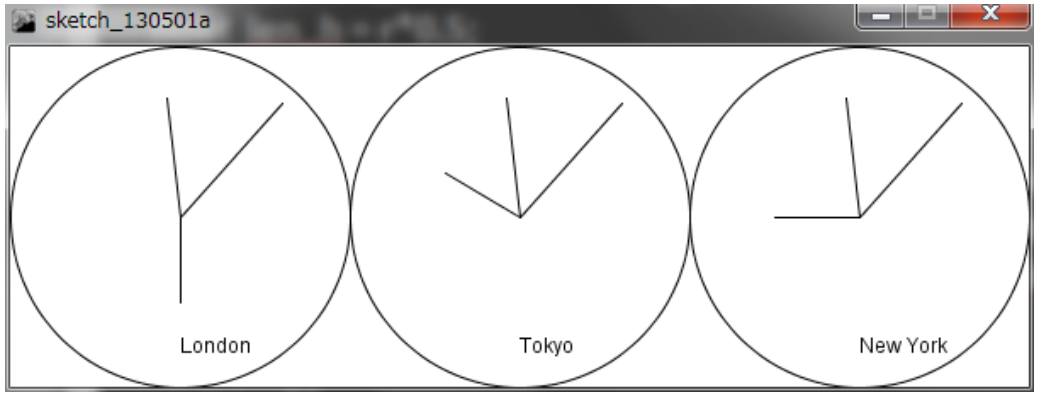
プログラミング演習I （第10回） 課題

- **基本①　スケッチ名： analogueClock3**
  - LondonとTokyoとNew Yorkの時計を表示するアナログ時計を横に並べてみましょう（時差考慮）
    - サマータイムで日本とロンドン間の時差が+8時間，日本とNYの間の時差が-13時間
  - アナログ時計を描く関数（ただし，この関数は下記のように整数値で中心座標(x,y)，半径(r)，時分秒(h,m,s)と，文字列で場所(location)を引数とし，時差については引数で加算減算せよ）を作成し，その関数を利用して3つの時計を描画するようにせよ

```
void analogueClock(int x, int y, int r, int h, int m, int s, String location);
```

```
// 中心(200,200)に半径80で，時差がないTokyoのアナログ時計を表示する場合
analogueClock(200, 200, 80, hour(), minute(), second(), "Tokyo" );
// 中心(600,200)に半径60で，時差1時間のSydneyのアナログ時計を表示する場合
analogueClock(150, 200, 60, hour()+1, minute(), second(), "Sydney" );
```



プログラミング演習I （第10回） 課題

- **基本②　スケッチ名：getLastDay**
  - 引数として入力された年，月から，その月の最終日を整数値として返す関数を作成せよ
  - 1，3，5，7，8，10，12月は31日，4，6，9，11月は30日が最終日である
  - 2月は通常28日が最終日だが，年が4で割り切れる場合に29日が最終日となり，さらに100で割り切れる場合に28日が最終日となり，さらに400で割り切れる場合に29日が最終日となる
  - 下記のプログラムを setup 内に入力し，標準出力を確認せよ

```
void setup(){
  for( int i=1; i<=12; i++ ){
    println( getLastDay( 2018, i ) );
  }
  for( int i=1990; i<=2010; i++ ){
    println( getLastDay( i, 2 ) );
  }
  for( int i=0; i<=2100; i+= 100 ){
    println( getLastDay( i, 2 ) );
  }
}
```

プログラミング演習I（第10回） 課題

- **基本③　スケッチ名：lifegame_func**
  - 誕生、生存、過疎、過密によってセルが生まれたり死んだりするライフゲームを関数で作ろう。
  - ライフゲームでは、対象とするセルの周囲8マスが生きているか死んでいるかを数え、その結果に応じてセルを生きている状態にするか、死んでいる状態にするかを切り替をえる。
  - 配布する lifegame_func.pde の checkNextDeadOrAlive 関数を完成させてライフログゲームを完成させよ
  - ただし，checkNextDeadOrAlive 関数は，調べたいセルの（x，y）座標を引数とし，関数の返り値として，次の状態が「生」の場合は1，「死」の場合は0を返すものとせよ
  - 下記URLの安定状態が幾つか観測されたら成功

    http://ja.wikipedia.org/wiki/ライフゲーム

プログラミング演習I （第10回） 課題

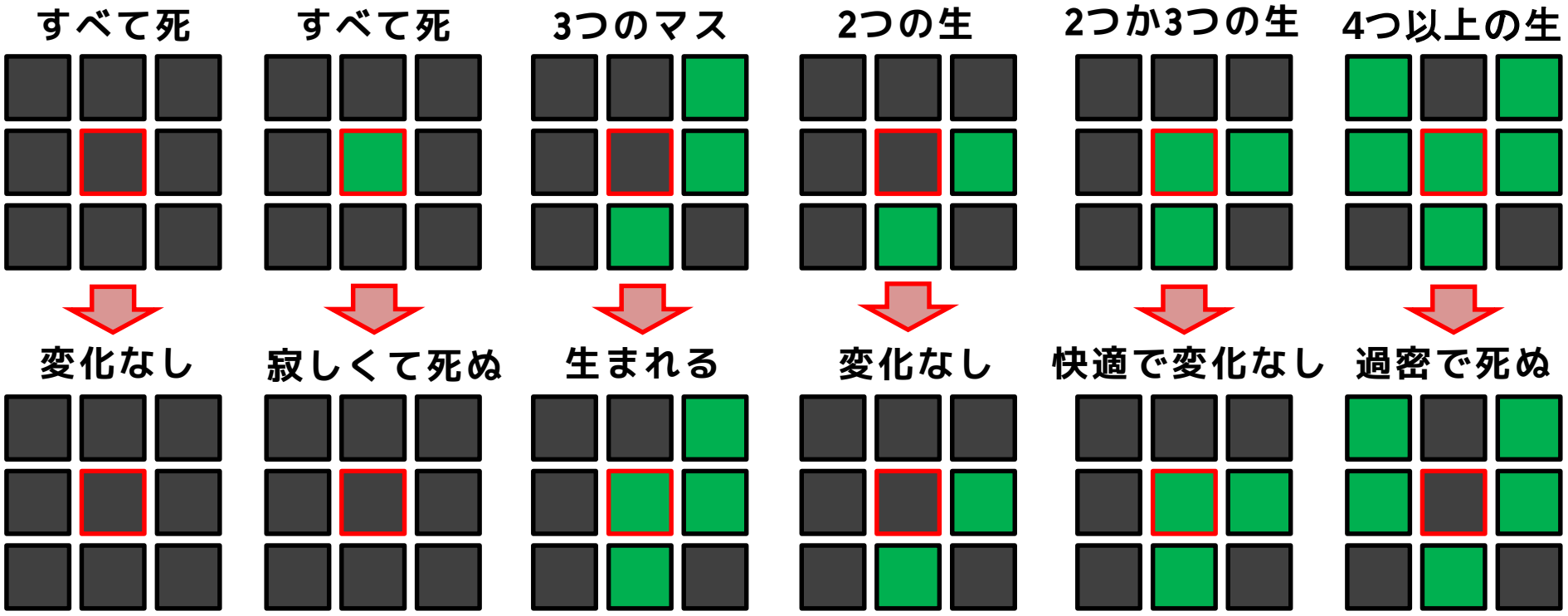
あるマス（赤フレーム）の縦・横・斜めの8マスの生死の状態（生の数）に注目する

【誕生】 死んでいるセルに隣接する生きたセルがちょうど3つならば次世代が誕生

【生存】 生きているセルに隣接する生きたセルが2つか3つならば次世代でも生存

【過疎】 生きているセルに隣接する生きたセルが1つ以下ならば過疎により死滅

【過密】 生きているセルに隣接する生きたセルが4つ以上ならば過密により死滅

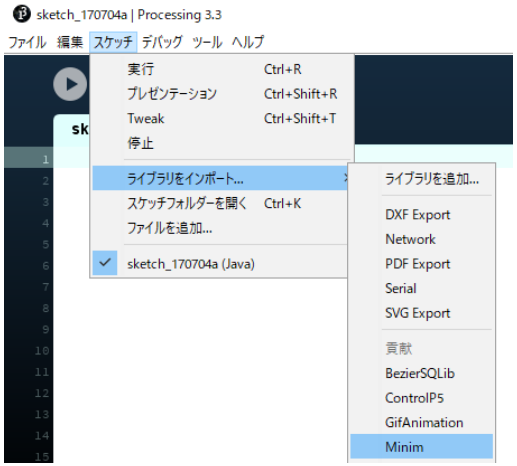


プログラミング演習I （第10回） 課題

- **発展①　スケッチ名：DoReMi**
  - 配布したプログラムの幹音（ドレミファソラシド）となる周波数を求める関数 getFrequency を作成し、キーボード操作の上下によって音程を上下させるプログラムを完成させよ。
  - まず準備段階としてminimを環境に導入せよ（後述）
  - 関数の引数は幹音のIDとし、返り値はその周波数の値（float）とせよ。
    - ド 261.6Hz, レ 293.7Hz, ミ 329.6Hz, ファ 349.2Hz
    - ソ 392.0Hz, ラ 440.0Hz, シ 493.9Hz
  - 幹音のIDが0のときは261.6Hzのド、1のときは293.7Hzのレとなるようにすること。なお、1オクターブ上がるとそれぞれ周波数は2倍に、1オクターブ下がると周波数は1/2倍になる。
  - 少なくとも3オクターブ分の結果を返すようにせよ。音はある程度聞こえていればOK。

minimの利用方法

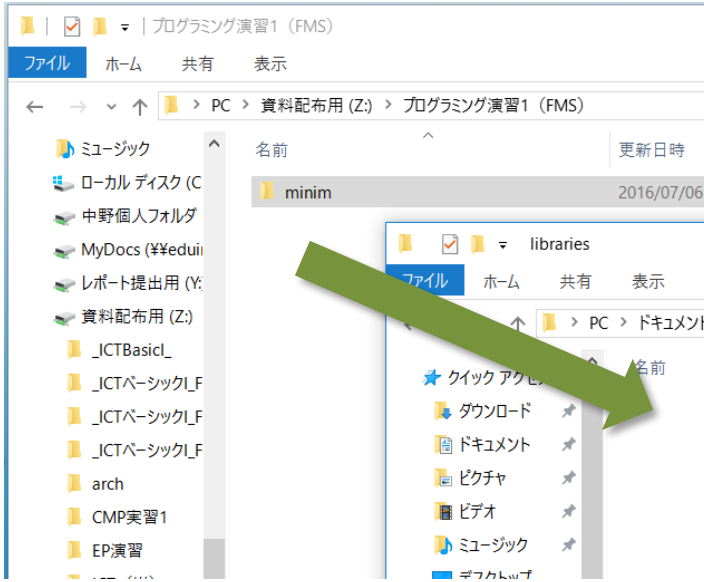
- スケッチ > ライブラリをインポート > Minim で利用できます
  - ない環境ではライブラリを追加しよう！
  - ライブラリを追加で基本的にはできるはず



- 無理な場合はProcessingを終了し、資料配布用（Zドライブ）のプログラミング演習1（FMS）フォルダから

Xドライブ:¥ドキュメント¥Processing¥library

に minim というフォルダをコピーする！



プログラミング演習I （第10回） 課題

- **発展②　スケッチ名：MaclaurinCos**
  - cos x は下記の式にマクローリン展開可能（テイラー展開のa=0のもの）で多項式近似可能である。このマクローリン展開を行うための関数Maclaurinを完成させよ。これを用いてcosx と近似できることを図示せよ。
  - ただし、Maclaurin関数は、入力をxの値と項数（n）とし、その時の値を返すようにせよ。
  - $cosx = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots$
  - $f(x,n) = (-1)^n \frac{x^{2n}}{(2n)!}$ とすると
  - $Maclaurin(x,n) = \sum_{i=0}^{n} f(x,i)$ となる