

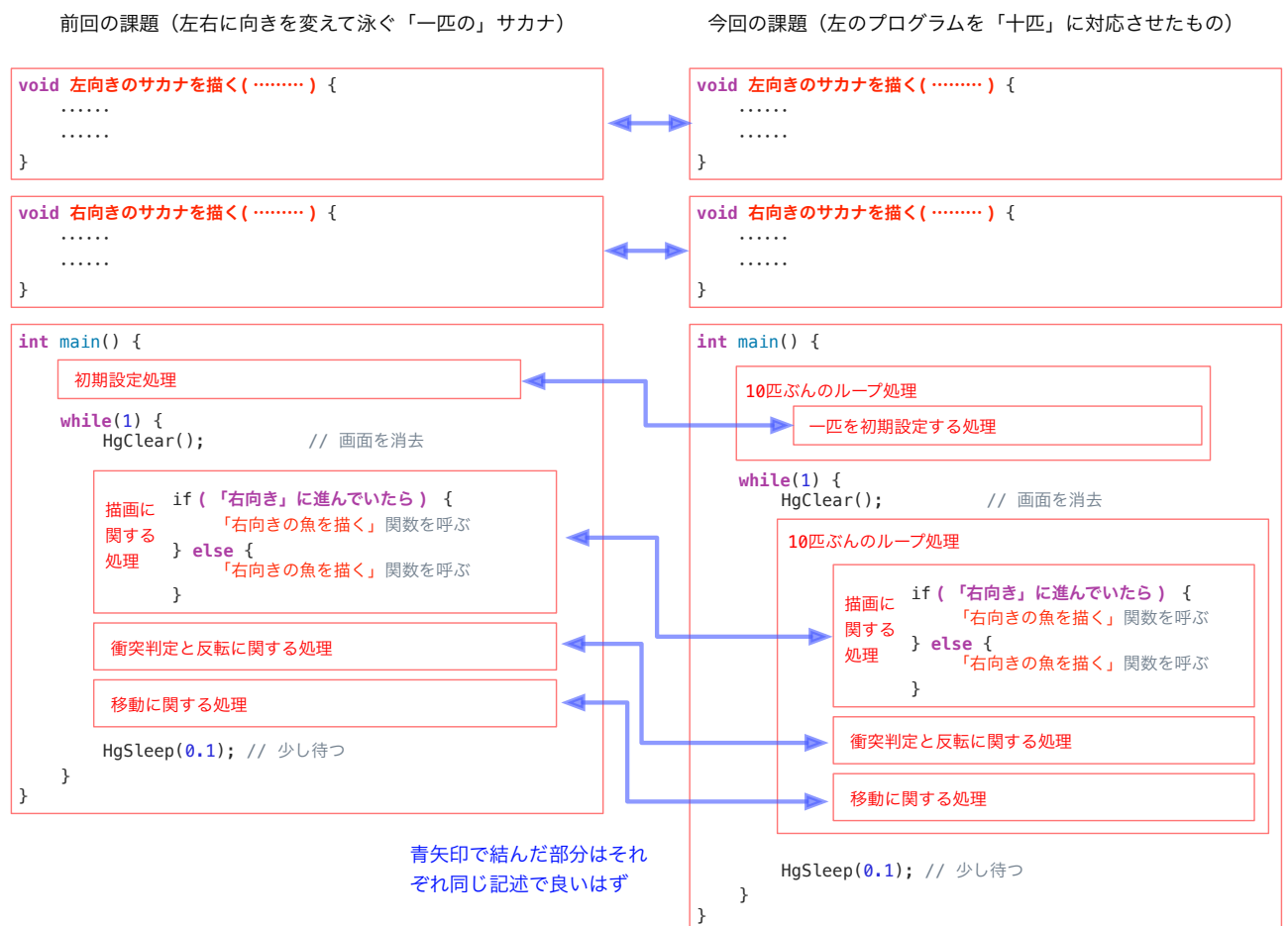
補助資料 #10（前半）

以下に教材の各段階ごとに、教室で説明する時に押さえているポイントなどについてコメントします。この補助資料と、元の教材を並列に見ながら少しずつ作業するようにしてください。

■ 課題 1. 配列を使ってサカナを複数出す

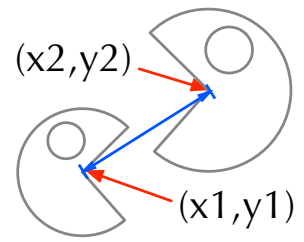
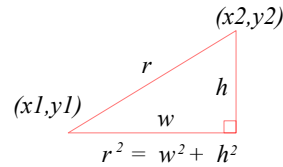
全体の構造としては下図のようになるはずですね。

左が前回の課題（左右に向きを変えて泳ぐ「一匹の」サカナ）です。右がこれを「十匹」にしたものです。プログラムのほとんどのパーツがそのまま使えていることが分かるでしょうか。



■ 課題 2. 群れになって泳ぐサカナ

水平方向と垂直方向の距離が分かっている二点間の直線距離を得るには、ピタゴラスの定理（右図参照、 $r^2 = w^2 + h^2$ ）がそのまま使えます。



つまり、二つのサカナの頭の座標情報、たとえば $(x1, y1)$, $(x2, y2)$ から、上の三角形における w , h の値を求められます。そこから r が得られるはずです。

二匹の衝突判定がそれでできたとして、すべての魚について衝突判定をするためにどうすれば良いか思いつかない分からない人は、例えば 4 匹だったらどれとどれを比べれば良いか考えてみるのが良いです。まず 1-2（1 番目と 2 番目の距離をチェック）、次に 1-3, 1-4, そして 2-1, 2-3, 2-4... といった順にやりますね。

つまり 4 匹中の 2 匹について衝突しているかチェックするすべての組み合わせは、右図に示したように $4 \times 4 = 16$ 通りチェックすることが考えられます。この「全ての組み合わせを試す処理が二重ループで実現できますね。」

1-1	1-2	1-3	1-4
2-1	2-2	2-3	2-4
3-1	3-2	3-3	3-4
4-1	4-2	4-3	4-4

しかし、まず自分自身との衝突は考えなくて良いのだから、1-1, 2-2 といったもののチェックはしなくて良いでしょう。

また、1 と 2 の衝突については、「1-2」の組み合わせで確かめれば良いだけで、その後もう一度「2-1」の組み合わせでチェックする必要はありません。つまり衝突チェックは右図の右上の 6 つぶん、1-2, 1-3, 1-4, 2-3, 2-4, 3-4 だけをやれば良いことになります。

では、これが 10 匹になれば、どのようなパターンで繰り返し処理をすれば（どんな二重ループを構成すれば）、必要な 45 通りのチェックが実施できるか、考えてください。