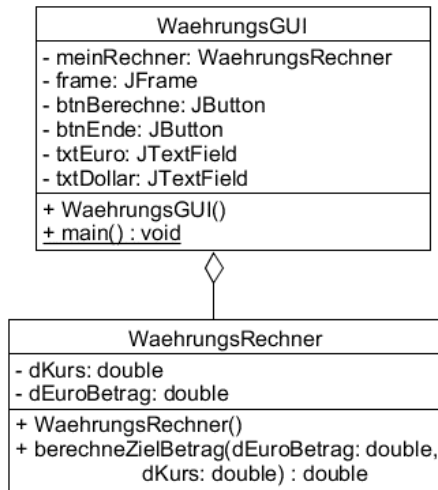


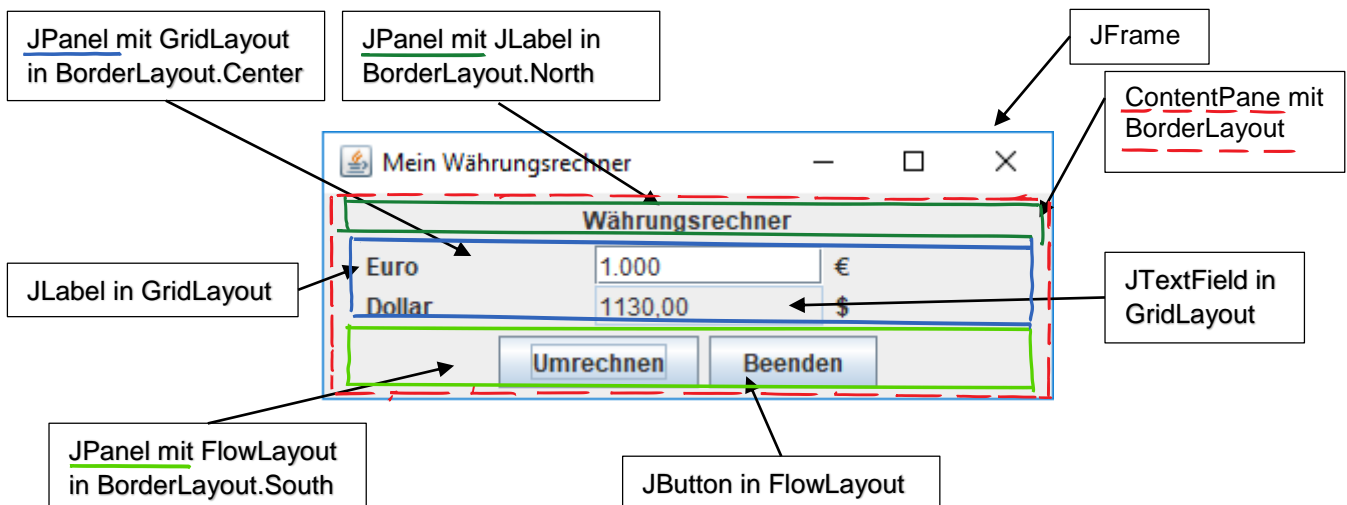
Einfache GUI Anwendung

Ziel: Erstellung einer einfachen GUI Anwendung: Kennenlernen des prinzipiellen Aufbaus eines JFrame mit JPanel, LayoutManagern und verschiedenen Dialogelementen.

1.1. Die zu erstellende Anwendung hat folgendes Klassendiagramm:



1.2. Die GUI der Anwendung hat folgende Struktur:



- 1.3. Erstellen Sie zunächst die Klasse WaehrungsGUI und zum Testen der GUI die Klasse WaehrungsStart.
- 1.4. Erstellen Sie eine MouseListener Klasse (abgeleitet von der MouseAdapter Klasse) für den „Beenden“ Button als innere Klasse von WaehrungGUI. In dieser Klasse muss nur die mouseClicked() Methode implementiert werden. Wenn der Button geklickt wird, soll das Programm mit dem Befehl `System.exit(0)` ; beendet werden.
- 1.5. Erstellen Sie eine MouseListener Klasse (abgeleitet von der MouseAdapter Klasse) für den „Umrechnen“ Button als innere Klasse von WaehrungGUI. In dieser Klasse muss nur die mouseClicked() Methode implementiert werden. Wenn der Button geklickt wird, geben Sie zum Testen zunächst nur einen Text im unteren JTextField aus.

1.6. Implementieren Sie im selben Package jetzt noch die Fachkonzept-Klasse

WaehrungsRechner.

1.7. Jetzt soll die `mouseClicked()` Methode von 1.5 abgeändert werden.

- aus dem Euro-Textfeld wird der String mit `getText()` ausgelesen.
- der String muss in einen double Wert konvertiert werden.
- dann kann mit einem Objekt der Fachkonzeptklasse der Zielbetrag berechnet werden.
- das Ergebnis wird mit `String.format()` für die Ausgabe im Dollar-Textfeld formatiert.

1.8. Wenn man das Programm testet, stellt man fest, dass es bei einer Fehleingabe im Euro-Textfeld möglicherweise mit einer Exception abstürzt. Aus diesem Grund ist es sinnvoller für dieses Textfeld statt der Klasse `JTextField` die Klasse

`JFormattedTextField` zu verwenden:

- ändern Sie die Verweisvariable von `JTextField` in `JFormattedTextField` ab
- erzeugen Sie das `JFormattedTextField` mit:
`new JFormattedTextField(NumberFormat.getNumberInstance())`
→ In diesem Textfeld können jetzt nur noch Zahlen eingegeben werden
- Der Wert aus dem Textfeld kann mit folgendem Befehl geholt und konvertiert werden:
`((Number)txtEuro.getValue()).doubleValue()`
- Testen Sie das geänderte Programm.

1.9. Eine weitere Verbesserung des Programms bestünde darin, dass die Umrechnung nicht nur dann ausgeführt wird, wenn man den Button klickt, sondern auch dann, wenn man im Euro-Textfeld die <Enter> Taste drückt. Hierfür gibt es selbstverständlich verschiedene Lösungen, eine sieht folgendermaßen aus:

Statt den beiden `MouseListener` Klassen wird eine `ActionListener` Klasse für die Ereignisbehandlung verwendet:

- Kommentieren Sie alles aus, was mit der `MouseListener` Ereignisbehandlung zu tun hatte.
- Definieren Sie als innere Klasse eine `ActionListener` Klasse die das `ActionListener` Interface implementiert:

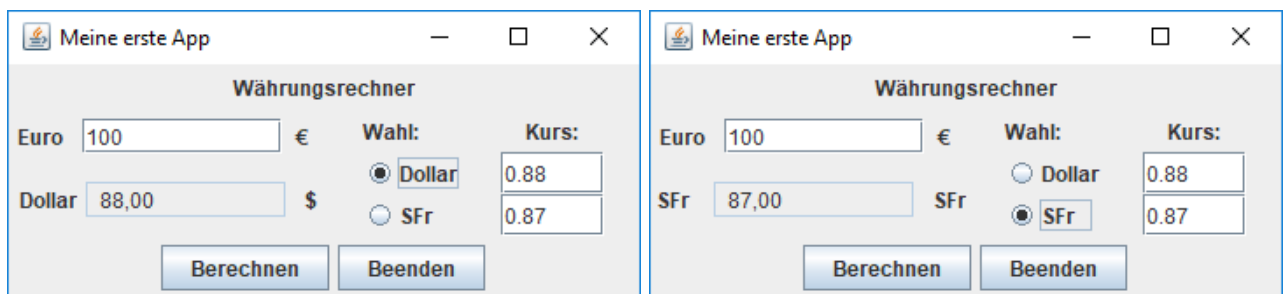
```
private class BtnActionListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(btnEnde))
        {
            System.exit(0);
        }
        else
        {
            // hier die Berechnung durchführen
        }
    }
}
```

- Registrieren Sie den `ActionListener` jetzt sowohl auf die beiden Buttons als auch auf das Euro-Textfeld und testen Sie, ob das geänderte Programm, die Anforderung erfüllt.

Arbeiten mit dem Eclipse Window Builder

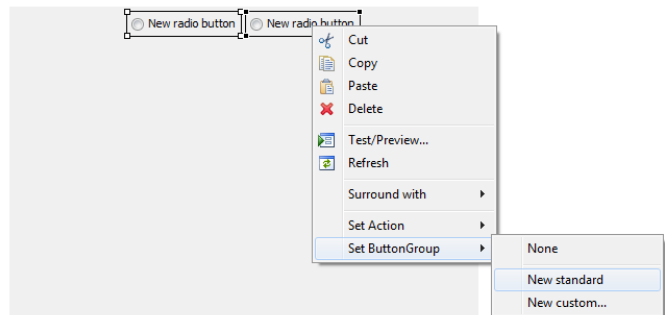
Es gibt verschiedene Tools um grafische Oberflächen mit Java produktiver und einfacher zu erstellen. Eclipse bietet hierfür den Window Builder (und weitere Alternativen).

- 2.1. Um sich mit dem Window Builder ein wenig vertraut zu machen, soll zunächst der Währungsrechner aus Aufgabe 1 jetzt mit dem Window Builder erstellt werden. Die Fachkonzeptklasse kann wiederverwendet werden.
- 2.2. Legen Sie eine Kopie der Lösung von 2.1 an und ergänzen Sie diese entsprechend dem nachfolgenden Bild.



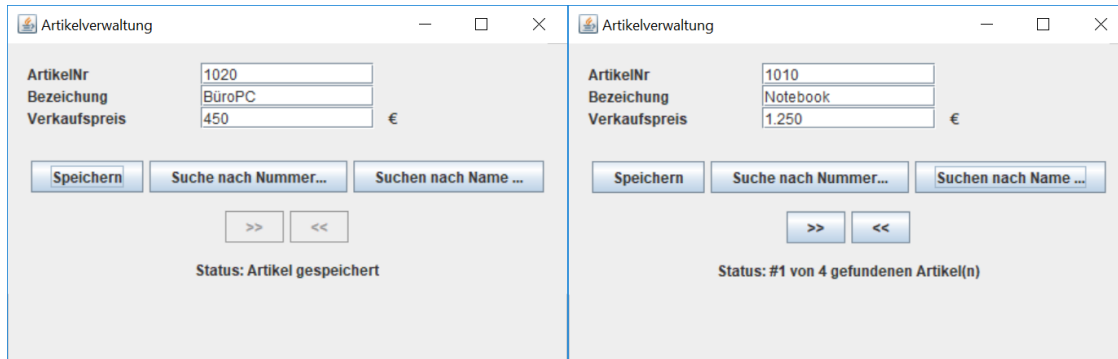
Wie zu erkennen ist, soll sich bei der Änderung der Devisenwahl auch die Beschriftungen bei der Ergebnisausgabe entsprechend anpassen.

- Für den Bereich „Devisenauswahl“ wurde eine extra JPanel angelegt, dass sich im BorderLayout.EAST befindet und selbst ein GridLayout enthält.
- Die RadioButton müssen zu einer Gruppe zusammengefügt werden, so dass immer nur eine Option wählbar ist.
- In der Endversion soll bei jeder Optionsänderung die Berechnung automatisch ausgeführt werden, ohne dass der Benutzer den „Berechnen“ Knopf anklicken muss.



- 2.3. Im Aufgabenblatt „A_Java_Arrays“ in Aufgabe 3.1 haben wir ein Programm zur Erfassung und Verwaltung von Artikel Daten erstellt (siehe UML Klassendiagramm dort). In Aufgabe 1 von Aufgabenblatte „A_JavaCollections“ haben wir die Container-Klasse dieses Programms noch verbessert. Für dieses Programm soll jetzt statt der bisherigen Konsoloberfläche eine GUI bereitgestellt werden (s.u.).

Die GUI soll im wesentlichen die selbe Funktionalität ermöglichen, wie die Konsolanwendung. Es können Artikel erfasst werden, Artikel anhand ihrer eindeutigen ArtikelNr und anhand einer nicht eindeutigen Bezeichnung gesucht werden.



- Mit der Speichern-Schaltfläche wird ein neuer Artikel erfasst und in der Container-Klasse abgespeichert. Die Vor- und Zurück-Schaltflächen sind dabei deaktiviert.
- Mit Klick auf „Suchen nach Nummer...“ öffnet sich ein kleines Dialogfenster zur Eingabe der gewünschten ArtikelNr und sofern ein passender Artikel gefunden wird, werden dessen Daten angezeigt.
Um nebenstehenden Dialog anzuzeigen und dessen Rückgabewert entgegenzunehmen wird folgender Befehl verwendet:



```
sNummer = (String) JOptionPane.showInputDialog("Bitte ArtikelNr  
eingeben:");
```

Die Vor- und Zurück-Schaltflächen bleiben auch in diesem Fall deaktiviert.

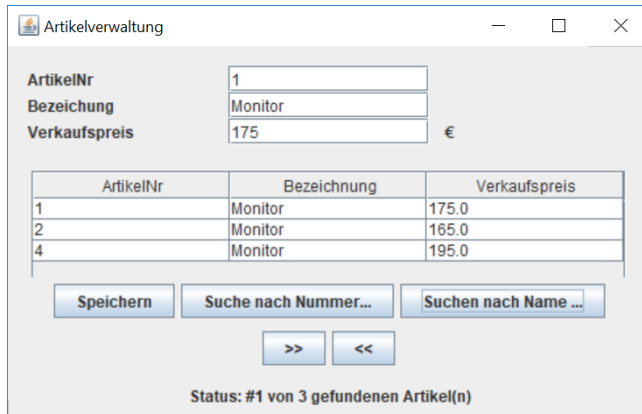
- Mit Klick auf „Suchen nach Name...“ öffnet sich ebenfalls ein entsprechendes Dialogfenster zur Eingabe der gesuchten Artikelbezeichnung. In diesem Fall kann es mehrere Artikel mit derselben Bezeichnung als Ergebnis geben. Falls mehr als ein Artikel gefunden wurde, werden die Vor- und Zurück-Schaltflächen aktiviert und mit jedem Klick darauf wird der nächste bzw. vorherige Artikel angezeigt. Das kann so gemacht werden, dass man nach dem letzten Artikel wieder zum ersten gelangt und umgekehrt.

Vorgehensweise:

- Erstellen Sie ein neues Package und kopieren Sie die Klassen von Moodle in das Package („KlassenArtikelContainer“).
- Erstellen Sie eine GUI-Klasse mit Hilfe des Eclipse Window Builder.
- Die Inhaltsebene enthält ein GridLayout mit 1 Spalte.
- Zeile 1 enthält ein JPanel mit einem GridLayout mit 3 Spalten für den Ein-/Ausgabebereich.
- Zeile 2 enthält ein JPanel für den Button-Bereich bestehend aus zwei JPanel für die obere und untere Button-Zeile.
Zeile 3 enthält nur ein JPanel mit JLabel für die Statusmeldungen
- Für die Ereignisverarbeitung bietet sich ein ActionListener in einer inneren Klasse an. Hier wird mit getSource() jeweils festgestellt von welcher Komponente das Ereignis ausgelöst wurde und dann können in einer entsprechenden Methode die notwendigen Aktionen ausgeführt werden.

2.4. Zusatzaufgabe für ganz Schnelle:

Erweitern Sie das Programm um eine Ausgabetabelle für den Fall, dass bei der „Suchen nach Name...“ mehr als ein Artikel gefunden wird.



ArtikelNr	Bezeichnung	Verkaufspreis
1	Monitor	175.0
2	Monitor	165.0
4	Monitor	195.0

- Falls danach „Speichern“ oder „Suchen nach Nummer..“ ausgeführt wird, soll die Tabelle wieder entfernt werden.
- Eine Tabelle, die nur Werte anzeigen soll, diese aber vom Benutzer nicht verändert werden können, ist relativ einfach zu erstellen. In Moodle liegt die Klasse `TestJTable.java`. Laden Sie diese herunter, testen diese und versuchen das Prinzip nachzuvollziehen.
- Versuchen Sie eine Ergebnistabelle in ihr Programm einzufügen.

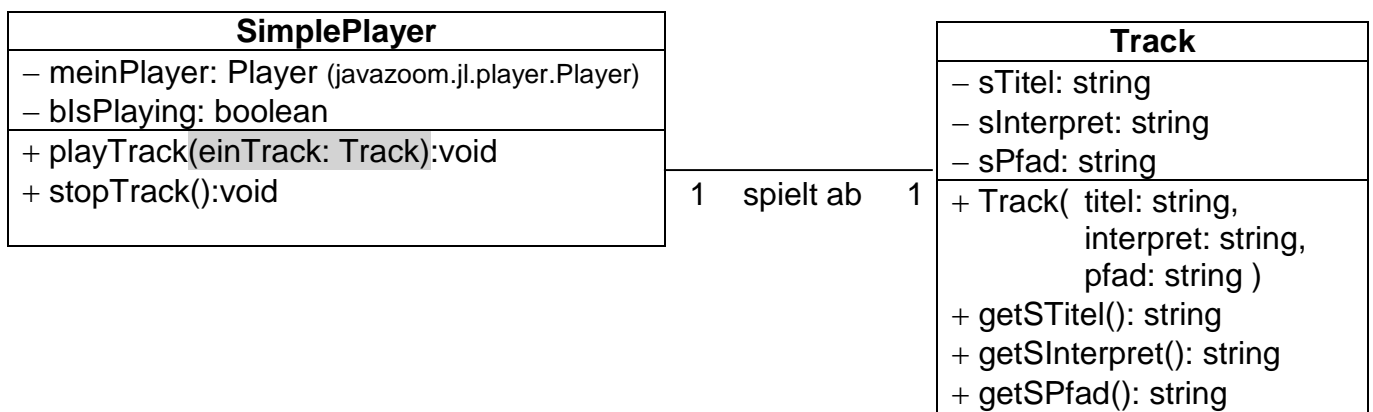
2.5. Einfacher mp3-Player mit kurzfristiger Assoziation

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

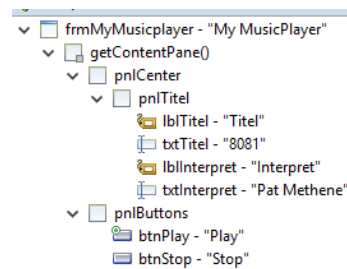
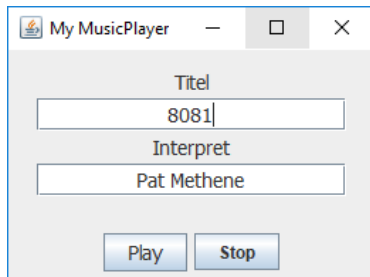
- Assoziationen können kurzfristig sein,
 - wenn ein anderes Objekt an das aktuelle Objekt als Parameter übergeben wird.
 - wenn ein Objekt in einer Methode einer anderen Klasse lokal erzeugt wird.
- Assoziationen können langfristig sein,
 - wenn das Objekt Referenzen auf die assoziierten Objekte speichert.

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`

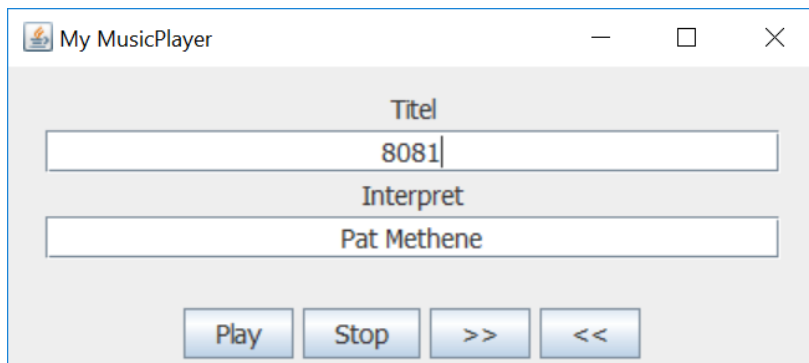
- Die Klasse verwendet die Bibliothek `j11.0.1.jar` (aus Moodle) zum Abspielen von .mp3 Dateien (s. <http://www.javazoom.net/javayer/sources.html>).
- Da die `playTrack` Methode relativ komplex ist, kann sie von Moodle kopiert werden (s. Moodle `Player_PlayTrack.txt`)
- Die Methode `stopTrack` prüft anhand des Attributs `isPlaying`, 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

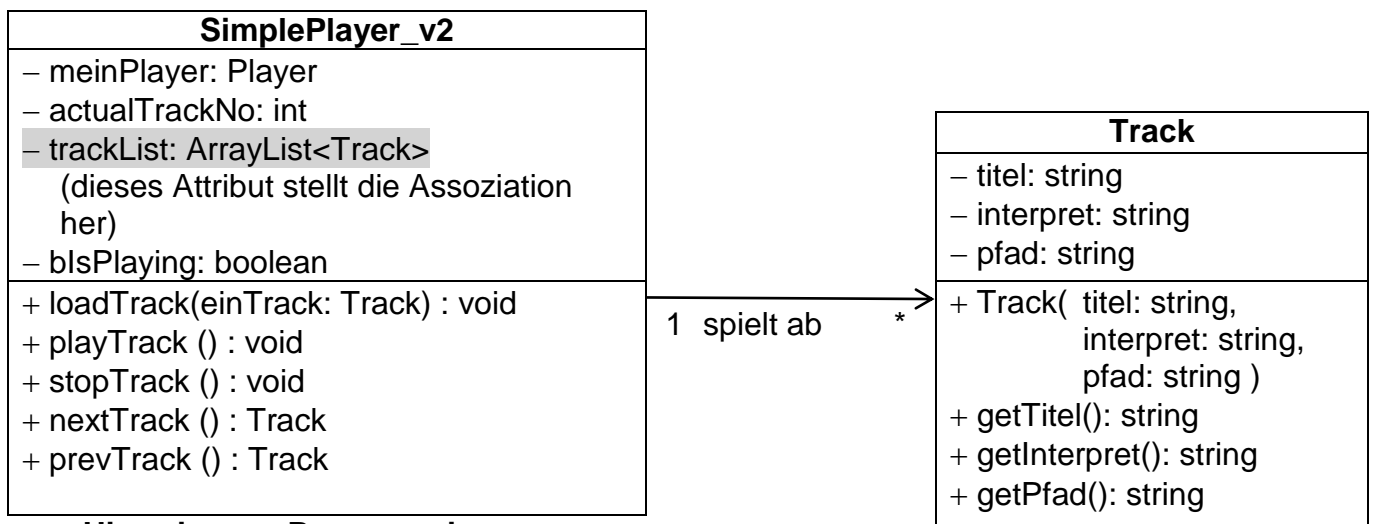
```
JOptionPane.showMessageDialog(frmMyMusicplayer, "mp3 Datei nicht gefunden!");
```

eine Fehlermeldung angezeigt werden.
- Programmieren Sie das Stop-Button-Ereignis um den Track zu beenden.

2.6. Verbessertes 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`

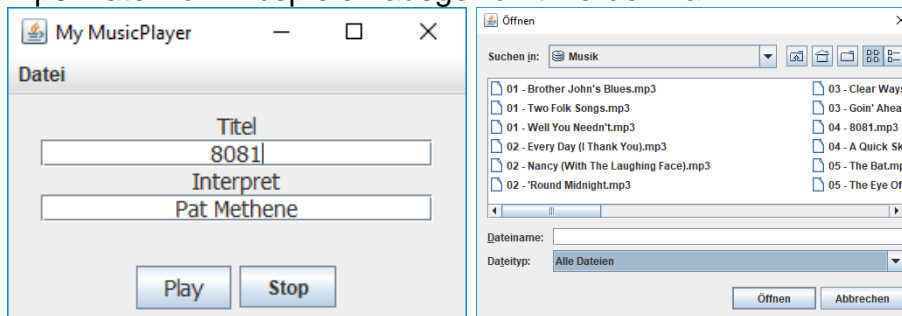
- Nehmen Sie als Ausgangspunkt für diese Klasse die alte Klasse `SimplePlayer`.
- Das Attribut `actualTrackNo` gibt an, welcher Track gerade ausgewählt ist.
- Die Methode `loadTrack()` fügt einen Track in `ArrayList` am Ende an.
Die `actualTrackNo` wird auf 0 gesetzt, d.h. der erste Track ist nach dem 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.

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.