

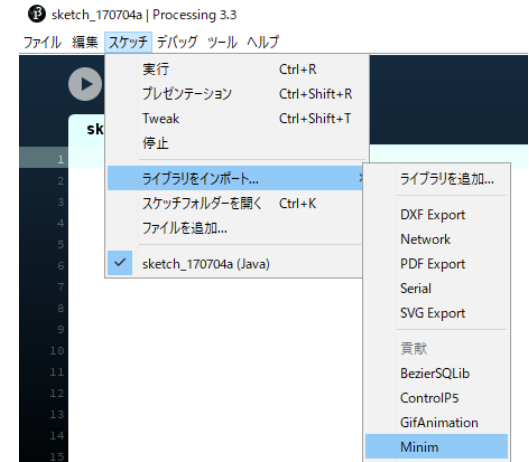
プログラミング演習I (第10回) 課題

• 基本① スケッチ名: basic_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
- プログラムとしていじるのは `getFrequency` 内部だけ。今回は `getFrequency` がかなり長くなってもよい。
- 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の周波数は2倍になる。例えば、1オクターブ上のドは $261.6 * 2$ となり、1オクターブ上のレは $293.7 * 2$ Hz となる。
- 少なくとも3オクターブ分 (少なくとも `toneID` が20まで) 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とする。

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



プログラミング演習I (第10回) 課題

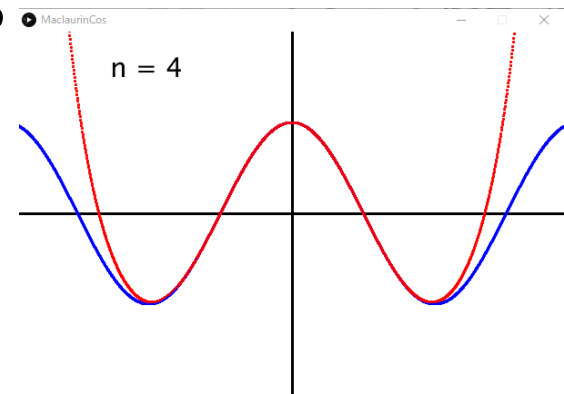
• 基本② スケッチ名: basic_MaclaurinCos

- $\cos x$ は下記の式にマクローリン展開 (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`は、階乗と累乗の組み合わせで表現できる。
- 階乗`calcFactorial`と累乗`calcPower`の関数を完成させることで、 $\cos x$ と近似できることを確認せよ。`calcPower`は`pow`を使わず実現しよう。

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

$$f(x, n) = (-1)^n \frac{x^{2n}}{(2n)!} \text{ とすると}$$

$$\text{Maclaurin}(x, n) = \sum_{i=0}^n f(x, i) \text{ となる}$$



プログラミング演習I (第10回) 課題

• 基本③ スケッチ名: basic_TsuruKame

- 配布したプログラムを利用して鶴亀算を完成させよ
- 「鶴は頭が1つ足が2本」「亀は頭が1つ足が4本」である
- ここで、頭の数と足の数に関数の引数として与えたときに、鶴が何羽いるのかを計算して返す `getNumberOfCrane` 関数を作成せよ
- また、そういった鶴と亀の条件が存在しないときには `-1` を返すようにせよ
- 鶴亀算は繰り返しで網羅するように調べるプログラムを組んで良いよ！
- 標準出力への出力結果がすべて `true` になれば、作成した関数はうまく動作している

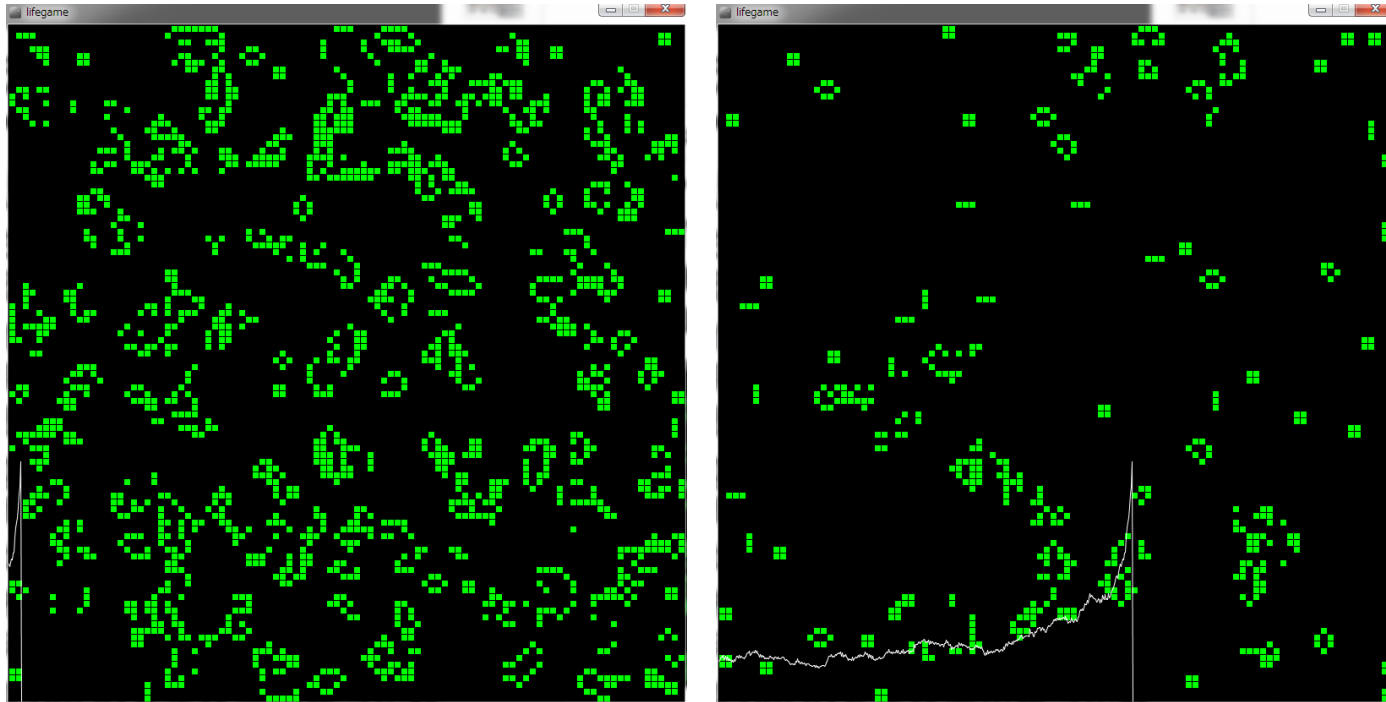
プログラミング演習I (第10回) 課題

• 発展① スケッチ名 : advanced_LifeGame

- 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（上端と下端、左端と右端はつながっているものとせよ）が活着ているか死んでるかを数え、その結果に応じて次のターンで、対象となるセルを活着ている状態にするか、死んでいる状態にするかを決めるものである（詳細は2ページ後を参照）。
- このライフゲームでは、drawLifeGame関数で、セルが活着ている場合（1のとき）は緑色の四角形を、死んでいる場合（0のとき）は黒色の四角形を描画することで表現する。
- また、checkNextDeadOrAlive 関数は、次のターンにそのセルが活着ているかどうかの判定を行うものであり、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。
- このdrawLifeGame関数とcheckNextDeadOrAlive関数を完成させてライフゲームを完成させよ

プログラミング演習I (第10回) 課題

- ライフゲームとはこんなもの
 - ある種の生命のシミュレーション
 - 誕生, 生存, 過疎, 過密で生死を繰り返す



下記動画も面白いので参考までに
ライフゲームの世界 <http://www.nicovideo.jp/mylist/34610498>

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

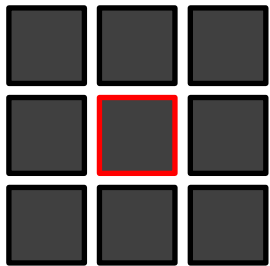
【生存】 生きているセルに隣接する生きたセルが2つか3つならば次世代でも生存

【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

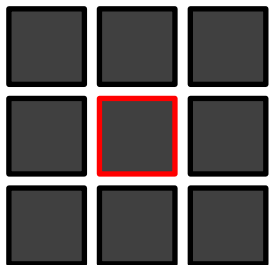
【過密】 生きているセルに隣接する生きたセルが4つ以上ならば過密により死滅

上記以外の場合は現状維持とする

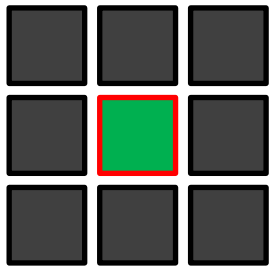
すべて死



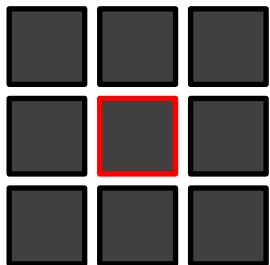
変化なし



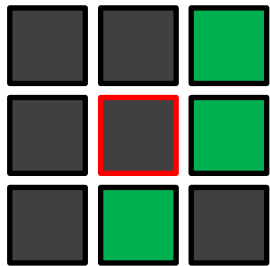
すべて死



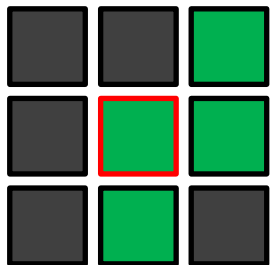
寂しくて死ぬ



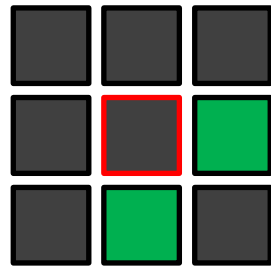
3つの生



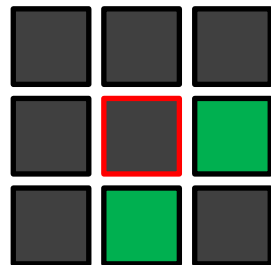
生まれる



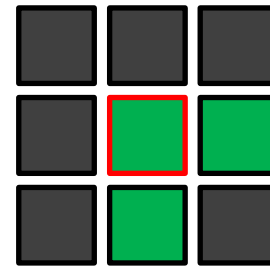
2つの生



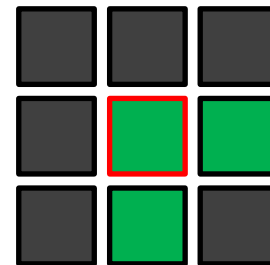
変化なし



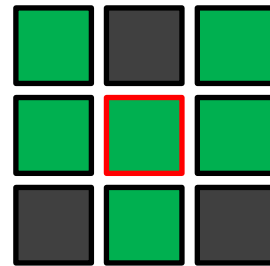
2つか3つの生



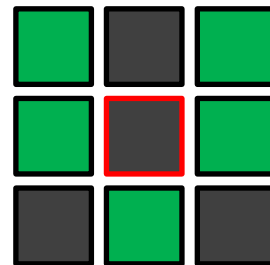
快適で変化なし



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②スケッチ名 : advanced_CalcPI

– フーリエ級数展開を用いると, 下記の式を用いて円周率の近似値を求めることが可能である

$$- \sum_{k=0}^N \frac{(-1)^k}{2k+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

– このNをいくつまで指定するかというのを引数とし, 上記の式に4を掛けた値を返り値として, double型で返す関数 `clacPI` を作成せよ

• 関数の定義 : `double calcPI(int N);`

– また, `calcPI` の引数を10, 100, 1000, 10000, 100000, 1000000, 10000000としていった時の結果 (円周率の近似値) を標準出力せよ

```
N= 10 3.2323157489299774
N= 100 3.151493337005377
N= 1000 3.1
N= 10000 3.1
N= 100000 3.1
N= 1000000 3.1
N=10000000 3.1
```