Helper classes

PmaskBjork - Event patterns for Bjorklund sequences. Note that Bjorlund sequences are also available in the parser. The following two examples are similar - but not identical - because of duration differences:

```
b.p(1, deg"0 1 2 3 4 5 6 7" << PmaskBjork(5,8))
```

```
b.p(1, mask"1*(5,8)" << deg"0 1 2 3 4 5 6 7")
```

Psine, **Psaw** - create time-based patterns, which need durations on the left hand side to produce values.

```
b.p(1, { BBandPass.ar(PinkNoise.ar(\namp.kr(0.4,0.1)!2),
 \bpf.kr(1000, 0.1), 0.1) })
```

```
b.pset(1->1, dur"1*32" << bpf"Psaw.exprange(4,0,40,4000)" <<
 namp"Psine.range(2,0.25,0.5,1)")
```

Pr, **Pr2**, **Per** - shortcuts for random generators (w/optional seed)

```
b.p(1, deg"0 2 4 7" << amp"Per(0.1,0.5)")
```

NOTE: when using SC code inside pattern parsing strings, like the above examples, it's important **not** to use any spaces!

Pattern filters

pattern.**degrade**, **scramble**, **rand**, **perfectShuffle**, **reverse**, **mirror**, **mirror1**, **mirror2**, **rotate**, **pyramid**, **permute** - Event filters that can be applied to patterns, similar to the equivalent Array methods.

```
// Compare:
```

```
b.p(1, deg"[0 0 4 4 5 5 4@2 3 3 2 2 1 1 0@2]@2")
```

```
b.p(1, deg"[0 0 4 4 5 5 4@2 3 3 2 2 1 1 0@2]@2".perfectShuffle)
```

```
b.p(1, deg"[0 0 4 4 5 5 4@2 3 3 2 2 1 1 0@2]@2".pyramid(9))
```

```
b.p(1, deg"[0 0 4 4 5 5 4@2 3 3 2 2 1 1 0@2]@2".scramble(12355)
 .degrade(0.7, 222))
```

Event key abbreviations

Common:

deg: \degree

ins, **inst**: \instrument

leg: \legato

mn, mid: \midinote

oct: \octave

sca, **scl**: \scale

slow, str: \stretch

Not abbreviated:

root, **note**, **freq**, tempo,

dur, lag, strum, **amp**, db,

pan, trig, group, latency

Rare:

ctr: \ctranspose,

det: \detune,

gtr: \gtranspose,

har: \harmonic,

mtr: \mtranspose,

sus: \sustain,

vel: \velocity,

toff: \timingOffset

Baking

Bake a result into a string and save to clipboard for pasting:

Bake((0..7)) → "0 1 2 3 4 5 6 7" is ready to paste

Bake(rrand(0,9!8), "") → e.g. "23632889"

Bake(Pbrown(-7,7, 2, 12)) →
 e.g. "0 1 -1 1 0 2 3 1 -1 -2 -4 -3"

Bake.cs(deg"<0 1> 2*(3,8) <3,5> 7") →

```
 Pbind('degree', Ppatlace([ Ppatlace([ 0, 1 ], inf), 2, 2, 2,
 Ptuple([ 3, 5 ]), 7 ], 2), 'dur', Pseq([ 0.25, 0.09375, 0.09375,
 0.0625, 0.25, 0.25 ], 2))
```

Binaural spatialization (experimental)

b.**spatialInit**, **spatialize**, **spatialFree**, **despatialize**

b.**spatial.set**, **spatial.map**, **spatial.playBuf**, **spatial.playPat**, **spatial.oncePat**, **spatial.stopPat**, spatial.free

Bacalao v0.5.27 cheatsheet (2/2)
https://github.com/totalgee/bacalao