メディアプログラミング演習―第７回（第３テーマ２日目）―

音関連の課題２

課題１では、簡単なキーボードを作成した。課題２では、電子的「楽譜」の演奏システムを作成する。単純化して、振幅（大きさ）一定な正弦波を発生する（純音）楽器一つを想定する。

(A)音データの生成

正弦波（サインカーブ(sine curve)での＜音データ＞は、

```
sine = new SineWave(440, 0.5, out.sampleRate());
```

により、440Hz の正弦波が生成できる.

(B)音階

Hz	262	294	330	349	392	440	494	523	587	659	699	784	880	988
音階	ド	レ	ミ	ファ	ソ	ラ	シ	ド	レ	ミ	ファ	ソ	ラ	シ
								オクターブ上						

ド レ ミ ファ ソ

音階記号　－＞S　A　B　C　D　E　F　G　a　b　c　d　e　f　g

(C)指定時間の遅延

指定した時間（ミリ秒で指定）遅延させるには、以下の関数を呼び出す。

```
void delaymsec(int m){
    int ms=millis();
    while( millis()<(ms+m) ) {};
  }
```

(D)音の発生と停止

(A)で生成した音データを、ＰＣの音源発生に送り、音を鳴らす関数が

`out.addSignal(＜音データ＞)` であり、

停止させる関数が、`out.removeSignal(＜音データ＞)`

である。

(E)サンプルプログラム

以上より、

```
sine = new SineWave(440, 0.5, out.sampleRate());音データ（ラ）の生成
out.addSignal(sine);        「低いラ」を鳴らす
delaymmsec(500);            0.5 秒待つ
out.removeSignal(sine);     音を停止する
delaymmsec(500);            0.5 秒待つ
```

により、0.5 秒ラを鳴らし、0.5 秒無音となる。これを含むサンプルプログラムが、

`ss-jihou.txt` である。

（F）楽譜のデータ構造

音は、音程記号と音符記号番号のペアの列であるので、

`char tone[]` および `int note[]`

を利用することとする。

音程記号は、前述の（B）に従い、`'S', 'A'` ～`'G', 'a'` ～ `'g'`、および、`'P'`：休止、`'T'`：終了、とする。音符記号番号は, 16 分音（休）符を 1 とし、その倍数とする。たとえば、4 分音符は 4、8 分音符は 2、符点 8 分音符は 3 となる。

```
char   tone= {'F','P','F','P','F','P','f','P','T' }
int    note= {  2,  2,  2,  2,  2,  2,  4,  0, 0  }
```

により、サンプル `ss-jihou` と同じ曲？を表す楽譜となる。

再生の速さ（ `int speed`）は、16 分音符の時間間隔とする、たとえば、

♩=60 の場合は、<60000: 分のミリ秒 >/<60: 4 分音符の数 >/<4: 16 分音符を単位>

とし、実際の時間間隔は、speed*note[] となる。

(H)再生プログラム

まず、音階記号と周波数の対応表（前述（B））が必要である

（`char ontei[], int Freq[]`）.

(1)プログラムの構造としては、以下の構造となる。

```
while(  ＜音＞  ) {   ＜音＞の処理  }  ;
```

具体的には、以下の流れとなる。

```
k=0;
while( tone[k]!='T' )
   {  ＜＜この部分に＜音＞の処理＞＞ ; k++;};
```

(2)続いて、＜音＞は、音階記号または休止記号 P であるので、

```
if( tone[k] == 'P' ) {  ＜休止の処理＞ }
  else{  ＜音階記号の場合の処理＞  };
```

の形となる。

(3)＜休止の処理＞は、休符であるので、

```
delaymsec( note[k]*speed );
```

となる。

(4) また、＜音階記号の場合＞は、

①音階記号 tone[k]を ontei[]から探し、対応する周波数を求め、

②その音データを作成、

③その音を鳴らし、⑤note[k]*speed 待ち、音を止める。

よって、

① `i=0; while( tone[k] != ontei[i] ) { i++;};`

② `sine = new SineWave(Freq[i] , 0.5, out.sampleRate());`

③ `out.addSignal(sine); delaymsec( note[k]*speed );`

となる。

課題

上記を参考に `ss-play-note` を完成させなさい。また、適切な楽曲の楽譜を作り、再生しなさい。