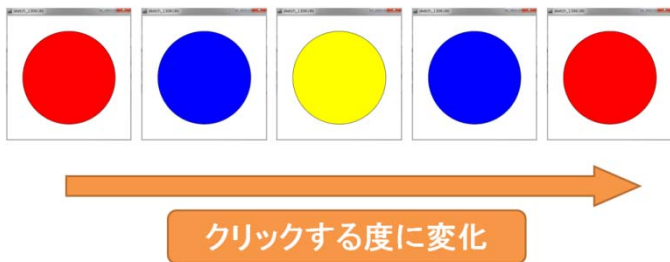


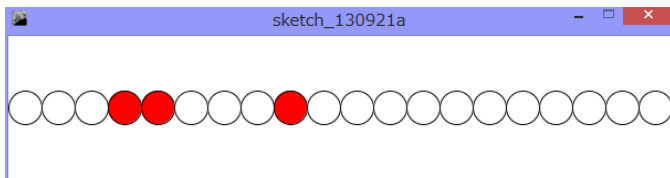
設問 1 (変数と条件分岐)

マウスでウインドウ内をクリックする度に 400x400 のウインドウの中心に設置された円の色 (直径 300) が赤→青→黄→青→赤→青→黄→青→赤…と変化するプログラムを作成せよ。



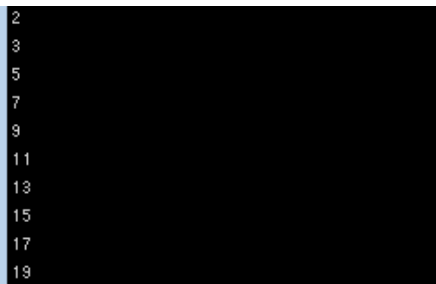
設問 2 (繰返しと配列)

復習 1-4 で扱った 1 次元の電光掲示板について、右から左にスクロールさせ、一番左端まで来ると右端から登場するようにせよ。また、ウインドウ内をクリックするとスクロール方向を左から右に変更し、右端まで来ると左端から登場するようにせよ。



設問 3 (メソッド/関数)

引数として指定した数字が素数かどうかを判定するメソッド (素数判定関数) を作成せよ。なお、返り値は `boolean` とし、1 とその数以外の約数がない場合は `true` を、それ以外の場合は `false` を返すようにせよ。また、その関数 (メソッド) を利用し、2 から 1000 までのすべての整数について素数かどうかを判定し、素数の場合はその値を `println` で標準出力するプログラムを作成せよ。

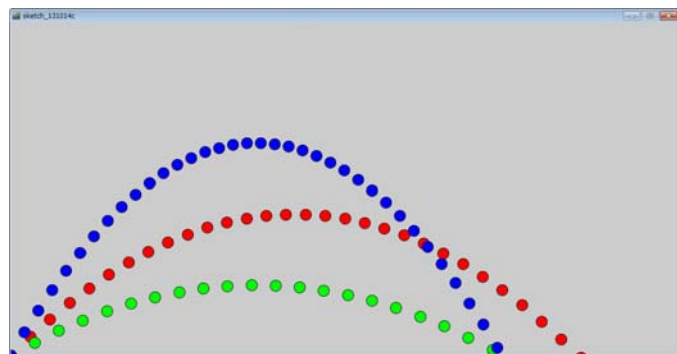


設問 4 (クラス)

1200x600 のウインドウ上に、地上で斜め上方向にボールを投げたときの様子をシミュレーションするプログラムを作成せよ。投射速度 100m/秒 (ただし、1m を 1 ピクセルとする) とし、3 種類の角度 (30 度, 45 度, 60 度) で同時に投げたときの比較結果を示せ。なお、0.5 秒毎の様子を描画せよ。

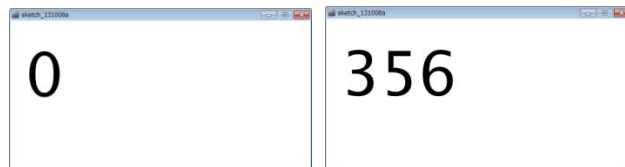
本プログラムの作成においては、ボールを表現する `Ball` クラスを作成し、角度、速度および描画する `x`, `y` 座標をインスタンス変数として持つようにせよ。また、コンストラクタで速度と角度を設定するようにせよ。さらに、引数を経過時間とする `update` というインスタンスメソッドを用意して `x` および `y` 座標を更新し、引数なしの `display` というインスタンスメソッドでボールを描画するようにせよ。

$$x = v_0 \cos \theta t$$
$$y = v_0 \sin \theta t - \frac{1}{2} g t^2$$



設問 5 (メソッド/関数)

マウスをクリックしてから現在までの経過時間をミリ秒単位で毎フレーム表示するプログラムを作成し、再度クリックすると停止して、経過時間を表示するプログラムを作成せよ。なお、本プログラムの作成においては、0 時 0 分 0 秒から現在まで何ミリ秒経過しているかという値を返すメソッド (引数は無く、返り値が経過ミリ秒数となる) を作成し、利用せよ。



設問 6 (繰返しとメソッド)

500x500 のウインドウに、50 ピクセルずつあけて九九の表を作成せよ。また、九九の表の数字が平方数の場合は、赤色の文字で、それ以外の文字は黒色の文字で描くようにして下さい。なお、平方数かどうかを判定するメソッド (引数を判定する数字、戻り値を boolean 型で、引数で指定した数字が平方数の場合は true を、平方数でない場合は false を返すものとせよ) も作成し、利用せよ。

なお、平方数の判定は単純に、入力された数字について 1 から順に 2 乗の値と一致するかどうかで判定せよ。



1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

設問 7

600x400 のウインドウを作成し、その左端と右端では跳ね返り、上端までくると下端から現れ、下端まで来ると上端から現れるような円を描画する `BounceBall` クラスを作成せよ。なお、`BounceBall` はインスタンス変数として `x` 方向、`y` 方向の速度を持ち (各方向について 1~10 の速度をもつものとする)、同じく `x` 座標と `y` 座標をもつようにせよ (各値はランダムに与えよ)。初期位置はコンストラクタ内で設定し、`move` メソッドによって移動し、`display` メソッドによって表示するようにせよ。

また、そのクラスを利用して 10 個の円が動きまわるプログラムを作成せよ。

設問 8 (宿題 2.1 に相当)

200x200 のウインドウに、(100, 30)の位置に直径が 20 ピクセル、(150, 130)の位置に直径が 30 ピクセルの円を描画し、直径が 20 ピクセルの円を描画するようにせよ。

- それぞれの円については、`Ball` クラスとして作成し、インスタンス変数として `x` 座標、`y` 座標、直径、透過度合いを持つようにせよ。
- `Ball` クラスのコンストラクタで `x` 座標、`y` 座標、直径を指定可能とせよ。
- `Ball` クラスには `display` という引数なしのインスタンスメソッドを作成し、各座標と直径、透過度合いに応じて円を描画するようにせよ。
- 2つの `Ball` オブジェクト同士の距離を計算するインスタンスメソッド (引数は `Ball` 変数 1 つで、戻り値は `int` 型の整数とせよ) と、静的メソッド (引数は `Ball` 変数 2 つで、戻り値は `int` 型の整数とせよ) を作成せよ。
- 小さい円の座標は、`draw` の度にマウスカーソルの位置に更新し、マウスを追尾するようにせよ。
- `draw` の度に小さい円と大きい円の距離をそれぞれ計算し、その距離を小さい円、大きい円の透過度にするよう代入せよ。なお、小さい円の透過度を設定する際は静的メソッドで、大きい円の透過度を設定する際はインスタンスメソッドを使用せよ。

設問 9 (宿題 3.1)

時刻情報を整数に変換する `toInt()` メソッドをオーバーライドした `Hourly2.pde` を書け。

設問 10 (宿題 4.1)

`Ball` クラスを拡張して、星形が弾むクラスを作れ。