音楽で楽しもう

琉球音階の自動作曲

沖縄の音階（琉球音階）を使って、プログラムで作曲してみよう。

琉球音階は、「ドレミファソラシ」の「レ」と「ラ」が存在しない音階である。次のプログラムでは、「琉球音階」という配列に、基本となる音階を入れた後で、それぞれの音をメロディオブジェクトに入れて演奏している。

```
琉球音階＝配列！"ド" "ミ" "ファ" "ソ" "シ" 作る。

僕の楽譜＝メロディ！作る。
僕の楽譜！（琉球音階！1 読む）追加。
僕の楽譜！（琉球音階！2 読む）追加。
僕の楽譜！（琉球音階！3 読む）追加。
僕の楽譜！（琉球音階！4 読む）追加。
僕の楽譜！（琉球音階！5 読む）追加。
僕の楽譜！演奏。
```

```
琉球音階＝配列！"ド" "ミ" "ファ" "ソ" "シ" 作る。

僕の楽譜＝メロディ！作る。
□□n|□僕の楽譜！（琉球音階□□n□読む）追加」！5回 繰り返す。
僕の楽譜！演奏。
```

配列である琉球音階からは、5種類の音を1から5の数字で取り出せる。乱数で1〜5の数を発生させることで、ランダムな音を演奏することも可能である。

```
琉球音階＝配列！"ド" "ミ" "ファ" "ソ" "シ" 作る。

僕の楽譜＝メロディ！作る。
「僕の楽譜！（琉球音階！（乱数（5））読む）追加」！5回 繰り返す。
僕の楽譜！演奏。
```

次のプログラムでは、乱数でAメロディとBメロディという短い旋律を作り、それらを組み合わせて曲を作っている□Aメロディ6個の音を、Bメロディは8個の音を、それぞれ乱数でつなげて作っている。そして□Aメロディの最後には「ド〜」を加えている。曲の構成としては□Aメロディを2回演奏し、Bメロディを演奏した後、最後にAメロディをもういちど演奏して終る。楽器には「マリンバ」を設定した。

```
琉球音階＝配列！"ド" "ミ" "ファ" "ソ" "シ" 作る。

Aメロディ＝メロディ！作る。
□Aメロディ！（琉球音階！（乱数（5））読む）追加」！6回 繰り返す。
Aメロディ！"ド〜" 追加。

Bメロディ＝メロディ！作る。
□Bメロディ！（琉球音階！（乱数（5））読む）追加」！8回 繰り返す。

僕の楽譜＝メロディ！作る。
「僕の楽譜□□Aメロディ）追加」！2回 繰り返す。
僕の楽譜□□Bメロディ）追加。
```

```
僕の楽譜□□Aメロディ）追加。

僕の楽器＝楽器！"マリンバ"　作る。
僕の楽譜！（僕の楽器）設定。
僕の楽譜！演奏。
```

ランダムな音階の自動作曲

今度は、一定範囲の音をランダムに作り出してみよう。ここでは、2オクターブの音階の音をランダムに演奏することにする。

次のプログラムでは、最初に乱数で作る音の基準となる音程を決めている。ここでは「ド」にした。そして音程を書いていくためのメロディを作り、「楽譜」という名前にしている。続いて、基準の音から2オクターブ分の音をランダムに作り、楽譜に50回追加している。「乱数（24）」は1〜24の値をランダムに返す。「音上げる」は指定された数だけ半音ずつ音程を上げる。最大24個の半音なので、2オクターブ上の音階まで表現できることになる。最後に、作った楽譜を演奏している。

```
音程＝メロディ！作る"ド"追加。
楽譜＝メロディ！作る。
「楽譜！（音程!（乱数（24））音上げる）追加」！50回 繰り返す。
楽譜！演奏。
```

フレーズを利用した自動作曲

配列に複数の音を入れて演奏してみよう。このプログラムは、複数パートの曲を作る場合にも応用できる。

次のプログラムでは、最初にメロディの元になるフレーズを配列に格納している。これらのフレーズは、演奏したときに同じ長さになるように揃えておく。次に、パートごとにメロディを作り、楽器を割り当てる。ここでは楽器番号で指定した。続いて、パートごとにランダムにフレーズをつなげてメロディーを作る。最後に各パートを集めてバンドを作り、テンポを決めて演奏している。

```
フレーズ＝配列！"ドレミファミレド〜""ミファソラソファミ〜""ド・ド・ド・ド・""ド8ド8レ8レ8ミ8ミ8ファ8ファ8ミレド〜"作る。

パート１＝メロディ！作る（楽器！10作る）設定。
パート２＝メロディ！作る（楽器！11作る）設定。
パート３＝メロディ！作る（楽器！12作る）設定。

「パート１！（フレーズ！（乱数（4））読む）追加」！8　繰り返す。
「パート２！（フレーズ！（乱数（4））読む）追加」！8　繰り返す。
「パート３！（フレーズ！（乱数（4））読む）追加」！8　繰り返す。

合奏＝バンド！（パート１）（パート２）（パート３）作る（68）テンポ　演奏。
```

リズムと組み合わせた自動作曲

最後に、メロディとリズムを組み合わせた自動作曲をしてみよう。プログラムを分かりやすくするために、ここでは作曲する部分をメロディオブジェクトとドラムオブジェクトにメソッドとして定義している。作成する曲は、笛と太鼓の二重奏である。

次のプログラムでは、最初に笛の楽器番号と曲の速さ、音符の数の変数を定義している。このように、調整したい値を変数として定義しておくと、後から変更するときに便利である。続いて、笛と太鼓用の音階の配列を用意している。

作曲は、メロディとドラムに定義したメソッドで行う。笛のメロディは、音符として7個の音階をランダムに選び、その長さをランダムに決めたものを繰り返すようにした。「↑」は「^」と同じ意味で、続く音階が1オクターブ上がる。「↑↑」は2オクターブ、「↑↑↑」は3オクターブ上がることを示す。太鼓のリズムは、3個の打楽器をランダムに選び、その長さをランダムに決めている。太鼓ではリズムを細かく刻むために、音符の長さは笛の半分（8分音符）に設定し、音符の数を倍に設定した。最後に笛と太鼓でバンドを作り、指定した曲の速さで演奏している。

```
笛楽器＝072。曲の速さ＝120。音符の数＝50。
笛の音＝配列！"↑↑レ""↑↑ミ""↑↑ファ""↑↑ラ""↑↑シ""↑↑↑レ""↑↑↑ミ" 作る。
太鼓の音＝配列！"ど""た""つ" 作る。
笛＝メロディ！作る（楽器！（笛楽器）作る）設定。
太鼓＝ドラム！作る。

笛：作曲□□□n| □
  音符＝笛の音！（乱数（7））読む。
  「乱数（2）>1」！なら「自分！（音符＋"4"）追加」そうでなければ「自分！（音符＋"2"）追加」実行。
 □□ □n□ 繰り返す」。
太鼓：作曲□□□n| □
  音符＝太鼓の音！（乱数（3））読む。
  「乱数（2）>1」！なら「自分！（音符＋"8"）追加」そうでなければ「自分！（音符＋"4"）追加」実行。
 □□ □n□ 繰り返す」。

笛パート＝笛！（音符の数）作曲。
太鼓パート＝太鼓！（音符の数＊2）作曲。
バンド！作る（笛パート）追加（太鼓パート）追加（曲の速さ）テンポ　演奏。
```

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/ch_music_compose33?rev=1578222403

Last update: **2020/01/05 20:06**