode (z.B. mit der Variante „anonyme Klasse“: im Designer Modus→Kontextmenü des Buttons→Add Event Handler…) um den Track abzuspielen. Da die playTrack() Methode eine Exception auslösen kann, muss diese abgefangen werden und gegebenenfalls mit
  `JOptionPane.showMessageDialog(frmMyMusicplayer,"mp3 Datei nicht gefunden!");`
  eine Fehlermeldung angezeigt werden.

- Programmieren Sie das Stop-Button-Ereignis um den Track zu beenden.

```
import java.io.BufferedInputStream;

 * @author stk
public class SimplePlayer
{
    private Player meinPlayer;

    public void playTrack(Track einTrack) throws Exception
    {
        try {
            FileInputStream fis     = new FileInputStream(einTrack.getPfad());
            BufferedInputStream bis = new BufferedInputStream(fis);
            meinPlayer = new Player(bis);
        }
        catch (Exception e) {
            throw(e);
        }

        // run in new thread to play in background
        try {
            new Thread() {
                public void run() {
                    try
                    {
                        meinPlayer.play();
                    }
                    catch (JavaLayerException e)
                    {
                        RuntimeException re = new RuntimeException();
                        throw  re;
                    }
                }
            }.start();
        }
        catch (Exception e) {
            throw(e);
        }

    }

    public void stopTrack()
    {
        this.meinPlayer.close();
    }

}
```

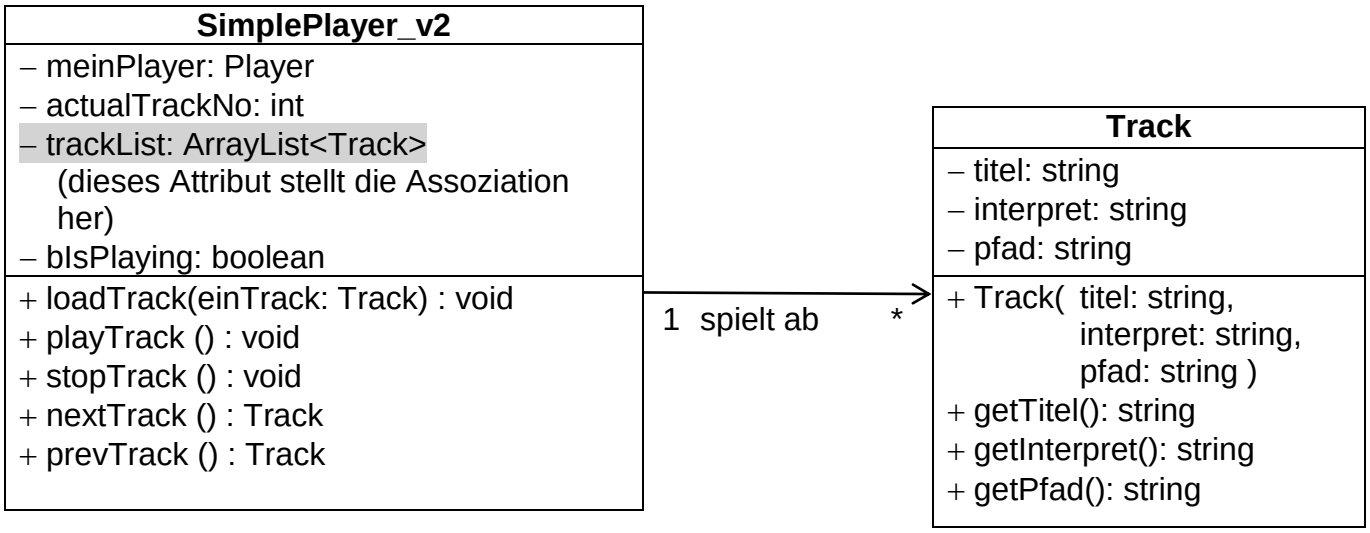
```
public class MP3AppGUI_Gb
{

    private JFrame frmMyMusicplayer;
    private JTextField txtTitel;
    private JTextField txtInterpret;
    private Track einTrack = new Track("8081", "Pat Methene",
                                "D:\\Users\\stk\\Music\\04 - 8081.mp3");
    private SimplePlayer einfacherPlayer = new SimplePlayer();

     * Launch the application.
    public static void main(String[] args)

     * Create the application.
    public MP3AppGUI_Gb()

     * Initialize the contents of the frame.
    private void initialize()
    {

    ...
        JButton btnPlay = new JButton("Play");
        btnPlay.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                try {
                    einfacherPlayer.playTrack(einTrack);
                }
                catch (Exception ex)
                {
                    JOptionPane.showMessageDialog(frmMyMusicplayer,
                                                  "mp3 Datei nicht gefunden!");
                }
            }
        });
        btnPlay.setFont(new Font("Tahoma", Font.PLAIN, 14));
        pnlButtons.add(btnPlay);

        JButton btnStop = new JButton("Stop");
        btnStop.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                einfacherPlayer.stopTrack();
            }
        });
        pnlButtons.add(btnStop);
    }

}
```

2.6. Verbesserter mp3-Player mit **langfristiger** Assoziation



Durch die ArrayList mit dem Namen **`trackList`** der Klasse Track, kann eine langfristige 1: * (eins zu viele) Assoziation abgebildet werden.

Im Attribut **`actualTrackNo`** wird der Index des aktuell im **SimplePlayer_v2** ausgewählten Tracks gespeichert.



Hinweise zur Programmierung:

An der Klasse `Track` sind keine Änderungen erforderlich

Implementierung der Klasse `SimplePlayer_v2`

- Nehmen Sie als Ausganspunkt für diese Klasse die alte Klasse `SimplePlayer`.
- Das Attribut `actualTrackNo` gibt an, welcher Track gerade ausgewählt ist.
- Die Methode `loadTrack( )` fügt einen Track in ArrayList am Ende an.
  Die `actualTrackNo` wird auf 0 gesetzt, d.h. der erste Track ist nach dem Laden der aktuelle.
- Die Methode `playTrack` besitzt jetzt keinen Parameter mehr. Sie soll den Track abspielen, der durch die `actualTrackNo` ausgewählt ist.
- Die Methode `nextTrack( )` prüft, ob es nach dem aktuell ausgewählten Track noch einen weiteren Track gibt.
  Wenn ja, dann wird dieser Track zum aktuellen Track gemacht und ein Verweis auf diesen Track wird als Rückgabewert zurückgegeben.
  Wenn nein, dann wird der erste Track in der Liste zum aktuellen.

- Die Methode `prevTrack( )` arbeitet analog zur Methode `nextTrack( )`

Implementierung der Klasse `PlayerGUI_v2`

- Fügen Sie der GUI die beiden zusätzlichen Schaltflächen hinzu.
- Laden Sie innerhalb des Konstruktors mehrere Track Objekte mit Hilfe der Methode `loadTrack( )` in das SimplePlayer_v2-Objekt. Zeigen Sie den Titel und Interpret des ersten Tracks auf der GUI an.
- Die Ereignismethode des play-Buttons muss angepasst werden, da die `playTrack`-Methode jetzt keinen Parameter mehr besitzt und wird deshalb einfach ohne Parameter aufgerufen.
- Die Buttons „vorwärts“ bzw. „rückwärts“ rufen einfach die entsprechenden Methoden der `SimplePlayer` Klasse auf und aktualisieren die „Titel“ und „Interpret“ Textfelder.

```
public class SimplePlayer_v2
{
    private Player meinPlayer;
    private ArrayList<Track> trackList = new ArrayList<Track>();
    private int actualTrackNo = 0;

    public void playTrack() throws Exception

    public void stopTrack()
    {
        this.meinPlayer.close();
    }

    public void loadTrack(Track einTrack)
    {
        this.trackList.add(einTrack);
        this.actualTrackNo = 0;
    }

    public Track nextTrack()
    {
        Track actualTrack = null;

        if (this.actualTrackNo < this.trackList.size() -1)
        {
            this.actualTrackNo++;
        }
        else
        {
            this.actualTrackNo = 0;
        }
        actualTrack = this.trackList.get(actualTrackNo);

        return actualTrack;
    }

    public Track prevTrack()
    {
        Track actualTrack = null;

        if (this.actualTrackNo > 0)
        {
            this.actualTrackNo--;
        }
        else
        {
            this.actualTrackNo =  this.trackList.size() -1;
        }
        actualTrack = this.trackList.get(actualTrackNo);

        return actualTrack;
    }
}
```

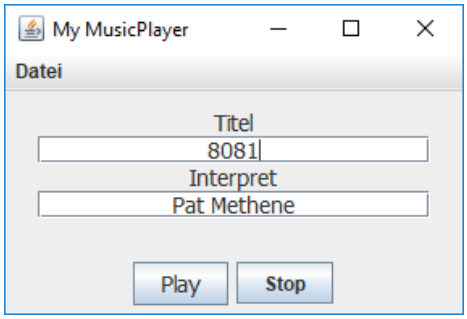
```
public class MP3AppGUI_Gb2_v2
{

    private JFrame frmMyMusicplayer;
    private JTextField txtTitel;
    private JTextField txtInterpret;
    private SimplePlayer_v2 einfacherPlayer = new SimplePlayer_v2();

     * Launch the application.
    public static void main(String[] args)

     * Create the application.
    public MP3AppGUI_Gb2_v2()
    {
        initialize();
        Track einTrack = new Track("8081", "Pat Methene",
                "D:\\Users\\stk\\Music\\04 - 8081.mp3");
        txtInterpret.setText(einTrack.getInterpret());
        txtTitel.setText(einTrack.getTitel());
        einfacherPlayer.loadTrack(einTrack);
        einfacherPlayer.loadTrack(new Track("Brother John's Blues", "Pat Methene",
        "D:\\Users\\stk\\Music\\01 - Brother John's Blues.mp3"));
        einfacherPlayer.loadTrack(new Track("The Eye Of The Hurricane",
        "Herbie Hankock", "D:\\Users\\stk\\Music\\05 - The Eye Of The Hurricane.mp3"));
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize()
    {
```

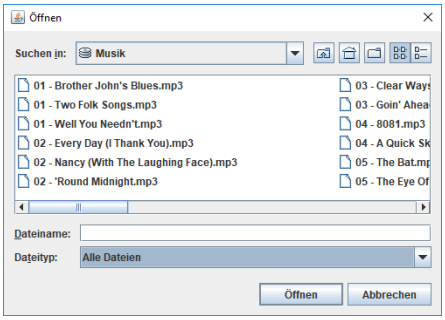
…

```
        JButton btnPlay = new JButton("Play");
        btnPlay.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                try {
                    einfacherPlayer.playTrack();
                }
                catch (Exception ex)
                {
                    JOptionPane.showMessageDialog(frmMyMusicplayer,"mp3 Datei
                }
            }
        });
        btnPlay.setFont(new Font("Tahoma", Font.PLAIN, 14));
        pnlButtons.add(btnPlay);

        JButton btnStop = new JButton("Stop");
        btnStop.setFont(new Font("Tahoma", Font.PLAIN, 14));
        btnStop.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                einfacherPlayer.stopTrack();
            }
        });
        pnlButtons.add(btnStop);

        JButton btnVorwaerts = new JButton(">>");
        btnVorwaerts.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                Track einTrack;
                einTrack = einfacherPlayer.nextTrack();
                txtInterpret.setText(einTrack.getInterpret());
                txtTitel.setText(einTrack.getTitel());
            }
        });
        btnVorwaerts.setFont(new Font("Tahoma", Font.PLAIN, 14));
        pnlButtons.add(btnVorwaerts);

        JButton btnRueckwaerts = new JButton("<<");
        btnRueckwaerts.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                Track einTrack;
                einTrack = einfacherPlayer.prevTrack();
                txtInterpret.setText(einTrack.getInterpret());
                txtTitel.setText(einTrack.getTitel());          }
        });
        btnRueckwaerts.setFont(new Font("Tahoma", Font.PLAIN, 14));
        pnlButtons.add(btnRueckwaerts);
    }
```

2.7. Zusatzaufgabe für die ganz Schnellen:

Das Programm von 2.5 soll erweitert werden, so dass über ein Datei Menü eine beliebige mp3-Datei zum Abspielen ausgewählt werden kann.





- Dazu wird das Fenster um eine `JMenuBar` erweitert (Window Builder verwenden).
- Die `JMenuBar` enthält ein `JMenu` („Datei“) und das `JMenu` ein `JMenuItem` („Track laden…“).
- Auf das `JMenuItem` kann eine ActionListener registriert werden. In der Ereignismethode soll dann ein `JFileChooser` Dialog geöffnet werden.
- Hat der Benutzer eine Datei ausgewählt, so kann diese dem Pfad des Track Objektes zugewiesen werden.