

プログラミング演習I (第10回) 課題

• 基本① 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② スケッチ名：**Zeller**

- 引数として入力された年, 月, 日 (いずれも整数で, 年は1582以上とする) から, 曜日を判定し, 曜日を数値として出力する関数 `calcDayWeek` を作成せよ.
- ただし, Zellerの公式では下の表の「Zeller」行のように, 土曜日だと0, 日曜日だと1...となるため, 下の表の「出力」行のように日曜日だと0, 月曜日だと1と返すようにせよ
- なお, 曜日を求めるにはツェラーの公式を利用せよ
- <https://ja.wikipedia.org/wiki/ツェラーの公式>
- さらに, setup内にて, 下記の関数呼び出しを行え

曜日	<u>日</u>	<u>月</u>	<u>火</u>	<u>水</u>	<u>木</u>	<u>金</u>	<u>土</u>
Zeller	1	2	3	4	5	6	0
出力	0	1	2	3	4	5	6

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**

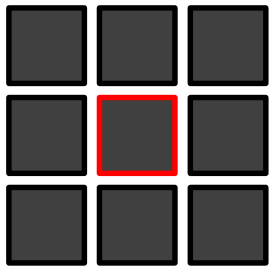
- 誕生、生存、過疎、過密によってセルが生まれたり死んだりするライフゲームを関数で作ろう。
- ライフゲームでは、対象とするセルの周囲8マスが生きているか死んでいるかを数え、その結果に応じてセルを生きている状態にするか、死んでいる状態にするかを切り替える。
- 配布する lifegame_func.pde の checkNextDeadOrAlive 関数を完成させてライフログゲームを完成させよ
- ただし、checkNextDeadOrAlive 関数は、調べたいセルの (x, y) 座標を引数とし、関数の戻り値として、次の状態が「生」の場合は1、「死」の場合は0を返すものとせよ
- 下記URL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

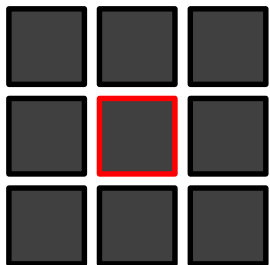
あるマス (赤フレーム) の縦・横・斜めの8マスの生死の状態 (生の数) に注目する

- 【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

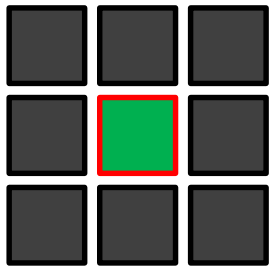
すべて死



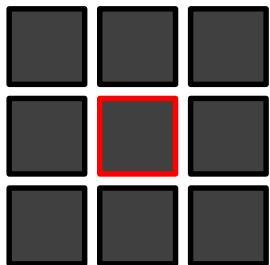
変化なし



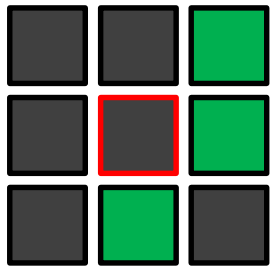
すべて死



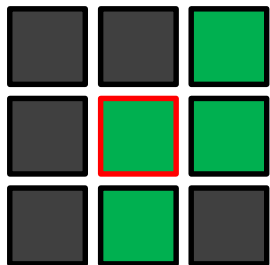
寂しくて死ぬ



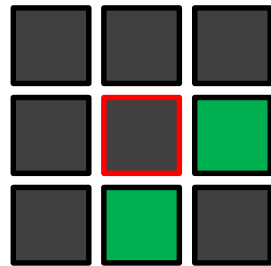
3つのマス



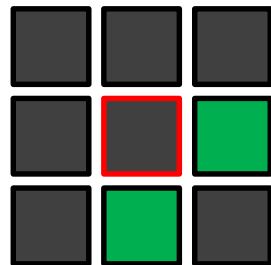
生まれる



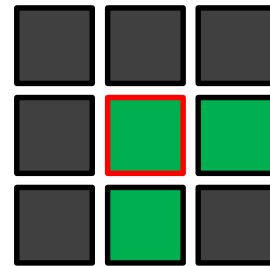
2つの生



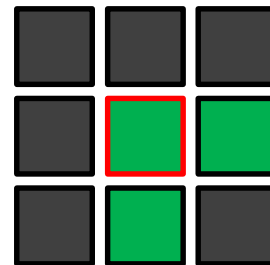
変化なし



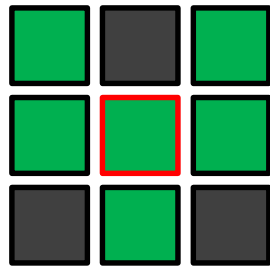
2つか3つの生



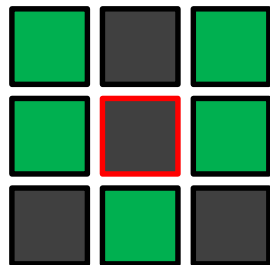
快適で変化なし



4つ以上の生



過密で死ぬ



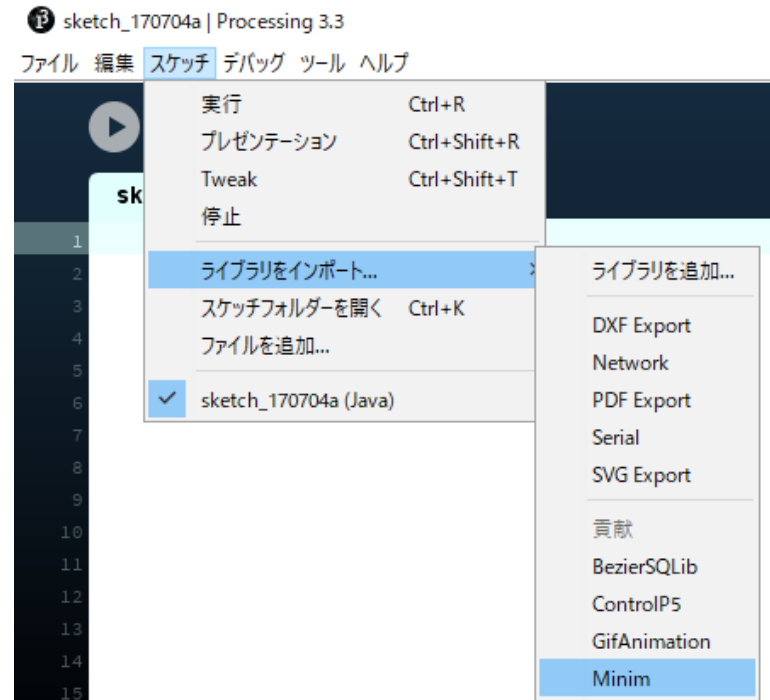
プログラミング演習I (第10回) 課題

• 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 392.0 Hz, ラ 440.0, 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



- ない自分の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！

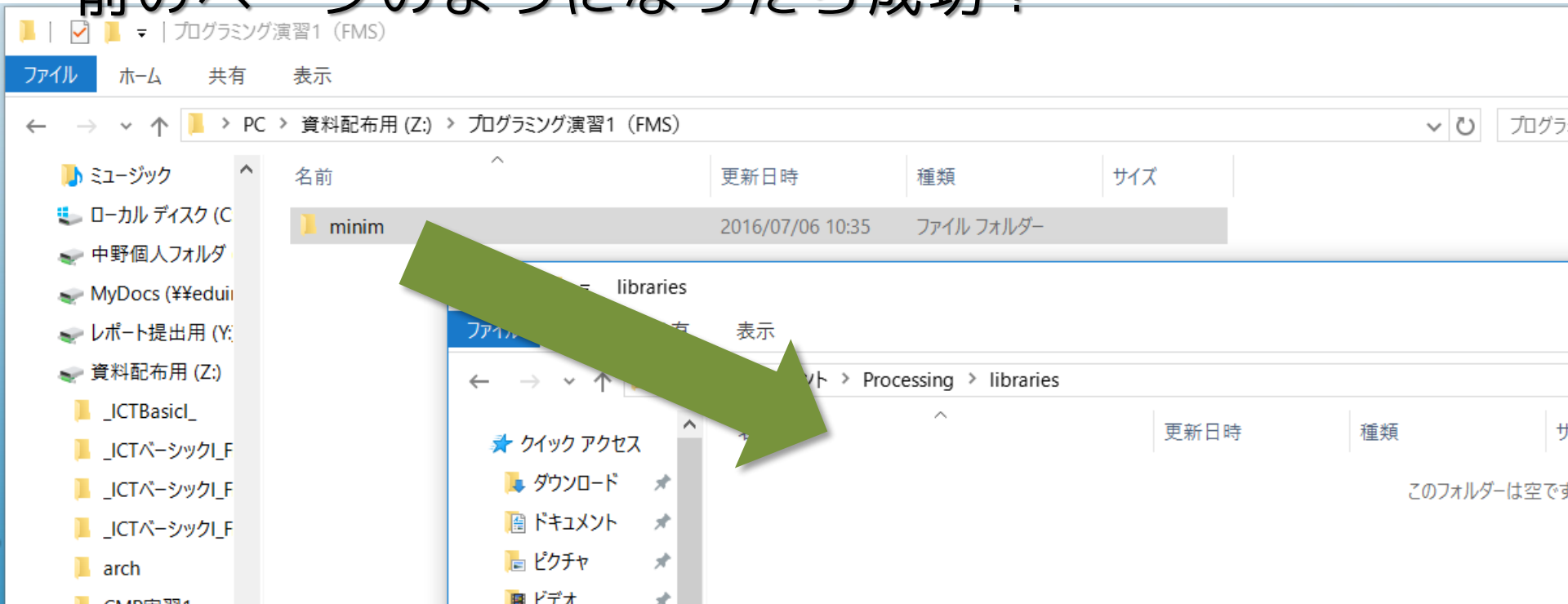
無理な場合は…

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！

前のページのようにになったら成功！



プログラミング演習I (第10回) 課題

• 発展② スケッチ名：showCalendar

- 基本課題①, ②で作成した関数と, 過去の課題で取り組んだカレンダーを表示するプログラムを組み合わせ, 任意の年月(ただし, 年の値は1582以上とする)のカレンダーを年月と曜日込みで標準出力するプログラムを作成せよ
- なお, プログラムでは年月が関数の引数として与えられると右図のように出力するようにせよ. 右図は下記の出力例である

- `showCalendar(2018, 5);`
- `showCalendar(2018, 2);`
- `showCalendar(2016, 1);`

2018年5月

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

2018年2月

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

2016年1月

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

確認テスト実施について

- 時間： 7月18日 13:30～15:10
- 場所： 203, 206教室

- 過去の基本課題の中から, 2問出します
 - 今年度は, 出す課題についても限定します
- テスト中は, 参考資料や, 過去に作成したファイル, インターネットを参照することはできません
 - 操作ログをチェックして確認する予定です
- 最終課題点 (30点) のうち, 15点をここで獲得することができます (残りの15点は発表会点)

確認テスト対象課題

- 第1回よりLaos
- 第2回よりmouseOutput
- 第3回よりcircle
- 第4回よりgachaoutput
- 第5回よりdice
- 第6回よりcalendar
- 第7回よりcafewall
- 第8回よりsort_chart
- 第9回よりkeijiban
- 第10回よりgetLastDay
- 第11回より（11日に案内します）

プログラミング演習発表会

- 7月23日 19:00-22:00 (5F ホール)
- 進め方
 - 研究室内でリーダーを決め、できるだけリーダーのPCにプログラムを集めてプレゼンを行う
 - 必ず事前にノートPCで動作を確認しておくこと (本番で動かなかつたら点数はありません)
 - その場で順序を決めその順序に応じて発表を行うこと
 - 発表時間は交代も含めて【**90秒厳守**】
 - 早く終わったからといって減点はしません
- 提出締め切り
 - プログラムはいつもの定められたフォルダに【**24日の15時まで**】に提出すること。
 - 提出が確認できない場合は発表点がつきません

レギュレーション

- Processingにより 「実世界のモノを模倣せよ」
 - 実世界の何らかの模倣であること (3点)
 - 動き (アニメーション) があり, マウスまたはキーボードの入力に対して, 何らかの反応 (動きなど) をするインタラクティブなものであること (3点)
 - プログラムは適切にインデントし, コメントを入れること (3点)
 - 昨年度までの条件分岐, 繰り返し, 配列, 関数を使えというレギュレーションはなし (確認テストで判定)
 - 組, 番号, 名前, 何の模倣かと, こだわりポイントを発表すること (3点)
 - プログラムが動作し, 発表時間が交代時間含めて1人90秒で終わること (厳守&強制終了) (3点)
- 起動 + 自己紹介があるので見せる時間は60秒程度!
- 必ず研究室単位の発表用PCで動作確認をし, 発表練習をしておくこと. バックアップPCは用意しておくべし!