

# 移調のためのプログラム

慶應義塾大学大学院 理工学研究科  
開放環境科学専攻 後期博士課程

横 山 暁

2004 年 4 月 23 日

## 1 Introduction

先月の研究会の際、西川氏が「J 言語を音楽にコードネーム移調のための tiny tool」というプログラムを紹介した。そこで、西川氏とは違った観点で移調のためのプログラムを作成してみたので紹介する。

## 2 Algorithm

### 2.1 入出力

プログラムは trans という名前であり、まず、# や  $b$  の記号 (# と  $b$ )、およびその数 (C dur, A moll の時には 0) と、変換したい音のドイツ音名を左引数に入力する。さらに変換後の記号およびその数 (C dur, A moll の時には 0) を右引数に入力する。(例えば、'#1C' trans '0') 実行すると、変換後のドイツ音名が出力される。

### 2.2 入力処理

入力された左引数を記号、数、ドイツ音名に分離する。これをそれぞれ、 $b_1, b_2, b_3$  に代入する。

$b_3$  に代入されたドイツ音名により num に数値を割り当てる。この数値は C が 0 で、半音ごとに 1 ずつ増えていく。(表 1 参照)

表 1 ドイツ音名と割り当てる数値

ドイツ音名	His,C	Cis,Des	D	Dis,Es	E,Fes	Eis,F	Fis,Ges	G	Gis,As	A	Ais,B	H,Ces
数値	0	1	2	3	4	5	6	7	8	9	10	11

さらに、b1 に代入された記号により、# であれば、b2 に代入された数 × 5 を、  
であれば b2 に代入された数 × 7 を、tb に代入する。C dur,A moll の時には何もしない。

そして、num-tb を計算し、その値を before\_num に代入する。

b3 に代入されるドイツ音名であるが、C,D,E,F,G,A,H およびその # および の 21 音名を  
きちんと認識させるため、スマートな方法ではないが文字列の長さによって場合分けを行っ  
ている。さらに、#系か 系かによってフラグを立てておく。

## 2.3 出力処理

右引数も左引数と同様に認識をさせている。a1 に記号を、a2 に数を代入し、記号によっ  
て # であれば、a2 に代入された数 × 5 を、  
であれば a2 に代入された数 × 7 を、ta に代  
入する。

最後に、before\_num+ta を行い result に代入する。ただし result の値が 0 以上 12  
未満になるように調整を行う。

この値と、フラグの値を元に、変換後のドイツ音名を出力する。出力されるドイツ音名が  
#系 系の 2 種類ある時、フラグが 1、つまり変換前のドイツ音名が #系であれば、#系の  
ドイツ音名を、フラグが 2、つまり返還前のドイツ音名が 系であれば、系のドイツ音名  
を出力するようになっている。

## 2.4 問題点

このプログラムにはいくつかの問題点があると考えられる。

例えば、ダブルシャープやダブルフラットの処理ができないこと、D dur の時の Fis の音  
を Ges と入力し、C dur で出力する時に Fes と出力されず E と出力されることなどといっ  
た問題である。

この問題は、フラグのとり方を細かくすれば回避できる可能性があるが非常に煩雑になる  
と予想されるため、今回は実装を見送った。

また、別個にドイツ音名を認識させているためプログラムがスマートで無いという問題も  
あると考えられる。

## 3 Application

いくつかの実行例を示す。

```
'#1C' trans '0'  
F  
'#1C' trans '#2'  
G
```

```
'0Fis' trans '#1'  
Cis  
'b2G' trans '#2'  
H  
'b2Fis' trans '#1'  
Dis
```

## 付録 A Script

```
NB. =====  
NB. Transpose of Music  
NB. This program is designed for J504b  
NB. Modified by Satoru Yokoyama 2005/03/30  
NB. =====  
  trans=:4 : 0  
before=:x.  
after=:y.  
tb=:before_num=:num=:frag=:0  
b1=:0{before  
b2=:".1{before  
b3=:2}.before  
if. b1='#' do.  
  tb=:tb+b2*7  
elseif. b1='b' do.  
  tb=:tb+b2*5  
elseif. b1='0' do.  
  b3=:1}.before  
end.  
if. (#b3)=1 do.  
  if. b3='C' do. num=:0  
  elseif. b3='D' do. num=:2  
  elseif. b3='E' do. num=:4  
  elseif. b3='F' do. num=:5  
  elseif. b3='G' do. num=:7  
  elseif. b3='A' do. num=:9  
  elseif. b3='B' do.  
    num=:10  
    frag=:2  
  elseif. b3='H' do. num=:11  
  end.  
elseif. (#b3)=2 do.  
  if. */b3='Es' do.  
    num=:3  
    frag=:2  
  elseif. */b3='As' do.  
    num=:8  
    frag=:2  
  end.  
end.
```

```

elseif. (#b3)=3 do.
  if. */b3='His' do.
    num=:0
    frag=:1
  elseif. */b3='Cis' do.
    num=:1
    frag=:1
  elseif. */b3='Des' do.
    num=:1
    frag=:2
  elseif. */b3='Dis' do.
    num=:3
    frag=:2
  elseif. */b3='Fes' do.
    num=:4
    frag=:2
  elseif. */b3='Eis' do.
    num=:5
    frag=:1
  elseif. */b3='Fis' do.
    num=:6
    frag=:1
  elseif. */b3='Ges' do.
    num=:6
    frag=:2
  elseif. */b3='Gis' do.
    num=:8
    frag=:1
  elseif. */b3='Ais' do.
    num=:10
    frag=:1
  elseif. */b3='Ces' do.
    num=:11
    frag=:2
  end.
end.
before_num=:num-tb

ta=:result=:0
a1=:0{after
if. a1='#' do.
  a2=:".1{after
  ta=:ta+a2*7
elseif. a1='b' do.
  a2=:".1{after
  ta=:ta+a2*5
elseif. a1='0' do. ta=:0
end.

result=:before_num+ta
while. result>11 do.

```

```
result=:result-12
end.
while. result<0 do.
result=:result+12
end.
if. result=0 do. ans=: 'C'
elseif. result=1 *. frag=2 do. ans=: 'Des'
elseif. result=1 do. ans=: 'Cis'
elseif. result=2 do. ans=: 'D'
elseif. result=3 *. frag=1 do. ans=: 'Dis'
elseif. result=3 do. ans=: 'Es'
elseif. result=4 do. ans=: 'E'
elseif. result=5 do. ans=: 'F'
elseif. result=6 *. frag=2 do. ans=: 'Ges'
elseif. result=6 do. ans=: 'Fis'
elseif. result=7 do. ans=: 'G'
elseif. result=8 *. frag=1 do. ans=: 'Ges'
elseif. result=8 do. ans=: 'As'
elseif. result=9 do. ans=: 'A'
elseif. result=10 *. frag=1 do. ans=: 'Ais'
elseif. result=10 do. ans=: 'B'
elseif. result=11 do. ans=: 'H'
end.
ans
)
```