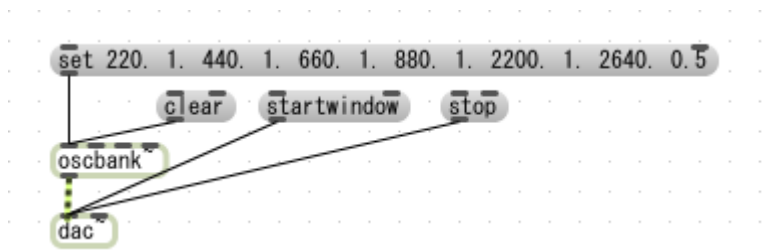


## 4 倍音加算した音で演奏する

### 4.1 オシレータバンクの利用

これまでの章で作成してきた倍音を加算して生成した音は、`oscbank~` オブジェクトを使って生成できる。



`oscbank~` オブジェクトは、`cycle~` (発振器) がたくさんあるようなオブジェクトである。set メッセージによって、中の発振器の周波数とその振幅を設定することができる。set メッセージの後ろの数字を set メッセージのアーギュメント (引数) とよぶ。この例の場合、アーギュメントは複数の数字が構成されている。このような複数の数字からなるデータを、MAX/MSP では、リストとよぶ。`oscbank~` オブジェクトの set メッセージの場合、アーギュメントは周波数とその振幅の対が単位となり、その対を複数指定することができる。この例の場合は、220Hz の振幅が 1, 440Hz の振幅が 1, 660Hz の振幅が 1, ... 2640Hz の振幅が 0.5 ということあらわす。

また、このパッチでは、`dac~` オブジェクトに `start` でなく、`startwindow` というメッセージが接続している。`startwindow` メッセージは、複数のパッチを開いているときでも、そのパッチの `dac~` だけをオンにできる。(start では、開いている全ての `dac~` がオンになってしまう。)

演習 4-1 これまで自分が作成した倍音加算の音を `oscbank~` を利用して作成してみよ。なお、`oscbank~` が生成した信号にも `adsr~` をかけられる。

### 4.2 リストの処理

`oscbank~` を利用してキー入力に対応して音階を変化させることを考える。そのためには、set のリストの周波数に対応する数値の部分を生じたい音階に対応して変化させなければならない。

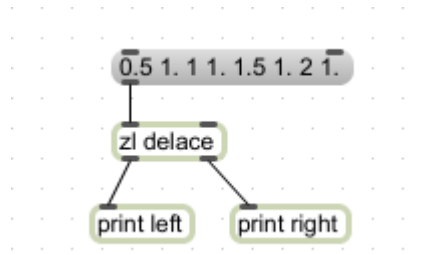
ここでは、例えば、0.5 倍音の振幅が 1, 1 倍音の振幅が 1, 1.5 倍音の振幅が 1 というふうに音色を指定することにする。これを 0.5 1. 1 1. 1.5 1. というリストであらわす。これで、440Hz のラの音を生成するときには、`set 220 1. 440 1. 660 1` というメッセージを生成し、`oscbank~` に入力すればよい。

必要なリスト処理は次のようになる。

1. 周波数に関する数値と振幅に関する数値を分解する。
2. 周波数に関する数値に周波数の値をかける。
3. 周波数, 振幅の対になったリストを再構成する。
4. リストの先頭に set という文字列をつける。

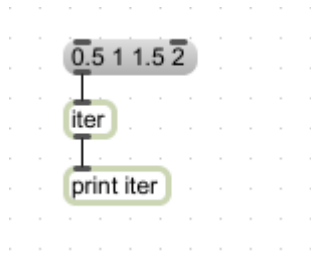
以下、ステップごとに説明する。

いろいろなりリスト処理を行うオブジェクトとして `z1` というオブジェクトが用意されている。

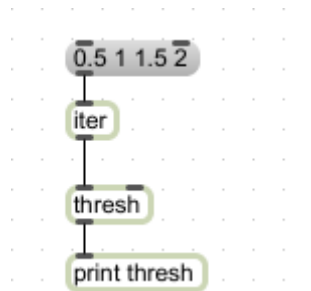


zl の delace モードでは、リストの内容が交互に分解されて、2つのリストが作成される。

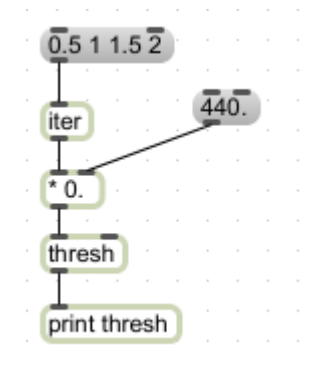
このようにして作成された左のリストは 0.5 1 1.5 2 となる。これを1つずつの要素に分解するオブジェクトが iter である。以下のパッチでは、リストをクリックすると処理が開始される。



バラバラになった要素をリストにまとめるオブジェクトが thresh である。



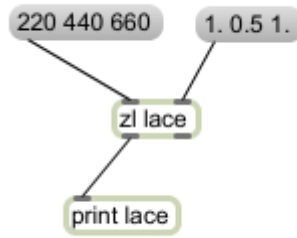
バラバラになった要素に演算をおこなうこともできる。



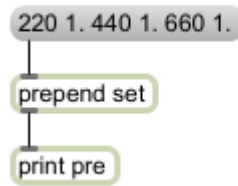
このパッチはまず、440 をクリックしてから、上のリストをクリックしなければならない。\* の引数の 0. という初期値は、この乗算が実数に対するものであることを表す。

演習 4-2 このパッチで \* の引数を削除して結果がどのように変わるか確認せよ。引数がない場合は、演算が整数に対するものであることを表している。

2つのリストの要素を交互に並べて1つのリストにするには zl オブジェクトの lace モードを利用する。



このパッチは、右のリストをクリックし、次に左のリストをクリックすると結果が出力される。  
 リストの先頭に要素を追加するには `prepend` オブジェクトを利用する。



演習 4-3 ここまでに紹介してきた技法を利用して倍音加算した音を使って演奏するパッチを作成せよ。