Task 2 - Music Individual Class

This class establishes the object representation of a music sample (all 3 melodies together). The *Music* object consists of the following fields:

1. *melody1* - The list of *Note* objects representing the first melody, as defined by the specifications of Task 1.
2. *melody2* - The list of *Note* objects representing the second melody, as defined by the specifications of Task 1.
3. *melody3* - The list of *Note* objects representing the third melody, as defined by the specifications of Task 1.
4. *str-representation* - The EasyABC string representation of the music sample, such that it can be played in EasyABC via a direct copy/paste.

The *Music* class features seven methods for displaying melodies. These methods include the functionality to display melodies by themselves, in pairs, or altogether. The *demo–music-class* demonstrates the usage of these methods.

Demo

```
[2]> ( demo--music-class )

"----------------DISPLAY MELODY1 TEST----------------"
Melody 1: F2 B/2 A2 E2 F D2 E/2 F2 G/2 B2 A/2 B A B/2 B G/2 F2
C/2 D/2 A2 G/2 A G/2


"----------------DISPLAY MELODY2 TEST----------------"
Melody 2: E2 B G/2 B G/2 A F2 B/2 F A2 D/2 F2 A E/2 G/2 D2 F2
G/2 A/2 A2 B2 B/2 C/2
```

"----------------DISPLAY MELODY3 TEST----------------"
Melody 3: G, G4, D2, C4, B2, F4, G4, D2, E2, C,

"----------------DISPLAY MELODIES 1&2 TEST----------------"
Melody 1: F2 B/2 A2 E2 F D2 E/2 F2 G/2 B2 A/2 B A B/2 B G/2 F2 C/2 D/2 A2 G/2 A G/2
Melody 2: E2 B G/2 B G/2 A F2 B/2 F A2 D/2 F2 A E/2 G/2 D2 F2 G/2 A/2 A2 B2 B/2 C/2

"----------------DISPLAY MELODIES 1&3 TEST----------------"
Melody 1: F2 B/2 A2 E2 F D2 E/2 F2 G/2 B2 A/2 B A B/2 B G/2 F2 C/2 D/2 A2 G/2 A G/2
Melody 3: G, G4, D2, C4, B2, F4, G4, D2, E2, C,

"----------------DISPLAY MELODIES 2&3 TEST----------------"
Melody 2: E2 B G/2 B G/2 A F2 B/2 F A2 D/2 F2 A E/2 G/2 D2 F2 G/2 A/2 A2 B2 B/2 C/2
Melody 3: G, G4, D2, C4, B2, F4, G4, D2, E2, C,

"----------------DISPLAY ALL MELODIES TEST----------------"
Melody 1: F2 B/2 A2 E2 F D2 E/2 F2 G/2 B2 A/2 B A B/2 B G/2 F2 C/2 D/2 A2 G/2 A G/2
Melody 2: E2 B G/2 B G/2 A F2 B/2 F A2 D/2 F2 A E/2 G/2 D2 F2 G/2 A/2 A2 B2 B/2 C/2
Melody 3: G, G4, D2, C4, B2, F4, G4, D2, E2, C,
NIL
[3]>

Demo Code

```
; Method that demos the display methods of the music class
( defmethod demo--music-class ()
    ( setf melody1-notes ( generate-melody1 ) )
    ( setf melody2-notes ( generate-melody2 melody1-notes ) )
    ( setf melody3-notes ( generate-bassline ) )

    ( setf sample
        ( make-instance 'music
            :melody1 melody1-notes
            :melody2 melody2-notes
            :melody3 melody3-notes
            :str-representation ""
        )
    )

    ( print "----------------DISPLAY MELODY1 TEST----------------")
    (terpri)
    ( display-melody1 sample )
    (terpri)
    ( print "----------------DISPLAY MELODY2 TEST----------------")
    (terpri)
    ( display-melody2 sample )
    (terpri)
    ( print "----------------DISPLAY MELODY3 TEST----------------")
    (terpri)
    ( display-melody3 sample )
    (terpri)
    ( print "----------------DISPLAY MELODIES 1&2 TEST----------------")
    (terpri)
    ( display-melody1&2 sample )
    (terpri)
    ( print "----------------DISPLAY MELODIES 1&3 TEST----------------")
    (terpri)
    ( display-melody1&3 sample )
    (terpri)
    ( print "----------------DISPLAY MELODIES 2&3 TEST----------------")
```

```
    (terpri)
    ( display-melody2&3 sample )
    (terpri)
    ( print "----------------DISPLAY ALL MELODIES TEST----------------")
    (terpri)
    ( display-all-melodies sample )
)
```

Code

```
; Class for music representation
; - includes the list of notes for melody1, the list of notes for melody2,
;   the list of notes for melody3, and the string representation of all
;   melodies that can be copied/pasted to EasyABC to play the whole sample
( defclass music ()
    (
        ( melody1 :accessor music-melody1 :initarg :melody1 )
        ( melody2 :accessor music-melody2 :initarg :melody2 )
        ( melody3 :accessor music-melody3 :initarg :melody3 )
        ( str-representation :accessor music-str-representation :initarg
:str-representation )
    )
)

; Method that displays melody1 of a music sample
( defmethod display-melody1 ( (m music) )
    ( format t "Melody 1: ~{~a~^ ~}~%" ( display-notes-list (
music-melody1 m ) ) )
)

; Method that displays melody2 of a music sample
( defmethod display-melody2 ( (m music) )
    ( format t "Melody 2: ~{~a~^ ~}~%" ( display-notes-list (
music-melody2 m ) ) )
)

; Method that displays melody3 of a music sample
( defmethod display-melody3 ( (m music) )
```

```
    ( format t "Melody 3: ~{~a~^ ~}~%" ( display-notes-list (
music-melody3 m ) ) )
)

; Method that displays melody1 and melody2 of a music sample
( defmethod display-melody1&2 ( (m music) )
    ( format t "Melody 1: ~{~a~^ ~}~%" ( display-notes-list (
music-melody1 m ) ) )
    ( format t "Melody 2: ~{~a~^ ~}~%" ( display-notes-list (
music-melody2 m ) ) )
)

; Method that displays melody1 and melody3 of a music sample
( defmethod display-melody1&3 ( ( m music ) )
    ( format t "Melody 1: ~{~a~^ ~}~%" ( display-notes-list (
music-melody1 m ) ) )
    ( format t "Melody 3: ~{~a~^ ~}~%" ( display-notes-list (
music-melody3 m ) ) )
)

; Method that displays melody2 and melody3 of a music sample
( defmethod display-melody2&3 ( ( m music ) )
    ( format t "Melody 2: ~{~a~^ ~}~%" ( display-notes-list (
music-melody2 m ) ) )
    ( format t "Melody 3: ~{~a~^ ~}~%" ( display-notes-list (
music-melody3 m ) ) )
)

; Method that displays all melodies of a music sample
( defmethod display-all-melodies ( ( m music ) )
    ( format t "Melody 1: ~{~a~^ ~}~%" ( display-notes-list (
music-melody1 m ) ) )
    ( format t "Melody 2: ~{~a~^ ~}~%" ( display-notes-list (
music-melody2 m ) ) )
    ( format t "Melody 3: ~{~a~^ ~}~%" ( display-notes-list (
music-melody3 m ) ) )
)
```