

```
-- Сочиняем электронную музыку на Haskell

* cabal install csound-catalog

* install Csound -> http://csound.github.io/download.html

* добавьте Csound в PATH, если само не добавилось

-- Добро пожаловать

ghci

-- # Hello, World!

import Csound.Base

dac (osc 220)
    -- Нажмите Ctrl+C для завершения программы!

-- # Настройка громкости

dac $ mul 0.45 $ osc 220

-- # Знак доллара

dac $ mul 0.45 $ osc 220

-- то же что и

dac (mul 0.45 (osc 220))

-- # Ау! Скрипачи, духовики

dac $ mul 0.5 $ testDrone 220

-- # Мыслим нотами

dac $ mul 0.5 $ testDrone $ cspch 7.02

-- # Практикуемся в Ре-мажоре

dac $ mean $ fmap (testDrone2 . cspch) [7.02, 7.09, 8.02, 8.06]

-- # Поиграем с метрономом?

dac $ ticks 4 135

-- Есть разные варианты тембра: `ticks2`, `ticks3`, `ticks4`

-- # Для повёрнутых на ритме!

dac $ nticks [3,2,2] 135
```

```

dac $ nticks [2, 3, 4, 5] 170

-- # Соединим метроном с гармонией

let drone = mean $ fmap (testDrone2 . cpspch) [7.02, 7.09, 8.02, 8.06]

let rhythm = nticks [3, 3, 2] 120

dac $ sum [drone, return $ fromMono rhythm]

-- # Дело в типах

:t drone

:t rhythm

-- # Подстроим уровни громкости

dac $ sum [mul 0.6 drone, return $ fromMono $ mul 1.3 rhythm]

-- # Как насчёт готовых синтов

import Csound.Patch

vdac $ mul 0.45 $ atMidi dreamPad

vdac $ mul 0.7 $ atMidi toneWheelOrgan

vdac $ mul 0.45 $ atMidi vibraphone1

-- # Попробуем разные звуки

vdac $ mul 0.45 $ atMidi vibraphone1


cathedralOrgan    dreamPad        noiz            whaleSongPad
vibraphone2       xylophone        simpleMarimba   bassClarinet
razorLead         fmDroneMedium    hammondOrgan    overtonePad
choirA            scrapeDahina     pwEnsemble      hulusi
epiano1           chalandiPlates   magicBanyan     nightPad
(vibhu 65)        (avatara 60)     (bhumis 50)     (rishi 70)

-- # Заменяем метроном на барабаны!

import Csound.Catalog.Drum.Tr808

```

```
-- bd   - base drum      sn - snare drum      chh - closed high hat
--      - бочка          - рабочий          - закрытый хэт
```

```
dac bd
```

```
dac sn
```

```
dac chh
```

```
-- # Создаём паттерны
```

```
import Csound.Sam
```

```
-- # pat -- евклидовы биты
```

```
-- pat :: [D] -> Sam -> Sam
```

```
-- pat beats shot = loop
```

```
dac $ pat [3, 3, 2] bd
```

```
-- # del - задержка на несколько долей
```

```
-- del :: D -> Sam -> Sam
```

```
-- del duration sam = delayedSam
```

```
dac $ sum [pat [3, 3, 2] bd, del 2 $ pat [4] sn ]
```

```
-- # str - изменим темп
```

```
-- str :: D -> Sam -> Sam
```

```
dac $ str 0.5 $ sum [ pat [3, 3, 2] bd, del 2 $ pat [4] sn ]
```

```
-- # pat' - добавим акценты
```

```
-- pat' :: [D] -> [D] -> Sam -> Sam
```

```
dac $ str 0.5 $ pat' [1, 0.5, 0.2, 0.1] [1] chh
```

```
dac $ str 0.5 $ sum [ pat [3, 3, 2] bd, del 2 $ pat [4] sn, pat' [1, 0.5, 0.2, 0.1] [1] chh ]
```

```
-- # Звуки невтопад
```

```
let toms = mul 0.25 $ sum [del 3 $ pat [5, 11, 7, 4] mtom, pat [4, 7, 1, 9] htom, del 7 $ pat [3, 7, 6] ltom]
```

```
let drums = str 0.5 $ sum [pat [3, 3, 2] bd, del 2 $ pat [4] sn, pat' [1, 0.5, 0.2, 0.1] [1] chh, toms, del 16 $ pat [15, 2, 3] rim]
```

```
dac drums
```

```
-- # Попробуем разные звуки
```

```
bd      - base drums      htom, mtom, ltom - high middle low toms
        бочка              томы
```

sn	- snare рабочий	cl	- claves палочки
chh	- closed high-hat закрытый хэт	rim	- rim-shot удар по ободу
ohh	- open high-hat открытый хэт	mar	- maracas маракасы
сум	- cymbal	hcon, mcon, lcon	- high, middle, low conga

ещё один набор барабашек в модуле Csound.Catalog.Drum.Minipops

-- # Давайте делать паузы...

```
-- lim :: D -> Sam -> Sam      - назначить длину сэмпла (limit)
--
-- mel :: [Sam] -> Sam          - играем один сэмпл за другим (melody)
--
-- rest :: D -> Sam            - пауза (число долей)
--
-- loop :: Sam -> Sam          - бесконечный повтор
```

```
let hhats = loop $ mel [lim 8 $ pat' [1, 0.5, 0.25, 0.1] [1] chh, rest 8]
dac $ hhats
```

-- # Добавим гармонию к барабанам

```
:t drone
:t drums
```

```
-- toSam :: ToSam a => a -> Sam      -- бесконечный сигнал
-- limSam :: ToSam a => D -> a -> Sam  -- конечный
```

```
let player = toSam $ atMidi vibraphone1
let performance = sum [mul 0.74 $ toSam drone, mul 1.2 drums, mul 0.5 player]
vdac performance
```

-- # Сделаем запись!

```
-- dumpWav :: String -> (Sig, Sig) -> SE (Sig, Sig)
```

```
vdac $ at (dumpWav "song2.wav") performance
```

-- # Послушаем

```
dac $ loopWav 1 "song2.wav"
```

```
dac $ loopWav (-1) "song2.wav"
```

```
dac $ loopWav (constSeq [1, 1, -1, 2, 1] 1) "song2.wav"
```

```

-- # Пример: Vibhu Vibes

-- # Глитч: Пульсирующий шум
dac $ mul (sqrSeq [1, 0.5, 0.25] 8) $ pink

-- # Глитч: Поиграемся со скоростью воспроизведения

let file = "loop.wav"
dac $ loopWav1 1 file

dac $ loopWav1 (-1) file

dac $ loopWav1 0.5 file

dac $ loopWav1 (-0.25) file

-- # Глитч: сэмпл барабанов с разными скоростями

let d1 = loopWav1 (-(constSeq [1, 2, 4, 2] 0.5)) file
let d2 = mul (constSeq [1, 0] 0.5) $ loopWav1 (-0.25) file
let noisyDrum = sum [d1, d2]

-- # Глитч: Добавим пульсацию и ревер
let glitchy = mixAt 0.2 smallRoom2 $ mul (sqrSeq [1, 0.5, 0.25] 8) noisyDrum
dac glitchy

-- # Дрон
vdac $ mul 0.5 $ atMidi nightPad

vdac $ mul 0.5 $ atMidi $ deepPad nightPad

-- # Смешаем пады
vdac $ mul 0.3 $ sum [atMidi dreamPad, atMidi $ deepPad fmDroneMedium]
vdac $ mul 0.3 $ sum [atMidi pwPad, atMidi $ deepPad whaleSongPad]

-- # Добавим пульсацию
let pulsar = sawSeq [1, 0.5, 0.25, 0.8, 0.4, 0.1, 0.8, 0.5] 8
vdac $ mul pulsar $ atMidi nightPad

-- # Итоговый вариант дрона
let p1 = atMidi whaleSongPad

```

```
let p2 = atMidi $ deepPad overtonePad

let p3 = mul pulsar $ atMidi nightPad

let pads = mul 0.3 $ sum [p1, p2, mul 2 p3]

vdac pads

-- # Барабаны и дрон

vdac $ sum [pads, return glitchy]

-- # Спасибо за внимание!
```