

MultimodSynth Reference

Table of Contents

MultimodSynth Reference	22
MultimodSynth - пространство имен	23
AbsClipper - класс	23
AbsClipper.Process - метод	23
AbsClipper.SetAmount - метод	24
AdsrEnvelope - класс	24
AdsrEnvelope - конструктор	25
AdsrEnvelope.State - свойство	25
AdsrEnvelope.CalcCoef - метод	26
AdsrEnvelope.OnSoundStop - метод	26
AdsrEnvelope.Process - метод	26
AdsrEnvelope.SetAttackRate - метод	27
AdsrEnvelope.SetAttackTargetRatio - метод	27
AdsrEnvelope.SetDecayRate - метод	27
AdsrEnvelope.SetDecayReleaseTargetRatio - метод	28
AdsrEnvelope.SetReleaseRate - метод	28
AdsrEnvelope.SetSustainLevel - метод	28
AdsrEnvelope.TriggerAttack - метод	29
AdsrEnvelope.TriggerRelease - метод	29
SoundStop - событие	29
EnvelopeState - перечисление	29
attackBase - поле	30
attackCoef - поле	30
attackRate - поле	30
attackTargetRatio - поле	31
currValue - поле	31
decayBase - поле	31
decayCoef - поле	31
decayRate - поле	32
decayReleaseTargetRatio - поле	32
releaseBase - поле	32
releaseCoef - поле	32

releaseRate - поле.....	33
state - поле.....	33
sustainLevel - поле.....	33
AudioComponent - класс	33
AudioComponent.SampleRate - свойство.....	34
AudioComponent.OnSampleRateChanged - метод	34
sampleRate - поле.....	35
AudioComponentWithParameters - класс.....	35
AudioComponentWithParameters - конструктор.....	36
AudioComponentWithParameters.Plugin - свойство.....	36
AudioComponentWithParameters.InitializeParameters - метод.....	36
AudioComponentWithParameters.InitializeParameters (ParameterFactory) - метод	37
parameterCategory - поле	37
parameterPrefix - поле	37
AudioProcessor - класс	37
AudioProcessor - конструктор.....	38
AudioProcessor.ProcessingMutex - свойство.....	38
AudioProcessor.Routing - свойство	39
AudioProcessor.SampleRate - свойство	39
AudioProcessor.Process - метод.....	39
plugin - поле.....	39
BitCrusher - класс.....	40
BitCrusher.Process - метод.....	40
BitCrusher.SetAmount - метод.....	41
steps - поле	41
Converters - класс	41
Converters.AsymmetryToString - метод.....	42
Converters.CentsToString - метод.....	42
Converters.DelayLfoRateToString - метод.....	43
Converters.DelayModeToString - метод	43
Converters.DelayTimeToString - метод.....	43
Converters.DistortionAmpToString - метод	44
Converters.DistortionLowpassCutoffToString - метод	44
Converters.DistortionModeToString - метод	44

Converters.EnvelopeCurveToString - метод.....	45
Converters.EnvelopeTimeToString - метод.....	45
Converters.FilterCutoffMultiplierToString - метод.....	45
Converters.FilterTypeToString - метод.....	46
Converters.InvertToString - метод	46
Converters.ModulationTypeToString - метод	47
Converters.OversamplingOrderToString - метод	47
Converters.PercentsToString - метод.....	47
Converters.SemitonesToString - метод.....	48
Converters.StereoAmountToString - метод	48
Converters.ToAsymmetry - метод.....	48
Converters.ToCents - метод	49
Converters.ToDelayLfoRate - метод.....	49
Converters.ToDelayMode - метод.....	49
Converters.ToDelayTime - метод	50
Converters.ToDistortionAmp - метод.....	50
Converters.ToDistortionLowpassCutoff - метод.....	51
Converters.ToDistortionMode - метод.....	51
Converters.ToEnvelopeTime - метод.....	51
Converters.ToFilterCutoffMultiplier - метод.....	52
Converters.ToFilterType - метод	52
Converters.ToModulationType - метод.....	52
Converters.ToOversamplingOrder - метод.....	53
Converters.ToSemitones - метод.....	53
Converters.ToStereoAmount - метод.....	53
Converters.ToWaveTable - метод	54
Converters.WaveTableToString - метод	54
CubicClipper - класс	54
CubicClipper.Process - метод.....	55
CubicClipper.SetAmount - метод.....	55
DCBlocker - класс	56
DCBlocker - конструктор	56
DCBlocker.OnSampleRateChanged - метод	57
DCBlocker.Process - метод	57

DCBlocker.SetResponseTimeCoefficient - метод	58
DefaultExp - поле	58
normalizationCoeff - поле.....	58
r - поле	58
responseTimeCoefficient - поле	59
xm1 - поле.....	59
ym1 - поле.....	59
BaseSampleRate - поле.....	59
DelayLine - класс	60
DelayLine.CalculateOutput - метод	60
DelayLine.OnSampleRateChanged - метод	61
DelayLine.Process - метод	61
DelayLine.Reset - метод.....	61
DelayLine.SetDelay - метод	62
DelayLine.SetFeedback - метод.....	62
buffer - поле.....	62
bufferLength - поле.....	62
delay - поле	63
feedback - поле	63
readOffset - поле	63
readPoint - поле.....	63
writePoint - поле.....	64
MaxTime - поле.....	64
DelayManager - класс	64
DelayManager - конструктор	65
DelayManager.FeedbackManager - свойство	65
DelayManager.InvertManager - свойство	66
DelayManager.LfoDepthManager - свойство.....	66
DelayManager.LfoRateManager - свойство	66
DelayManager.MixManager - свойство.....	66
DelayManager.ModeManager - свойство	67
DelayManager.StereoAmountManager - свойство.....	67
DelayManager.TimeManager - свойство.....	67
DelayManager.ChangeDelay - метод.....	67

DelayManager.InitializeParameters - метод	68
DelayManager.OnSampleRateChanged - метод.....	68
DelayManager.Process - метод	68
DelayManager.SetFeedback - метод.....	69
DelayManager.SetInvert - метод.....	69
DelayManager.SetLfoDepth - метод.....	70
DelayManager.SetLfoRate - метод.....	70
DelayManager.SetMixTarget - метод.....	70
DelayManager.SetMode - метод.....	71
DelayManager.SetStereoAmountTarget - метод	71
DelayManager.SetTimeTarget - метод	72
DelayManager.UpdateMix - метод.....	72
DelayManager.UpdateStereoAmount - метод.....	72
DelayManager.UpdateTime - метод.....	73
StereoMode - перечисление.....	73
currentDelay - поле.....	73
delay - поле	74
dryCoeff - поле.....	74
feedback - поле	74
lfo - поле	74
lfoDepth - поле.....	74
maxDelay - поле.....	75
mixFilter - поле.....	75
mode - поле	75
pingPongDelay - поле.....	75
stereoAmount - поле	76
stereoAmountFilter - поле	76
stereoOffsetDelay - поле	76
timeFilter - поле	76
variousTimeDelay - поле.....	77
wetCoeff - поле	77
wetSign - поле	77
MaxLfoDepth - поле.....	77
MaxTime - поле.....	78

DistortionManager - класс	78
DistortionManager - конструктор	79
DistortionManager.AmountManager - свойство	79
DistortionManager.AmpManager - свойство	79
DistortionManager.AsymmetryManager - свойство	80
DistortionManager.LowPassCutoffManager - свойство	80
DistortionManager.MixManager - свойство	80
DistortionManager.ModeManager - свойство	80
DistortionManager.ChangeDistortion - метод	81
DistortionManager.InitializeParameters - метод	81
DistortionManager.OnSampleRateChanged - метод	81
DistortionManager.Process - метод	82
DistortionManager.SetAmount - метод	82
DistortionManager.SetAmpTarget - метод	82
DistortionManager.SetAsymmetryTarget - метод	83
DistortionManager.SetLowPassCutoff - метод	83
DistortionManager.SetMode - метод	84
DistortionManager.UpdateAmp - метод	84
DistortionManager.UpdateAsymmetry - метод	84
DistortionManager.UpdateMix - метод	85
DistortionMode - перечисление	85
absClipper - поле	85
amount - поле	86
amp - поле	86
ampFilter - поле	86
asymmetryFilter - поле	86
bitCrusher - поле	87
cubicClipper - поле	87
currentDistortion - поле	87
dcBlocker - поле	87
dcOffset - поле	88
dryCoeff - поле	88
lowPass - поле	88
mixFilter - поле	88

mode - поле	89
sampleRateReducer - поле	89
softClipper - поле	89
wetCoeff - поле	89
Downsampler - класс	89
Downsampler.Order - свойство	90
Downsampler.Process - метод	90
X2OversamplingCoefficients - поле	91
X4OversamplingCoefficients - поле	91
X8OversamplingCoefficients - поле	91
buffer - поле	91
coefficients - поле	92
currBufferIndex - поле	92
order - поле	92
Envelope - класс	92
Envelope - конструктор	93
Envelope.IsActive - свойство	93
Envelope.OnSampleRateChanged - метод	94
Envelope.Process - метод	94
Envelope.SetAmplitude - метод	94
Envelope.SetAttackCurve - метод	95
Envelope.SetAttackTime - метод	95
Envelope.SetDecayReleaseCurve - метод	95
Envelope.SetDecayTime - метод	96
Envelope.SetReleaseTime - метод	96
Envelope.SetSustainLevel - метод	96
Envelope.TriggerAttack - метод	97
Envelope.TriggerRelease - метод	97
amplitude - поле	97
attackCurve - поле	97
attackTargetRatio - поле	98
attackTime - поле	98
decayReleaseCurve - поле	98
decayReleaseTargetRatio - поле	98

decayTime - поле	99
envelope - поле.....	99
releaseTime - поле.....	99
sustainLevel - поле.....	99
EnvelopesManager - класс.....	99
EnvelopesManager - конструктор	100
EnvelopesManager.AttackCurveManager - свойство	101
EnvelopesManager.AttackTimeManager - свойство	101
EnvelopesManager.DecayReleaseCurveManager - свойство	101
EnvelopesManager.DecayTimeManager - свойство	101
EnvelopesManager.EnvelopeAmplitudeManager - свойство	102
EnvelopesManager.ReleaseTimeManager - свойство.....	102
EnvelopesManager.SustainLevelManager - свойство	102
EnvelopesManager.CreateNewEnvelope - метод	102
EnvelopesManager.InitializeParameters - метод.....	103
EnvelopesManager.Process - метод.....	103
EnvelopesManager.SetAttackCurve - метод	103
EnvelopesManager.SetAttackTime - метод.....	104
EnvelopesManager.SetDecayReleaseCurve - метод	104
EnvelopesManager.SetDecayTime - метод	104
EnvelopesManager.SetReleaseTime - метод.....	105
EnvelopesManager.SetSustainLevel - метод.....	105
EnvelopesManager.UpdateEnvelopeAmplitude - метод	105
attackCurve - поле	106
attackTime - поле.....	106
decayReleaseCurve - поле	106
decayTime - поле	106
envelopeAmplitude - поле	107
envelopeAmplitudeFilter - поле	107
envelopes - поле	107
releaseTime - поле	107
sustainLevel - поле	108
Filter - класс	108
Filter - конструктор.....	109

Filter.CalculateCutoff - метод.....	109
Filter.OnSampleRateChanged - метод.....	109
Filter.Process - метод.....	109
Filter.Reset - метод.....	110
Filter.SetCurve - метод	110
Filter.SetCutoffMultiplier - метод.....	110
Filter.SetFilterType - метод.....	111
Filter.SetNoteFrequency - метод.....	111
Filter.SetTrackingCoeff - метод.....	111
FilterModulationLookup - поле	112
cutoff - поле	112
cutoffMultiplier - поле	112
filter - поле	113
noteFrequency - поле	113
trackingCoeff - поле	113
BaseCutoff - поле	113
FiltersManager - класс	113
FiltersManager - конструктор.....	114
FiltersManager.CurveManager - свойство	115
FiltersManager.CutoffManager - свойство.....	115
FiltersManager.FilterTypeManager - свойство.....	115
FiltersManager.TrackingCoeffManager - свойство	115
FiltersManager.CreateNewFilter - метод	116
FiltersManager.InitializeParameters - метод.....	116
FiltersManager.Process - метод	116
FiltersManager.SetCurve - метод	116
FiltersManager.SetCutoffMultiplierTarget - метод	117
FiltersManager.SetFilterType - метод	117
FiltersManager.SetTrackingCoeff - метод	118
FiltersManager.UpdateCutoffMultiplier - метод.....	118
curve - поле.....	118
cutoffMultiplier - поле	118
cutoffMultiplierFilter - поле	119
filterType - поле	119

filters - поле	119
trackingCoeff - поле	119
MidiNoteEventArgs - класс	120
MidiNoteEventArgs - конструктор	120
MidiNoteEventArgs.Note - свойство	121
MidiNoteEventArgs.PressedNotesCount - свойство	121
MidiProcessor - класс.....	121
MidiProcessor - конструктор	122
MidiProcessor.ChannelCount - свойство.....	122
MidiProcessor.OnNoteOff - метод	122
MidiProcessor.OnNoteOn - метод.....	123
MidiProcessor.PressNoteFromUI - метод.....	123
MidiProcessor.Process - метод.....	124
MidiProcessor.ReleaseNoteFromUI - метод.....	124
NoteOff - событие.....	124
NoteOn - событие	125
plugin - поле.....	125
pressedNotesCount - поле.....	125
Oscillator - класс	125
Oscillator.OnSampleRateChanged - метод.....	126
Oscillator.Process - метод.....	126
Oscillator.Reset - метод.....	127
Oscillator.SetNoteFrequency - метод.....	127
Oscillator.SetPitchMultiplier - метод.....	127
Oscillator.SetWaveTable - метод.....	128
Oscillator.UpdateCoefficients - метод.....	128
frequency - поле	128
noteFrequency - поле	129
phaseIncrement - поле	129
phasor - поле.....	129
pitchMultiplier - поле.....	129
waveTable - поле	129
OscillatorsManager - класс	130
OscillatorsManager - конструктор.....	130

OscillatorsManager.PitchFineManager - свойство.....	131
OscillatorsManager.PitchSemiManager - свойство.....	131
OscillatorsManager.WaveTableManager - свойство.....	131
OscillatorsManager.CreateNewOscillator - метод	132
OscillatorsManager.InitializeParameters - метод.....	132
OscillatorsManager.Process - метод.....	132
OscillatorsManager.SetPitchFine - метод.....	132
OscillatorsManager.SetPitchSemi - метод.....	133
OscillatorsManager.SetWaveTable - метод.....	133
OscillatorsManager.UpdatePitchMultiplier - метод.....	134
oscillators - поле	134
pitchFine - поле.....	134
pitchMultiplier - поле.....	134
pitchMultiplierFilter - поле	135
pitchSemi - поле.....	135
waveTable - поле	135
ParameterFactory - класс.....	135
ParameterFactory - конструктор.....	136
ParameterFactory.CreateParameterManager - метод.....	136
namePrefix - поле	137
parameterCategory - поле	137
plugin - поле.....	137
programs - поле	138
ParameterFilter - класс	138
ParameterFilter - конструктор.....	139
ParameterFilter.OnSampleRateChanged - метод.....	139
ParameterFilter.Process - метод	139
ParameterFilter.SetResponseTimeCoefficient - метод.....	140
ParameterFilter.SetTarget - метод	140
a - поле	140
b - поле.....	141
isActive - поле	141
responseTimeCoefficient - поле	141
target - поле	141

value - поле	141
valueChangedAction - поле	142
BaseSampleRate - поле	142
DefaultExp - поле	142
PingPongDelay - класс	142
PingPongDelay - конструктор	143
PingPongDelay.OnSampleRateChanged - метод	143
PingPongDelay.Process - метод	144
PingPongDelay.Reset - метод	144
PingPongDelay.SetDelay - метод	145
PingPongDelay.SetFeedback - метод	145
PingPongDelay.SetStereo - метод	145
delayL - поле	146
delayR - поле	146
feedback - поле	146
leftCoeff - поле	146
rightCoeff - поле	147
Plugin - класс	147
Plugin - конструктор	147
Plugin.AudioProcessor - свойство	148
Plugin.MidiProcessor - свойство	148
Plugin.Programs - свойство	148
Plugin.CreateAudioProcessor - метод	148
Plugin.CreateEditor - метод	149
Plugin.CreateMidiProcessor - метод	149
Plugin.CreatePrograms - метод	149
PluginCommandStub - класс	150
PluginCommandStub.CreateInstance - метод	150
PluginEditor - класс	150
PluginEditor - конструктор	151
PluginEditor.Bounds - свойство	151
PluginEditor.KnobMode - свойство	152
PluginEditor.Close - метод	152
PluginEditor.GetElementPixelSize - метод	152

PluginEditor.KeyDown - метод	152
PluginEditor.KeyUp - метод	153
PluginEditor.Open - метод	153
PluginEditor.ProcessIdle - метод	154
bounds - поле	154
hwndSource - поле	154
instance - поле	155
plugin - поле	155
PluginPrograms - класс	155
PluginPrograms - конструктор	156
PluginPrograms.ParameterCategories - свойство	156
PluginPrograms.ParameterInfos - свойство	156
PluginPrograms.CreateProgramCollection - метод	156
plugin - поле	157
Routing - класс	157
Routing - конструктор	158
Routing.DelayManager - свойство	158
Routing.DistortionManager - свойство	158
Routing.Downsampler - свойство	158
Routing.MasterVolumeManager - свойство	159
Routing.OversamplingOrderManager - свойство	159
Routing.VoicesManager - свойство	159
Routing.InitializeParameters - метод	159
Routing.MidiProcessor_NoteOff - метод	160
Routing.MidiProcessor_NoteOn - метод	160
Routing.OnSampleRateChanged - метод	161
Routing.Process - метод	161
Routing.SetOversamplingOrder - метод	161
Routing.UpdateMasterVolume - метод	162
Routing.UpdateSampleRates - метод	162
masterVolume - поле	162
masterVolumeFilter - поле	163
samplesForOversampling - поле	163
SampleRateReductor - класс	163

SampleRateReducer.OnSampleRateChanged - метод.....	164
SampleRateReducer.Process - метод.....	164
SampleRateReducer.SetAmount - метод.....	165
SampleRateReducer.UpdateCoefficients - метод.....	165
holdTime - поле	165
phaseIncrement - поле	165
phasor - поле.....	166
sample - поле	166
SineLFO - класс.....	166
SineLFO - конструктор	167
SineLFO.OnSampleRateChanged - метод.....	167
SineLFO.Process - метод.....	167
SineLFO.SetFrequency - метод	168
SineLFO.UpdateCoefficients - метод	168
coeff - поле	168
cos - поле	168
frequency - поле	169
sin - поле	169
SoftClipper - класс.....	169
SoftClipper.Process - метод	170
SoftClipper.SetAmount - метод.....	170
denominator - поле	171
normalizationCoeff - поле.....	171
treshold - поле	171
StereoOffsetDelay - класс	171
StereoOffsetDelay - конструктор	172
StereoOffsetDelay.OnSampleRateChanged - метод.....	172
StereoOffsetDelay.Process - метод	173
StereoOffsetDelay.Reset - метод.....	173
StereoOffsetDelay.SetDelay - метод	173
StereoOffsetDelay.SetFeedback - метод.....	174
StereoOffsetDelay.SetStereo - метод.....	174
delayL - поле	175
delayR - поле	175

offsetBufferL - поле.....	175
offsetBufferR - поле	175
SvfFilter - класс.....	176
SvfFilter - конструктор.....	176
SvfFilter.OnSampleRateChanged - метод.....	177
SvfFilter.Process - метод.....	177
SvfFilter.Reset - метод.....	178
SvfFilter.SetCutoff - метод.....	178
SvfFilter.SetGain - метод.....	178
SvfFilter.SetQ - метод	179
SvfFilter.SetType - метод	179
SvfFilter.UpdateCoefficients - метод.....	179
FilterType - перечисление.....	180
a1 - поле	180
a2 - поле	180
a3 - поле	181
cutoff - поле	181
gain - поле	181
ic1eq - поле	181
ic2eq - поле	182
m0 - поле	182
m1 - поле	182
m2 - поле	182
q - поле.....	183
type - поле	183
Utilities - класс	183
Utilities - конструктор.....	184
Utilities.GetOrGenerateWaveTables - метод.....	184
Utilities.HalfSquareGenerator - метод	184
Utilities.MidiNoteToFrequency - метод.....	185
Utilities.QuarterSquareGenerator - метод.....	185
Utilities.ReadParameters - метод	185
Utilities.SawGenerator - метод.....	186
Utilities.SineGenerator - метод	186

Utilities.SquareGenerator - метод.....	187
Utilities.TriangleGenerator - метод.....	187
Utilities.WriteParameters - метод.....	188
WaveTables - поле	188
DefaultSampleRate - поле	188
VariousTimeDelay - класс	189
VariousTimeDelay - конструктор.....	189
VariousTimeDelay.OnSampleRateChanged - метод.....	190
VariousTimeDelay.Process - метод.....	190
VariousTimeDelay.Reset - метод.....	190
VariousTimeDelay.SetDelay - метод.....	191
VariousTimeDelay.SetFeedback - метод	191
VariousTimeDelay.SetStereo - метод	192
delay - поле	192
delayL - поле	192
delayR - поле.....	192
leftDelayCoeff - поле.....	193
rightDelayCoeff - поле	193
Voice - класс.....	193
Voice - конструктор	194
Voice.IsActive - свойство	195
Voice.Modulation - свойство.....	195
Voice.Note - свойство.....	195
Voice.OnSampleRateChanged - метод	195
Voice.PlayNote - метод.....	196
Voice.Process - метод.....	196
Voice.TriggerRelease - метод	196
ModulationType - перечисление	197
envA - поле.....	197
envB - поле.....	197
envFilter - поле	197
filter - поле	198
fmAmountMultiplier - поле	198
noteVelocity - поле	198

oscA - поле	198
oscB - поле	199
VoicesManager - класс.....	199
VoicesManager - конструктор	200
VoicesManager.FilterCutoffEnvelopeManager - свойство.....	200
VoicesManager.FilterManager - свойство.....	200
VoicesManager.ModulationTypeManager - свойство	201
VoicesManager.OscAManager - свойство	201
VoicesManager.OscAVolumeEnvelopeManager - свойство	201
VoicesManager.OscBManager - свойство	201
VoicesManager.OscBVolumeEnvelopeManager - свойство.....	201
VoicesManager.CreateVoice - метод.....	202
VoicesManager.InitializeParameters - метод.....	202
VoicesManager.OnSampleRateChanged - метод.....	202
VoicesManager.PlayNote - метод.....	203
VoicesManager.Process - метод.....	203
VoicesManager.ReleaseNote - метод.....	203
VoicesManager.SetModulationType - метод	204
VoicesManager.StopVoice - метод.....	204
activeVoices - поле.....	204
freeVoicesIndices - поле	205
modulationType - поле	205
noteToVoicesMapping - поле	205
voicesPool - поле	205
MaxVoicesCount - поле	206
WaveTableOscillator - класс	206
WaveTableOscillator(GeneratorFunction, Double, Double) - конструктор	206
WaveTableOscillator - конструктор.....	207
WaveTableOscillator.Clone - метод.....	207
WaveTableOscillator.Deserialize - метод	207
WaveTableOscillator.Normalize - метод	208
WaveTableOscillator.PrepareSamples - метод.....	208
WaveTableOscillator.Process - метод.....	209
WaveTableOscillator.Serialize - метод	209

WaveTableOscillator.SetPhaseIncrement - метод	209
WaveTable - класс	210
Length - поле	210
PhaseIncrement - поле	211
Samples - поле	211
GeneratorFunction - делегат	211
phaseIncrement - поле	212
waveTable - поле	212
waveTables - поле	212
waveTablesAmount - поле	212
DefaultSampleRate - поле	213
IDelay - интерфейс	213
IDelay.Process - метод	213
IDelay.Reset - метод	214
IDelay.SetDelay - метод	214
IDelay.SetFeedback - метод	214
IDelay.SetStereo - метод	215
IDistortion - интерфейс	215
IDistortion.Process - метод	215
IDistortion.SetAmount - метод	216
MidiNote - структура	216
MidiNote.NoteNo - свойство	216
MidiNote.Velocity - свойство	217
MultimodSynth.UI - пространство имен	218
BipolarKnob - класс	218
BipolarKnob - конструктор	223
BipolarKnob.Rotate - метод	223
EditorView - класс	223
EditorView.BindDelay - метод	229
EditorView.BindDistortion - метод	229
EditorView.BindFilter - метод	229
EditorView.BindKeyboard - метод	230
EditorView.BindMasterSettings - метод	230
EditorView.BindOscA - метод	230

EditorView.BindOscB - метод	231
EditorView.BindParameters - метод.....	231
EditorView.BindToPlugin - метод	232
EditorView.OpenButton_Click - метод	232
EditorView.SaveButton_Click - метод	232
plugin - поле.....	233
Knob - класс	233
Knob - конструктор.....	238
Knob.DisplayValue - свойство.....	238
Knob.DisplayValueConverter - свойство	239
Knob.Stroke - свойство	239
Knob.AttachTo - метод.....	239
Knob.Ellipse_MouseDown - метод.....	240
Knob.Ellipse_MouseMove - метод	240
Knob.Ellipse_MouseUp - метод.....	240
Knob.OnPropertyChanged - метод.....	241
Knob.PopupPlacementCallback - метод.....	241
Knob.Rotate - метод	242
Knob.SetCursorPos - метод	242
Knob.SetValue - метод.....	243
Knob.UpdateDisplayValue - метод	243
Knob.UpdateParameterValue - метод.....	243
PropertyChanged - событие	243
accumulatedValue - поле.....	244
displayValue - поле	244
leftButtonPressed - поле	244
manager - поле	244
startMousePosition - поле	245
startScreenMousePosition - поле	245
stroke - поле.....	245
value - поле	245
Delta - поле	245
Index	247

MultimodSynth Reference

Пространства имен

[MultimodSynth²³](#), [MultimodSynth.UI²¹⁸](#)

MultimodSynth - пространство имен

Классы

[AbsClipper](#)₂₃, [AdsrEnvelope](#)₂₄, [AudioComponent](#)₃₃, [AudioComponentWithParameters](#)₃₅,
[AudioProcessor](#)₃₇, [BitCrusher](#)₄₀, [Converters](#)₄₁, [CubicClipper](#)₅₄, [DCBlocker](#)₅₆, [DelayLine](#)₆₀,
[DelayManager](#)₆₄, [DistortionManager](#)₇₈, [Downsampler](#)₈₉, [Envelope](#)₉₂, [EnvelopesManager](#)₉₉, [Filter](#)₁₀₈,
[FiltersManager](#)₁₁₃, [MidiNoteEventArgs](#)₁₂₀, [MidiProcessor](#)₁₂₁, [Oscillator](#)₁₂₅, [OscillatorsManager](#)₁₃₀,
[ParameterFactory](#)₁₃₅, [ParameterFilter](#)₁₃₈, [PingPongDelay](#)₁₄₂, [Plugin](#)₁₄₇, [PluginCommandStub](#)₁₅₀,
[PluginEditor](#)₁₅₀, [PluginPrograms](#)₁₅₅, [Routing](#)₁₅₇, [SampleRateReductor](#)₁₆₃, [SineLFO](#)₁₆₆, [SoftClipper](#)₁₆₉,
[StereoOffsetDelay](#)₁₇₁, [SvfFilter](#)₁₇₆, [Utilities](#)₁₈₃, [VariousTimeDelay](#)₁₈₉, [Voice](#)₁₉₃, [VoicesManager](#)₁₉₉,
[WaveTableOscillator](#)₂₀₆

Интерфейсы

[IDelay](#)₂₁₃, [IDistortion](#)₂₁₅

Структуры

[MidiNote](#)₂₁₆

AbsClipper - класс

Представляет собой вариант эффекта сатурации.

[System.Object](#)

[MultimodSynth.AbsClipper](#)

C#

```
internal class AbsClipper : IDistortion
```

Требования

Пространство имен:[MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [Process](#)₂₃, [SetAmount](#)₂₄,
[ToString](#) (унаследован от [Object](#))

См. также

Ошибка! Недопустимый объект гиперссылки.

AbsClipper.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

Реализации

[IDistortion.Process](#)²¹⁵

См. также

Применяется: [AbsClipper](#)²³

AbsClipper.SetAmount - метод

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]). У данного варианта сатурации нет силы.

C#

```
public void SetAmount(  
    float value  
)
```

Параметры

value

Значение "силы" эффекта.

Реализации

[IDistortion.SetAmount](#)²¹⁶

См. также

Применяется: [AbsClipper](#)²³

AdsrEnvelope - класс

Реализация ADSR огибающей.

[System.Object](#)

[MultimodSynth.AdsrEnvelope](#)

C#

```
internal class AdsrEnvelope
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[AdsrEnvelope](#)²⁵

Свойства

[State](#)²⁵

Методы

[CalcCoef](#)²⁶, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSoundStop](#)²⁶, [Process](#)²⁶, [SetAttackRate](#)²⁷, [SetAttackTargetRatio](#)²⁷, [SetDecayRate](#)²⁷, [SetDecayReleaseTargetRatio](#)²⁸, [SetReleaseRate](#)²⁸, [SetSustainLevel](#)²⁸, [ToString](#) (унаследован от [Object](#)), [TriggerAttack](#)²⁹, [TriggerRelease](#)²⁹

События

[SoundStop](#)²⁹

Перечисления

[EnvelopeState](#)²⁹

Поля

[attackBase](#)³⁰, [attackCoef](#)³⁰, [attackRate](#)³⁰, [attackTargetRatio](#)³¹, [currValue](#)³¹, [decayBase](#)³¹, [decayCoef](#)³¹, [decayRate](#)³², [decayReleaseTargetRatio](#)³², [releaseBase](#)³², [releaseCoef](#)³², [releaseRate](#)³³, [state](#)³³, [sustainLevel](#)³³

См. также

Ошибка! Недопустимый объект гиперссылки.

AdsrEnvelope - конструктор

Инициализирует новый объект типа AdsrEnvelope.

C#

```
public AdsrEnvelope()
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.State - свойство

Текущее состояние огибающей.

C#

```
public EnvelopeState State {get;}
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.CalcCoef - метод

Возвращает коэффициент для вычисления значений огибающей.

C#

```
private double CalcCoef(  
    double rate,  
    double targetRatio  
)
```

Параметры

rate

targetRatio

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.OnSoundStop - метод

Метод, вызываемый при остановке генератора огибающей.

C#

```
private void OnSoundStop()
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.Process - метод

Генерация нового значения огибающей.

C#

```
public double Process()
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetAttackRate - метод

Устанавливает новое значение "скорости" стадии атаки.

C#

```
public void SetAttackRate(  
    double rate  
)
```

Параметры

rate

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetAttackTargetRatio - метод

Устанавливает разность целевого и максимального значения стадии атаки. Используется для управления изгибом огибающей.

C#

```
public void SetAttackTargetRatio(  
    double targetRatio  
)
```

Параметры

targetRatio

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetDecayRate - метод

Устанавливает новое значение "скорости" стадии спада.

C#

```
public void SetDecayRate(  
    double rate  
)
```

Параметры

rate

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetDecayReleaseTargetRatio - метод

Устанавливает разность целевого и максимального значения стадий спада и затухания.

Используется для управления изгибом огибающей.

C#

```
public void SetDecayReleaseTargetRatio(  
    double targetRatio  
)
```

Параметры

targetRatio

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetReleaseRate - метод

Устанавливает новое значение длительности стадии затухания (в сэмплах).

C#

```
public void SetReleaseRate(  
    double rate  
)
```

Параметры

rate

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.SetSustainLevel - метод

Устанавливает новое значение уровня стадии поддержки.

C#

```
public void SetSustainLevel(  
    double level
```

```
)
```

Параметры

level

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.TriggerAttack - метод

Метод, начинаящий стадию атаки.

C#

```
public void TriggerAttack()
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AdsrEnvelope.TriggerRelease - метод

Метод, начинаящий стадию затухания.

C#

```
public void TriggerRelease()
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

SoundStop - событие

Событие, возникающее при остановке генератора огибающей.

C#

```
public event EventHandler SoundStop
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

EnvelopeState - перечисление

Указывает стадию огибающей.

Константа	Значение	Описание
Attack	1	
Decay	2	
Idle	0	
Release	4	
Sustain	3	

Требования

Пространство имен:[MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [AdsrEnvelope](#)²⁴

attackBase - поле

Значение, которое прибавляется на стадии атаки.

C#

```
private double attackBase
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

attackCoef - поле

Коэффициент для стадии атаки.

C#

```
private double attackCoef
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

attackRate - поле

Длительность стадии атаки (в сэмплах).

C#

```
private double attackRate
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

attackTargetRatio - поле

Разность целевого значения и максимального значения стадии атаки.

C#

```
private double attackTargetRatio
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

currValue - поле

Текущее значение огибающей.

C#

```
private double currValue
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

decayBase - поле

Значение, которое прибавляется на стадии спада.

C#

```
private double decayBase
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

decayCoef - поле

Коэффициент для стадии спада.

C#

```
private double decayCoef
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

decayRate - поле

Длительность стадии спада (в сэмплах).

C#

```
private double decayRate
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

decayReleaseTargetRatio - поле

Разность целевого значения и максимального значения стадий спада и затухания.

C#

```
private double decayReleaseTargetRatio
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

releaseBase - поле

Значение, которое прибавляется на стадии затухания.

C#

```
private double releaseBase
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

releaseCoef - поле

Коэффициент для стадии затухания.

C#

```
private double releaseCoef
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

releaseRate - поле

Длительность стадии затухания (в сэмплах).

C#

```
private double releaseRate
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

state - поле

Текущее состояние огибающей.

C#

```
private EnvelopeState state
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

sustainLevel - поле

Уровень стадии поддержки.

C#

```
private double sustainLevel
```

См. также

Применяется: [AdsrEnvelope](#)²⁴

AudioComponent - класс

Базовый класс, предоставляющий своим наследникам информацию о частоте дискретизации.

[System.Object](#)

MultimodSynth.AudioComponent

[MultimodSynth.AudioComponentWithParameters](#)³⁵

[MultimodSynth.DCBlocker](#)⁵⁶

[MultimodSynth.DelayLine](#)⁶⁰

[MultimodSynth.Downsampler](#)⁸⁹

[MultimodSynth.Envelope](#)⁹²

[MultimodSynth.Filter](#)¹⁰⁸

[MultimodSynth.Oscillator](#)¹²⁵

[MultimodSynth.ParameterFilter](#)¹³⁸

[MultimodSynth.PingPongDelay](#)¹⁴²

[MultimodSynth.SampleRateReductor](#)¹⁶³

[MultimodSynth.SineLFO](#)¹⁶⁶
[MultimodSynth.StereoOffsetDelay](#)¹⁷¹
[MultimodSynth.SvfFilter](#)¹⁷⁶
[MultimodSynth.VariousTimeDelay](#)¹⁸⁹
[MultimodSynth.Voice](#)¹⁹³

C#

```
internal abstract class AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

[SampleRate](#)³⁴

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)³⁴,
[ToString](#) (унаследован от [Object](#))

Поля

[sampleRate](#)³⁵

AudioComponent.SampleRate - свойство

Частота дискретизации, на которой работает данный компонент

C#

```
public float SampleRate {get; set;}
```

См. также

Применяется: [AudioComponent](#)³³

AudioComponent.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected virtual void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

34

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [AudioComponent](#)³³

sampleRate - поле**C#**

```
private float sampleRate
```

См. также

Применяется: [AudioComponent](#)³³

AudioComponentWithParameters - класс

Базовый класс, предоставляющий своим наследникам доступ к созданию параметров.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

MultimodSynth.AudioComponentWithParameters

[MultimodSynth.DelayManager](#)⁶⁴

[MultimodSynth.DistortionManager](#)⁷⁸

[MultimodSynth.EnvelopesManager](#)⁹⁹

[MultimodSynth.FiltersManager](#)¹¹³

[MultimodSynth.OscillatorsManager](#)¹³⁰

[MultimodSynth.Routing](#)¹⁵⁷

[MultimodSynth.VoicesManager](#)¹⁹⁹

C#

```
internal abstract class AudioComponentWithParameters : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[AudioComponentWithParameters](#)³⁶

Свойства

[Plugin](#)³⁶, [SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),

[GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)³⁷, [MemberwiseClone](#) (унаследован от [Object](#)),

[OnSampleRateChanged](#)³⁴ (унаследован от [AudioComponent](#)), [ToString](#) (унаследован от [Object](#))

Поля

[parameterCategory](#)³⁷, [parameterPrefix](#)³⁷

AudioComponentWithParameters - конструктор

Инициализирует объект класса AudioComponentWithParameters, связанный с переданным плагином.

C#

```
public AudioComponentWithParameters(  
    Plugin plugin,  
    string parameterPrefix,  
    string parameterCategory = "plugin"  
)
```

Параметры

plugin

parameterPrefix

parameterCategory

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

AudioComponentWithParameters.Plugin - свойство

Объект класса Plugin, который содержит данный компонент.

C#

```
public Plugin Plugin {get;}
```

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

AudioComponentWithParameters.InitializeParameters - метод

Инициализирует параметры. Производные классы вызывают этот метод где-то в конструкторе (обычно в конце).

C#

```
protected void InitializeParameters()
```

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

AudioComponentWithParameters.InitializeParameters (ParameterFactory) - метод

Производные классы переопределяют этот метод, чтобы в нём создать свои параметры с помощью посланного им объекта класса ParameterFactory.

C#

```
protected abstract void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

parameterCategory - поле

Категория, которой принадлежат создаваемые параметры.

C#

```
private string parameterCategory
```

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

parameterPrefix - поле

C#

```
private string parameterPrefix
```

См. также

Применяется: [AudioComponentWithParameters](#)³⁵

AudioProcessor - класс

Реализация интерфейса IVstPluginAudioProcessor.

C#

```
internal class AudioProcessor : VstPluginAudioProcessorBase
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[AudioProcessor](#)³⁸

Свойства

[ProcessingMutex](#)³⁸, [Routing](#)³⁹, [SampleRate](#)³⁹

Методы

[Process](#)³⁹

Поля

[plugin](#)³⁹

AudioProcessor - конструктор

Инициализирует новый объект типа AudioProcessor, принадлежащий заданному плагину.

C#

```
public AudioProcessor(  
    Plugin plugin  
)
```

Параметры

plugin

См. также

Применяется: [AudioProcessor](#)³⁷

AudioProcessor.ProcessingMutex - свойство

Мьютекс, используемый для синхронизации UI-потока и основного потока.

C#

```
public Mutex ProcessingMutex {get; set;}
```

См. также

Применяется: [AudioProcessor](#)³⁷

AudioProcessor.Routing - свойство

Ссылка на объект, предстающий собой всю цепочку создания и обработки звука.

C#

```
public Routing Routing {get; private set;}
```

См. также

Применяется: [AudioProcessor](#)³⁷

AudioProcessor.SampleRate - свойство

Текущая частота дискретизации.

C#

```
public override float SampleRate {get; set;}
```

См. также

Применяется: [AudioProcessor](#)³⁷

AudioProcessor.Process - метод

Метод, обрабатывающий входные данные, поступающие от плагина, и генерирующий новые выходные данные.

C#

```
public override void Process(  
    VstAudioBuffer[] inChannels,  
    VstAudioBuffer[] outChannels  
)
```

Параметры

inChannels

Входные каналы.

outChannels

Выходные каналы.

См. также

Применяется: [AudioProcessor](#)³⁷

plugin - поле

Ссылка на плагин, которому принадлежит этот компонент.

C#

```
private Plugin plugin
```

См. также

Применяется: [AudioProcessor](#)³⁷

BitCrusher - класс

Представляет собой эффект ухудшения качества звука путём уменьшения глубины квантования.

[System.Object](#)

[MultimodSynth.BitCrusher](#)

C#

```
internal class BitCrusher : IDistortion
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [Process](#)⁴⁰, [SetAmount](#)⁴¹, [ToString](#) (унаследован от [Object](#))

Поля

[steps](#)⁴¹

BitCrusher.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

Реализации

40

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

[IDistortion.Process](#)²¹⁵

См. также

Применяется: [BitCrusher](#)⁴⁰

BitCrusher.SetAmount - метод

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]). Чем больше сила, тем меньше глубина квантования.

C#

```
public void SetAmount(  
    float value  
)
```

Параметры

value

Значение "силы" эффекта.

Реализации

[IDistortion.SetAmount](#)²¹⁶

См. также

Применяется: [BitCrusher](#)⁴⁰

steps - поле

Количество возможных уровней входного сигнала.

C#

```
private float steps
```

См. также

Применяется: [BitCrusher](#)⁴⁰

Converters - класс

Содержит методы для преобразования нормализованных значений параметра в реальные и в отображаемые значения.

[System.Object](#)

[MultimodSynth.Converters](#)

C#

```
internal static class Converters
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[AsymmetryToString](#)⁴², [CentsToString](#)⁴², [DelayLfoRateToString](#)⁴³, [DelayModeToString](#)⁴³,
[DelayTimeToString](#)⁴³, [DistortionAmpToString](#)⁴⁴, [DistortionLowpassCutoffToString](#)⁴⁴,
[DistortionModeToString](#)⁴⁴, [EnvelopeCurveToString](#)⁴⁵, [EnvelopeTimeToString](#)⁴⁵, [Equals](#) (унаследован от
Object), [FilterCutoffMultiplierToString](#)⁴⁵, [FilterTypeToString](#)⁴⁶, [Finalize](#) (унаследован от Object),
[GetHashCode](#) (унаследован от Object), [GetType](#) (унаследован от Object), [InvertToString](#)⁴⁶,
[MemberwiseClone](#) (унаследован от Object), [ModulationTypeToString](#)⁴⁷, [OversamplingOrderToString](#)⁴⁷,
[PercentsToString](#)⁴⁷, [SemitonesToString](#)⁴⁸, [StereoAmountToString](#)⁴⁸, [ToAsymmetry](#)⁴⁸, [ToCents](#)⁴⁹,
[ToDelayLfoRate](#)⁴⁹, [ToDelayMode](#)⁴⁹, [ToDelayTime](#)⁵⁰, [ToDistortionAmp](#)⁵⁰, [ToDistortionLowpassCutoff](#)⁵¹,
[ToDistortionMode](#)⁵¹, [ToEnvelopeTime](#)⁵¹, [ToFilterCutoffMultiplier](#)⁵², [ToFilterType](#)⁵²,
[ToModulationType](#)⁵², [ToOversamplingOrder](#)⁵³, [ToSemitones](#)⁵³, [ToStereoAmount](#)⁵³, [ToString](#)
(унаследован от Object), [ToWaveTable](#)⁵⁴, [WaveTableToString](#)⁵⁴

Converters.AsymmetryToString - метод

Возвращает строковое представление количества ассиметрии эффекта дисторшн.

C#

```
public static string AsymmetryToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.CentsToString - метод

Возвращает строковое представление подстройки по частоте (в центах).

C#

```
public static string CentsToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.DelayLfoRateToString - метод

Возвращает строковое представление частоты генератора низких частот эффекта дилэй.

C#

```
public static string DelayLfoRateToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.DelayModeToString - метод

Возвращает строковое представление режима эффекта дилэй.

C#

```
public static string DelayModeToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.DelayTimeToString - метод

Возвращает строковое представление времени задержки эффекта дилэй.

C#

```
public static string DelayTimeToString(  
    double value  
)
```

Параметры*value***См. также**Применяется: [Converters₄₁](#)**Converters.DistortionAmpToString - метод**

Возвращает строковое представление увеличения громкости эффекта дисторшн.

C#

```
public static string DistortionAmpToString(  
    double value  
)
```

Параметры*value***См. также**Применяется: [Converters₄₁](#)**Converters.DistortionLowpassCutoffToString - метод**

Возвращает строковое представление частоты среза фильтра эффекта дисторшн.

C#

```
public static string DistortionLowpassCutoffToString(  
    double value  
)
```

Параметры*value***См. также**Применяется: [Converters₄₁](#)**Converters.DistortionModeToString - метод**

Возвращает строковое представление режима эффекта дисторшн.

C#

```
public static string DistortionModeToString(  
    double value
```

```
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.EnvelopeCurveToString - метод

Возвращает строковое представление изгиба стадии огибающей.

C#

```
public static string EnvelopeCurveToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.EnvelopeTimeToString - метод

Возвращает строковое представление длительности стадии огибающей.

C#

```
public static string EnvelopeTimeToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.FilterCutoffMultiplierToString - метод

Возвращает строковое представление множителя частоты среза фильтра.

C#

```
public static string FilterCutoffMultiplierToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)₄₁

Converters.FilterTypeToString - метод

Возвращает строковое представление типа фильтра.

C#

```
public static string FilterTypeToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)₄₁

Converters.InvertToString - метод

Возвращает строковое представление параметра инверсии эффекта дилэй.

C#

```
public static string InvertToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)₄₁

Converters.ModulationTypeToString - метод

Возвращает строковое представление типа модуляции.

C#

```
public static string ModulationTypeToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters₄₁](#)

Converters.OversamplingOrderToString - метод

Возвращает строковое представление множителя частоты дискретизации.

C#

```
public static string OversamplingOrderToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters₄₁](#)

Converters.PercentToString - метод

Возвращает строковое представление значения в процентах.

C#

```
public static string PercentToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.SemitonesToString - метод

Возвращает строковое представление подстройки по частоте (в полутонах).

C#

```
public static string SemitonesToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.StereoAmountToString - метод

Возвращает строковое представление стерео-эффекта эффекта дилэй.

C#

```
public static string StereoAmountToString(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToAsymmetry - метод

Конвертирует нормализованное значение параметра в количество асимметрии эффекта дисторшн.

C#

```
public static double ToAsymmetry(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToCents - метод

Конвертирует нормализованное значение параметра в подстройку по частоте (в центах).

C#

```
public static int ToCents(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToDoubleDelayLfoRate - метод

Конвертирует нормализованное значение параметра в частоту генератора низких частот эффекта дилэй.

C#

```
public static double ToDelayLfoRate(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToDoubleDelayMode - метод

Конвертирует нормализованное значение параметра в режим эффекта дилэй.

C#

```
public static StereoMode ToDelayMode(  
    double value  
)
```

```
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToDelayTime - метод

Конвертирует нормализованное значение параметра в время задержки эффекта дилэй.

C#

```
public static double ToDelayTime(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToDistortionAmp - метод

Конвертирует нормализованное значение параметра в увеличение уровня громкости эффекта дисторшн.

C#

```
public static double ToDistortionAmp(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToDistortionLowpassCutoff - метод

Конвертирует нормализованное значение параметра в частоту среза фильтра эффекта дисторшн.

C#

```
public static double ToDistortionLowpassCutoff(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters₄₁](#)

Converters.ToDistortionMode - метод

Конвертирует нормализованное значение параметра в режим эффекта дисторшн.

C#

```
public static DistortionMode ToDistortionMode(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters₄₁](#)

Converters.ToEnvelopeTime - метод

Конвертирует нормализованное значение параметра в длительность стадии огибающей.

C#

```
public static double ToEnvelopeTime(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToFilterCutoffMultiplier - метод

Конвертирует нормализованное значение параметра в множитель частоты среза фильтра.

C#

```
public static double ToFilterCutoffMultiplier(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToFilterType - метод

Конвертирует нормализованное значение параметра в тип фильтра.

C#

```
public static FilterType ToFilterType(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters](#)⁴¹

Converters.ToModulationType - метод

Конвертирует нормализованное значение параметра в тип модуляции.

C#

```
public static ModulationType ToModulationType(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.ToOversamplingOrder - метод

Конвертирует нормализованное значение параметра в множитель частоты дискретизации.

C#

```
public static int ToOversamplingOrder(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.ToSemitones - метод

Конвертирует нормализованное значение параметра в подстройку по частоте (в полутонах).

C#

```
public static int ToSemitones(  
    double value  
)
```

Параметры

value

См. также

Применяется: [Converters⁴¹](#)

Converters.ToStereoAmount - метод

Конвертирует нормализованное значение параметра в стерео-эффект эфекта дилэй.

C#

```
public static double ToStereoAmount(  
    double value  
)
```

Параметры*value***См. так же**Применяется: [Converters](#)⁴¹**Converters.ToWaveTable - метод**

Конвертирует нормализованное значение параметра в объект WaveTableOscillator.

C#

```
public static WaveTableOscillator ToWaveTable(  
    double value  
)
```

Параметры*value***См. так же**Применяется: [Converters](#)⁴¹**Converters.WaveTableToString - метод**

Возвращает строковое представление выбранного объекта WaveTableOscillator.

C#

```
public static string WaveTableToString(  
    double value  
)
```

Параметры*value***См. так же**Применяется: [Converters](#)⁴¹**CubicClipper - класс**

Представляет собой вариант эффекта сатурации.

[System.Object](#)[MultimodSynth.CubicClipper](#)**C#**

```
internal class CubicClipper : IDistortion
```

Требования

Пространство имен: MultimodSynth²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [Process⁵⁵](#), [SetAmount⁵⁵](#), [ToString](#) (унаследован от [Object](#))

См. также

Ошибка! Недопустимый объект гиперссылки.

CubicClipper.Process - метод

Обработка новых входных данных.

C#

```
public float Process(
    float input
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

Реализации

[IDistortion.Process²¹⁵](#)

См. также

Применяется: [CubicClipper⁵⁴](#)

CubicClipper.SetAmount - метод

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]). У этого варианта сатурации нет "силы".

C#

```
public void SetAmount(
    float value
)
```

Параметры

value

Значение "силы" эффекта.

Реализации

[IDistortion.SetAmount](#)²¹⁶

См. также

Применяется: [CubicClipper](#)⁵⁴

DCBlocker - класс

Представляет собой фильтр, убирающий постоянное амплитудное смещение.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.DCBlocker](#)

C#

```
internal class DCBlocker : AudioComponent
```

Требования

Пространство имен:[MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[DCBlocker](#)⁵⁶

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)⁵⁷,
[Process](#)⁵⁷, [SetResponseTimeCoefficient](#)⁵⁸, [ToString](#) (унаследован от [Object](#))

Поля

[BaseSampleRate](#)⁵⁹, [DefaultExp](#)⁵⁸, [normalizationCoeff](#)⁵⁸, [r](#)⁵⁸, [responseTimeCoefficient](#)⁵⁹, [xm1](#)⁵⁹, [ym1](#)⁵⁹

См. также

Ошибка! Недопустимый объект гиперссылки.

DCBlocker - конструктор

Инициализирует новый объект класса DCBlocker.

C#

```
public DCBlocker(  
    float responseTimeCoefficient = 1  
)
```

Параметры

responseTimeCoefficient

Коэффициент скорости отклика фильтра

См. также

Применяется: [DCBlocker](#)⁵⁶

DCBlocker.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [DCBlocker](#)⁵⁶

DCBlocker.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float x  
)
```

Параметры

x

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [DCBlocker](#)₅₆

DCBlocker.SetResponseTimeCoefficient - метод

Устанавливает коэффициент времени отклика фильтра.

C#

```
public void SetResponseTimeCoefficient(  
    float value  
)
```

Параметры

value

См. также

Применяется: [DCBlocker](#)₅₆

DefaultExp - поле

Коэффициент для генерации коэффициента скорости отклика фильтра.

C#

```
private double DefaultExp
```

См. также

Применяется: [DCBlocker](#)₅₆

normalizationCoeff - поле

Коэффициент для нормализации выходного сигнала.

C#

```
private float normalizationCoeff
```

См. также

Применяется: [DCBlocker](#)₅₆

r - поле

Коэффициент фильтра.

C#

```
private float r
```

См. также

Применяется: [DCBlocker](#)⁵⁶

responseTimeCoefficient - поле

Коэффициент скорости отклика.

C#

```
private float responseTimeCoefficient
```

См. также

Применяется: [DCBlocker](#)⁵⁶

xm1 - поле

Предыдущие значения входного и выходного сигнала.

C#

```
private float xm1
```

См. также

Применяется: [DCBlocker](#)⁵⁶

ym1 - поле

Предыдущие значения входного и выходного сигнала.

C#

```
private float ym1
```

См. также

Применяется: [DCBlocker](#)⁵⁶

BaseSampleRate - поле

Частота дискретизации, для которой вычислен коэффициент DefaultExp.

C#

```
private const float BaseSampleRate = 44100
```

См. также

Применяется: [DCBlocker](#)⁵⁶

DelayLine - класс

Представляет собой линию задержки.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

MultimodSynth.DelayLine

C#

```
internal class DelayLine : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[CalculateOutput](#)⁶⁰, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)⁶¹, [Process](#)⁶¹, [Reset](#)⁶¹, [SetDelay](#)⁶², [SetFeedback](#)⁶², [ToString](#) (унаследован от [Object](#))

Поля

[MaxTime](#)⁶⁴, [buffer](#)⁶², [bufferLength](#)⁶², [delay](#)⁶³, [feedback](#)⁶³, [readOffset](#)⁶³, [readPoint](#)⁶³, [writePoint](#)⁶⁴

См. также

Ошибка! Недопустимый объект гиперссылки.

DelayLine.CalculateOutput - метод

Возвращает значение в точке чтения (используется линейная интерполяция).

C#

```
public float CalculateOutput()
```

Возвращаемые значения

Значение в точке чтения.

См. также

Применяется: [DelayLine](#)⁶⁰

DelayLine.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [DelayLine₆₀](#)

DelayLine.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [DelayLine₆₀](#)

DelayLine.Reset - метод

Очищает буфер линии задержки.

C#

```
public void Reset()
```

См. также

Применяется: [DelayLine₆₀](#)

DelayLine.SetDelay - метод

Устанавливает новое значение времени задержки (в сэмплах).

C#

```
public void SetDelay(  
    float delay  
)
```

Параметры

delay

Время задержки.

См. также

Применяется: [DelayLine₆₀](#)

DelayLine.SetFeedback - метод

Устанавливает новое значение коэффициента обратной связи.

C#

```
public void SetFeedback(  
    float value  
)
```

Параметры

value

Коэффициент обратной связи.

См. также

Применяется: [DelayLine₆₀](#)

buffer - поле

Буфер, используемый для хранения предыдущих сэмплов.

C#

```
private float[] buffer
```

См. также

Применяется: [DelayLine₆₀](#)

bufferLength - поле

Длина буфера.

C#

```
private int bufferLength
```

См. также

Применяется: [DelayLine₆₀](#)

delay - поле

Время задержки (в сэмплах).

C#

```
private float delay
```

См. также

Применяется: [DelayLine₆₀](#)

feedback - поле

Коэффициент обратной связи.

C#

```
private float feedback
```

См. также

Применяется: [DelayLine₆₀](#)

readOffset - поле

Сдвиг точки чтения относительно указателя для чтения. Используется для нецелочисленного времени задержки.

C#

```
private float readOffset
```

См. также

Применяется: [DelayLine₆₀](#)

readPoint - поле

Указатель на ячейку буфера для чтения.

C#

```
private int readPoint
```

См. также

Применяется: [DelayLine₆₀](#)

writePoint - поле

Указатель на ячейку буфера для записи.

C#

```
private int writePoint
```

См. также

Применяется: [DelayLine₆₀](#)

MaxTime - поле

Максимальное время задержки (в секундах).

C#

```
private const float MaxTime = 1
```

См. также

Применяется: [DelayLine₆₀](#)

DelayManager - класс

Компонент плагина, отвечающий за эффект "дилэй".

[System.Object](#)

[MultimodSynth.AudioComponent₃₃](#)

[MultimodSynth.AudioComponentWithParameters₃₅](#)

MultimodSynth.DelayManager

C#

```
internal class DelayManager : AudioComponentWithParameters
```

Требования

Пространство имен: [MultimodSynth₂₃](#)

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[DelayManager](#)⁶⁵

Свойства

[FeedbackManager](#)⁶⁵, [InvertManager](#)⁶⁶, [LfoDepthManager](#)⁶⁶, [LfoRateManager](#)⁶⁶, [MixManager](#)⁶⁶, [ModeManager](#)⁶⁷, [Plugin](#)³⁶ (унаследован от [AudioComponentWithParameters](#)), [SampleRate](#)³⁴ (унаследован от [AudioComponent](#)), [StereoAmountManager](#)⁶⁷, [TimeManager](#)⁶⁷

Методы

[ChangeDelay](#)⁶⁷, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)⁶⁸, [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)⁶⁸, [Process](#)⁶⁸, [SetFeedback](#)⁶⁹, [SetInvert](#)⁶⁹, [SetLfoDepth](#)⁷⁰, [SetLfoRate](#)⁷⁰, [SetMixTarget](#)⁷⁰, [SetMode](#)⁷¹, [SetStereoAmountTarget](#)⁷¹, [SetTimeTarget](#)⁷², [ToString](#) (унаследован от [Object](#)), [UpdateMix](#)⁷², [UpdateStereoAmount](#)⁷², [UpdateTime](#)⁷³

Перечисления

[StereoMode](#)⁷³

Поля

[MaxLfoDepth](#)⁷⁷, [MaxTime](#)⁷⁸, [currentDelay](#)⁷³, [delay](#)⁷⁴, [dryCoeff](#)⁷⁴, [feedback](#)⁷⁴, [lfo](#)⁷⁴, [lfoDepth](#)⁷⁴, [maxDelay](#)⁷⁵, [mixFilter](#)⁷⁵, [mode](#)⁷⁵, [pingPongDelay](#)⁷⁵, [stereoAmount](#)⁷⁶, [stereoAmountFilter](#)⁷⁶, [stereoOffsetDelay](#)⁷⁶, [timeFilter](#)⁷⁶, [variousTimeDelay](#)⁷⁷, [wetCoeff](#)⁷⁷, [wetSign](#)⁷⁷

DelayManager - конструктор

Инициализирует новый объект класса DelayManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public DelayManager(  
    Plugin plugin,  
    string parameterPrefix = "DL"  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [DelayManager](#)⁶⁴

DelayManager.FeedbackManager - свойство

Объект, управляющий параметром коэффициента обратной связи.

C#

```
public VstParameterManager FeedbackManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.InvertManager - свойство

Объект, управляющий параметром инвертирования выходного сигнала.

C#

```
public VstParameterManager InvertManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.LfoDepthManager - свойство

Объект, управляющий параметром амплитуды генератора низких частот.

C#

```
public VstParameterManager LfoDepthManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.LfoRateManager - свойство

Объект, управляющий параметром частоты генератора низких частот.

C#

```
public VstParameterManager LfoRateManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.MixManager - свойство

Объект, управляющий параметром количества входного и выходного сигналов.

C#

```
public VstParameterManager MixManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.ModeManager - свойство

Объект, управляющий параметром типа эффекта.

C#

```
public VstParameterManager ModeManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.StereoAmountManager - свойство

Объект, управляющий параметром стерео.

C#

```
public VstParameterManager StereoAmountManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.TimeManager - свойство

Объект, управляющий параметром времени задержки.

C#

```
public VstParameterManager TimeManager {get; set;}
```

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.ChangeDelay - метод

Изменяет текущий объект дилэя на новый.

C#

```
private void ChangeDelay(
```

```
    IDelay newDelay  
)
```

Параметры

newDelay

Новый объект дилэя.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.Process - метод

Обработка новых входных данных.

C#

```
public void Process(
    float inputL,
    float inputR,
    out float outputL,
    out float outputR
)
```

Параметры

inputL

Левый канал входного сигнала

inputR

Правый канал входного сигнала

outputL

Левый канал выходного сигнала.

outputR

Правый канал выходного сигнала.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.SetFeedback - метод

Обработчик изменения коэффициента обратной связи.

C#

```
private void SetFeedback(
    float value
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.SetInvert - метод

Обработчик изменения параметра инвертирования выходного сигнала.

C#

```
private void SetInvert(
    float value
)
```

Параметры*value***См. также**Применяется: [DelayManager₆₄](#)**DelayManager.SetLfoDepth - метод**

Обработчик изменения амплитуды генератора низких частот.

C#

```
private void SetLfoDepth(  
    float value  
)
```

Параметры*value***См. также**Применяется: [DelayManager₆₄](#)**DelayManager.SetLfoRate - метод**

Обработчик изменения частоты генератора низких частот.

C#

```
private void SetLfoRate(  
    float value  
)
```

Параметры*value***См. также**Применяется: [DelayManager₆₄](#)**DelayManager.SetMixTarget - метод**

Обработчик изменения параметра количества выходного и входного сигнала.

C#

```
private void SetMixTarget(  
    float target  
)
```

Параметры

target

Нормированное новое значение параметра.

См. так же

Применяется: [DelayManager₆₄](#)

DelayManager.SetMode - метод

Обработчик изменения типа дилэя.

C#

```
private void SetMode(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. так же

Применяется: [DelayManager₆₄](#)

DelayManager.SetStereoAmountTarget - метод

Обработчик изменения параметра стерео-эффекта.

C#

```
private void SetStereoAmountTarget(  
    float target  
)
```

Параметры

target

Нормированное новое значение параметра.

См. так же

Применяется: [DelayManager₆₄](#)

DelayManager.SetTimeTarget - метод

Обработчик изменения времени задержки.

C#

```
private void SetTimeTarget(  
    float target  
)
```

Параметры

target

Нормированное новое значение параметра.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.UpdateMix - метод

Обработчик изменения "сглаженного" значения количества выходного и входного сигнала.

C#

```
private void UpdateMix(  
    float value  
)
```

Параметры

value

Новое значение.

См. также

Применяется: [DelayManager₆₄](#)

DelayManager.UpdateStereoAmount - метод

Обработчик изменения "сглаженного" значения параметра стерео-эффекта.

C#

```
private void UpdateStereoAmount(  
    float value  
)
```

Параметры

value

Новое значение стерео-эффекта.

См. также

Применяется: [DelayManager](#)₆₄

DelayManager.UpdateTime - метод

Обработчик изменения "сглаженного" значения времени задержки.

C#

```
private void UpdateTime(  
    float value  
)
```

Параметры

value

Новое значение времени задержки.

См. также

Применяется: [DelayManager](#)₆₄

StereoMode - перечисление

Указывает тип эффекта дилэй.

Константа	Значение	Описание
None	0	
PingPong	3	
StereoOffset	1	
VariousTime	2	

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [DelayManager](#)₆₄

currentDelay - поле

Текущий объект эффекта дилэй.

C#

```
private IDelay currentDelay
```

См. также

Применяется: [DelayManager](#)₆₄

delay - поле

Текущее время задержки (в секундах).

C#

```
private float delay
```

См. также

Применяется: [DelayManager₆₄](#)

dryCoeff - поле

Текущий коэффициент "чистого" сигнала без эффекта.

C#

```
private float dryCoeff
```

См. также

Применяется: [DelayManager₆₄](#)

feedback - поле

Текущий коэффициент обратной связи.

C#

```
private float feedback
```

См. также

Применяется: [DelayManager₆₄](#)

lfo - поле

Генератор низких частот.

C#

```
private SineLFO lfo
```

См. также

Применяется: [DelayManager₆₄](#)

lfoDepth - поле

Текущая амплитуда генератора низких частот (в диапазоне [0, 1]).

C#

```
private float lfoDepth
```

См. также

Применяется: [DelayManager₆₄](#)

maxDelay - поле

Максимальное значение времени задержки (в сэмплах).

C#

```
private float maxDelay
```

См. также

Применяется: [DelayManager₆₄](#)

mixFilter - поле

Фильтр низких частот, используемый для сглаживания параметра количества входного сигнала и выходного.

C#

```
private ParameterFilter mixFilter
```

См. также

Применяется: [DelayManager₆₄](#)

mode - поле

Текущий тип эффекта дилэй.

C#

```
private StereoMode mode
```

См. также

Применяется: [DelayManager₆₄](#)

pingPongDelay - поле

Объект класа PingPongDelay.

C#

```
private readonly PingPongDelay pingPongDelay
```

См. также

Применяется: [DelayManager₆₄](#)

stereoAmount - поле

Текущее значение коэффициента, показывающего, как сильно отличаются левый и правый канал выходного сигнала (значение в диапазоне [-1, 1]).

C#

```
private float stereoAmount
```

См. также

Применяется: [DelayManager₆₄](#)

stereoAmountFilter - поле

Фильтр низких частот, используемый для сглаживания параметра стерео.

C#

```
private ParameterFilter stereoAmountFilter
```

См. также

Применяется: [DelayManager₆₄](#)

stereoOffsetDelay - поле

Объект класса StereoOffsetDelay.

C#

```
private readonly StereoOffsetDelay stereoOffsetDelay
```

См. также

Применяется: [DelayManager₆₄](#)

timeFilter - поле

Фильтр низких частот, используемый для сглаживания параметра времени задержки.

C#

```
private ParameterFilter timeFilter
```

См. также

Применяется: [DelayManager₆₄](#)

variousTimeDelay - поле

Объект класса VariousTimeDelay.

C#

```
private readonly VariousTimeDelay variousTimeDelay
```

См. также

Применяется: [DelayManager₆₄](#)

wetCoeff - поле

Текущий коэффициент сигнала с применённым к нему эффектом.

C#

```
private float wetCoeff
```

См. также

Применяется: [DelayManager₆₄](#)

wetSign - поле

Знак сигнала с применённым к нему эффектом.

C#

```
private int wetSign
```

См. также

Применяется: [DelayManager₆₄](#)

MaxLfoDepth - поле

Максимальная амплитуда генератора низких частот.

C#

```
private const float MaxLfoDepth = 0,05
```

См. также

Применяется: [DelayManager](#)⁶⁴

MaxTime - поле

Максимальное время задержки в секундах.

C#

```
private const float MaxTime = 1
```

См. также

Применяется: [DelayManager](#)⁶⁴

DistortionManager - класс

Компонент плагина, отвечающий за эффект "дисторшн".

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.AudioComponentWithParameters](#)³⁵

[MultimodSynth.DistortionManager](#)

C#

```
internal class DistortionManager : AudioComponentWithParameters
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[DistortionManager](#)⁷⁹

Свойства

[AmountManager](#)⁷⁹, [AmpManager](#)⁷⁹, [AsymmetryManager](#)⁸⁰, [LowPassCutoffManager](#)⁸⁰, [MixManager](#)⁸⁰,

[ModeManager](#)⁸⁰, [Plugin](#)³⁶ (унаследован от [AudioComponentWithParameters](#)), [SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[ChangeDistortion](#)⁸¹, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)⁸¹, [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)⁸¹, [Process](#)⁸², [SetAmount](#)⁸², [SetAmpTarget](#)⁸²,

[SetAsymmetryTarget⁸³](#), [SetLowPassCutoff⁸³](#), [SetMode⁸⁴](#), [ToString](#) (унаследован от [Object](#)), [UpdateAmp⁸⁴](#), [UpdateAsymmetry⁸⁴](#), [UpdateMix⁸⁵](#)

Перечисления

[DistortionMode⁸⁵](#)

Поля

[absClipper⁸⁵](#), [amount⁸⁶](#), [amp⁸⁶](#), [ampFilter⁸⁶](#), [asymmetryFilter⁸⁶](#), [bitCrusher⁸⁷](#), [cubicClipper⁸⁷](#), [currentDistortion⁸⁷](#), [dcBlocker⁸⁷](#), [dcOffset⁸⁸](#), [dryCoeff⁸⁸](#), [lowPass⁸⁸](#), [mixFilter⁸⁸](#), [mode⁸⁹](#), [sampleRateReductor⁸⁹](#), [softClipper⁸⁹](#), [wetCoeff⁸⁹](#)

DistortionManager - конструктор

Инициализирует новый объект класса DistortionManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public DistortionManager(
    Plugin plugin,
    string parameterPrefix = "D"
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [DistortionManager⁷⁸](#)

DistortionManager.AmountManager - свойство

Объект, управляющий параметром силы эффекта.

C#

```
public VstParameterManager AmountManager {get; private set;}
```

См. также

Применяется: [DistortionManager⁷⁸](#)

DistortionManager.AmpManager - свойство

Объект, управляющий параметром громкости входного сигнала.

C#

```
public VstParameterManager AmpManager {get; private set;}
```

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.AsymmetryManager - свойство

Объект, управляющий параметром постоянного амплитудного сдвига.

C#

```
public VstParameterManager AsymmetryManager {get; private set;}
```

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.LowPassCutoffManager - свойство

Объект, управляющий параметром частоты среза фильтра низких частот.

C#

```
public VstParameterManager LowPassCutoffManager {get; private set;}
```

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.MixManager - свойство

Объект, управляющий параметром количества выходного и входного сигналов.

C#

```
public VstParameterManager MixManager {get; private set;}
```

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.ModeManager - свойство

Объект, управляющий параметром типа эффекта.

C#

```
public VstParameterManager ModeManager {get; private set;}
```

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.ChangeDistortion - метод

Изменяет текущий объект дисторшна на новый.

C#

```
private void ChangeDistortion(  
    IDistortion newDistortion  
)
```

Параметры

newDistortion

Новый объект дисторшна.

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры*newSampleRate*

Новая частота дискретизации.

См. такжеПрименяется: [DistortionManager](#)⁷⁸

DistortionManager.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры*input*

Входящий сигнал

Возвращаемые значения

Выходящий сигнал

См. такжеПрименяется: [DistortionManager](#)⁷⁸

DistortionManager.SetAmount - метод

Обработчик изменения силы эффекта.

C#

```
private void SetAmount(  
    float value  
)
```

Параметры*value*

Нормированное новое значение параметра.

См. такжеПрименяется: [DistortionManager](#)⁷⁸

DistortionManager.SetAmpTarget - метод

Обработчик изменения уровня входного сигнала.

C#

```
private void SetAmpTarget(  
    float value  
)
```

Параметры*value*

Нормированное новое значение параметра.

См. так же

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.SetAsymmetryTarget - метод

Обработчик изменения коэффициента постоянного амплитудного сдвига.

C#

```
private void SetAsymmetryTarget(  
    float value  
)
```

Параметры*value*

Нормированное новое значение параметра.

См. так же

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.SetLowPassCutoff - метод

Обработчик изменения частоты среза фильтра низких частот.

C#

```
private void SetLowPassCutoff(  
    float value  
)
```

Параметры*value*

Нормированное новое значение параметра.

См. так же

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.SetMode - метод

Обработчик изменения типа дисторшна.

C#

```
private void SetMode(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.UpdateAmp - метод

Обработчик изменения "сглаженного" значения уровня входного сигнала.

C#

```
private void UpdateAmp(  
    float value  
)
```

Параметры

value

Новое значение уровня входного сигнала.

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.UpdateAsymmetry - метод

Обработчик изменения "сглаженного" значения постоянного амплитудного сдвига.

C#

```
private void UpdateAsymmetry(  
    float value  
)
```

Параметры

value

Новое значение постоянного амплитудного сдвига.

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionManager.UpdateMix - метод

Обработчик изменения "сглаженного" значения количества выходного и входного сигнала.

C#

```
private void UpdateMix(  
    float value  
)
```

Параметры

value

Новое значение.

См. также

Применяется: [DistortionManager](#)⁷⁸

DistortionMode - перечисление

Указывает тип эффекта дисторшн.

Константа	Значение	Описание
AbsClipping	1	
BitCrush	4	
CubicClipping	3	
None	0	
SampleRateReduction	5	
SoftClipping	2	

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [DistortionManager](#)⁷⁸

absClipper - поле

Объект класса AbsClipper.

C#

```
private readonly AbsClipper absClipper
```

См. также

Применяется: [DistortionManager](#)⁷⁸

amount - поле

Текущая сила эффекта дисторшн.

C#

```
private float amount
```

См. также

Применяется: [DistortionManager](#)⁷⁸

amp - поле

Текущий уровень громкости входного сигнала.

C#

```
private float amp
```

См. также

Применяется: [DistortionManager](#)⁷⁸

ampFilter - поле

Фильтр низких частот, используемый для сглаживания параметра уровня громкости входного сигнала.

C#

```
private ParameterFilter ampFilter
```

См. также

Применяется: [DistortionManager](#)⁷⁸

asymmetryFilter - поле

Фильтр низких частот, используемый для сглаживания параметра постоянного амплитудного сдвига.

C#

```
private ParameterFilter asymmetryFilter
```

См. также

Применяется: [DistortionManager](#)⁷⁸

bitCrusher - поле

Объект класса BitCrusher.

C#

```
private readonly BitCrusher bitCrusher
```

См. также

Применяется: [DistortionManager](#)⁷⁸

cubicClipper - поле

Объект класса CubicClipper.

C#

```
private readonly CubicClipper cubicClipper
```

См. также

Применяется: [DistortionManager](#)⁷⁸

currentDistortion - поле

Текущий объект эффекта дисторшн.

C#

```
private IDistortion currentDistortion
```

См. также

Применяется: [DistortionManager](#)⁷⁸

dcBlocker - поле

Фильтр высоких частот для устранения постоянного амплитудного смещения.

C#

```
private DCBlocker dcBlocker
```

См. также

Применяется: [DistortionManager](#)⁷⁸

dcOffset - поле

Текущее значение постоянного амплитудного смещения.

C#

```
private float dcOffset
```

См. также

Применяется: [DistortionManager](#)⁷⁸

dryCoeff - поле

Текущий коэффициент "чистого" сигнала без эффекта.

C#

```
private float dryCoeff
```

См. также

Применяется: [DistortionManager](#)⁷⁸

lowPass - поле

Фильтр низких частот, применяемый к входному сигналу.

C#

```
private SvfFilter lowPass
```

См. также

Применяется: [DistortionManager](#)⁷⁸

mixFilter - поле

Фильтр низких частот, используемый для сглаживания параметра количества входного и выходного сигналов.

C#

```
private ParameterFilter mixFilter
```

См. также

Применяется: [DistortionManager](#)⁷⁸

mode - поле

Текущий тип эффекта дисторшн.

C#

```
private DistortionMode mode
```

См. также

Применяется: [DistortionManager](#)⁷⁸

sampleRateReducer - поле

Объект класса SampleRateReducer.

C#

```
private readonly SampleRateReducer sampleRateReducer
```

См. также

Применяется: [DistortionManager](#)⁷⁸

softClipper - поле

Объект класса SoftClipper.

C#

```
private readonly SoftClipper softClipper
```

См. также

Применяется: [DistortionManager](#)⁷⁸

wetCoeff - поле

Текущий коэффициент сигнала с применённым к нему эффектом.

C#

```
private float wetCoeff
```

См. также

Применяется: [DistortionManager](#)⁷⁸

Downsampler - класс

Фильтр низких частот, используемый для понижения частоты дискретизации.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.Downsampler](#)

C#

```
internal class Downsample : AudioComponent
```

Требования

Пространство имен:[MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

[Order](#)⁹⁰, [SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)³⁴ (унаследован от [AudioComponent](#)), [Process](#)⁹⁰, [ToString](#) (унаследован от [Object](#))

Поля

[X2OversamplingCoefficients](#)⁹¹, [X4OversamplingCoefficients](#)⁹¹, [X8OversamplingCoefficients](#)⁹¹, [buffer](#)⁹¹, [coefficients](#)⁹², [currBufferIndex](#)⁹², [order](#)⁹²

Downsample.Order - свойство

Отношение частоты дискретизации выходного сигнала к входному.

C#

```
public int Order {get; set;}
```

См. также

Применяется: [Downsample](#)⁸⁹

Downsample.Process - метод

Обработка новых входных данных.

C#

```
public double Process(  
    params double[] inputs  
)
```

Параметры

inputs

Массив, содержащий как минимум [Order](#) сэмплов.

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [Downsampler](#)⁸⁹

X2OversamplingCoefficients - поле

Коэффициенты, используемые для понижения частоты дискретизации в 2 раза.

C#

```
public static double[] X2oversamplingCoefficients
```

См. также

Применяется: [Downsampler](#)⁸⁹

X4OversamplingCoefficients - поле

Коэффициенты, используемые для понижения частоты дискретизации в 4 раза.

C#

```
public static double[] X4oversamplingCoefficients
```

См. также

Применяется: [Downsampler](#)⁸⁹

X8OversamplingCoefficients - поле

Коэффициенты, используемые для понижения частоты дискретизации в 8 раз.

C#

```
public static readonly double[] X80versamplingCoefficients
```

См. также

Применяется: [Downsampler](#)⁸⁹

buffer - поле

Циклический буфер, хранящий последние n входных сэмплов.

C#

```
private double[] buffer
```

См. также

Применяется: [Downsampler](#)₈₉

coefficients - поле

Текущие коэффициенты фильтра.

C#

```
private double[] coefficients
```

См. также

Применяется: [Downsampler](#)₈₉

currBufferIndex - поле

Указывает на индекс для записи в циклический буфер.

C#

```
private int currBufferIndex
```

См. также

Применяется: [Downsampler](#)₈₉

order - поле

Отношение частоты дискретизации выходного сигнала к входному.

C#

```
private int order
```

См. также

Применяется: [Downsampler](#)₈₉

Envelope - класс

Компонент голоса плагина, представляющий ADSR-огибающую.

[System.Object](#)

[MultimodSynth.AudioComponent](#)₃₃

[MultimodSynth.Envelope](#)

C#

```
internal class Envelope : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Envelope](#)⁹³

Свойства

[IsActive](#)⁹³, [SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)⁹⁴,
[Process](#)⁹⁴, [SetAmplitude](#)⁹⁴, [SetAttackCurve](#)⁹⁵, [SetAttackTime](#)⁹⁵, [SetDecayReleaseCurve](#)⁹⁵,
[SetDecayTime](#)⁹⁶, [SetReleaseTime](#)⁹⁶, [SetSustainLevel](#)⁹⁶, [ToString](#) (унаследован от [Object](#)), [TriggerAttack](#)⁹⁷,
[TriggerRelease](#)⁹⁷

Поля

[amplitude](#)⁹⁷, [attackCurve](#)⁹⁷, [attackTargetRatio](#)⁹⁸, [attackTime](#)⁹⁸, [decayReleaseCurve](#)⁹⁸,
[decayReleaseTargetRatio](#)⁹⁸, [decayTime](#)⁹⁹, [envelope](#)⁹⁹, [releaseTime](#)⁹⁹, [sustainLevel](#)⁹⁹

Envelope - конструктор

Инициализирует новый объект класса Envelope.

C#

```
public Envelope()
```

См. также

Применяется: [Envelope](#)⁹²

Envelope.IsActive - свойство

Показывает, активна ли огибающая в данный момент времени.

C#

```
public bool IsActive {get;}
```

См. также

Применяется: [Envelope](#)⁹²

Envelope.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [Envelope](#)₉₂

Envelope.Process - метод

Обработка новых входных данных.

C#

```
public float Process()
```

Возвращаемые значения

Новое значение "высоты" огибающей

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetAmplitude - метод

Устанавливает новое значение амплитуды огибающей.

C#

```
public void SetAmplitude(  
    float value  
)
```

Параметры

value

Амплитуда (в диапазоне [0, 1]).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetAttackCurve - метод

Устанавливает новое значение изгиба огибающей в стадии атаки.

C#

```
public void SetAttackCurve(  
    float value  
)
```

Параметры

value

Изгиб (в диапазоне [0, 1]).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetAttackTime - метод

Устанавливает новое значение длительности стадии атаки (в секундах).

C#

```
public void SetAttackTime(  
    float value  
)
```

Параметры

value

Время (в секундах).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetDecayReleaseCurve - метод

Устанавливает новое значение изгиба огибающей в стадиях спада и затухания.

C#

```
public void SetDecayReleaseCurve(  
    float value  
)
```

Параметры

value

Изгиб (в диапазоне [0, 1]).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetDecayTime - метод

Устанавливает новое значение длительности стадии спада (в секундах).

C#

```
public void SetDecayTime(  
    float value  
)
```

Параметры

value

Время (в секундах).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetReleaseTime - метод

Устанавливает новое значение длительности стадии затухания (в секундах).

C#

```
public void SetReleaseTime(  
    float value  
)
```

Параметры

value

Время (в секундах).

См. также

Применяется: [Envelope](#)₉₂

Envelope.SetSustainLevel - метод

Устанавливает новое значение уровня поддержки.

C#

```
public void SetSustainLevel(  
    float value  
)
```

Параметры

value

Уровень (в диапазоне [0, 1]).

См. также

Применяется: [Envelope](#)₉₂

Envelope.TriggerAttack - метод

Начинает стадию атаки.

C#

```
public void TriggerAttack()
```

См. также

Применяется: [Envelope](#)₉₂

Envelope.TriggerRelease - метод

Начинает стадию затухания.

C#

```
public void TriggerRelease()
```

См. также

Применяется: [Envelope](#)₉₂

amplitude - поле

Максимальное значение уровня огибающей.

C#

```
private float amplitude
```

См. также

Применяется: [Envelope](#)₉₂

attackCurve - поле

Текущее значение изгиба огибающей в стадии атаки (в диапазоне [0, 1]).

C#

```
private float attackCurve
```

См. также

Применяется: [Envelope](#)₉₂

attackTargetRatio - поле

Значение, показывающее форму стадии атаки.

C#

```
private double attackTargetRatio
```

См. также

Применяется: [Envelope](#)₉₂

attackTime - поле

Текущее значение длительности стадии атаки (в секундах).

C#

```
private float attackTime
```

См. также

Применяется: [Envelope](#)₉₂

decayReleaseCurve - поле

Текущее значение изгиба огибающей в стадии спада и затухания (в диапазоне [0, 1]).

C#

```
private float decayReleaseCurve
```

См. также

Применяется: [Envelope](#)₉₂

decayReleaseTargetRatio - поле

Значение, показывающее форму стадии спада и затухания.

C#

```
private double decayReleaseTargetRatio
```

См. также

Применяется: [Envelope](#)₉₂

decayTime - поле

Текущее значение длительности стадии спада с 100% до 0% (в секундах).

C#

```
private float decayTime
```

См. также

Применяется: [Envelope](#)₉₂

envelope - поле

Объект огибающей.

C#

```
private AdsREnvelope envelope
```

См. также

Применяется: [Envelope](#)₉₂

releaseTime - поле

Текущее значение длительности стадии затухания с 100% до 0% (в секундах).

C#

```
private float releaseTime
```

См. также

Применяется: [Envelope](#)₉₂

sustainLevel - поле

Текущее значение уровня периода поддержки.

C#

```
private float sustainLevel
```

См. также

Применяется: [Envelope](#)₉₂

EnvelopesManager - класс

Компонент плагина, управляющий одной огибающей многих голосов.

System.Object

MultimodSynth.AudioComponent³³

MultimodSynth.AudioComponentWithParameters³⁵

MultimodSynth.EnvelopesManager

C#

```
internal class EnvelopesManager : AudioComponentWithParameters
```

Требования

Пространство имен: MultimodSynth²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

EnvelopesManager¹⁰⁰

Свойства

AttackCurveManager¹⁰¹, AttackTimeManager¹⁰¹, DecayReleaseCurveManager¹⁰¹, DecayTimeManager¹⁰¹, EnvelopeAmplitudeManager¹⁰², Plugin³⁶ (унаследован от AudioComponentWithParameters), ReleaseTimeManager¹⁰², SampleRate³⁴ (унаследован от AudioComponent), SustainLevelManager¹⁰²

Методы

CreateNewEnvelope¹⁰², Equals (унаследован от Object), Finalize (унаследован от Object), GetHashCode (унаследован от Object), GetType (унаследован от Object), InitializeParameters¹⁰³, MemberwiseClone (унаследован от Object), OnSampleRateChanged³⁴ (унаследован от AudioComponent), Process¹⁰³, SetAttackCurve¹⁰³, SetAttackTime¹⁰⁴, SetDecayReleaseCurve¹⁰⁴, SetDecayTime¹⁰⁴, SetReleaseTime¹⁰⁵, SetSustainLevel¹⁰⁵, ToString (унаследован от Object), UpdateEnvelopeAmplitude¹⁰⁵

Поля

attackCurve¹⁰⁶, attackTime¹⁰⁶, decayReleaseCurve¹⁰⁶, decayTime¹⁰⁶, envelopeAmplitude¹⁰⁷, envelopeAmplitudeFilter¹⁰⁷, envelopes¹⁰⁷, releaseTime¹⁰⁷, sustainLevel¹⁰⁸

EnvelopesManager - конструктор

Инициализирует новый объект класса EnvelopesManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public EnvelopesManager(  
    Plugin plugin,  
    string parameterPrefix = "E"  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.AttackCurveManager - свойство

Объект, управляющий параметром изгиба стадии атаки.

C#

```
public VstParameterManager AttackCurveManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.AttackTimeManager - свойство

Объект, управляющий параметром длительности стадии атаки.

C#

```
public VstParameterManager AttackTimeManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.DecayReleaseCurveManager - свойство

Объект, управляющий параметром изгиба стадий спада и затухания.

C#

```
public VstParameterManager DecayReleaseCurveManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.DecayTimeManager - свойство

Объект, управляющий параметром длительности стадии спада.

C#

```
public VstParameterManager DecayTimeManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.EnvelopeAmplitudeManager - свойство

Объект, управляющий параметром амплитуды огибающей.

C#

```
public VstParameterManager EnvelopeAmplitudeManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.ReleaseTimeManager - свойство

Объект, управляющий параметром длительности стадии затухания.

C#

```
public VstParameterManager ReleaseTimeManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SustainLevelManager - свойство

Объект, управляющий параметром уровня стадии поддержки.

C#

```
public VstParameterManager SustainLevelManager {get; private set;}
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.CreateNewEnvelope - метод

Возвращает новый объект огибающей, связанный с этим объектом.

C#

```
public Envelope CreateNewEnvelope()
```

Возвращаемые значения

Новая огибающая.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.Process - метод

Метод, выполняющий обновление всех сглаживающих фильтров.

C#

```
public void Process()
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetAttackCurve - метод

Обработчик изменения изгиба стадии атаки.

C#

```
private void SetAttackCurve(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. так же

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetAttackTime - метод

Обработчик изменения длительности стадии атаки.

C#

```
private void SetAttackTime(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. так же

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetDecayReleaseCurve - метод

Обработчик изменения изгиба стадии спада и затухания.

C#

```
private void SetDecayReleaseCurve(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. так же

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetDecayTime - метод

Обработчик изменения длительности стадии спада.

C#

```
private void SetDecayTime(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetReleaseTime - метод

Обработчик изменения длительности стадии затухания.

C#

```
private void SetReleaseTime(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.SetSustainLevel - метод

Обработчик изменения уровня стадии поддержки.

C#

```
private void SetSustainLevel(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

EnvelopesManager.UpdateEnvelopeAmplitude - метод

Обработчик изменения "сглаженного" значения амплитуды огибающей.

C#

```
private void UpdateEnvelopeAmplitude(  
    float value  
)
```

Параметры

value

Новое значение амплитуды огибающей.

См. также

Применяется: [EnvelopesManager](#)⁹⁹

attackCurve - поле

Текущее значение изгиба огибающей в стадии атаки.

C#

```
private float attackCurve
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

attackTime - поле

Текущая длительность стадии атаки огибающей (в секундах).

C#

```
private float attackTime
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

decayReleaseCurve - поле

Текущее значение изгиба огибающей в стадии спада и затухания.

C#

```
private float decayReleaseCurve
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

decayTime - поле

Текущая длительность стадии спада огибающей (в секундах).

C#

```
private float decayTime
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

envelopeAmplitude - поле

Текущее значение амплитуды огибающей.

C#

```
private float envelopeAmplitude
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

envelopeAmplitudeFilter - поле

Фильтр низких частот, используемый для сглаживания параметра амплитуды огибающей.

C#

```
private ParameterFilter envelopeAmplitudeFilter
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

envelopes - поле

Список огибающих, связанных с этим менеджером огибающих.

C#

```
private List<Envelope> envelopes
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

releaseTime - поле

Текущая длительность стадии затухания огибающей (в секундах).

C#

```
private float releaseTime
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

sustainLevel - поле

Текущее значение уровня стадии поддержкигибающей.

C#

```
private float sustainLevel
```

См. также

Применяется: [EnvelopesManager](#)⁹⁹

Filter - класс

Компонент голоса плагина, представляющий собой SVF-фильтр.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.Filter](#)

C#

```
internal class Filter : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Filter](#)¹⁰⁹

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[CalculateCutoff](#)¹⁰⁹, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹⁰⁹, [Process](#)¹⁰⁹, [Reset](#)¹¹⁰, [SetCurve](#)¹¹⁰, [SetCutoffMultiplier](#)¹¹⁰, [SetFilterType](#)¹¹¹, [SetNoteFrequency](#)¹¹¹, [SetTrackingCoeff](#)¹¹¹, [ToString](#) (унаследован от [Object](#))

Поля

[BaseCutoff](#)¹¹³, [FilterModulationLookup](#)¹¹², [cutoff](#)¹¹², [cutoffMultiplier](#)¹¹², [filter](#)¹¹³, [noteFrequency](#)¹¹³, [trackingCoeff](#)¹¹³

Filter - конструктор

C#

```
private static Filter()
```

См. также

Применяется: [Filter](#)₁₀₈

Filter.CalculateCutoff - метод

Обновляет текущую частоту среза.

C#

```
private void CalculateCutoff()
```

См. также

Применяется: [Filter](#)₁₀₈

Filter.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [Filter](#)₁₀₈

Filter.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input,  
    float cutoffModulation = 0  
)
```

Параметры*input*

Входной сигнал.

cutoffModulation

Модуляция частоты среза фильтра.

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [Filter₁₀₈](#)

Filter.Reset - метод

Сбрасывает текущее состояние фильтра.

C#

```
public void Reset()
```

См. также

Применяется: [Filter₁₀₈](#)

Filter.SetCurve - метод

Устанавливает новое значение "ширины" фильтра.

C#

```
public void SetCurve(  
    float value  
)
```

Параметры*value***См. также**

Применяется: [Filter₁₀₈](#)

Filter.SetCutoffMultiplier - метод

Устанавливает новое значение множителя частоты среза.

C#

```
public void SetCutoffMultiplier(  
    float value
```

```
)
```

Параметры

value

См. также

Применяется: [Filter](#)₁₀₈

Filter.SetFilterType - метод

Устанавливает новое значение типа фильтра.

C#

```
public void SetFilterType(  
    FilterType value  
)
```

Параметры

value

См. также

Применяется: [Filter](#)₁₀₈

Filter.SetNoteFrequency - метод

Устанавливает новое значение частоты играемой ноты.

C#

```
public void SetNoteFrequency(  
    float value  
)
```

Параметры

value

См. также

Применяется: [Filter](#)₁₀₈

Filter.SetTrackingCoeff - метод

Устанавливает новое значение коэффициента отслеживания частоты играемой ноты.

C#

```
public void SetTrackingCoeff(  
    float value  
)
```

Параметры

value

См. также

Применяется: [Filter₁₀₈](#)

FilterModulationLookup - поле

Массив заранее сгенерированных коэффициентов. Используется для ускорения вычислений.

C#

```
private static float[] FilterModulationLookup
```

См. также

Применяется: [Filter₁₀₈](#)

cutoff - поле

Текущая частота среза фильтра.

C#

```
private float cutoff
```

См. также

Применяется: [Filter₁₀₈](#)

cutoffMultiplier - поле

Множитель частоты срезы фильтра.

C#

```
private float cutoffMultiplier
```

См. также

Применяется: [Filter₁₀₈](#)

filter - поле

Объект фильтра.

C#

```
private SvfFilter filter
```

См. также

Применяется: [Filter₁₀₈](#)

noteFrequency - поле

Частота играемой ноты.

C#

```
private float noteFrequency
```

См. также

Применяется: [Filter₁₀₈](#)

trackingCoeff - поле

Коэффициент отслеживания частоты играемой ноты.

C#

```
private float trackingCoeff
```

См. также

Применяется: [Filter₁₀₈](#)

BaseCutoff - поле

Базовая частота среза фильтра.

C#

```
private const float BaseCutoff = 65,41
```

См. также

Применяется: [Filter₁₀₈](#)

FiltersManager - класс

Компонент плагина, управляющий одним фильтром многих голосов.

System.Object

MultimodSynth.AudioComponent³³

MultimodSynth.AudioComponentWithParameters³⁵

MultimodSynth.FiltersManager

C#

```
internal class FiltersManager : AudioComponentWithParameters
```

Требования

Пространство имен: MultimodSynth²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

*FiltersManager*¹¹⁴

Свойства

CurveManager¹¹⁵, CutoffManager¹¹⁵, FilterTypeManager¹¹⁵, Plugin³⁶ (унаследован от AudioComponentWithParameters), SampleRate³⁴ (унаследован от AudioComponent), TrackingCoeffManager¹¹⁵

Методы

CreateNewFilter¹¹⁶, Equals (унаследован от Object), Finalize (унаследован от Object), GetHashCode (унаследован от Object), GetType (унаследован от Object), InitializeParameters¹¹⁶, MemberwiseClone (унаследован от Object), OnSampleRateChanged³⁴ (унаследован от AudioComponent), Process¹¹⁶, SetCurve¹¹⁶, SetCutoffMultiplierTarget¹¹⁷, SetFilterType¹¹⁷, SetTrackingCoeff¹¹⁸, ToString (унаследован от Object), UpdateCutoffMultiplier¹¹⁸

Поля

curve¹¹⁸, cutoffMultiplier¹¹⁸, cutoffMultiplierFilter¹¹⁹, filterType¹¹⁹, filters¹¹⁹, trackingCoeff¹¹⁹

FiltersManager - конструктор

Инициализирует новый объект класса FiltersManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public FiltersManager(  
    Plugin plugin,  
    string parameterPrefix  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.CurveManager - свойство

Объект, управляющий параметром "ширины" фильтра.

C#

```
public VstParameterManager CurveManager {get; private set;}
```

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.CutoffManager - свойство

Объект, управляющий параметром множителя частоты среза фильтра.

C#

```
public VstParameterManager CutoffManager {get; private set;}
```

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.FilterTypeManager - свойство

Объект, управляющий параметром типа фильтра.

C#

```
public VstParameterManager FilterTypeManager {get; private set;}
```

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.TrackingCoeffManager - свойство

Объект, управляющий параметром коэффициента отслеживания частоты играющей ноты.

C#

```
public VstParameterManager TrackingCoeffManager {get; private set;}
```

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.CreateNewFilter - метод

Возвращает новый объект фильтра, связанный с этим объектом.

C#

```
public Filter CreateNewFilter()
```

Возвращаемые значения

Новый фильтр.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.Process - метод

Метод, выполняющий обновление всех сглаживающих фильтров.

C#

```
public void Process()
```

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.SetCurve - метод

Обработчик изменения "ширины" фильтра.

C#

```
private void SetCurve(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.SetCutoffMultiplierTarget - метод

Обработчик изменения множителя частоты среза фильтра.

C#

```
private void SetCutoffMultiplierTarget(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.SetFilterType - метод

Обработчик изменения длительности типа фильтра.

C#

```
private void SetFilterType(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.SetTrackingCoeff - метод

Обработчик изменения коэффициента отслеживания частоты играющей ноты.

C#

```
private void SetTrackingCoeff(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [FiltersManager](#)₁₁₃

FiltersManager.UpdateCutoffMultiplier - метод

Обработчик изменения "сглаженного" множителя частоты среза фильтра.

C#

```
private void UpdateCutoffMultiplier(  
    float value  
)
```

Параметры

value

Новое значение множителя частоты среза фильтра.

См. также

Применяется: [FiltersManager](#)₁₁₃

curve - поле

Текущая "ширина" фильтра.

C#

```
private float curve
```

См. также

Применяется: [FiltersManager](#)₁₁₃

cutoffMultiplier - поле

Текущий множитель частоты среза фильтра.

C#

```
private float cutoffMultiplier
```

См. также

Применяется: [FiltersManager](#)₁₁₃

cutoffMultiplierFilter - поле

Фильтр низких частот, используемый для сглаживания параметра множителя частоты среза фильтра.

C#

```
private ParameterFilter cutoffMultiplierFilter
```

См. также

Применяется: [FiltersManager](#)₁₁₃

filterType - поле

Текущий тип фильтра.

C#

```
private FilterType filterType
```

См. также

Применяется: [FiltersManager](#)₁₁₃

filters - поле

Список фильтров, связанных с этим менеджером фильтров.

C#

```
private List<Filter> filters
```

См. также

Применяется: [FiltersManager](#)₁₁₃

trackingCoeff - поле

Текущий коэффициент отслеживания частоты играющей ноты.

C#

```
private float trackingCoeff
```

См. также

Применяется: [FiltersManager](#)₁₁₃

MidiNoteEventArgs - класс

Аргумент событий MidiProcessor.NoteOn и MidiProcessor.NoteOff.

[System.Object](#)

[System.EventArgs](#)

MultimodSynth.MidiNoteEventArgs

C#

```
public class MidiNoteEventArgs : EventArgs
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[MidiNoteEventArgs](#)₁₂₀

Свойства

[Note](#)₁₂₁, [PressedNotesCount](#)₁₂₁

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),

[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [ToString](#) (унаследован от [Object](#))

MidiNoteEventArgs - конструктор

Инициализирует новый объект типа MidiNoteEventArgs с заданными параметрами.

C#

```
public MidiNoteEventArgs(  
    byte noteNo,  
    byte velocity,  
    int pressedNotesCount  
)
```

Параметры

noteNo

velocity

pressedNotesCount

См. также

Применяется: [MidiNoteEventArgs](#)¹²⁰

MidiNoteEventArgs.Note - свойство

Играемая нота.

C#

```
public MidiNote Note {get; protected set;}
```

См. также

Применяется: [MidiNoteEventArgs](#)¹²⁰

MidiNoteEventArgs.PressedNotesCount - свойство

Количество нажатых нот.

C#

```
public int PressedNotesCount {get; protected set;}
```

См. также

Применяется: [MidiNoteEventArgs](#)¹²⁰

MidiProcessor - класс

Реализация интерфейса IVstMidiProcessor.

[System.Object](#)

[MultimodSynth.MidiProcessor](#)

C#

```
internal class MidiProcessor : IVstMidiProcessor
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[MidiProcessor](#)¹²²

Свойства

[ChannelCount](#)¹²²

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnNoteOff](#)¹²², [OnNoteOn](#)¹²³, [PressNoteFromUI](#)¹²³, [Process](#)¹²⁴, [ReleaseNoteFromUI](#)¹²⁴, [ToString](#) (унаследован от [Object](#))

События

[NoteOff](#)¹²⁴, [NoteOn](#)¹²⁵

Поля

[plugin](#)¹²⁵, [pressedNotesCount](#)¹²⁵

MidiProcessor - конструктор

Инициализирует новых объект типа MidiProcessor, принадлежащий заданному плагину.

C#

```
public MidiProcessor(  
    Plugin plugin  
)
```

Параметры

plugin

См. также

Применяется: [MidiProcessor](#)¹²¹

MidiProcessor.ChannelCount - свойство

Количество каналов.

C#

```
public int ChannelCount {get;}
```

См. также

Применяется: [MidiProcessor](#)¹²¹

MidiProcessor.OnNoteOff - метод

Метод, вызывающий событие NoteOff с заданными параметрами.

C#

```
private void OnNoteOff(  
    byte noteNo,  
    byte velocity  
)
```

Параметры

noteNo

velocity

См. также

Применяется: [MidiProcessor](#)₁₂₁

MidiProcessor.OnNoteOn - метод

Метод, вызывающий событие NoteOn с заданными параметрами.

C#

```
private void OnNoteOn(  
    byte noteNo,  
    byte velocity  
)
```

Параметры

noteNo

velocity

См. также

Применяется: [MidiProcessor](#)₁₂₁

MidiProcessor.PressNoteFromUI - метод

Метод, "нажимающий" клавишу из UI.

C#

```
public void PressNoteFromUI(  
    byte noteNo,  
    byte velocity  
)
```

Параметры

noteNo

velocity

См. также

Применяется: [MidiProcessor](#)₁₂₁

MidiProcessor.Process - метод

Обработка новых входных данных.

C#

```
public void Process(  
    VstEventCollection events  
)
```

Параметры

events

См. также

Применяется: [MidiProcessor](#)₁₂₁

MidiProcessor.ReleaseNoteFromUI - метод

Метод, "отпускающий" клавишу из UI.

C#

```
public void ReleaseNoteFromUI(  
    byte noteNo,  
    byte velocity  
)
```

Параметры

noteNo

velocity

См. также

Применяется: [MidiProcessor](#)₁₂₁

NoteOff - событие

Событие, возникающее при отпускании клавиши.

C#

```
public event EventHandler<MidiNoteEventArgs> NoteOff
```

См. также

Применяется: [MidiProcessor](#)₁₂₁

NoteOn - событие

Событие, возникающее при нажатии клавиши.

C#

```
public event EventHandler<MidiNoteEventArgs> NoteOn
```

См. также

Применяется: [MidiProcessor](#)₁₂₁

plugin - поле

Ссылка на плагин, которому принадлежит этот компонент.

C#

```
private Plugin plugin
```

См. также

Применяется: [MidiProcessor](#)₁₂₁

pressedNotesCount - поле

Текущее количество нажатых клавиш.

C#

```
private int pressedNotesCount
```

См. также

Применяется: [MidiProcessor](#)₁₂₁

Oscillator - класс

Компонент голоса плагина, представляющий осциллятор.

[System.Object](#)

[MultimodSynth.AudioComponent](#)₃₃

MultimodSynth.Oscillator**C#**

```
internal class Oscillator : AudioComponent
```

Требования

Пространство имен:[MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹²⁶,
[Process](#)¹²⁶, [Reset](#)¹²⁷, [SetNoteFrequency](#)¹²⁷, [SetPitchMultiplier](#)¹²⁷, [SetWaveTable](#)¹²⁸, [ToString](#)
(унаследован от [Object](#)), [UpdateCoefficients](#)¹²⁸

Поля

[frequency](#)¹²⁸, [noteFrequency](#)¹²⁹, [phaseIncrement](#)¹²⁹, [phasor](#)¹²⁹, [pitchMultiplier](#)¹²⁹, [waveTable](#)¹²⁹

Oscillator.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [Oscillator](#)¹²⁵

Oscillator.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float phaseModulation = 0  
)
```

Параметры*phaseModulation*

Фазовая модуляция.

Возвращаемые значения

Выходной сигнал.

См. такжеПрименяется: [Oscillator](#)₁₂₅**Oscillator.Reset - метод**

Сбрасывает текущее состояние осциллятора.

C#

```
public void Reset()
```

См. такжеПрименяется: [Oscillator](#)₁₂₅**Oscillator.SetNoteFrequency - метод**

Устанавливает новое значение частоты играемой ноты.

C#

```
public void SetNoteFrequency(  
    float value  
)
```

Параметры*value***См. также**Применяется: [Oscillator](#)₁₂₅**Oscillator.SetPitchMultiplier - метод**

Устанавливает новое значение множителя частоты.

C#

```
public void SetPitchMultiplier(  
    float value  
)
```

Параметры*value***См. так же**Применяется: [Oscillator](#)₁₂₅**Oscillator.SetWaveTable - метод**

Устанавливает новый объект класса WaveTable.

C#

```
public void SetWaveTable(  
    WaveTableOscillator waveTable  
)
```

Параметры*waveTable***См. так же**Применяется: [Oscillator](#)₁₂₅**Oscillator.UpdateCoefficients - метод**

Обновляет все коэффициенты.

C#

```
private void UpdateCoefficients()
```

См. так жеПрименяется: [Oscillator](#)₁₂₅**frequency - поле**

Играемая частота.

C#

```
private float frequency
```

См. так жеПрименяется: [Oscillator](#)₁₂₅

noteFrequency - поле

Частота играемой ноты.

C#

```
private float noteFrequency
```

См. также

Применяется: [Oscillator](#)₁₂₅

phaseIncrement - поле

Изменение фазы за 1 сэмпл.

C#

```
private float phaseIncrement
```

См. также

Применяется: [Oscillator](#)₁₂₅

phasor - поле

Текущая фаза осциллятора.

C#

```
private float phasor
```

См. также

Применяется: [Oscillator](#)₁₂₅

pitchMultiplier - поле

Множитель частоты.

C#

```
private float pitchMultiplier
```

См. также

Применяется: [Oscillator](#)₁₂₅

waveTable - поле

Ссылка на используемый объект класса WaveTable.

C#

```
private WaveTableOscillator waveTable
```

См. также

Применяется: [Oscillator](#)¹²⁵

OscillatorsManager - класс

Компонент плагина, управляющий одним осциллятором многих голосов.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.AudioComponentWithParameters](#)³⁵

MultimodSynth.OscillatorsManager

C#

```
internal class OscillatorsManager : AudioComponentWithParameters
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[OscillatorsManager](#)¹³⁰

Свойства

[PitchFineManager](#)¹³¹, [PitchSemiManager](#)¹³¹, [Plugin](#)³⁶ (унаследован от [AudioComponentWithParameters](#)),
[SampleRate](#)³⁴ (унаследован от [AudioComponent](#)), [WaveTableManager](#)¹³¹

Методы

[CreateNewOscillator](#)¹³², [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)¹³², [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)³⁴ (унаследован от [AudioComponent](#)), [Process](#)¹³²,
[SetPitchFine](#)¹³², [SetPitchSemi](#)¹³³, [SetWaveTable](#)¹³³, [ToString](#) (унаследован от [Object](#)),
[UpdatePitchMultiplier](#)¹³⁴

Поля

[oscillators](#)¹³⁴, [pitchFine](#)¹³⁴, [pitchMultiplier](#)¹³⁴, [pitchMultiplierFilter](#)¹³⁵, [pitchSemi](#)¹³⁵, [waveTable](#)¹³⁵

OscillatorsManager - конструктор

Инициализирует новый объект класса OscillatorsManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#**130**

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

```
public OscillatorsManager(  
    Plugin plugin,  
    string parameterPrefix  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.PitchFineManager - свойство

Объект, управляющий параметром подстройки частоты (в центах).

C#

```
public VstParameterManager PitchFineManager {get; private set;}
```

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.PitchSemiManager - свойство

Объект, управляющий параметром подстройки частоты (в полутонах).

C#

```
public VstParameterManager PitchSemiManager {get; private set;}
```

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.WaveTableManager - свойство

Объект, управляющий параметром текущей используемой таблицы семплов.

C#

```
public VstParameterManager WaveTableManager {get; private set;}
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

OscillatorsManager.CreateNewOscillator - метод

Возвращает новый объект осциллятора, связанный с этим объектом.

C#

```
public Oscillator CreateNewOscillator()
```

Возвращаемые значения

Новая огибающая.

См. также

Применяется: [OscillatorsManager](#)₁₃₀

OscillatorsManager.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(  
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [OscillatorsManager](#)₁₃₀

OscillatorsManager.Process - метод

Метод, выполняющий обновление всех сглаживающих фильтров.

C#

```
public void Process()
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

OscillatorsManager.SetPitchFine - метод

Обработчик изменения подстройки частоты (в центах).

C#

```
private void SetPitchFine(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.SetPitchSemi - метод

Обработчик изменения подстройки частоты в полутонах.

C#

```
private void SetPitchSemi(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.SetWaveTable - метод

Обработчик изменения текущей таблицы сэмплов.

C#

```
private void SetWaveTable(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [OscillatorsManager](#)¹³⁰

OscillatorsManager.UpdatePitchMultiplier - метод

Обработчик изменения "сглаженного" значения общей подстройки частоты.

C#

```
private void UpdatePitchMultiplier(  
    float value  
)
```

Параметры

value

Новое значение подстройки частоты.

См. также

Применяется: [OscillatorsManager](#)₁₃₀

oscillators - поле

Список осцилляторов, связанных с этим менеджером осцилляторов.

C#

```
private List<Oscillator> oscillators
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

pitchFine - поле

Коэффициент высоты играемой ноты (изменяется параметром подстройки в полутонах).

C#

```
private float pitchFine
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

pitchMultiplier - поле

Общий коэффициент высоты играемой ноты.

C#

```
private float pitchMultiplier
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

pitchMultiplierFilter - поле

Фильтр низких частот, используемый для сглаживания изменений частоты играемой ноты.

C#

```
private ParameterFilter pitchMultiplierFilter
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

pitchSemi - поле

Другой коэффициент высоты играемой ноты (изменяется параметром подстройки в центах).

C#

```
private float pitchSemi
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

waveTable - поле

Ссылка на текущую используемую таблицу сэмплов.

C#

```
private WaveTableOscillator waveTable
```

См. также

Применяется: [OscillatorsManager](#)₁₃₀

ParameterFactory - класс

Представляет собой фабрику для удобного создания параметров.

[System.Object](#)

[MultimodSynth.ParameterFactory](#)

C#

```
internal class ParameterFactory
```

Требования

Пространство имен: MultimodSynth²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[ParameterFactory](#)¹³⁶

Методы

[CreateParameterManager](#)¹³⁶, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [ToString](#) (унаследован от [Object](#))

Поля

[namePrefix](#)¹³⁷, [parameterCategory](#)¹³⁷, [plugin](#)¹³⁷, [programs](#)¹³⁸

ParameterFactory - конструктор

Инициализирует новый объект типа ParameterFactory, принадлежащий заданному плагину и имеющий переданную категорию и префикс имени параметра.

C#

```
public ParameterFactory(
    Plugin plugin,
    string category = "",
    string namePrefix = ""
)
```

Параметры

plugin

category

namePrefix

См. также

Применяется: [ParameterFactory](#)¹³⁵

ParameterFactory.CreateParameterManager - метод

Создаёт новый параметр и регистрирует его.

C#

```
public VstParameterManager CreateParameterManager(
    string name = null,
    float defaultValue = 0,
    Action<float> valueChangedHandler = null
)
```

Параметры

name

Имя параметра.

defaultValue

Начальное значение.

valueChangedHandler

Обработчик изменения значения параметра.

Возвращаемые значения

Объект, управляющий созданным параметром.

См. также

Применяется: [ParameterFactory](#)¹³⁵

namePrefix - поле

Префикс названия параметра.

C#

```
private string namePrefix
```

См. также

Применяется: [ParameterFactory](#)¹³⁵

parameterCategory - поле

Категория параметров.

C#

```
private VstParameterCategory parameterCategory
```

См. также

Применяется: [ParameterFactory](#)¹³⁵

plugin - поле

Ссылка на плагин, которому принадлежит этот компонент.

C#

```
private Plugin plugin
```

См. также

Применяется: [ParameterFactory](#)¹³⁵

programs - поле

Ссылка на компонент PluginPrograms данного плагина.

C#

```
private PluginPrograms programs
```

См. также

Применяется: [ParameterFactory](#)¹³⁵

ParameterFilter - класс

Фильтр низких частот, используемый для сглаживания изменения параметров.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.ParameterFilter](#)

C#

```
internal class ParameterFilter : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[ParameterFilter](#)¹³⁹

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹³⁹,
[Process](#)¹³⁹, [SetResponseTimeCoefficient](#)¹⁴⁰, [SetTarget](#)¹⁴⁰, [ToString](#) (унаследован от [Object](#))

Поля

[BaseSampleRate](#)¹⁴², [DefaultExp](#)¹⁴², [a](#)¹⁴⁰, [b](#)¹⁴¹, [isActive](#)¹⁴¹, [responseTimeCoefficient](#)¹⁴¹, [target](#)¹⁴¹,
[value](#)¹⁴¹, [valueChangedAction](#)¹⁴²

См. также

Ошибка! Недопустимый объект гиперссылки.

ParameterFilter - конструктор

Инициализирует новый объект класса ParameterFilter, выполняющий заданное действие при каждом изменении сглаженного значения.

C#

```
public ParameterFilter(  
    Action<float> valueChangedAction,  
    float initialValue = 0,  
    float responseTimeCoefficient = 1  
)
```

Параметры

valueChangedAction

Действие, выполняемое при изменении сглаженного значения.

initialValue

Начальное значение фильтра.

responseTimeCoefficient

Коэффициент скорости отклика фильтра

См. также

Применяется: [ParameterFilter](#)¹³⁸

ParameterFilter.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [ParameterFilter](#)¹³⁸

ParameterFilter.Process - метод

Выполнение очередного шага сглаживания.

C#

```
public void Process()
```

См. также

Применяется: [ParameterFilter](#)₁₃₈

ParameterFilter.SetResponseTimeCoefficient - метод

Устанавливает коэффициент времени отклика фильтра.

C#

```
public void SetResponseTimeCoefficient(  
    float value  
)
```

Параметры

value

См. также

Применяется: [ParameterFilter](#)₁₃₈

ParameterFilter.SetTarget - метод

Устанавливает целевое значение фильтра.

C#

```
public void SetTarget(  
    float value  
)
```

Параметры

value

См. также

Применяется: [ParameterFilter](#)₁₃₈

а - поле

Коэффициенты фильтра.

C#

```
private float a
```

См. также

Применяется: [ParameterFilter](#)₁₃₈

b - поле

Коэффициенты фильтра.

C#

```
private float b
```

См. также

Применяется: [ParameterFilter₁₃₈](#)

isActive - поле

Показывает, активен ли фильтр в данный момент.

C#

```
private bool isActive
```

См. также

Применяется: [ParameterFilter₁₃₈](#)

responseTimeCoefficient - поле

Коэффициент скорости отклика.

C#

```
private float responseTimeCoefficient
```

См. также

Применяется: [ParameterFilter₁₃₈](#)

target - поле

Целевое значение фильтра.

C#

```
private float target
```

См. также

Применяется: [ParameterFilter₁₃₈](#)

value - поле

Текущее слаженное значение.

C#

```
private float value
```

См. также

Применяется: [ParameterFilter](#)¹³⁸

valueChangedAction - поле

Действие, выполняемое при изменении текущего значения.

C#

```
private Action<float> valueChangedAction
```

См. также

Применяется: [ParameterFilter](#)¹³⁸

BaseSampleRate - поле

Частота дискретизации, для которой вычислен коэффициент DefaultExp.

C#

```
private const double BaseSampleRate = 44100
```

См. также

Применяется: [ParameterFilter](#)¹³⁸

DefaultExp - поле

Коэффициент для генерации коэффициента скорости отклика фильтра.

C#

```
private const double DefaultExp = 0,00159956063081846
```

См. также

Применяется: [ParameterFilter](#)¹³⁸

PingPongDelay - класс

Представляет собой эффект пинг-понг дилэй.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

MultimodSynth.PingPongDelay

C#

```
internal class PingPongDelay : AudioComponent,  
    IDelay
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[PingPongDelay](#)¹⁴³

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹⁴³,
[Process](#)¹⁴⁴, [Reset](#)¹⁴⁴, [SetDelay](#)¹⁴⁵, [SetFeedback](#)¹⁴⁵, [SetStereo](#)¹⁴⁵, [ToString](#) (унаследован от [Object](#))

Поля

[delayL](#)¹⁴⁶, [delayR](#)¹⁴⁶, [feedback](#)¹⁴⁶, [leftCoeff](#)¹⁴⁶, [rightCoeff](#)¹⁴⁷

PingPongDelay - конструктор

Инициализирует новый объект типа PingPongDelay.

C#

```
public PingPongDelay()
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

PingPongDelay.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

143

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [PingPongDelay](#)¹⁴²

PingPongDelay.Process - метод

Обработка новых входных данных.

C#

```
public void Process(  
    float inputL,  
    float inputR,  
    out float outputL,  
    out float outputR  
)
```

Параметры

inputL

Левый канал входного сигнала

inputR

Правый канал входного сигнала

outputL

Левый канал выходного сигнала.

outputR

Правый канал выходного сигнала.

Реализации

[IDelay.Process](#)²¹³

См. также

Применяется: [PingPongDelay](#)¹⁴²

PingPongDelay.Reset - метод

Очищает буфер дилэя.

C#

```
public void Reset()
```

Реализации

[IDelay.Reset](#)²¹⁴

См. также

Применяется: [PingPongDelay](#)¹⁴²

PingPongDelay.SetDelay - метод

Устанавливает новое значение времени задержки (в сэмплах).

C#

```
public void SetDelay(  
    float delay  
)
```

Параметры

delay

Реализации

[IDelay.SetDelay](#)₂₁₄

См. также

Применяется: [PingPongDelay](#)₁₄₂

PingPongDelay.SetFeedback - метод

Устанавливает новое значение коэффициента обратной связи.

C#

```
public void SetFeedback(  
    float value  
)
```

Параметры

value

Коэффициент обратной связи.

Реализации

[IDelay.SetFeedback](#)₂₁₄

См. также

Применяется: [PingPongDelay](#)₁₄₂

PingPongDelay.SetStereo - метод

Устанавливает новое значение коэффициента стерео-эффекта.

C#

```
public void SetStereo(  
    float value  
)
```

Параметры

value

Коэффициент стерео-эффекта.

Реализации

[IDelay.SetStereo](#)²¹⁵

См. также

Применяется: [PingPongDelay](#)¹⁴²

delayL - поле

Линия задержки для левого канала.

C#

```
private DelayLine delayL
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

delayR - поле

Линия задержки для правого канала.

C#

```
private DelayLine delayR
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

feedback - поле

Коэффициент обратной связи.

C#

```
private float feedback
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

leftCoeff - поле

Коэффициент начальной громкости левого канала.

C#

```
private float leftCoeff
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

rightCoeff - поле

Коэффициент начальной громкости правого канала.

C#

```
private float rightCoeff
```

См. также

Применяется: [PingPongDelay](#)¹⁴²

Plugin - класс

Представляет собой плагин.

C#

```
internal class Plugin : VstPluginWithInterfaceManagerBase
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Plugin](#)¹⁴⁷

Свойства

[AudioProcessor](#)¹⁴⁸, [MidiProcessor](#)¹⁴⁸, [Programs](#)¹⁴⁸

Методы

[CreateAudioProcessor](#)¹⁴⁸, [CreateEditor](#)¹⁴⁹, [CreateMidiProcessor](#)¹⁴⁹, [CreatePrograms](#)¹⁴⁹

Plugin - конструктор

Инициализирует новый объект типа Plugin.

C#

```
public Plugin()
```

См. также

Применяется: [Plugin](#)₁₄₇

Plugin.AudioProcessor - свойство

Ссылка на компонент AudioProcessor плагина.

C#

```
public AudioProcessor AudioProcessor {get;}
```

См. также

Применяется: [Plugin](#)₁₄₇

Plugin.MidiProcessor - свойство

Ссылка на компонент MidiProcessor плагина.

C#

```
public MidiProcessor MidiProcessor {get;}
```

См. также

Применяется: [Plugin](#)₁₄₇

Plugin.Programs - свойство

Ссылка на компонент PluginPrograms плагина.

C#

```
public PluginPrograms Programs {get;}
```

См. также

Применяется: [Plugin](#)₁₄₇

Plugin.CreateAudioProcessor - метод

Метод, возвращающий реализацию интерфейса IVstPluginAudioProcessor.

C#

```
protected override IVstPluginAudioProcessor CreateAudioProcessor(  
    IVstPluginAudioProcessor instance  
)
```

Параметры*instance***См. также**Применяется: [Plugin](#)₁₄₇**Plugin.CreateEditor - метод**

Метод, возвращающий реализацию интерфейса IVstPluginEditor.

C#

```
protected override IVstPluginEditor CreateEditor(  
    IVstPluginEditor instance  
)
```

Параметры*instance***См. также**Применяется: [Plugin](#)₁₄₇**Plugin.CreateMidiProcessor - метод**

Метод, возвращающий реализацию интерфейса IVstNidiProcessor.

C#

```
protected override IVstMidiProcessor CreateMidiProcessor(  
    IVstMidiProcessor instance  
)
```

Параметры*instance***См. также**Применяется: [Plugin](#)₁₄₇**Plugin.CreatePrograms - метод**

Метод, возвращающий реализацию интерфейса IVstPluginPrograms.

C#

```
protected override IVstPluginPrograms CreatePrograms(
    IVstPluginPrograms instance
)
```

Параметры

instance

См. также

Применяется: [Plugin](#)₁₄₇

PluginCommandStub - класс

Публичный класс, являющийся точкой связывания managed и unmanaged кода.

C#

```
public class PluginCommandStub : StdPluginCommandStub
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[CreatePluginInstance](#)₁₅₀

PluginCommandStub.CreatePluginInstance - метод

Возвращает новый объект типа IVstPlugin.

C#

```
protected override IVstPlugin CreatePluginInstance()
```

См. также

Применяется: [PluginCommandStub](#)₁₅₀

PluginEditor - класс

Реализация интерфейса IVstPluginEditor, отвечающего за создание пользовательского интерфейса.

[System.Object](#)

[MultimodSynth.PluginEditor](#)

C#

```
internal class PluginEditor : IVstPluginEditor
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[PluginEditor](#)¹⁵¹

Свойства

[Bounds](#)¹⁵¹, [KnobMode](#)¹⁵²

Методы

[Close](#)¹⁵², [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetElementPixelSize](#)¹⁵²,
[GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [KeyDown](#)¹⁵², [KeyUp](#)¹⁵³,
[MemberwiseClone](#) (унаследован от [Object](#)), [Open](#)¹⁵³, [ProcessIdle](#)¹⁵⁴, [ToString](#) (унаследован от [Object](#))

Поля

[bounds](#)¹⁵⁴, [hwndSource](#)¹⁵⁴, [instance](#)¹⁵⁵, [plugin](#)¹⁵⁵

PluginEditor - конструктор

Инициализирует новый объект типа PluginEditor, принадлежащий переданному плагину.

C#

```
public PluginEditor(  
    Plugin plugin  
)
```

Параметры

plugin

См. также

Применяется: [PluginEditor](#)¹⁵⁰

PluginEditor.Bounds - свойство

Возвращает размеры текущего окна.

C#

```
public Rectangle Bounds {get;}
```

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.KnobMode - свойство

Тип переключателей редактора.

C#

```
public VstKnobMode KnobMode {get; set;}
```

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.Close - метод

Закрывает редактор.

C#

```
public void Close()
```

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.GetElementPixelSize - метод

Возвращает размер элемента в экранных пикселях.

C#

```
private Size GetElementPixelSize(  
    UIElement element  
)
```

Параметры

element

См. также

Применяется: [PluginEditor](#)₁₅₀

Ошибка! Недопустимый объект гиперссылки.

PluginEditor.KeyDown - метод

Обрабатывает нажатия клавиш (не используется).

C#

```
public bool KeyDown(  
    byte ascii,  
    VstVirtualKey virtualKey,  
    VstModifierKeys modifiers  
)
```

Параметры

ascii

virtualKey

modifiers

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.KeyUp - метод

Обрабатывает отпускания клавиш клавиатуры (не используется).

C#

```
public bool KeyUp(  
    byte ascii,  
    VstVirtualKey virtualKey,  
    VstModifierKeys modifiers  
)
```

Параметры

ascii

virtualKey

modifiers

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.Open - метод

Открывает и прикрепляет элемент управления к переданному hWnd.

C#

```
public void Open()
```

```
    IntPtr hWnd  
)
```

Параметры

hWnd

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginEditor.ProcessIdle - метод

Метод, вызываемый хостом, когда процессор бездействует (не используется).

C#

```
public void ProcessIdle()
```

См. также

Применяется: [PluginEditor](#)₁₅₀

bounds - поле

Текущие границы окна редактора.

C#

```
private Nullable<Rectangle> bounds
```

См. также

Применяется: [PluginEditor](#)₁₅₀

hwndSource - поле

Объект типа HwndSource.

C#

```
private HwndSource hwndSource
```

См. также

Применяется: [PluginEditor](#)₁₅₀

instance - поле

Ссылка на окно редактора.

C#

```
private EditorView instance
```

См. также

Применяется: [PluginEditor](#)₁₅₀

plugin - поле

Ссылка на плагин, которому принадлежит этот компонент.

C#

```
private Plugin plugin
```

См. также

Применяется: [PluginEditor](#)₁₅₀

PluginPrograms - класс

Реализация интерфейса IVstPluginPrograms, отвечающего за работу с программами.

C#

```
internal class PluginPrograms : VstPluginProgramsBase
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[PluginPrograms](#)₁₅₆

Свойства

[ParameterCategories](#)₁₅₆, [ParameterInfos](#)₁₅₆

Методы

[CreateProgramCollection](#)₁₅₆

Поля

[plugin](#)₁₅₇

PluginPrograms - конструктор

Инициализирует новый объект типа PluginPrograms, принадлежащий заданному плагину.

C#

```
public PluginPrograms(  
    Plugin plugin  
)
```

Параметры

plugin

См. также

Применяется: [PluginPrograms₁₅₅](#)

PluginPrograms.ParameterCategories - свойство

Ссылка на коллекцию категорий параметров.

C#

```
public VstParameterCategoryCollection ParameterCategories {get; protected set;}
```

См. также

Применяется: [PluginPrograms₁₅₅](#)

PluginPrograms.ParameterInfos - свойство

Ссылка на коллекцию данных о параметрах.

C#

```
public VstParameterInfoCollection ParameterInfos {get; protected set;}
```

См. также

Применяется: [PluginPrograms₁₅₅](#)

PluginPrograms.CreateProgramCollection - метод

Создает коллекцию программ.

C#

```
protected override VstProgramCollection CreateProgramCollection()
```

См. также

Применяется: [PluginPrograms](#)¹⁵⁵

plugin - поле

Ссылка на плагин, которому принадлежит этот компонент.

C#

```
private Plugin plugin
```

См. также

Применяется: [PluginPrograms](#)¹⁵⁵

Routing - класс

Компонент плагина, предстающий собой всю цепочку создания обработки звука.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.AudioComponentWithParameters](#)³⁵

MultimodSynth.Routing

C#

```
internal class Routing : AudioComponentWithParameters
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Routing](#)¹⁵⁸

Свойства

[DelayManager](#)¹⁵⁸, [DistortionManager](#)¹⁵⁸, [Downsampler](#)¹⁵⁸, [MasterVolumeManager](#)¹⁵⁹,

[OversamplingOrderManager](#)¹⁵⁹, [Plugin](#)¹⁵⁶ (унаследован от [AudioComponentWithParameters](#)), [SampleRate](#)³⁴

(унаследован от [AudioComponent](#)), [VoicesManager](#)¹⁵⁹

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),

[GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)¹⁵⁹, [MemberwiseClone](#) (унаследован от [Object](#)),

[MidiProcessor_NoteOff](#)¹⁶⁰, [MidiProcessor_NoteOn](#)¹⁶⁰, [OnSampleRateChanged](#)¹⁶¹, [Process](#)¹⁶¹,

[SetOversamplingOrder](#)¹⁶¹, [ToString](#) (унаследован от [Object](#)), [UpdateMasterVolume](#)¹⁶²,

[UpdateSampleRates](#)¹⁶²

Поля

[masterVolume](#)¹⁶², [masterVolumeFilter](#)¹⁶³, [samplesForOversampling](#)¹⁶³

Routing - конструктор

Инициализирует новый объект класса Routing, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public Routing(  
    Plugin plugin,  
    string parameterPrefix = "M_"  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.DelayManager - свойство

Эффект дилэй.

C#

```
public DelayManager DelayManager {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.DistortionManager - свойство

Эффект дисторшн.

C#

```
public DistortionManager DistortionManager {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.Downsampler - свойство

Фильтр для снижения частоты дискретизации.

C#

```
public Downampler Downampler {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.MasterVolumeManager - свойство

Объект, управляющий параметром уровня громкости выходного сигнала.

C#

```
public VstParameterManager MasterVolumeManager {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.OversamplingOrderManager - свойство

Объект, управляющий параметром коэффициента повышения частоты дискретизации.

C#

```
public VstParameterManager OversamplingOrderManager {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.VoicesManager - свойство

Менеджер всех голосов.

C#

```
public VoicesManager VoicesManager {get; private set;}
```

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.InitializeParameters - метод

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters()
```

```
    ParameterFactory factory  
)
```

Параметры

factory

Фабрика параметров

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.MidiProcessor_NoteOff - метод

Обработчик события отпускания клавиши.

C#

```
private void MidiProcessor_NoteOff(  
    object sender,  
    MidiNoteEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.MidiProcessor_NoteOn - метод

Обработчик события нажатия на клавишу.

C#

```
private void MidiProcessor_NoteOn(  
    object sender,  
    MidiNoteEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.Process - метод

Генерация новых выходных данных.

C#

```
public void Process(  
    out float left,  
    out float right  
)
```

Параметры

left

Левый канал выходного сигнала.

right

Правый канал выходного сигнала.

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.SetOversamplingOrder - метод

Обработчик изменения коэффициента повышения частоты дискретизации.

C#

```
private void SetOversamplingOrder(  
    float value
```

```
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.UpdateMasterVolume - метод

Обработчик изменения "сглаженного" значения уровня громкости выходного сигнала.

C#

```
private void UpdateMasterVolume(  
    float value  
)
```

Параметры

value

Новое значение уровня громкости.

См. также

Применяется: [Routing](#)¹⁵⁷

Routing.UpdateSampleRates - метод

Обновляет частоты дискретизации компонентов, подверженных оверсэмплингу.

C#

```
private void UpdateSampleRates()
```

См. также

Применяется: [Routing](#)¹⁵⁷

masterVolume - поле

Уровень громкости выходного сигнала.

C#

```
private float masterVolume
```

См. также

Применяется: [Routing](#)¹⁵⁷

masterVolumeFilter - поле

Фильтр низких частот, используемый для сглаживания параметра уровня громкости выходного сигнала.

C#

```
private ParameterFilter masterVolumeFilter
```

См. также

Применяется: [Routing](#)¹⁵⁷

samplesForOversampling - поле

Буфер для хранения последних n семплов (для оверсэмплинга).

C#

```
new private double[] samplesForOversampling
```

См. также

Применяется: [Routing](#)¹⁵⁷

SampleRateReductor - класс

Представляет собой эффект ухудшения качества звука путём уменьшения частоты дискретизации.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.SampleRateReductor](#)

C#

```
internal class SampleRateReductor : AudioComponent,  
IDistortion
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)₁₆₄, [Process](#)₁₆₄, [SetAmount](#)₁₆₅, [ToString](#) (унаследован от [Object](#)), [UpdateCoefficients](#)₁₆₅

Поля

[holdTime](#)₁₆₅, [phaseIncrement](#)₁₆₅, [phasor](#)₁₆₆, [sample](#)₁₆₆

SampleRateReductor.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [SampleRateReductor](#)₁₆₃

SampleRateReductor.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

Реализации

[IDistortion.Process](#)₂₁₅

См. также

Применяется: [SampleRateReductor](#)₁₆₃

SampleRateReducer.SetAmount - метод

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]). Чем больше сила, тем меньше частота дискретизации.

C#

```
public void SetAmount(  
    float value  
)
```

Параметры

value

Значение "силы" эффекта.

Реализации

[IDistortion.SetAmount](#)²¹⁶

См. также

Применяется: [SampleRateReducer](#)¹⁶³

SampleRateReducer.UpdateCoefficients - метод

Обновление инкремента фазы при изменении некоторых параметров.

C#

```
private void UpdateCoefficients()
```

См. также

Применяется: [SampleRateReducer](#)¹⁶³

holdTime - поле

Время, в течении которого проигрывается записанный сэмпл.

C#

```
private float holdTime
```

См. также

Применяется: [SampleRateReducer](#)¹⁶³

phaseIncrement - поле

Инкремент фазы.

C#

```
private float phaseIncrement
```

См. также

Применяется: [SampleRateReducer](#)₁₆₃

phasor - поле

Фазовый аккумулятор.

C#

```
private float phasor
```

См. также

Применяется: [SampleRateReducer](#)₁₆₃

sample - поле

Записанный сэмпл, который будет повторяться некоторое время.

C#

```
private float sample
```

См. также

Применяется: [SampleRateReducer](#)₁₆₃

SineLFO - класс

Реализация низкочастотного генератора синусоиды.

[System.Object](#)

[MultimodSynth.AudioComponent](#)₃₃

[MultimodSynth.SineLFO](#)

C#

```
internal class SineLFO : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[SineLFO](#)₁₆₇

Свойства

[SampleRate₃₄](#) (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged₁₆₇](#), [Process₁₆₇](#), [SetFrequency₁₆₈](#), [ToString](#) (унаследован от [Object](#)), [UpdateCoefficients₁₆₈](#)

Поля

[coeff₁₆₈](#), [cos₁₆₈](#), [frequency₁₆₉](#), [sin₁₆₉](#)

SineLFO - конструктор

Инициализирует новый объект типа SineLFO.

C#

```
public SineLFO()
```

См. также

Применяется: [SineLFO₁₆₆](#)

SineLFO.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [SineLFO₁₆₆](#)

SineLFO.Process - метод

Генерация нового значения.

C#

```
public float Process()
```

См. также

Применяется: [SineLFO](#)₁₆₆

SineLFO.SetFrequency - метод

Устанавливает новое значение частоты.

C#

```
public void SetFrequency(  
    float value  
)
```

Параметры

value

См. также

Применяется: [SineLFO](#)₁₆₆

SineLFO.UpdateCoefficients - метод

Выполняет обновление коэффициента.

C#

```
private void UpdateCoefficients()
```

См. также

Применяется: [SineLFO](#)₁₆₆

coeff - поле

Коэффициент для генерации.

C#

```
private float coeff
```

См. также

Применяется: [SineLFO](#)₁₆₆

cos - поле

Текущее значение косинусоиды.

C#

```
private float cos
```

См. также

Применяется: [SineLFO](#)₁₆₆

frequency - поле

Частота генератора.

C#

```
private float frequency
```

См. также

Применяется: [SineLFO](#)₁₆₆

sin - поле

Текущее значение синусоиды.

C#

```
private float sin
```

См. также

Применяется: [SineLFO](#)₁₆₆

SoftClipper - класс

Представляет собой эффект сатурации с изменяемым порогом.

[System.Object](#)

[MultimodSynth.SoftClipper](#)

C#

```
internal class SoftClipper : IDistortion
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [Process₁₇₀](#), [SetAmount₁₇₀](#), [ToString](#) (унаследован от [Object](#))

Поля

[denominator₁₇₁](#), [normalizationCoeff₁₇₁](#), [threshold₁₇₁](#)

См. также

Ошибка! Недопустимый объект гиперссылки.

SoftClipper.Process - метод

Обработка новых входных данных.

C#

```
public float Process(  
    float input  
)
```

Параметры

input

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

Реализации

[IDistortion.Process₂₁₅](#)

См. также

Применяется: [SoftClipper₁₆₉](#)

SoftClipper.SetAmount - метод

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]).

C#

```
public void SetAmount(  
    float value  
)
```

Параметры

value

Значение "силы" эффекта.

Реализации

[IDistortion.SetAmount₂₁₆](#)

См. также

Применяется: [SoftClipper](#)¹⁶⁹

denominator - поле

Временная переменная, использующаяся для ускорения вычислений.

C#

```
private float denominator
```

См. также

Применяется: [SoftClipper](#)¹⁶⁹

normalizationCoeff - поле

Коэффициент нормализации.

C#

```
private float normalizationCoeff
```

См. также

Применяется: [SoftClipper](#)¹⁶⁹

threshold - поле

Порог, начиная с которого начинает действовать эффект.

C#

```
private float threshold
```

См. также

Применяется: [SoftClipper](#)¹⁶⁹

StereoOffsetDelay - класс

Представляет собой эффект дилей с предварительной задержкой в левом или правом канале.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.StereoOffsetDelay](#)

C#

```
internal class StereoOffsetDelay : AudioComponent,
```

IDelay

Требования

Пространство имен: MultimodSynth²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

StereoOffsetDelay¹⁷²

Свойства

SampleRate³⁴ (унаследован от AudioComponent)

Методы

Equals (унаследован от Object), Finalize (унаследован от Object), GetHashCode (унаследован от Object),
GetType (унаследован от Object), MemberwiseClone (унаследован от Object), OnSampleRateChanged¹⁷²,
Process¹⁷³, Reset¹⁷³, SetDelay¹⁷³, SetFeedback¹⁷⁴, SetStereo¹⁷⁴, ToString (унаследован от Object)

Поля

delayL¹⁷⁵, delayR¹⁷⁵, offsetBufferL¹⁷⁵, offsetBufferR¹⁷⁵

StereoOffsetDelay - конструктор

Инициализирует новый объект типа StereoOffsetDelay.

C#

```
public StereoOffsetDelay()
```

См. также

Применяется: StereoOffsetDelay¹⁷¹

StereoOffsetDelay.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

172

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

Применяется: [StereoOffsetDelay](#)¹⁷¹

StereoOffsetDelay.Process - метод

Обработка новых входных данных.

C#

```
public void Process(  
    float inputL,  
    float inputR,  
    out float outputL,  
    out float outputR  
)
```

Параметры

inputL

Левый канал входного сигнала

inputR

Правый канал входного сигнала

outputL

Левый канал выходного сигнала.

outputR

Правый канал выходного сигнала.

Реализации

[IDelay.Process](#)²¹³

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

StereoOffsetDelay.Reset - метод

Очищает буфер дилэя.

C#

```
public void Reset()
```

Реализации

[IDelay.Reset](#)²¹⁴

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

StereoOffsetDelay.SetDelay - метод

Устанавливает новое значение времени задержки (в сэмплах).

C#

```
public void SetDelay(  
    float value  
)
```

Параметры

value

Время задержки.

Реализации

[IDelay.SetDelay](#)₂₁₄

См. также

Применяется: [StereoOffsetDelay](#)₁₇₁

StereoOffsetDelay.SetFeedback - метод

Устанавливает новое значение коэффициента обратной связи.

C#

```
public void SetFeedback(  
    float value  
)
```

Параметры

value

Коэффициент обратной связи.

Реализации

[IDelay.SetFeedback](#)₂₁₄

См. также

Применяется: [StereoOffsetDelay](#)₁₇₁

StereoOffsetDelay.SetStereo - метод

Устанавливает новое значение коэффициента стерео-эффекта.

C#

```
public void SetStereo(  
    float value  
)
```

Параметры

value

Коэффициент стерео-эффекта.

Реализации

[IDelay.SetStereo](#)²¹⁵

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

delayL - поле

Линия задержки для левого канала.

C#

```
private DelayLine delayL
```

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

delayR - поле

Линия задержки для правого канала.

C#

```
private DelayLine delayR
```

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

offsetBufferL - поле

Предварительная линия задержки для левого канала.

C#

```
private DelayLine offsetBufferL
```

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

offsetBufferR - поле

Предварительная линия задержки для правого канала.

C#

```
private DelayLine offsetBufferR
```

См. также

Применяется: [StereoOffsetDelay](#)¹⁷¹

SvfFilter - класс

Реализация SVF (State Variable Filter).

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.SvfFilter](#)

C#

```
internal class SvfFilter : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[SvfFilter](#)¹⁷⁶

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹⁷⁷,
[Process](#)¹⁷⁷, [Reset](#)¹⁷⁸, [SetCutoff](#)¹⁷⁸, [SetGain](#)¹⁷⁸, [SetQ](#)¹⁷⁹, [SetType](#)¹⁷⁹, [ToString](#) (унаследован от [Object](#)),
[UpdateCoefficients](#)¹⁷⁹

Перечисления

[FilterType](#)¹⁸⁰

Поля

[a1](#)¹⁸⁰, [a2](#)¹⁸⁰, [a3](#)¹⁸¹, [cutoff](#)¹⁸¹, [gain](#)¹⁸¹, [ic1eq](#)¹⁸¹, [ic2eq](#)¹⁸², [m0](#)¹⁸², [m1](#)¹⁸², [m2](#)¹⁸², [q](#)¹⁸³, [type](#)¹⁸³

См. также

Ошибка! Недопустимый объект гиперссылки.

SvfFilter - конструктор

Инициализирует новый объект типа SvfFilter с заданными параметрами.

C#

```
public SvfFilter(
    FilterType type = FilterType.None,
    float cutoff = 20000,
    float q = 1,
    float gain = 0
)
```

Параметры*type**cutoff**q**gain***См. также**Применяется: [SvfFilter](#)¹⁷⁶**SvfFilter.OnSampleRateChanged - метод**

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(
    float newSampleRate
)
```

Параметры*newSampleRate*

Новая частота дискретизации.

См. такжеПрименяется: [SvfFilter](#)¹⁷⁶**SvfFilter.Process - метод**

Обработка новых входных данных.

C#

```
public float Process(
    float vθ
)
```

Параметры*v0*

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [SvfFilter](#)¹⁷⁶

SvfFilter.Reset - метод

Сбрасывает текущее состояние фильтра.

C#

```
public void Reset()
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

SvfFilter.SetCutoff - метод

Устанавливает новое значение частоты среза фильтра.

C#

```
public void SetCutoff(  
    float value  
)
```

Параметры*value***См. также**

Применяется: [SvfFilter](#)¹⁷⁶

SvfFilter.SetGain - метод

Устанавливает новое значение увеличения уровня громкости.

C#

```
public void SetGain(  
    float value  
)
```

Параметры*value***См. также**Применяется: [SvfFilter](#)¹⁷⁶**SvfFilter.SetQ - метод**

Устанавливает новое значение "ширины" фильтра.

C#

```
public void SetQ(  
    float value  
)
```

Параметры*value***См. также**Применяется: [SvfFilter](#)¹⁷⁶**SvfFilter.SetType - метод**

Устанавливает новое значение типа фильтра.

C#

```
public void SetType(  
    FilterType value  
)
```

Параметры*value***См. также**Применяется: [SvfFilter](#)¹⁷⁶**SvfFilter.UpdateCoefficients - метод**

Выполняет обновление всех коэффициентов.

C#

```
private void UpdateCoefficients()
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

FilterType - перечисление

Указывает тип фильтра.

Константа	Значение	Описание
All	6	
Band	2	
Bell	7	
High	3	
HighShelf	9	
Low	1	
LowShelf	8	
None	0	
Notch	4	
Peak	5	

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [SvfFilter](#)¹⁷⁶

a1 - поле

Коэффициент фильтра.

C#

```
private float a1
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

a2 - поле

Коэффициент фильтра.

C#

```
private float a2
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

a3 - поле

Коэффициент фильтра.

C#

```
private float a3
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

cutoff - поле

Частота среза фильтра.

C#

```
private float cutoff
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

gain - поле

Коэффициент фильтра.

C#

```
private float gain
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

ic1eq - поле

Первый аккумулятор.

C#

```
private float ic1eq
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

ic2eq - поле

Второй аккумулятор.

C#

```
private float ic2eq
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

m0 - поле

Коэффициент фильтра.

C#

```
private float m0
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

m1 - поле

Коэффициент фильтра.

C#

```
private float m1
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

m2 - поле

Коэффициент фильтра.

C#

```
private float m2
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

q - поле

"Ширина" фильтра.

C#

```
private float q
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

type - поле

Текущий тип фильтра.

C#

```
private FilterType type
```

См. также

Применяется: [SvfFilter](#)¹⁷⁶

Utilities - класс

Класс, содержащий различные вспомогательные методы.

[System.Object](#)

[MultimodSynth.Utilities](#)

C#

```
internal static class Utilities
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Utilities](#)¹⁸⁴

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),

[GetOrGenerateWaveTables](#)¹⁸⁴, [GetType](#) (унаследован от [Object](#)), [HalfSquareGenerator](#)¹⁸⁴,

[MemberwiseClone](#) (унаследован от [Object](#)), [MidiNoteToFrequency](#)¹⁸⁵, [QuarterSquareGenerator](#)¹⁸⁵,

[ReadParameters](#)¹⁸⁵, [SawGenerator](#)¹⁸⁶, [SineGenerator](#)¹⁸⁶, [SquareGenerator](#)¹⁸⁷, [ToString](#) (унаследован от [Object](#)), [TriangleGenerator](#)¹⁸⁷, [WriteParameters](#)¹⁸⁸

Поля

[DefaultSampleRate](#)¹⁸⁸, [WaveTables](#)¹⁸⁸

Utilities - конструктор

Статический конструктор,читывающий сгенерированные таблицы сэмплов из файла.

C#

```
private static Utilities()
```

См. также

Применяется: [Utilities](#)¹⁸³

Utilities.GetOrGenerateWaveTables - метод

Метод, обеспечивающий создание таблиц сэмплов.

C#

```
private static void GetOrGenerateWaveTables()
```

См. также

Применяется: [Utilities](#)¹⁸³

Utilities.HalfSquareGenerator - метод

Функция-генератор квадратной волны с применением широтно-импульсной модуляции.

C#

```
private static double HalfSquareGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

phase

freq

maxFreq

См. также

Применяется: [Utilities](#)₁₈₃

Utilities.MidiNoteToFrequency - метод

Метод, переводящий номер клавиши в частоту.

C#

```
public static double MidiNoteToFrequency(  
    int note  
)
```

Параметры*note***См. также**

Применяется: [Utilities](#)₁₈₃

Utilities.QuarterSquareGenerator - метод

Функция-генератор квадратной волны с применением широтно-импульсной модуляции.

C#

```
private static double QuarterSquareGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры*phase**freq**maxFreq***См. также**

Применяется: [Utilities](#)₁₈₃

Utilities.ReadParameters - метод

Метод, используемый для считывания значений параметров из файла.

C#

```
public static void ReadParameters(  
    Stream stream,  
    VstParameterCollection activeParameters  
)
```

Параметры

stream

activeParameters

См. также

Применяется: [Utilities](#)¹⁸³

Utilities.SawGenerator - метод

Функция-генератор пилообразной волны.

C#

```
private static double SawGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

phase

freq

maxFreq

См. также

Применяется: [Utilities](#)¹⁸³

Utilities.SineGenerator - метод

Функция-генератор синусоиды.

C#

```
private static double SineGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

phase

freq

maxFreq

См. также

Применяется: [Utilities](#)₁₈₃

Utilities.SquareGenerator - метод

Функция-генератор квадратной волны.

C#

```
private static double SquareGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

phase

freq

maxFreq

См. также

Применяется: [Utilities](#)₁₈₃

Utilities.TriangleGenerator - метод

Функция-генератор треугольной волны.

C#

```
private static double TriangleGenerator(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

187

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

*phase**freq**maxFreq***См. также**Применяется: [Utilities](#)₁₈₃**Utilities.WriteParameters - метод**

Метод, используемый для записи значений параметров в файл.

C#

```
public static void WriteParameters(
    Stream stream,
    VstParameterCollection activeParameters
)
```

Параметры*stream**activeParameters***См. также**Применяется: [Utilities](#)₁₈₃**WaveTables - поле**

Массив таблиц сэмплов.

C#

```
public static WaveTableOscillator[] WaveTables
```

См. такжеПрименяется: [Utilities](#)₁₈₃**DefaultSampleRate - поле**

Стандартная частота дискретизации.

C#

```
private const double DefaultSampleRate = 44100
```

См. также

Применяется: [Utilities](#)¹⁸³

VariousTimeDelay - класс

Представляет собой эффект дилэй с отличающимся временем задержки в левом и правом канале.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.VariousTimeDelay](#)

C#

```
internal class VariousTimeDelay : AudioComponent,  
    IDelay
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[VariousTimeDelay](#)¹⁸⁹

Свойства

[SampleRate](#)³⁴ (унаследован от [AudioComponent](#))

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)¹⁹⁰,
[Process](#)¹⁹⁰, [Reset](#)¹⁹⁰, [SetDelay](#)¹⁹¹, [SetFeedback](#)¹⁹¹, [SetStereo](#)¹⁹², [ToString](#) (унаследован от [Object](#))

Поля

[delay](#)¹⁹², [delayL](#)¹⁹², [delayR](#)¹⁹², [leftDelayCoeff](#)¹⁹³, [rightDelayCoeff](#)¹⁹³

VariousTimeDelay - конструктор

Инициализирует новый объект типа VariousTimeDelay.

C#

```
public VariousTimeDelay()
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.Process - метод

Обработка новых входных данных.

C#

```
public void Process(  
    float inputL,  
    float inputR,  
    out float outputL,  
    out float outputR  
)
```

Параметры

inputL

Левый канал входного сигнала

inputR

Правый канал входного сигнала

outputL

Левый канал выходного сигнала.

outputR

Правый канал выходного сигнала.

Реализации

[IDelay.Process](#)²¹³

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.Reset - метод

Очищает буфер дилэя.

C#

```
public void Reset()
```

Реализации

[IDelay.Reset](#)²¹⁴

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.SetDelay - метод

Устанавливает новое значение времени задержки (в сэмплах).

C#

```
public void SetDelay(  
    float value  
)
```

Параметры

value

Время задержки.

Реализации

[IDelay.SetDelay](#)²¹⁴

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.SetFeedback - метод

Устанавливает новое значение коэффициента обратной связи.

C#

```
public void SetFeedback(  
    float value  
)
```

Параметры

value

Коэффициент обратной связи.

Реализации

[IDelay.SetFeedback](#)²¹⁴

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

VariousTimeDelay.SetStereo - метод

Устанавливает новое значение коэффициента стерео-эффекта.

C#

```
public void SetStereo(  
    float value  
)
```

Параметры

value

Коэффициент стерео-эффекта.

Реализации

[IDelay.SetStereo](#)²¹⁵

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

delay - поле

Базовое время задержки (в сэмплах).

C#

```
private float delay
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

delayL - поле

Линия задержки для левого канала.

C#

```
private DelayLine delayL
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

delayR - поле

Линия задержки для правого канала.

C#

```
private DelayLine delayR
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

leftDelayCoeff - поле

Коэффициент времени задержки для левого канала.

C#

```
private float leftDelayCoeff
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

rightDelayCoeff - поле

Коэффициент времени задержки для правого канала.

C#

```
private float rightDelayCoeff
```

См. также

Применяется: [VariousTimeDelay](#)¹⁸⁹

Voice - класс

Компонент плагина, представляющий собой один голос.

[System.Object](#)

[MultimodSynth.AudioComponent](#)³³

[MultimodSynth.Voice](#)

C#

```
internal class Voice : AudioComponent
```

Требования

Пространство имен: [MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

Voice¹⁹⁴

Свойства

IsActive¹⁹⁵, Modulation¹⁹⁵, Note¹⁹⁵, SampleRate³⁴ (унаследован от AudioComponent)

Методы

Equals (унаследован от Object), Finalize (унаследован от Object), GetHashCode (унаследован от Object), GetType (унаследован от Object), MemberwiseClone (унаследован от Object), OnSampleRateChanged¹⁹⁵, PlayNote¹⁹⁶, Process¹⁹⁶, ToString (унаследован от Object), TriggerRelease¹⁹⁶

Перечисления

ModulationType¹⁹⁷

Поля

envA¹⁹⁷, envB¹⁹⁷, envFilter¹⁹⁷, filter¹⁹⁸, fmAmountMultiplier¹⁹⁸, noteVelocity¹⁹⁸, oscA¹⁹⁸, oscB¹⁹⁹

Voice - конструктор

Инициализирует новый объект класса Voice, имеющий переданные компоненты.

C#

```
public Voice(  
    Plugin plugin,  
    Oscillator oscA,  
    Oscillator oscB,  
    Filter filter,  
    Envelope envA,  
    Envelope envB,  
    Envelope envFilter  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

oscA

Осциллятор А.

oscB

Осциллятор В.

filter

Фильтр.

envA

Огибающая уровня осциллятора А.

envB

Огибающая уровня оциллятора В.

envFilter

Огибающая частоты среза фильтра.

См. также

Применяется: [Voice](#)₁₉₃

Voice.IsActive - свойство

Указывает, активен ли данный голос.

C#

```
public bool IsActive {get; private set;}
```

См. также

Применяется: [Voice](#)₁₉₃

Voice.Modulation - свойство

Текущий тип модуляции.

C#

```
public ModulationType Modulation {get; set;}
```

См. также

Применяется: [Voice](#)₁₉₃

Voice.Note - свойство

Текущая играемая нота.

C#

```
public MidiNote Note {get; private set;}
```

См. также

Применяется: [Voice](#)₁₉₃

Voice.OnSampleRateChanged - метод

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(  
    float newSampleRate  
)
```

Параметры

newSampleRate

Новая частота дискретизации.

См. также

Применяется: [Voice₁₉₃](#)

Voice.PlayNote - метод

Играет переданную ноту.

C#

```
public void PlayNote(  
    MidiNote note  
)
```

Параметры

note

Нота, которую необходимо проиграть.

См. также

Применяется: [Voice₁₉₃](#)

Voice.Process - метод

Генерация новых выходных данных.

C#

```
public float Process()
```

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [Voice₁₉₃](#)

Voice.TriggerRelease - метод

Прекращает проигрывание ноты.

C#

```
public void TriggerRelease()
```

См. также

Применяется: [Voice](#)₁₉₃

ModulationType - перечисление

Указывает тип взаимодействия двух осцилляторов.

Константа	Значение	Описание
AmplitudeModulationA	3	
AmplitudeModulationB	4	
None	0	
PhaseModulationA	1	
PhaseModulationB	2	

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [Voice](#)₁₉₃

envA - поле

Огибающая громкости осциллятора A.

C#

```
private Envelope envA
```

См. также

Применяется: [Voice](#)₁₉₃

envB - поле

Огибающая громкости осциллятора B.

C#

```
private Envelope envB
```

См. также

Применяется: [Voice](#)₁₉₃

envFilter - поле

Огибающая частоты среза фильтра.

C#

```
private Envelope envFilter
```

См. также

Применяется: [Voice](#)₁₉₃

filter - поле

Фильтр.

C#

```
private Filter filter
```

См. также

Применяется: [Voice](#)₁₉₃

fmAmountMultiplier - поле

Максимальная "сила" частотной модуляции.

C#

```
private float fmAmountMultiplier
```

См. также

Применяется: [Voice](#)₁₉₃

noteVelocity - поле

Сила нажатия текущей играемой ноты.

C#

```
private float noteVelocity
```

См. также

Применяется: [Voice](#)₁₉₃

oscA - поле

Оscиллятор A.

C#

```
private Oscillator oscA
```

См. также

Применяется: [Voice](#)₁₉₃

oscB - поле

Оscиллятор B.

C#

```
private Oscillator oscB
```

См. также

Применяется: [Voice](#)₁₉₃

VoiceManager - класс

Компонент плагина, управляющий всем голосами.

[System.Object](#)

[MultimodSynth.AudioComponent](#)₃₃

[MultimodSynth.AudioComponentWithParameters](#)₃₅

[MultimodSynth.VoiceManager](#)

C#

```
internal class VoiceManager : AudioComponentWithParameters
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[VoiceManager](#)₂₀₀

Свойства

[FilterCutoffEnvelopeManager](#)₂₀₀, [FilterManager](#)₂₀₀, [ModulationTypeManager](#)₂₀₁, [OscAManager](#)₂₀₁,

[OscAVolumeEnvelopeManager](#)₂₀₁, [OscBManager](#)₂₀₁, [OscBVolumeEnvelopeManager](#)₂₀₁, [Plugin](#)₃₆

(унаследован от [AudioComponentWithParameters](#)), [SampleRate](#)₃₄ (унаследован от [AudioComponent](#))

Методы

[CreateVoice](#)₂₀₂, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [InitializeParameters](#)₂₀₂, [MemberwiseClone](#) (унаследован от [Object](#)), [OnSampleRateChanged](#)₂₀₂, [PlayNote](#)₂₀₃, [Process](#)₂₀₃, [ReleaseNote](#)₂₀₃, [SetModulationType](#)₂₀₄, [StopVoice](#)₂₀₄, [ToString](#) (унаследован от [Object](#))

Поля

`MaxVoicesCount`²⁰⁶, `activeVoices`²⁰⁴, `freeVoicesIndices`²⁰⁵, `modulationType`²⁰⁵,
`noteToVoicesMapping`²⁰⁵, `voicesPool`²⁰⁵

VoicesManager - конструктор

Инициализирует новый объект класса VoicesManager, принадлежащий переданному плагину и имеющий переданный префикс названия параметров.

C#

```
public VoicesManager(  
    Plugin plugin,  
    string parameterPrefix  
)
```

Параметры

plugin

Плагин, которому принадлежит создаваемый объект.

parameterPrefix

Префикс названия параметров.

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.FilterCutoffEnvelopeManager - свойство

Менеджер всех огибающих частоты среза фильтра во всех голосах.

C#

```
public EnvelopesManager FilterCutoffEnvelopeManager {get; set;}
```

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.FilterManager - свойство

Менеджер фильра во всех голосах.

C#

```
public FiltersManager FilterManager {get; set;}
```

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.ModulationTypeManager - свойство

Объект, управляющий параметром типа модуляции.

C#

```
public VstParameterManager ModulationTypeManager {get; private set;}
```

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.OscAManager - свойство

Менеджер всех осцилляторов А во всех голосах.

C#

```
public OscillatorsManager OscAManager {get; set;}
```

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.OscAVolumeEnvelopeManager - свойство

Менеджер всех огибающих уровня осциллятора А во всех голосах.

C#

```
public EnvelopesManager OscAVolumeEnvelopeManager {get; set;}
```

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.OscBManager - свойство

Менеджер всех осцилляторов В во всех голосах.

C#

```
public OscillatorsManager OscBManager {get; set;}
```

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.OscBVolumeEnvelopeManager - свойство

Менеджер всех огибающих уровня осциллятора В во всех голосах.

C#

```
public EnvelopesManager OscBVolumeEnvelopeManager {get; set;}
```

См. такжеПрименяется: [VoicesManager](#)₁₉₉**VoicesManager.CreateVoice - метод**

Возвращает новый объект голоса, связанный с этим объектом.

C#

```
private Voice CreateVoice()
```

См. такжеПрименяется: [VoicesManager](#)₁₉₉**VoicesManager.InitializeParameters - метод**

Инициализирует параметры с помощью переданной фабрики параметров.

C#

```
protected override void InitializeParameters(
    ParameterFactory factory
)
```

Параметры*factory*

Фабрика параметров

См. такжеПрименяется: [VoicesManager](#)₁₉₉**VoicesManager.OnSampleRateChanged - метод**

Обработчик изменения частоты дискретизации.

C#

```
protected override void OnSampleRateChanged(
    float newSampleRate
)
```

Параметры*newSampleRate*

Новая частота дискретизации.

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.PlayNote - метод

Играет переданную ноту.

C#

```
public void PlayNote(  
    MidiNote note  
)
```

Параметры

note

Нота, которую необходимо проиграть.

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.Process - метод

Генерация новых выходных данных.

C#

```
public float Process()
```

Возвращаемые значения

Выходной сигнал.

См. также

Применяется: [VoicesManager](#)¹⁹⁹

VoicesManager.ReleaseNote - метод

Отпускает переданную ноту.

C#

```
public void ReleaseNote(  
    MidiNote note  
)
```

Параметры

note

Нота, которую необходимо отпустить.

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.SetModulationType - метод

Обработчик изменения типа модуляции.

C#

```
private void SetModulationType(  
    float value  
)
```

Параметры

value

Нормированное новое значение параметра.

См. также

Применяется: [VoicesManager](#)₁₉₉

VoicesManager.StopVoice - метод

Останавливает переданный голос и помечает его как неиспользуемый.

C#

```
private void StopVoice(  
    Voice voice  
)
```

Параметры

voice

См. также

Применяется: [VoicesManager](#)₁₉₉

activeVoices - поле

Список активных голосов.

C#

```
private List<Voice> activeVoices
```

См. также

Применяется: [VoicesManager](#)₁₉₉

freeVoicesIndices - поле

Отсортированное множество номеров свободных голосов.

C#

```
private SortedSet<int> freeVoicesIndices
```

См. также

Применяется: [VoicesManager](#)₁₉₉

modulationType - поле

Текущий тип модуляции.

C#

```
private ModulationType modulationType
```

См. также

Применяется: [VoicesManager](#)₁₉₉

noteToVoicesMapping - поле

Отображение номера ноты в список голосов, играющих эту ноту.

C#

```
private Dictionary<byte, List<Voice>> noteToVoicesMapping
```

См. также

Применяется: [VoicesManager](#)₁₉₉

voicesPool - поле

Список ссылок на все используемые и неиспользуемые голоса.

C#

```
private List<Voice> voicesPool
```

См. также

Применяется: [VoicesManager](#)₁₉₉

MaxVoicesCount - поле

Максимальное количество голосов.

C#

```
private const int MaxVoicesCount = 32
```

См. также

Применяется: [VoicesManager](#)₁₉₉

WaveTableOscillator - класс

Реализация осциллятора, основанного на таблицах сэмплов.

[System.Object](#)

[MultimodSynth.WaveTableOscillator](#)

C#

```
public class WaveTableOscillator
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[WaveTableOscillator](#)₂₀₆

Методы

[Clone](#)₂₀₇, [Deserialize](#)₂₀₇, [Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)), [GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [Normalize](#)₂₀₈, [PrepareSamples](#)₂₀₈, [Process](#)₂₀₉, [Serialize](#)₂₀₉, [SetPhaseIncrement](#)₂₀₉, [ToString](#) (унаследован от [Object](#))

Классы

[WaveTable](#)₂₁₀

Делегаты

[GeneratorFunction](#)₂₁₁

Поля

[DefaultSampleRate](#)₂₁₃, [phaseIncrement](#)₂₁₂, [waveTable](#)₂₁₂, [waveTables](#)₂₁₂, [waveTablesAmount](#)₂₁₂

WaveTableOscillator(GeneratorFunction, Double, Double) - конструктор

Инициализирует новый объект типа WaveTableOscillator с заданными параметрами.

C#

```
public WaveTableOscillator(  
    GeneratorFunction generator,  
    double startFrequency,  
    double endFrequency  
)
```

Параметры

generator

Функция-генератор.

startFrequency

Минимальная частота, на которой планируется генерировать звук.

endFrequency

Максимальная частота, на которой планируется генерировать звук.

См. также

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator - конструктор

Инициализирует пустой объект типа WaveTableOscillator.

C#

```
private WaveTableOscillator()
```

См. также

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.Clone - метод

Клонирует текущую таблицу сэмплов.

C#

```
public WaveTableOscillator Clone()
```

Возвращаемые значения

Новая таблица сэмплов.

См. также

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.Deserialize - метод

Метод, восстанавливает объект WaveTableOscillator из потока.

C#

```
public static WaveTableOscillator Deserialize(  
    Stream stream  
)
```

Параметры

stream

См. также

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.Normalize - метод

Метод, нормализующий все таблицы этого осциллятора.

C#

```
private void Normalize()
```

См. также

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.PrepareSamples - метод

Метод, подготавливающий массив сэмплов.

C#

```
private static float[] PrepareSamples(  
    GeneratorFunction generator,  
    double freq,  
    double maxFreq  
)
```

Параметры

generator

Функция-генератор.

freq

maxFreq

Возвращаемые значения

Массив сэмплов.

См. так же

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.Process - метод

Метод, возвращающий сэмпл по заданной фазе.

C#

```
public float Process(  
    float phase  
)
```

Параметры

phase

Фаза.

Возвращаемые значения

Выходной сэмпл.

См. так же

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.Serialize - метод

Метод, записывающий объект WaveTableOscillator в поток.

C#

```
public static void Serialize(  
    Stream stream,  
    WaveTableOscillator obj  
)
```

Параметры

stream

obj

См. так же

Применяется: [WaveTableOscillator](#)²⁰⁶

WaveTableOscillator.SetPhaseIncrement - метод

Устанавливает новое значение инкремента фазы.

C#

```
public void SetPhaseIncrement(  
    double phaseIncrement  
)
```

Параметры

phaseIncrement

Инкремент фазы.

См. так же

Применяется: [WaveTableOscillator](#)₂₀₆

WaveTable - класс

Представляет собой одну таблицу сэмплов, записанную с заданным инкрементом фазы.

[System.Object](#)

[MultimodSynth.WaveTableOscillator.WaveTable](#)

C#

```
private class WaveTable
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Equals](#) (унаследован от [Object](#)), [Finalize](#) (унаследован от [Object](#)), [GetHashCode](#) (унаследован от [Object](#)),
[GetType](#) (унаследован от [Object](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [ToString](#) (унаследован от [Object](#))

Поля

[Length](#)₂₁₀, [PhaseIncrement](#)₂₁₁, [Samples](#)₂₁₁

См. так же

Применяется: [WaveTableOscillator](#)₂₀₆

Length - поле

Длина таблицы.

C#

```
public int Length
```

См. так же

Применяется: [WaveTable](#)₂₁₀

PhaseIncrement - поле

Инкремент фазы, с которым эта таблица была сгенерирована.

C#

```
public float PhaseIncrement
```

См. также

Применяется: [WaveTable](#)²¹⁰

Samples - поле

Сэмплы.

C#

```
public float[] Samples
```

См. также

Применяется: [WaveTable](#)²¹⁰

GeneratorFunction - делегат

Представляет собой функцию-генератор для создания таблицы.

C#

```
public delegate double GeneratorFunction(  
    double phase,  
    double freq,  
    double maxFreq  
)
```

Параметры

phase

Текущая фаза.

freq

Минимальная частота, на которой будет проигрываться создаваемая таблица.

maxFreq

Максимальная частота, на которой будет проигрываться создаваемая таблица.

Возвращаемые значения

Значение функции-генератора при заданных параметрах.

Требования

Пространство имен:[MultimodSynth](#)²³

Сборка: MultimodSynth (in MultimodSynth.dll)

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

phaseIncrement - поле

Текущий инкремент фазы.

C#

```
private float phaseIncrement
```

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

waveTable - поле

Текущая таблица.

C#

```
private WaveTable waveTable
```

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

waveTables - поле

Список всех таблиц.

C#

```
private WaveTable[] waveTables
```

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

waveTablesAmount - поле

Общее количество таблиц.

C#

```
private int waveTablesAmount
```

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

DefaultSampleRate - поле

Стандартная частота дискретизации.

C#

```
private const double DefaultSampleRate = 44100
```

См. также

Применяется: [WaveTableOscillator](#)₂₀₆

IDelay - интерфейс

Интерфейс, представляющий собой эффект дилэй.

C#

```
internal interface IDelay
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Process](#)₂₁₃, [Reset](#)₂₁₄, [SetDelay](#)₂₁₄, [SetFeedback](#)₂₁₄, [SetStereo](#)₂₁₅

IDelay.Process - метод

Обработка новых входных данных.

C#

```
void Process(
    float inputL,
    float inputR,
    out float outputL,
    out float outputR
)
```

Параметры

inputL

Левый канал входного сигнала

inputR

Правый канал входного сигнала

outputL

Левый канал выходного сигнала.

outputR

Правый канал выходного сигнала.

См. также

Применяется: [IDelay](#)₂₁₃

IDelay.Reset - метод

Очищает буфер дилэя.

C#

```
void Reset()
```

См. также

Применяется: [IDelay](#)₂₁₃

IDelay.SetDelay - метод

Устанавливает новое значение времени задержки (в сэмплах).

C#

```
void SetDelay(  
    float value  
)
```

Параметры

value

Время задержки.

См. также

Применяется: [IDelay](#)₂₁₃

IDelay.SetFeedback - метод

Устанавливает новое значение коэффициента обратной связи.

C#

```
void SetFeedback(  
    float value  
)
```

Параметры

value

Коэффициент обратной связи.

См. также

Применяется: [IDelay](#)₂₁₃

IDelay.SetStereo - метод

Устанавливает новое значение коэффициента стерео-эффекта.

C#

```
void SetStereo(  
    float value  
)
```

Параметры

value

Коэффициент стерео-эффекта.

См. также

Применяется: [IDelay](#)₂₁₃

IDistortion - интерфейс

Интерфейс, представляющий собой эффект дисторшн.

C#

```
internal interface IDistortion
```

Требования

Пространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Методы

[Process](#)₂₁₅, [SetAmount](#)₂₁₆

IDistortion.Process - метод

Обработка новых входных данных.

C#

```
float Process(  
    float input  
)
```

Параметры*input*

Входной сигнал.

Возвращаемые значения

Выходной сигнал.

См. такжеПрименяется: [IDistortion](#)₂₁₅**IDistortion.SetAmount - метод**

Устанавливает новое значение "силы" эффекта (в диапазоне [0, 1]).

C#

```
void SetAmount(  
    float value  
)
```

Параметры*value*

Значение "силы" эффекта.

См. такжеПрименяется: [IDistortion](#)₂₁₅**MidiNote - структура**

Представляет собой одну ноту.

C#

```
public struct MidiNote
```

ТребованияПространство имен: [MultimodSynth](#)₂₃

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства[NoteNo](#)₂₁₆, [Velocity](#)₂₁₇**MidiNote.NoteNo - свойство**

Номер ноты.

C#

```
public byte NoteNo {get; set;}
```

См. также

Применяется: [MidiNote](#)₂₁₆

MidiNote.Velocity - свойство

Сила нажатия клавиши.

C#

```
public byte Velocity {get; set;}
```

См. также

Применяется: [MidiNote](#)₂₁₆

MultimodSynth.UI - пространство имен

Классы

[BipolarKnob](#)²¹⁸, [EditorView](#)²²³, [Knob](#)²³³

BipolarKnob - класс

Представляет собой вариант ручки, стандартное значение которой находится на середине.

[System.Object](#)

[System.Windows.Threading.DispatcherObject](#)

[System.Windows.DependencyObject](#)

[System.Windows.Media.Visual](#)

[System.Windows.UIElement](#)

[System.Windows.FrameworkElement](#)

[System.Windows.Controls.Control](#)

[System.Windows.Controls.ContentControl](#)

[System.Windows.Controls.UserControl](#)

[MultimodSynth.UI.Knob](#)²³³

MultimodSynth.UI.BipolarKnob

C#

```
public class BipolarKnob : Knob
```

Требования

Пространство имен: [MultimodSynth.UI](#)²¹⁸

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[BipolarKnob](#)²²³

Свойства

[ActualHeight](#) (унаследован от [FrameworkElement](#)), [ActualWidth](#) (унаследован от [FrameworkElement](#)), [AllowDrop](#) (унаследован от [UIElement](#)), [AreAnyTouchesCaptured](#) (унаследован от [UIElement](#)), [AreAnyTouchesCapturedWithin](#) (унаследован от [UIElement](#)), [AreAnyTouchesDirectlyOver](#) (унаследован от [UIElement](#)), [AreAnyTouchesOver](#) (унаследован от [UIElement](#)), [Background](#) (унаследован от [Control](#)), [BindingGroup](#) (унаследован от [FrameworkElement](#)), [BitmapEffect](#) (унаследован от [UIElement](#)), [BitmapEffectInput](#) (унаследован от [UIElement](#)), [BorderBrush](#) (унаследован от [Control](#)), [BorderThickness](#) (унаследован от [Control](#)), [CacheMode](#) (унаследован от [UIElement](#)), [Clip](#) (унаследован от [UIElement](#)), [ClipToBounds](#) (унаследован от [UIElement](#)), [CommandBindings](#) (унаследован от [UIElement](#)), [Content](#) (унаследован от [ContentControl](#)), [ContentStringFormat](#) (унаследован от [ContentControl](#)), [ContentTemplate](#) (унаследован от [ContentControl](#)), [ContentTemplateSelector](#) (унаследован от [ContentControl](#)), [ContextMenu](#) (унаследован от [FrameworkElement](#)), [Cursor](#) (унаследован от [FrameworkElement](#)), [DataContext](#) (унаследован от [FrameworkElement](#)), [DependencyObjectType](#) (унаследован от [DependencyObject](#)), [DesiredSize](#) (унаследован от [UIElement](#)), [Dispatcher](#) (унаследован от [DispatcherObject](#)), [DisplayValue](#)²³⁸ (унаследован от [Knob](#)), [DisplayValueConverter](#)²³⁹ (унаследован от [Knob](#)), [Effect](#) (унаследован от [UIElement](#)), [FlowDirection](#)

(унаследован от [FrameworkElement](#)), [FocusVisualStyle](#) (унаследован от [FrameworkElement](#)), [Focusable](#) (унаследован от [UIElement](#)), [FontFamily](#) (унаследован от [Control](#)), [FontSize](#) (унаследован от [Control](#)), [FontStretch](#) (унаследован от [Control](#)), [FontStyle](#) (унаследован от [Control](#)), [FontWeight](#) (унаследован от [Control](#)), [ForceCursor](#) (унаследован от [FrameworkElement](#)), [Foreground](#) (унаследован от [Control](#)), [HasAnimatedProperties](#) (унаследован от [UIElement](#)), [HasContent](#) (унаследован от [ContentControl](#)), [Height](#) (унаследован от [FrameworkElement](#)), [HorizontalAlignment](#) (унаследован от [FrameworkElement](#)), [HorizontalContentAlignment](#) (унаследован от [Control](#)), [InputBindings](#) (унаследован от [UIElement](#)), [InputScope](#) (унаследован от [FrameworkElement](#)), [IsArrangeValid](#) (унаследован от [UIElement](#)), [.IsEnabled](#) (унаследован от [UIElement](#)), [IsFocused](#) (унаследован от [UIElement](#)), [IsHitTestVisible](#) (унаследован от [UIElement](#)), [IsInitialized](#) (унаследован от [FrameworkElement](#)), [IsInputMethodEnabled](#) (унаследован от [UIElement](#)), [IsKeyboardFocusWithin](#) (унаследован от [UIElement](#)), [IsKeyboardFocused](#) (унаследован от [UIElement](#)), [IsLoaded](#) (унаследован от [FrameworkElement](#)), [IsManipulationEnabled](#) (унаследован от [UIElement](#)), [IsMeasureValid](#) (унаследован от [UIElement](#)), [IsMouseCaptureWithin](#) (унаследован от [UIElement](#)), [IsMouseCaptured](#) (унаследован от [UIElement](#)), [IsMouseDirectlyOver](#) (унаследован от [UIElement](#)), [IsMouseOver](#) (унаследован от [UIElement](#)), [IsSealed](#) (унаследован от [DependencyObject](#)), [IsStylusCaptureWithin](#) (унаследован от [UIElement](#)), [IsStylusCaptured](#) (унаследован от [UIElement](#)), [IsStylusDirectlyOver](#) (унаследован от [UIElement](#)), [IsStylusOver](#) (унаследован от [UIElement](#)), [IsTabStop](#) (унаследован от [Control](#)), [IsVisible](#) (унаследован от [UIElement](#)), [Language](#) (унаследован от [FrameworkElement](#)), [LayoutTransform](#) (унаследован от [FrameworkElement](#)), [Margin](#) (унаследован от [FrameworkElement](#)), [MaxHeight](#) (унаследован от [FrameworkElement](#)), [MaxWidth](#) (унаследован от [FrameworkElement](#)), [MinHeight](#) (унаследован от [FrameworkElement](#)), [MinWidth](#) (унаследован от [FrameworkElement](#)), [Name](#) (унаследован от [FrameworkElement](#)), [Opacity](#) (унаследован от [UIElement](#)), [OpacityMask](#) (унаследован от [UIElement](#)), [OverridesDefaultStyle](#) (унаследован от [FrameworkElement](#)), [Padding](#) (унаследован от [Control](#)), [Parent](#) (унаследован от [FrameworkElement](#)), [PersistId](#) (унаследован от [UIElement](#)), [RenderSize](#) (унаследован от [UIElement](#)), [RenderTransform](#) (унаследован от [UIElement](#)), [RenderTransformOrigin](#) (унаследован от [UIElement](#)), [Resources](#) (унаследован от [FrameworkElement](#)), [SnapsToDevicePixels](#) (унаследован от [UIElement](#)), [Stroke](#)²³⁹ (унаследован от [Knob](#)), [Style](#) (унаследован от [FrameworkElement](#)), [TabIndex](#) (унаследован от [Control](#)), [Tag](#) (унаследован от [FrameworkElement](#)), [Template](#) (унаследован от [Control](#)), [TemplatedParent](#) (унаследован от [FrameworkElement](#)), [ToolTip](#) (унаследован от [FrameworkElement](#)), [TouchesCaptured](#) (унаследован от [UIElement](#)), [TouchesCapturedWithin](#) (унаследован от [UIElement](#)), [TouchesDirectlyOver](#) (унаследован от [UIElement](#)), [TouchesOver](#) (унаследован от [UIElement](#)), [Triggers](#) (унаследован от [FrameworkElement](#)), [Uid](#) (унаследован от [UIElement](#)), [UseLayoutRounding](#) (унаследован от [FrameworkElement](#)), [VerticalAlignment](#) (унаследован от [FrameworkElement](#)), [VerticalContentAlignment](#) (унаследован от [Control](#)), [Visibility](#) (унаследован от [UIElement](#)), [Width](#) (унаследован от [FrameworkElement](#))

Методы

[AddChild](#) (унаследован от [ContentControl](#)), [AddHandler](#) (унаследован от [UIElement](#)), [AddLogicalChild](#) (унаследован от [FrameworkElement](#)), [AddText](#) (унаследован от [ContentControl](#)), [AddToEventRoute](#) (унаследован от [UIElement](#)), [AddVisualChild](#) (унаследован от [Visual](#)), [ApplyAnimationClock](#) (унаследован от [UIElement](#)), [ApplyTemplate](#) (унаследован от [FrameworkElement](#)), [Arrange](#) (унаследован от [UIElement](#)), [ArrangeCore](#) (унаследован от [FrameworkElement](#)), [ArrangeOverride](#) (унаследован от [Control](#)), [AttachTo](#)₂₃₉ (унаследован от [Knob](#)), [BeginAnimation](#) (унаследован от [UIElement](#)), [BeginInit](#) (унаследован от [FrameworkElement](#)), [BeginStoryboard](#) (унаследован от [FrameworkElement](#)), [BringIntoView](#) (унаследован от [FrameworkElement](#)), [CaptureMouse](#) (унаследован от [UIElement](#)), [CaptureStylus](#) (унаследован от [UIElement](#)), [CaptureTouch](#) (унаследован от [UIElement](#)), [CheckAccess](#) (унаследован от [DispatcherObject](#)), [ClearValue](#) (унаследован от [DependencyObject](#))

`DependencyObject`), `CoerceValue` (унаследован от `DependencyObject`), `EndInit` (унаследован от `FrameworkElement`), `Equals` (унаследован от `DependencyObject`), `Finalize` (унаследован от `Object`), `FindCommonVisualAncestor` (унаследован от `Visual`), `FindName` (унаследован от `FrameworkElement`), `FindResource` (унаследован от `FrameworkElement`), `Focus` (унаследован от `UIElement`), `GetAnimationBaseValue` (унаследован от `UIElement`), `GetBindingExpression` (унаследован от `FrameworkElement`), `GetHashCode` (унаследован от `DependencyObject`), `GetLayoutClip` (унаследован от `FrameworkElement`), `GetLocalValueEnumerator` (унаследован от `DependencyObject`), `GetTemplateChild` (унаследован от `FrameworkElement`), `GetType` (унаследован от `Object`), `GetUIParentCore` (унаследован от `FrameworkElement`), `GetValue` (унаследован от `DependencyObject`), `GetVisualChild` (унаследован от `FrameworkElement`), `HitTestCore` (унаследован от `UIElement`), `InputHitTest` (унаследован от `UIElement`), `InvalidateArrange` (унаследован от `UIElement`), `InvalidateMeasure` (унаследован от `UIElement`), `InvalidateProperty` (унаследован от `DependencyObject`), `InvalidateVisual` (унаследован от `UIElement`), `IsAncestorOf` (унаследован от `Visual`), `IsDescendantOf` (унаследован от `Visual`), `Measure` (унаследован от `UIElement`), `MeasureCore` (унаследован от `FrameworkElement`), `MeasureOverride` (унаследован от `Control`), `MemberwiseClone` (унаследован от `Object`), `MoveFocus` (унаследован от `FrameworkElement`), `OnAccessKey` (унаследован от `UIElement`), `OnApplyTemplate` (унаследован от `FrameworkElement`), `OnChildDesiredSizeChanged` (унаследован от `UIElement`), `OnContentChanged` (унаследован от `ContentControl`), `OnContentStringFormatChanged` (унаследован от `ContentControl`), `OnContentTemplateChanged` (унаследован от `ContentControl`), `OnContentTemplateSelectorChanged` (унаследован от `ContentControl`), `OnContextMenuClosing` (унаследован от `FrameworkElement`), `OnContextMenuOpening` (унаследован от `FrameworkElement`), `OnCreateAutomationPeer` (унаследован от `UserControl`), `OnDpiChanged` (унаследован от `Visual`), `OnDragEnter` (унаследован от `UIElement`), `OnDragLeave` (унаследован от `UIElement`), `OnDragOver` (унаследован от `UIElement`), `OnDrop` (унаследован от `UIElement`), `OnGiveFeedback` (унаследован от `UIElement`), `OnGotFocus` (унаследован от `FrameworkElement`), `OnGotKeyboardFocus` (унаследован от `UIElement`), `OnGotMouseCapture` (унаследован от `UIElement`), `OnGotStylusCapture` (унаследован от `UIElement`), `OnGotTouchCapture` (унаследован от `UIElement`), `OnInitialized` (унаследован от `FrameworkElement`), `OnIsKeyboardFocusWithinChanged` (унаследован от `UIElement`), `OnIsKeyboardFocusedChanged` (унаследован от `UIElement`), `OnIsMouseCaptureWithinChanged` (унаследован от `UIElement`), `OnIsMouseCapturedChanged` (унаследован от `UIElement`), `OnIsMouseDirectlyOverChanged` (унаследован от `UIElement`), `OnIsStylusCaptureWithinChanged` (унаследован от `UIElement`), `OnIsStylusCapturedChanged` (унаследован от `UIElement`), `OnIsStylusDirectlyOverChanged` (унаследован от `UIElement`), `OnKeyDown` (унаследован от `UIElement`), `OnKeyUp` (унаследован от `UIElement`), `OnLostFocus` (унаследован от `UIElement`), `OnLostKeyboardFocus` (унаследован от `UIElement`), `OnLostMouseCapture` (унаследован от `UIElement`), `OnLostStylusCapture` (унаследован от `UIElement`), `OnLostTouchCapture` (унаследован от `UIElement`), `OnManipulationBoundaryFeedback` (унаследован от `UIElement`), `OnManipulationCompleted` (унаследован от `UIElement`), `OnManipulationDelta` (унаследован от `UIElement`), `OnManipulationInertiaStarting` (унаследован от `UIElement`), `OnManipulationStarted` (унаследован от `UIElement`), `OnManipulationStarting` (унаследован от `UIElement`), `OnMouseDoubleClick` (унаследован от `Control`), `OnMouseDown` (унаследован от `UIElement`), `OnMouseEnter` (унаследован от `UIElement`), `OnMouseLeave` (унаследован от `UIElement`), `OnMouseLeftButtonDown` (унаследован от `UIElement`), `OnMouseLeftButtonUp` (унаследован от `UIElement`), `OnMouseMove` (унаследован от `UIElement`), `OnMouseRightButtonDown` (унаследован от `UIElement`), `OnMouseRightButtonUp` (унаследован от `UIElement`), `OnMouseUp` (унаследован от `UIElement`), `OnMouseWheel` (унаследован от `UIElement`), `OnPreviewDragEnter` (унаследован от `UIElement`), `OnPreviewDragLeave` (унаследован от `UIElement`), `OnPreviewDragOver` (унаследован от `UIElement`),

[OnPreviewDrop](#) (унаследован от [UIElement](#)), [OnPreviewGiveFeedback](#) (унаследован от [UIElement](#)),
[OnPreviewGotKeyboardFocus](#) (унаследован от [UIElement](#)), [OnPreviewKeyDown](#) (унаследован от [UIElement](#)),
[OnPreviewKeyUp](#) (унаследован от [UIElement](#)), [OnPreviewLostKeyboardFocus](#) (унаследован от [UIElement](#)),
[OnPreviewMouseDoubleClick](#) (унаследован от [Control](#)), [OnPreviewMouseDown](#) (унаследован от [UIElement](#)),
[OnPreviewMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewMouseLeftButtonUp](#)
(унаследован от [UIElement](#)), [OnPreviewMouseMove](#) (унаследован от [UIElement](#)),
[OnPreviewMouseRightButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewMouseRightButtonUp](#)
(унаследован от [UIElement](#)), [OnPreviewMouseUp](#) (унаследован от [UIElement](#)), [OnPreviewMouseWheel](#)
(унаследован от [UIElement](#)), [OnPreviewQueryContinueDrag](#) (унаследован от [UIElement](#)),
[OnPreviewStylusButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewStylusButtonUp](#) (унаследован от
[UIElement](#)), [OnPreviewStylusDown](#) (унаследован от [UIElement](#)), [OnPreviewStylusInAirMove](#) (унаследован от
[UIElement](#)), [OnPreviewStylusInRange](#) (унаследован от [UIElement](#)), [OnPreviewStylusMove](#) (унаследован от
[UIElement](#)), [OnPreviewStylusOutOfRange](#) (унаследован от [UIElement](#)), [OnPreviewStylusSystemGesture](#)
(унаследован от [UIElement](#)), [OnPreviewStylusUp](#) (унаследован от [UIElement](#)), [OnPreviewTextInput](#)
(унаследован от [UIElement](#)), [OnPreviewTouchDown](#) (унаследован от [UIElement](#)), [OnPreviewTouchMove](#)
(унаследован от [UIElement](#)), [OnPreviewTouchUp](#) (унаследован от [UIElement](#)), [OnQueryContinueDrag](#)
(унаследован от [UIElement](#)), [OnQueryCursor](#) (унаследован от [UIElement](#)), [OnRender](#) (унаследован от [UIElement](#)),
[OnRenderSizeChanged](#) (унаследован от [FrameworkElement](#)), [OnStyleChanged](#) (унаследован от
[FrameworkElement](#)), [OnStylusButtonDown](#) (унаследован от [UIElement](#)), [OnStylusButtonUp](#) (унаследован от
[UIElement](#)), [OnStylusDown](#) (унаследован от [UIElement](#)), [OnStylusEnter](#) (унаследован от [UIElement](#)),
[OnStylusInAirMove](#) (унаследован от [UIElement](#)), [OnStylusInRange](#) (унаследован от [UIElement](#)), [OnStylusLeave](#)
(унаследован от [UIElement](#)), [OnStylusMove](#) (унаследован от [UIElement](#)), [OnStylusOutOfRange](#) (унаследован от
[UIElement](#)), [OnStylusSystemGesture](#) (унаследован от [UIElement](#)), [OnStylusUp](#) (унаследован от [UIElement](#)),
[OnTemplateChanged](#) (унаследован от [Control](#)), [OnTextInput](#) (унаследован от [UIElement](#)), [OnToolTipClosing](#)
(унаследован от [FrameworkElement](#)), [OnToolTipOpening](#) (унаследован от [FrameworkElement](#)), [OnTouchDown](#)
(унаследован от [UIElement](#)), [OnTouchEnter](#) (унаследован от [UIElement](#)), [OnTouchLeave](#) (унаследован от
[UIElement](#)), [OnTouchMove](#) (унаследован от [UIElement](#)), [OnTouchUp](#) (унаследован от [UIElement](#)),
[OnVisualChildrenChanged](#) (унаследован от [Visual](#)), [OnVisualParentChanged](#) (унаследован от
[FrameworkElement](#)), [ParentLayoutInvalidated](#) (унаследован от [FrameworkElement](#)), [PointFromScreen](#)
(унаследован от [Visual](#)), [PointToScreen](#) (унаследован от [Visual](#)), [PredictFocus](#) (унаследован от
[FrameworkElement](#)), [RaiseEvent](#) (унаследован от [UIElement](#)), [ReadLocalValue](#) (унаследован от
[DependencyObject](#)), [RegisterName](#) (унаследован от [FrameworkElement](#)), [ReleaseAllTouchCaptures](#)
(унаследован от [UIElement](#)), [ReleaseMouseCapture](#) (унаследован от [UIElement](#)), [ReleaseStylusCapture](#)
(унаследован от [UIElement](#)), [ReleaseTouchCapture](#) (унаследован от [UIElement](#)), [RemoveHandler](#) (унаследован
от [UIElement](#)), [RemoveLogicalChild](#) (унаследован от [FrameworkElement](#)), [RemoveVisualChild](#) (унаследован от
[Visual](#)), [Rotate₂₂₃](#), [SetBinding](#) (унаследован от [FrameworkElement](#)), [SetCurrentValue](#) (унаследован от
[DependencyObject](#)), [SetResourceReference](#) (унаследован от [FrameworkElement](#)), [SetValue₂₄₃](#) (унаследован от
[Knob](#)), [ShouldSerializeCommandBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeContent](#) (унаследован
от [ContentControl](#)), [ShouldSerializeInputBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeProperty](#)
(унаследован от [DependencyObject](#)), [ShouldSerializeResources](#) (унаследован от [FrameworkElement](#)),
[ShouldSerializeStyle](#) (унаследован от [FrameworkElement](#)), [ShouldSerializeTriggers](#) (унаследован от
[FrameworkElement](#)), [ToString](#) (унаследован от [Control](#)), [TransformToAncestor](#) (унаследован от [Visual](#)),
[TransformToDescendant](#) (унаследован от [Visual](#)), [TransformToVisual](#) (унаследован от [Visual](#)), [TranslatePoint](#)
(унаследован от [UIElement](#)), [TryFindResource](#) (унаследован от [FrameworkElement](#)), [UnregisterName](#)
(унаследован от [FrameworkElement](#)), [UpdateDefaultStyle](#) (унаследован от [FrameworkElement](#)),

[UpdateDisplayValue](#)²⁴³ (унаследован от [Knob](#)), [UpdateLayout](#) (унаследован от [UIElement](#)), [VerifyAccess](#) (унаследован от [DispatcherObject](#))

События

[ContextMenuClosing](#) (унаследован от [FrameworkElement](#)), [ContextMenuOpening](#) (унаследован от [FrameworkElement](#)), [DataContextChanged](#) (унаследован от [FrameworkElement](#)), [DragEnter](#) (унаследован от [UIElement](#)), [DragLeave](#) (унаследован от [UIElement](#)), [DragOver](#) (унаследован от [UIElement](#)), [Drop](#) (унаследован от [UIElement](#)), [FocusableChanged](#) (унаследован от [UIElement](#)), [GiveFeedback](#) (унаследован от [UIElement](#)), [GotFocus](#) (унаследован от [UIElement](#)), [GotKeyboardFocus](#) (унаследован от [UIElement](#)), [GotMouseCapture](#) (унаследован от [UIElement](#)), [GotStylusCapture](#) (унаследован от [UIElement](#)), [GotTouchCapture](#) (унаследован от [UIElement](#)), [Initialized](#) (унаследован от [FrameworkElement](#)), [IsEnabledChanged](#) (унаследован от [UIElement](#)), [IsHitTestVisibleChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusWithinChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusedChanged](#) (унаследован от [UIElement](#)), [IsMouseCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsMouseCapturedChanged](#) (унаследован от [UIElement](#)), [IsMouseDirectlyOverChanged](#) (унаследован от [UIElement](#)), [IsStylusCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsStylusCapturedChanged](#) (унаследован от [UIElement](#)), [IsStylusDirectlyOverChanged](#) (унаследован от [UIElement](#)), [IsVisibleChanged](#) (унаследован от [UIElement](#)), [KeyDown](#) (унаследован от [UIElement](#)), [KeyUp](#) (унаследован от [UIElement](#)), [LayoutUpdated](#) (унаследован от [UIElement](#)), [Loaded](#) (унаследован от [FrameworkElement](#)), [LostFocus](#) (унаследован от [UIElement](#)), [LostKeyboardFocus](#) (унаследован от [UIElement](#)), [LostMouseCapture](#) (унаследован от [UIElement](#)), [LostStylusCapture](#) (унаследован от [UIElement](#)), [LostTouchCapture](#) (унаследован от [UIElement](#)), [ManipulationBoundaryFeedback](#) (унаследован от [UIElement](#)), [ManipulationCompleted](#) (унаследован от [UIElement](#)), [ManipulationDelta](#) (унаследован от [UIElement](#)), [ManipulationInertiaStarting](#) (унаследован от [UIElement](#)), [ManipulationStarted](#) (унаследован от [UIElement](#)), [ManipulationStarting](#) (унаследован от [UIElement](#)), [MouseDoubleClick](#) (унаследован от [Control](#)), [MouseDown](#) (унаследован от [UIElement](#)), [MouseEnter](#) (унаследован от [UIElement](#)), [MouseLeave](#) (унаследован от [UIElement](#)), [MouseLeftButtonDown](#) (унаследован от [UIElement](#)), [MouseLeftButtonUp](#) (унаследован от [UIElement](#)), [MouseMove](#) (унаследован от [UIElement](#)), [MouseRightButtonDown](#) (унаследован от [UIElement](#)), [MouseRightButtonUp](#) (унаследован от [UIElement](#)), [MouseUp](#) (унаследован от [UIElement](#)), [MouseWheel](#) (унаследован от [UIElement](#)), [PreviewDragEnter](#) (унаследован от [UIElement](#)), [PreviewDragLeave](#) (унаследован от [UIElement](#)), [PreviewDragOver](#) (унаследован от [UIElement](#)), [PreviewDrop](#) (унаследован от [UIElement](#)), [PreviewGiveFeedback](#) (унаследован от [UIElement](#)), [PreviewGotKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewKeyDown](#) (унаследован от [UIElement](#)), [PreviewKeyUp](#) (унаследован от [UIElement](#)), [PreviewLostKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewMouseDoubleClick](#) (унаследован от [Control](#)), [PreviewMouseDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseMove](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseUp](#) (унаследован от [UIElement](#)), [PreviewMouseWheel](#) (унаследован от [UIElement](#)), [PreviewQueryContinueDrag](#) (унаследован от [UIElement](#)), [PreviewStylusButtonDown](#) (унаследован от [UIElement](#)), [PreviewStylusButtonUp](#) (унаследован от [UIElement](#)), [PreviewStylusDown](#) (унаследован от [UIElement](#)), [PreviewStylusInAirMove](#) (унаследован от [UIElement](#)), [PreviewStylusInRange](#) (унаследован от [UIElement](#)), [PreviewStylusMove](#) (унаследован от [UIElement](#)), [PreviewStylusOutOfRange](#) (унаследован от [UIElement](#)), [PreviewStylusSystemGesture](#) (унаследован от [UIElement](#)), [PreviewStylusUp](#) (унаследован от [UIElement](#)), [PreviewTextInput](#) (унаследован от [UIElement](#)), [PreviewTouchDown](#) (унаследован от [UIElement](#)), [PreviewTouchMove](#) (унаследован от [UIElement](#)), [PreviewTouchUp](#) (унаследован от [UIElement](#)), [PropertyChanged](#)²⁴³ (унаследован от [Knob](#)), [QueryContinueDrag](#) (унаследован от [UIElement](#)), [QueryCursor](#)

(унаследован от [UIElement](#)), [RequestBringIntoView](#) (унаследован от [FrameworkElement](#)), [SizeChanged](#) (унаследован от [FrameworkElement](#)), [SourceUpdated](#) (унаследован от [FrameworkElement](#)), [StylusButtonDown](#) (унаследован от [UIElement](#)), [StylusButtonUp](#) (унаследован от [UIElement](#)), [StylusDown](#) (унаследован от [UIElement](#)), [StylusEnter](#) (унаследован от [UIElement](#)), [StylusInAirMove](#) (унаследован от [UIElement](#)), [StylusInRange](#) (унаследован от [UIElement](#)), [StylusLeave](#) (унаследован от [UIElement](#)), [StylusMove](#) (унаследован от [UIElement](#)), [StylusOutOfRange](#) (унаследован от [UIElement](#)), [StylusSystemGesture](#) (унаследован от [UIElement](#)), [StylusUp](#) (унаследован от [UIElement](#)), [TargetUpdated](#) (унаследован от [FrameworkElement](#)), [TextInput](#) (унаследован от [UIElement](#)), [ToolTipClosing](#) (унаследован от [FrameworkElement](#)), [ToolTipOpening](#) (унаследован от [FrameworkElement](#)), [TouchDown](#) (унаследован от [UIElement](#)), [TouchEnter](#) (унаследован от [UIElement](#)), [TouchLeave](#) (унаследован от [UIElement](#)), [TouchMove](#) (унаследован от [UIElement](#)), [TouchUp](#) (унаследован от [UIElement](#)), [Unloaded](#) (унаследован от [FrameworkElement](#))

BipolarKnob - конструктор

Инициализирует новый объект типа BipolarKnob.

C#

```
public BipolarKnob()
```

См. также

Применяется: [BipolarKnob](#)₂₁₈

BipolarKnob.Rotate - метод

Обрабатывает вращение элементов.

C#

```
protected override void Rotate(  
    double normalizedValue  
)
```

Параметры

normalizedValue

См. также

Применяется: [BipolarKnob](#)₂₁₈

EditorView - класс

Представляет собой окно редактора плагина.

[System.Object](#)

[System.Windows.Threading.DispatcherObject](#)

[System.Windows.DependencyObject](#)

[System.Windows.Media.Visual](#)

```

System.Windows.UIElement
System.Windows.FrameworkElement
System.Windows.Controls.Control
System.Windows.Controls.ContentControl
System.Windows.Controls.UserControl
MultimodSynth.UI.EditorView

```

C#

```

public class EditorView : UserControl,
    IComponentConnector

```

Требования

Пространство имен: MultimodSynth.UI₂₁₈

Сборка: MultimodSynth (in MultimodSynth.dll)

Свойства

ActualHeight (унаследован от FrameworkElement), **ActualWidth** (унаследован от FrameworkElement), **AllowDrop** (унаследован от UIElement), **AreAnyTouchesCaptured** (унаследован от UIElement), **AreAnyTouchesCapturedWithin** (унаследован от UIElement), **AreAnyTouchesDirectlyOver** (унаследован от UIElement), **AreAnyTouchesOver** (унаследован от UIElement), **Background** (унаследован от Control), **BindingGroup** (унаследован от FrameworkElement), **BitmapEffect** (унаследован от UIElement), **BitmapEffectInput** (унаследован от UIElement), **BorderBrush** (унаследован от Control), **BorderThickness** (унаследован от Control), **CacheMode** (унаследован от UIElement), **Clip** (унаследован от UIElement), **ClipToBounds** (унаследован от UIElement), **CommandBindings** (унаследован от UIElement), **Content** (унаследован от ContentControl), **ContentStringFormat** (унаследован от ContentControl), **ContentTemplate** (унаследован от ContentControl), **ContentTemplateSelector** (унаследован от ContentControl), **ContextMenu** (унаследован от FrameworkElement), **Cursor** (унаследован от FrameworkElement), **DataContext** (унаследован от FrameworkElement), **DependencyObjectType** (унаследован от DependencyObject), **DesiredSize** (унаследован от UIElement), **Dispatcher** (унаследован от DispatcherObject), **Effect** (унаследован от UIElement), **FlowDirection** (унаследован от FrameworkElement), **FocusVisualStyle** (унаследован от FrameworkElement), **Focusable** (унаследован от UIElement), **FontFamily** (унаследован от Control), **FontSize** (унаследован от Control), **FontStretch** (унаследован от Control), **FontStyle** (унаследован от Control), **FontWeight** (унаследован от Control), **ForceCursor** (унаследован от FrameworkElement), **Foreground** (унаследован от Control), **HasAnimatedProperties** (унаследован от UIElement), **HasContent** (унаследован от ContentControl), **Height** (унаследован от FrameworkElement), **HorizontalAlignment** (унаследован от FrameworkElement), **HorizontalContentAlignment** (унаследован от Control), **InputBindings** (унаследован от UIElement), **InputScope** (унаследован от FrameworkElement), **IsArrangeValid** (унаследован от UIElement), **.IsEnabled** (унаследован от UIElement), **IsFocused** (унаследован от UIElement), **IsHitTestVisible** (унаследован от UIElement), **IsInitialized** (унаследован от FrameworkElement), **IsInputMethodEnabled** (унаследован от UIElement), **IsKeyboardFocusWithin** (унаследован от UIElement), **IsKeyboardFocused** (унаследован от UIElement), **IsLoaded** (унаследован от FrameworkElement), **IsManipulationEnabled** (унаследован от UIElement), **IsMeasureValid** (унаследован от UIElement), **IsMouseCaptureWithin** (унаследован от UIElement), **IsMouseCaptured** (унаследован от UIElement), **IsMouseDirectlyOver** (унаследован от UIElement), **IsMouseOver** (унаследован от UIElement), **IsSealed** (унаследован от DependencyObject), **IsStylusCaptureWithin** (унаследован от UIElement), **IsStylusCaptured** (унаследован от UIElement), **IsStylusDirectlyOver** (унаследован от UIElement), **IsStylusOver** (унаследован от UIElement)

[UIElement](#), [IsTabStop](#) (унаследован от [Control](#)), [IsVisible](#) (унаследован от [UIElement](#)), [Language](#) (унаследован от [FrameworkElement](#)), [LayoutTransform](#) (унаследован от [FrameworkElement](#)), [Margin](#) (унаследован от [FrameworkElement](#)), [MaxHeight](#) (унаследован от [FrameworkElement](#)), [MaxWidth](#) (унаследован от [FrameworkElement](#)), [MinHeight](#) (унаследован от [FrameworkElement](#)), [MinWidth](#) (унаследован от [FrameworkElement](#)), [Name](#) (унаследован от [FrameworkElement](#)), [Opacity](#) (унаследован от [UIElement](#)), [OpacityMask](#) (унаследован от [UIElement](#)), [OverridesDefaultStyle](#) (унаследован от [FrameworkElement](#)), [Padding](#) (унаследован от [Control](#)), [Parent](#) (унаследован от [FrameworkElement](#)), [PersistId](#) (унаследован от [UIElement](#)), [RenderSize](#) (унаследован от [UIElement](#)), [RenderTransform](#) (унаследован от [UIElement](#)), [RenderTransformOrigin](#) (унаследован от [UIElement](#)), [Resources](#) (унаследован от [FrameworkElement](#)), [SnapsToDevicePixels](#) (унаследован от [UIElement](#)), [Style](#) (унаследован от [FrameworkElement](#)), [TabIndex](#) (унаследован от [Control](#)), [Tag](#) (унаследован от [FrameworkElement](#)), [Template](#) (унаследован от [Control](#)), [TemplatedParent](#) (унаследован от [FrameworkElement](#)), [ToolTip](#) (унаследован от [FrameworkElement](#)), [TouchesCaptured](#) (унаследован от [UIElement](#)), [TouchesCapturedWithin](#) (унаследован от [UIElement](#)), [TouchesDirectlyOver](#) (унаследован от [UIElement](#)), [TouchesOver](#) (унаследован от [UIElement](#)), [Triggers](#) (унаследован от [FrameworkElement](#)), [Uid](#) (унаследован от [UIElement](#)), [UseLayoutRounding](#) (унаследован от [FrameworkElement](#)), [VerticalAlignment](#) (унаследован от [FrameworkElement](#)), [VerticalContentAlignment](#) (унаследован от [Control](#)), [Visibility](#) (унаследован от [UIElement](#)), [Width](#) (унаследован от [FrameworkElement](#))

Методы

[AddChild](#) (унаследован от [ContentControl](#)), [AddHandler](#) (унаследован от [UIElement](#)), [AddLogicalChild](#) (унаследован от [FrameworkElement](#)), [AddText](#) (унаследован от [ContentControl](#)), [AddToEventRoute](#) (унаследован от [UIElement](#)), [AddVisualChild](#) (унаследован от [Visual](#)), [ApplyAnimationClock](#) (унаследован от [UIElement](#)), [ApplyTemplate](#) (унаследован от [FrameworkElement](#)), [Arrange](#) (унаследован от [UIElement](#)), [ArrangeCore](#) (унаследован от [FrameworkElement](#)), [ArrangeOverride](#) (унаследован от [Control](#)), [BeginAnimation](#) (унаследован от [UIElement](#)), [BeginInit](#) (унаследован от [FrameworkElement](#)), [BeginStoryboard](#) (унаследован от [FrameworkElement](#)), [BindDelay](#)²²⁹, [BindDistortion](#)²²⁹, [BindFilter](#)²²⁹, [BindKeyboard](#)²³⁰, [BindMasterSettings](#)²³⁰, [BindOscA](#)²³⁰, [BindOscB](#)²³¹, [BindParameters](#)²³¹, [BindToPlugin](#)²³², [BringIntoView](#) (унаследован от [FrameworkElement](#)), [CaptureMouse](#) (унаследован от [UIElement](#)), [CaptureStylus](#) (унаследован от [UIElement](#)), [CaptureTouch](#) (унаследован от [UIElement](#)), [CheckAccess](#) (унаследован от [DispatcherObject](#)), [ClearValue](#) (унаследован от [DependencyObject](#)), [CoerceValue](#) (унаследован от [DependencyObject](#)), [EndInit](#) (унаследован от [FrameworkElement](#)), [Equals](#) (унаследован от [DependencyObject](#)), [Finalize](#) (унаследован от [Object](#)), [FindCommonVisualAncestor](#) (унаследован от [Visual](#)), [FindName](#) (унаследован от [FrameworkElement](#)), [FindResource](#) (унаследован от [FrameworkElement](#)), [Focus](#) (унаследован от [UIElement](#)), [GetAnimationBaseValue](#) (унаследован от [UIElement](#)), [GetBindingExpression](#) (унаследован от [FrameworkElement](#)), [GetHashCode](#) (унаследован от [DependencyObject](#)), [GetLayoutClip](#) (унаследован от [FrameworkElement](#)), [GetLocalValueEnumerator](#) (унаследован от [DependencyObject](#)), [GetTemplateChild](#) (унаследован от [FrameworkElement](#)), [GetType](#) (унаследован от [Object](#)), [GetUIParentCore](#) (унаследован от [FrameworkElement](#)), [GetValue](#) (унаследован от [DependencyObject](#)), [GetVisualChild](#) (унаследован от [FrameworkElement](#)), [HitTestCore](#) (унаследован от [UIElement](#)), [InputHitTest](#) (унаследован от [UIElement](#)), [InvalidateArrange](#) (унаследован от [UIElement](#)), [InvalidateMeasure](#) (унаследован от [UIElement](#)), [InvalidateProperty](#) (унаследован от [DependencyObject](#)), [InvalidateVisual](#) (унаследован от [UIElement](#)), [IsAncestorOf](#) (унаследован от [Visual](#)), [IsDescendantOf](#) (унаследован от [Visual](#)), [Measure](#) (унаследован от [UIElement](#)), [MeasureCore](#) (унаследован от [FrameworkElement](#)), [MeasureOverride](#) (унаследован от [Control](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [MoveFocus](#) (унаследован от [FrameworkElement](#)), [OnAccessKey](#) (унаследован от [UIElement](#)), [OnApplyTemplate](#) (унаследован от [FrameworkElement](#)),

`OnChildDesiredSizeChanged` (унаследован от `UIElement`), `OnContentChanged` (унаследован от `ContentControl`), `OnContentStringFormatChanged` (унаследован от `ContentControl`), `OnContentTemplateChanged` (унаследован от `ContentControl`), `OnContentTemplateSelectorChanged` (унаследован от `ContentControl`), `OnContextMenuClosing` (унаследован от `FrameworkElement`), `OnContextMenuOpening` (унаследован от `FrameworkElement`), `OnCreateAutomationPeer` (унаследован от `UserControl`), `OnDpiChanged` (унаследован от `Visual`), `OnDragEnter` (унаследован от `UIElement`), `OnDragLeave` (унаследован от `UIElement`), `OnDragOver` (унаследован от `UIElement`), `OnDrop` (унаследован от `UIElement`), `OnGiveFeedback` (унаследован от `UIElement`), `OnGotFocus` (унаследован от `FrameworkElement`), `OnGotKeyboardFocus` (унаследован от `UIElement`), `OnGotMouseCapture` (унаследован от `UIElement`), `OnGotStylusCapture` (унаследован от `UIElement`), `OnGotTouchCapture` (унаследован от `UIElement`), `OnInitialized` (унаследован от `FrameworkElement`), `OnIsKeyboardFocusWithinChanged` (унаследован от `UIElement`), `OnIsKeyboardFocusedChanged` (унаследован от `UIElement`), `OnIsMouseCaptureWithinChanged` (унаследован от `UIElement`), `OnIsMouseCapturedChanged` (унаследован от `UIElement`), `OnIsMouseDirectlyOverChanged` (унаследован от `UIElement`), `OnIsStylusCaptureWithinChanged` (унаследован от `UIElement`), `OnIsStylusCapturedChanged` (унаследован от `UIElement`), `OnIsStylusDirectlyOverChanged` (унаследован от `UIElement`), `OnKeyDown` (унаследован от `UIElement`), `OnKeyUp` (унаследован от `UIElement`), `OnLostFocus` (унаследован от `UIElement`), `OnLostKeyboardFocus` (унаследован от `UIElement`), `OnLostMouseCapture` (унаследован от `UIElement`), `OnLostStylusCapture` (унаследован от `UIElement`), `OnLostTouchCapture` (унаследован от `UIElement`), `OnManipulationBoundaryFeedback` (унаследован от `UIElement`), `OnManipulationCompleted` (унаследован от `UIElement`), `OnManipulationDelta` (унаследован от `UIElement`), `OnManipulationInertiaStarting` (унаследован от `UIElement`), `OnManipulationStarted` (унаследован от `UIElement`), `OnManipulationStarting` (унаследован от `UIElement`), `OnMouseDoubleClick` (унаследован от `Control`), `OnMouseDown` (унаследован от `UIElement`), `OnMouseEnter` (унаследован от `UIElement`), `OnMouseLeave` (унаследован от `UIElement`), `OnMouseLeftButtonDown` (унаследован от `UIElement`), `OnMouseLeftButtonUp` (унаследован от `UIElement`), `OnMouseMove` (унаследован от `UIElement`), `OnMouseRightButtonDown` (унаследован от `UIElement`), `OnMouseRightButtonUp` (унаследован от `UIElement`), `OnMouseWheel` (унаследован от `UIElement`), `OnPreviewDragEnter` (унаследован от `UIElement`), `OnPreviewDragLeave` (унаследован от `UIElement`), `OnPreviewDragOver` (унаследован от `UIElement`), `OnPreviewDrop` (унаследован от `UIElement`), `OnPreviewGiveFeedback` (унаследован от `UIElement`), `OnPreviewGotKeyboardFocus` (унаследован от `UIElement`), `OnPreviewKeyDown` (унаследован от `UIElement`), `OnPreviewKeyUp` (унаследован от `UIElement`), `OnPreviewLostKeyboardFocus` (унаследован от `UIElement`), `OnPreviewMouseDoubleClick` (унаследован от `Control`), `OnPreviewMouseDown` (унаследован от `UIElement`), `OnPreviewMouseLeftButtonDown` (унаследован от `UIElement`), `OnPreviewMouseLeftButtonUp` (унаследован от `UIElement`), `OnPreviewMouseMove` (унаследован от `UIElement`), `OnPreviewMouseRightButtonDown` (унаследован от `UIElement`), `OnPreviewMouseRightButtonUp` (унаследован от `UIElement`), `OnPreviewMouseUp` (унаследован от `UIElement`), `OnPreviewMouseWheel` (унаследован от `UIElement`), `OnPreviewQueryContinueDrag` (унаследован от `UIElement`), `OnPreviewStylusButtonDown` (унаследован от `UIElement`), `OnPreviewStylusButtonUp` (унаследован от `UIElement`), `OnPreviewStylusDown` (унаследован от `UIElement`), `OnPreviewStylusInAirMove` (унаследован от `UIElement`), `OnPreviewStylusInRange` (унаследован от `UIElement`), `OnPreviewStylusMove` (унаследован от `UIElement`), `OnPreviewStylusOutOfRange` (унаследован от `UIElement`), `OnPreviewStylusSystemGesture` (унаследован от `UIElement`), `OnPreviewStylusUp` (унаследован от `UIElement`), `OnPreviewTextInput` (унаследован от `UIElement`), `OnPreviewTouchDown` (унаследован от `UIElement`), `OnPreviewTouchMove` (унаследован от `UIElement`), `OnPreviewTouchUp` (унаследован от `UIElement`), `OnPropertyChanged`

(унаследован от [FrameworkElement](#)), [OnQueryContinueDrag](#) (унаследован от [UIElement](#)), [OnQueryCursor](#) (унаследован от [UIElement](#)), [OnRender](#) (унаследован от [UIElement](#)), [OnRenderSizeChanged](#) (унаследован от [FrameworkElement](#)), [OnStyleChanged](#) (унаследован от [FrameworkElement](#)), [OnStylusButtonDown](#) (унаследован от [UIElement](#)), [OnStylusButtonUp](#) (унаследован от [UIElement](#)), [OnStylusDown](#) (унаследован от [UIElement](#)), [OnStylusEnter](#) (унаследован от [UIElement](#)), [OnStylusInAirMove](#) (унаследован от [UIElement](#)), [OnStylusInRange](#) (унаследован от [UIElement](#)), [OnStylusLeave](#) (унаследован от [UIElement](#)), [OnStylusMove](#) (унаследован от [UIElement](#)), [OnStylusOutOfRange](#) (унаследован от [UIElement](#)), [OnStylusSystemGesture](#) (унаследован от [UIElement](#)), [OnStylusUp](#) (унаследован от [UIElement](#)), [OnTemplateChanged](#) (унаследован от [Control](#)), [OnTextInput](#) (унаследован от [UIElement](#)), [OnToolTipClosing](#) (унаследован от [FrameworkElement](#)), [OnToolTipOpening](#) (унаследован от [FrameworkElement](#)), [OnTouchDown](#) (унаследован от [UIElement](#)), [OnTouchEnter](#) (унаследован от [UIElement](#)), [OnTouchLeave](#) (унаследован от [UIElement](#)), [OnTouchMove](#) (унаследован от [UIElement](#)), [OnTouchUp](#) (унаследован от [UIElement](#)), [OnVisualChildrenChanged](#) (унаследован от [Visual](#)), [OnVisualParentChanged](#) (унаследован от [FrameworkElement](#)), [OpenButton_Click](#)₂₃₂, [ParentLayoutInvalidated](#) (унаследован от [FrameworkElement](#)), [PointFromScreen](#) (унаследован от [Visual](#)), [PointToScreen](#) (унаследован от [Visual](#)), [PredictFocus](#) (унаследован от [FrameworkElement](#)), [RaiseEvent](#) (унаследован от [UIElement](#)), [ReadLocalValue](#) (унаследован от [DependencyObject](#)), [RegisterName](#) (унаследован от [FrameworkElement](#)), [ReleaseAllTouchCaptures](#) (унаследован от [UIElement](#)), [ReleaseMouseCapture](#) (унаследован от [UIElement](#)), [ReleaseStylusCapture](#) (унаследован от [UIElement](#)), [ReleaseTouchCapture](#) (унаследован от [UIElement](#)), [RemoveHandler](#) (унаследован от [UIElement](#)), [RemoveLogicalChild](#) (унаследован от [FrameworkElement](#)), [RemoveVisualChild](#) (унаследован от [Visual](#)), [SaveButton_Click](#)₂₃₂, [SetBinding](#) (унаследован от [FrameworkElement](#)), [SetCurrentValue](#) (унаследован от [DependencyObject](#)), [SetResourceReference](#) (унаследован от [FrameworkElement](#)), [SetValue](#) (унаследован от [DependencyObject](#)), [ShouldSerializeCommandBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeContent](#) (унаследован от [ContentControl](#)), [ShouldSerializeInputBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeProperty](#) (унаследован от [DependencyObject](#)), [ShouldSerializeResources](#) (унаследован от [FrameworkElement](#)), [ShouldSerializeStyle](#) (унаследован от [FrameworkElement](#)), [ShouldSerializeTriggers](#) (унаследован от [FrameworkElement](#)), [ToString](#) (унаследован от [Control](#)), [TransformToAncestor](#) (унаследован от [Visual](#)), [TransformToDescendant](#) (унаследован от [Visual](#)), [TransformToVisual](#) (унаследован от [Visual](#)), [TranslatePoint](#) (унаследован от [UIElement](#)), [TryFindResource](#) (унаследован от [FrameworkElement](#)), [UnregisterName](#) (унаследован от [FrameworkElement](#)), [UpdateDefaultStyle](#) (унаследован от [FrameworkElement](#)), [UpdateLayout](#) (унаследован от [UIElement](#)), [VerifyAccess](#) (унаследован от [DispatcherObject](#))

События

[ContextMenuClosing](#) (унаследован от [FrameworkElement](#)), [ContextMenuOpening](#) (унаследован от [FrameworkElement](#)), [DataContextChanged](#) (унаследован от [FrameworkElement](#)), [DragEnter](#) (унаследован от [UIElement](#)), [DragLeave](#) (унаследован от [UIElement](#)), [DragOver](#) (унаследован от [UIElement](#)), [Drop](#) (унаследован от [UIElement](#)), [FocusableChanged](#) (унаследован от [UIElement](#)), [GiveFeedback](#) (унаследован от [UIElement](#)), [GotFocus](#) (унаследован от [UIElement](#)), [GotKeyboardFocus](#) (унаследован от [UIElement](#)), [GotMouseCapture](#) (унаследован от [UIElement](#)), [GotStylusCapture](#) (унаследован от [UIElement](#)), [GotTouchCapture](#) (унаследован от [UIElement](#)), [Initialized](#) (унаследован от [FrameworkElement](#)), [IsEnabledChanged](#) (унаследован от [UIElement](#)), [IsHitTestVisibleChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusWithinChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusedChanged](#) (унаследован от [UIElement](#)), [IsMouseCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsMouseCapturedChanged](#) (унаследован от [UIElement](#)), [IsMouseDirectlyOverChanged](#) (унаследован от [UIElement](#)), [IsStylusCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsStylusCapturedChanged](#) (унаследован от [UIElement](#)), [IsStylusDirectlyOverChanged](#)

(унаследован от [UIElement](#)), [IsVisibleChanged](#) (унаследован от [UIElement](#)), [KeyDown](#) (унаследован от [UIElement](#)), [KeyUp](#) (унаследован от [UIElement](#)), [LayoutUpdated](#) (унаследован от [UIElement](#)), [Loaded](#) (унаследован от [FrameworkElement](#)), [LostFocus](#) (унаследован от [UIElement](#)), [LostKeyboardFocus](#) (унаследован от [UIElement](#)), [LostMouseCapture](#) (унаследован от [UIElement](#)), [LostStylusCapture](#) (унаследован от [UIElement](#)), [LostTouchCapture](#) (унаследован от [UIElement](#)), [ManipulationBoundaryFeedback](#) (унаследован от [UIElement](#)), [ManipulationCompleted](#) (унаследован от [UIElement](#)), [ManipulationDelta](#) (унаследован от [UIElement](#)), [ManipulationInertiaStarting](#) (унаследован от [UIElement](#)), [ManipulationStarted](#) (унаследован от [UIElement](#)), [ManipulationStarting](#) (унаследован от [UIElement](#)), [MouseDoubleClick](#) (унаследован от [Control](#)), [MouseDown](#) (унаследован от [UIElement](#)), [MouseEnter](#) (унаследован от [UIElement](#)), [MouseLeave](#) (унаследован от [UIElement](#)), [MouseLeftButtonDown](#) (унаследован от [UIElement](#)), [MouseLeftButtonUp](#) (унаследован от [UIElement](#)), [MouseMove](#) (унаследован от [UIElement](#)), [MouseRightButtonDown](#) (унаследован от [UIElement](#)), [MouseRightButtonUp](#) (унаследован от [UIElement](#)), [MouseUp](#) (унаследован от [UIElement](#)), [MouseWheel](#) (унаследован от [UIElement](#)), [PreviewDragEnter](#) (унаследован от [UIElement](#)), [PreviewDragLeave](#) (унаследован от [UIElement](#)), [PreviewDragOver](#) (унаследован от [UIElement](#)), [PreviewDrop](#) (унаследован от [UIElement](#)), [PreviewGiveFeedback](#) (унаследован от [UIElement](#)), [PreviewGotKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewKeyDown](#) (унаследован от [UIElement](#)), [PreviewKeyUp](#) (унаследован от [UIElement](#)), [PreviewLostKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewMouseDoubleClick](#) (унаследован от [Control](#)), [PreviewMouseDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseMove](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseUp](#) (унаследован от [UIElement](#)), [PreviewMouseWheel](#) (унаследован от [UIElement](#)), [PreviewQueryContinueDrag](#) (унаследован от [UIElement](#)), [PreviewStylusButtonDown](#) (унаследован от [UIElement](#)), [PreviewStylusButtonUp](#) (унаследован от [UIElement](#)), [PreviewStylusDown](#) (унаследован от [UIElement](#)), [PreviewStylusInAirMove](#) (унаследован от [UIElement](#)), [PreviewStylusInRange](#) (унаследован от [UIElement](#)), [PreviewStylusMove](#) (унаследован от [UIElement](#)), [PreviewStylusOutOfRange](#) (унаследован от [UIElement](#)), [PreviewStylusSystemGesture](#) (унаследован от [UIElement](#)), [PreviewStylusUp](#) (унаследован от [UIElement](#)), [PreviewTextInput](#) (унаследован от [UIElement](#)), [PreviewTouchDown](#) (унаследован от [UIElement](#)), [PreviewTouchMove](#) (унаследован от [UIElement](#)), [PreviewTouchUp](#) (унаследован от [UIElement](#)), [QueryContinueDrag](#) (унаследован от [UIElement](#)), [QueryCursor](#) (унаследован от [UIElement](#)), [RequestBringIntoView](#) (унаследован от [FrameworkElement](#)), [SizeChanged](#) (унаследован от [FrameworkElement](#)), [SourceUpdated](#) (унаследован от [FrameworkElement](#)), [StylusButtonDown](#) (унаследован от [UIElement](#)), [StylusButtonUp](#) (унаследован от [UIElement](#)), [StylusDown](#) (унаследован от [UIElement](#)), [StylusEnter](#) (унаследован от [UIElement](#)), [StylusInAirMove](#) (унаследован от [UIElement](#)), [StylusInRange](#) (унаследован от [UIElement](#)), [StylusLeave](#) (унаследован от [UIElement](#)), [StylusMove](#) (унаследован от [UIElement](#)), [StylusOutOfRange](#) (унаследован от [UIElement](#)), [StylusSystemGesture](#) (унаследован от [UIElement](#)), [StylusUp](#) (унаследован от [UIElement](#)), [TargetUpdated](#) (унаследован от [FrameworkElement](#)), [TextInput](#) (унаследован от [UIElement](#)), [ToolTipClosing](#) (унаследован от [FrameworkElement](#)), [ToolTipOpening](#) (унаследован от [FrameworkElement](#)), [TouchDown](#) (унаследован от [UIElement](#)), [TouchEnter](#) (унаследован от [UIElement](#)), [TouchLeave](#) (унаследован от [UIElement](#)), [TouchMove](#) (унаследован от [UIElement](#)), [TouchUp](#) (унаследован от [UIElement](#)), [Unloaded](#) (унаследован от [FrameworkElement](#))

Поля

[plugin](#)²³³

EditorView.BindDelay - метод

Метод, привязывающий параметры эффекта дилэй плагина к редактору.

C#

```
private void BindDelay(  
    DelayManager delay  
)
```

Параметры

delay

См. также

Применяется: [EditorView](#)²²³

EditorView.BindDistortion - метод

Метод, привязывающий параметры эффекта дисторшн плагина к редактору.

C#

```
private void BindDistortion(  
    DistortionManager distortion  
)
```

Параметры

distortion

См. также

Применяется: [EditorView](#)²²³

EditorView.BindFilter - метод

Метод, привязывающий параметры фильтра плагина к редактору.

C#

```
private void BindFilter(  
    FiltersManager filter,  
    EnvelopesManager env  
)
```

Параметры

filter

env

См. также

Применяется: [EditorView](#)₂₂₃

EditorView.BindKeyboard - метод

Метод, привязывающий плагин к клавиатуре редактора.

C#

```
private void BindKeyboard(  
    MidiProcessor midiProcessor  
)
```

Параметры

midiProcessor

См. также

Применяется: [EditorView](#)₂₂₃

EditorView.BindMasterSettings - метод

Метод, привязывающий некие другие параметры плагина к редактору.

C#

```
private void BindMasterSettings(  
    Routing routing  
)
```

Параметры

routing

См. также

Применяется: [EditorView](#)₂₂₃

EditorView.BindOscA - метод

Метод, привязывающий параметры первого осциллятора плагина к редактору.

C#

```
private void BindOscA(  
    OscillatorsManager oscA,  
    EnvelopesManager envA  
)
```

Параметры*oscA**envA***См. также**Применяется: [EditorView](#)₂₂₃

EditorView.BindOscB - метод

Метод, привязывающий параметры второго осциллятора плагина к редактору.

C#

```
private void BindOscB(
    OscillatorsManager oscB,
    EnvelopesManager envB
)
```

Параметры*oscB**envB***См. также**Применяется: [EditorView](#)₂₂₃

EditorView.BindParameters - метод

Метод, привязывающий параметры плагина к редактору.

C#

```
private void BindParameters(
    Routing routing
)
```

Параметры*routing***См. также**Применяется: [EditorView](#)₂₂₃

EditorView.BindToPlugin - метод

Метод, привязывающий параметры плагина к редактору.

C#

```
internal void BindToPlugin(  
    Plugin plugin  
)
```

Параметры

plugin

См. также

Применяется: [EditorView](#)₂₂₃

EditorView.OpenButton_Click - метод

Обработчик события нажатия на кнопку "Открыть".

C#

```
private void OpenButton_Click(  
    object sender,  
    RoutedEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [EditorView](#)₂₂₃

EditorView.SaveButton_Click - метод

Обработчик события нажатия на кнопку "Сохранить".

C#

```
private void SaveButton_Click(  
    object sender,  
    RoutedEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [EditorView](#)²²³

plugin - поле

Ссылка на плагин, связанный с этим редактором.

C#

```
private Plugin plugin
```

См. также

Применяется: [EditorView](#)²²³

Knob - класс

Interaction logic for Knob.xaml

System.Object
System.Windows.Threading.DispatcherObject
System.Windows.DependencyObject
System.Windows.Media.Visual
System.Windows.UIElement
System.Windows.FrameworkElement
System.Windows.Controls.Control
System.Windows.Controls.ContentControl
System.Windows.Controls.UserControl
MultimodSynth.UI.Knob
MultimodSynth.UI.BipolarKnob²¹⁸

C#

```
public class Knob : UserControl,  
INotifyPropertyChanged,  
IComponentConnector
```

Требования

Пространство имен: [MultimodSynth.UI](#)²¹⁸

Сборка: MultimodSynth (in MultimodSynth.dll)

Конструкторы

[Knob](#)²³⁸

Свойства

233

Generated with unregistered version of VSdocman

Your own footer text will only be shown in registered version.

[ActualHeight](#) (унаследован от [FrameworkElement](#)), [ActualWidth](#) (унаследован от [FrameworkElement](#)), [AllowDrop](#) (унаследован от [UIElement](#)), [AreAnyTouchesCaptured](#) (унаследован от [UIElement](#)), [AreAnyTouchesCapturedWithin](#) (унаследован от [UIElement](#)), [AreAnyTouchesDirectlyOver](#) (унаследован от [UIElement](#)), [AreAnyTouchesOver](#) (унаследован от [UIElement](#)), [Background](#) (унаследован от [Control](#)), [BindingGroup](#) (унаследован от [FrameworkElement](#)), [BitmapEffect](#) (унаследован от [UIElement](#)), [BitmapEffectInput](#) (унаследован от [UIElement](#)), [BorderBrush](#) (унаследован от [Control](#)), [BorderThickness](#) (унаследован от [Control](#)), [CacheMode](#) (унаследован от [UIElement](#)), [Clip](#) (унаследован от [UIElement](#)), [ClipToBounds](#) (унаследован от [UIElement](#)), [CommandBindings](#) (унаследован от [UIElement](#)), [Content](#) (унаследован от [ContentControl](#)), [ContentStringFormat](#) (унаследован от [ContentControl](#)), [ContentTemplate](#) (унаследован от [ContentControl](#)), [ContentTemplateSelector](#) (унаследован от [ContentControl](#)), [ContextMenu](#) (унаследован от [FrameworkElement](#)), [Cursor](#) (унаследован от [FrameworkElement](#)), [DataContext](#) (унаследован от [FrameworkElement](#)), [DependencyObjectType](#) (унаследован от [DependencyObject](#)), [DesiredSize](#) (унаследован от [UIElement](#)), [Dispatcher](#) (унаследован от [DispatcherObject](#)), [DisplayValue₂₃₈](#), [DisplayValueConverter₂₃₉](#), [Effect](#) (унаследован от [UIElement](#)), [FlowDirection](#) (унаследован от [FrameworkElement](#)), [FocusVisualStyle](#) (унаследован от [FrameworkElement](#)), [Focusable](#) (унаследован от [UIElement](#)), [FontFamily](#) (унаследован от [Control](#)), [FontSize](#) (унаследован от [Control](#)), [FontStretch](#) (унаследован от [Control](#)), [FontStyle](#) (унаследован от [Control](#)), [FontWeight](#) (унаследован от [Control](#)), [ForceCursor](#) (унаследован от [FrameworkElement](#)), [Foreground](#) (унаследован от [Control](#)), [HasAnimatedProperties](#) (унаследован от [UIElement](#)), [HasContent](#) (унаследован от [ContentControl](#)), [Height](#) (унаследован от [FrameworkElement](#)), [HorizontalAlignment](#) (унаследован от [FrameworkElement](#)), [HorizontalContentAlignment](#) (унаследован от [Control](#)), [InputBindings](#) (унаследован от [UIElement](#)), [InputScope](#) (унаследован от [FrameworkElement](#)), [IsArrangeValid](#) (унаследован от [UIElement](#)), [IsEnabled](#) (унаследован от [UIElement](#)), [IsFocused](#) (унаследован от [UIElement](#)), [IsHitTestVisible](#) (унаследован от [UIElement](#)), [IsInitialized](#) (унаследован от [FrameworkElement](#)), [IsInputMethodEnabled](#) (унаследован от [UIElement](#)), [IsKeyboardFocusWithin](#) (унаследован от [UIElement](#)), [IsKeyboardFocused](#) (унаследован от [UIElement](#)), [IsLoaded](#) (унаследован от [FrameworkElement](#)), [IsManipulationEnabled](#) (унаследован от [UIElement](#)), [IsMeasureValid](#) (унаследован от [UIElement](#)), [IsMouseCaptureWithin](#) (унаследован от [UIElement](#)), [IsMouseCaptured](#) (унаследован от [UIElement](#)), [IsMouseDirectlyOver](#) (унаследован от [UIElement](#)), [IsMouseOver](#) (унаследован от [UIElement](#)), [IsSealed](#) (унаследован от [DependencyObject](#)), [IsStylusCaptureWithin](#) (унаследован от [UIElement](#)), [IsStylusCaptured](#) (унаследован от [UIElement](#)), [IsStylusDirectlyOver](#) (унаследован от [UIElement](#)), [IsStylusOver](#) (унаследован от [UIElement](#)), [IsTabStop](#) (унаследован от [Control](#)), [IsVisible](#) (унаследован от [UIElement](#)), [Language](#) (унаследован от [FrameworkElement](#)), [LayoutTransform](#) (унаследован от [FrameworkElement](#)), [Margin](#) (унаследован от [FrameworkElement](#)), [MaxHeight](#) (унаследован от [FrameworkElement](#)), [MaxWidth](#) (унаследован от [FrameworkElement](#)), [MinHeight](#) (унаследован от [FrameworkElement](#)), [MinWidth](#) (унаследован от [FrameworkElement](#)), [Name](#) (унаследован от [FrameworkElement](#)), [Opacity](#) (унаследован от [UIElement](#)), [OpacityMask](#) (унаследован от [UIElement](#)), [OverridesDefaultStyle](#) (унаследован от [FrameworkElement](#)), [Padding](#) (унаследован от [Control](#)), [Parent](#) (унаследован от [FrameworkElement](#)), [PersistId](#) (унаследован от [UIElement](#)), [RenderSize](#) (унаследован от [UIElement](#)), [RenderTransform](#) (унаследован от [UIElement](#)), [RenderTransformOrigin](#) (унаследован от [UIElement](#)), [Resources](#) (унаследован от [FrameworkElement](#)), [SnapsToDevicePixels](#) (унаследован от [UIElement](#)), [Stroke₂₃₉](#), [Style](#) (унаследован от [FrameworkElement](#)), [TabIndex](#) (унаследован от [Control](#)), [Tag](#) (унаследован от [FrameworkElement](#)), [Template](#) (унаследован от [Control](#)), [TemplatedParent](#) (унаследован от [FrameworkElement](#)), [ToolTip](#) (унаследован от [FrameworkElement](#)), [TouchesCaptured](#) (унаследован от [UIElement](#)), [TouchesCapturedWithin](#) (унаследован от [UIElement](#)), [TouchesDirectlyOver](#) (унаследован от [UIElement](#)), [TouchesOver](#) (унаследован от [UIElement](#)), [Triggers](#) (унаследован от [FrameworkElement](#)), [Uid](#) (унаследован от [UIElement](#)), [UseLayoutRounding](#) (унаследован от [FrameworkElement](#)), [VerticalAlignment](#) (унаследован от [FrameworkElement](#)),

[VerticalContentAlignment](#) (унаследован от [Control](#)), [Visibility](#) (унаследован от [UIElement](#)), [Width](#) (унаследован от [FrameworkElement](#))

Методы

[AddChild](#) (унаследован от [ContentControl](#)), [AddHandler](#) (унаследован от [UIElement](#)), [AddLogicalChild](#) (унаследован от [FrameworkElement](#)), [AddText](#) (унаследован от [ContentControl](#)), [AddToEventRoute](#) (унаследован от [UIElement](#)), [AddVisualChild](#) (унаследован от [Visual](#)), [ApplyAnimationClock](#) (унаследован от [UIElement](#)), [ApplyTemplate](#) (унаследован от [FrameworkElement](#)), [Arrange](#) (унаследован от [UIElement](#)), [ArrangeCore](#) (унаследован от [FrameworkElement](#)), [ArrangeOverride](#) (унаследован от [Control](#)), [AttachTo₂₃₉](#), [BeginAnimation](#) (унаследован от [UIElement](#)), [BeginInit](#) (унаследован от [FrameworkElement](#)), [BeginStoryboard](#) (унаследован от [FrameworkElement](#)), [BringIntoView](#) (унаследован от [FrameworkElement](#)), [CaptureMouse](#) (унаследован от [UIElement](#)), [CaptureStylus](#) (унаследован от [UIElement](#)), [CaptureTouch](#) (унаследован от [UIElement](#)), [CheckAccess](#) (унаследован от [DispatcherObject](#)), [ClearValue](#) (унаследован от [DependencyObject](#)), [CoerceValue](#) (унаследован от [DependencyObject](#)), [Ellipse_MouseDown₂₄₀](#), [Ellipse_MouseMove₂₄₀](#), [Ellipse_MouseUp₂₄₀](#), [EndInit](#) (унаследован от [FrameworkElement](#)), [Equals](#) (унаследован от [DependencyObject](#)), [Finalize](#) (унаследован от [Object](#)), [FindCommonVisualAncestor](#) (унаследован от [Visual](#)), [FindName](#) (унаследован от [FrameworkElement](#)), [FindResource](#) (унаследован от [FrameworkElement](#)), [Focus](#) (унаследован от [UIElement](#)), [GetAnimationBaseValue](#) (унаследован от [UIElement](#)), [GetBindingExpression](#) (унаследован от [FrameworkElement](#)), [GetHashCode](#) (унаследован от [DependencyObject](#)), [GetLayoutClip](#) (унаследован от [FrameworkElement](#)), [GetLocalValueEnumerator](#) (унаследован от [DependencyObject](#)), [GetTemplateChild](#) (унаследован от [FrameworkElement](#)), [GetType](#) (унаследован от [Object](#)), [GetUIParentCore](#) (унаследован от [FrameworkElement](#)), [GetValue](#) (унаследован от [DependencyObject](#)), [GetVisualChild](#) (унаследован от [FrameworkElement](#)), [HitTestCore](#) (унаследован от [UIElement](#)), [InputHitTest](#) (унаследован от [UIElement](#)), [InvalidateArrange](#) (унаследован от [UIElement](#)), [InvalidateMeasure](#) (унаследован от [UIElement](#)), [InvalidateProperty](#) (унаследован от [DependencyObject](#)), [InvalidateVisual](#) (унаследован от [UIElement](#)), [IsAncestorOf](#) (унаследован от [Visual](#)), [IsDescendantOf](#) (унаследован от [Visual](#)), [Measure](#) (унаследован от [UIElement](#)), [MeasureCore](#) (унаследован от [FrameworkElement](#)), [MeasureOverride](#) (унаследован от [Control](#)), [MemberwiseClone](#) (унаследован от [Object](#)), [MoveFocus](#) (унаследован от [FrameworkElement](#)), [OnAccessKey](#) (унаследован от [UIElement](#)), [OnApplyTemplate](#) (унаследован от [FrameworkElement](#)), [OnChildDesiredSizeChanged](#) (унаследован от [UIElement](#)), [OnContentChanged](#) (унаследован от [ContentControl](#)), [OnContentStringFormatChanged](#) (унаследован от [ContentControl](#)), [OnContentTemplateChanged](#) (унаследован от [ContentControl](#)), [OnContentTemplateSelectorChanged](#) (унаследован от [ContentControl](#)), [OnContextMenuClosing](#) (унаследован от [FrameworkElement](#)), [OnContextMenuOpening](#) (унаследован от [FrameworkElement](#)), [OnCreateAutomationPeer](#) (унаследован от [UserControl](#)), [OnDpiChanged](#) (унаследован от [Visual](#)), [OnDragEnter](#) (унаследован от [UIElement](#)), [OnDragLeave](#) (унаследован от [UIElement](#)), [OnDragOver](#) (унаследован от [UIElement](#)), [OnDrop](#) (унаследован от [UIElement](#)), [OnGiveFeedback](#) (унаследован от [UIElement](#)), [OnGotFocus](#) (унаследован от [FrameworkElement](#)), [OnGotKeyboardFocus](#) (унаследован от [UIElement](#)), [OnGotMouseCapture](#) (унаследован от [UIElement](#)), [OnGotStylusCapture](#) (унаследован от [UIElement](#)), [OnGotTouchCapture](#) (унаследован от [UIElement](#)), [OnInitialized](#) (унаследован от [FrameworkElement](#)), [OnIsKeyboardFocusWithinChanged](#) (унаследован от [UIElement](#)), [OnIsKeyboardFocusedChanged](#) (унаследован от [UIElement](#)), [OnIsMouseCaptureWithinChanged](#) (унаследован от [UIElement](#)), [OnIsMouseCapturedChanged](#) (унаследован от [UIElement](#)), [OnIsMouseDirectlyOverChanged](#) (унаследован от [UIElement](#)), [OnIsStylusCaptureWithinChanged](#) (унаследован от [UIElement](#)), [OnIsStylusCapturedChanged](#) (унаследован от [UIElement](#)), [OnIsStylusDirectlyOverChanged](#) (унаследован от [UIElement](#)), [OnKeyDown](#) (унаследован от [UIElement](#)),

[OnKeyUp](#) (унаследован от [UIElement](#)), [OnLostFocus](#) (унаследован от [UIElement](#)), [OnLostKeyboardFocus](#) (унаследован от [UIElement](#)), [OnLostMouseCapture](#) (унаследован от [UIElement](#)), [OnLostStylusCapture](#) (унаследован от [UIElement](#)), [OnLostTouchCapture](#) (унаследован от [UIElement](#)),
[OnManipulationBoundaryFeedback](#) (унаследован от [UIElement](#)), [OnManipulationCompleted](#) (унаследован от [UIElement](#)), [OnManipulationDelta](#) (унаследован от [UIElement](#)), [OnManipulationInertiaStarting](#) (унаследован от [UIElement](#)), [OnManipulationStarted](#) (унаследован от [UIElement](#)), [OnManipulationStarting](#) (унаследован от [UIElement](#)), [OnMouseDoubleClick](#) (унаследован от [Control](#)), [OnMouseDown](#) (унаследован от [UIElement](#)), [OnMouseEnter](#) (унаследован от [UIElement](#)), [OnMouseLeave](#) (унаследован от [UIElement](#)),
[OnMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [OnMouseLeftButtonUp](#) (унаследован от [UIElement](#)), [OnMouseMove](#) (унаследован от [UIElement](#)), [OnMouseRightButtonDown](#) (унаследован от [UIElement](#)),
[OnMouseRightButtonUp](#) (унаследован от [UIElement](#)), [OnMouseUp](#) (унаследован от [UIElement](#)), [OnMouseWheel](#) (унаследован от [UIElement](#)), [OnPreviewDragEnter](#) (унаследован от [UIElement](#)),
[OnPreviewDragLeave](#) (унаследован от [UIElement](#)), [OnPreviewDragOver](#) (унаследован от [UIElement](#)), [OnPreviewDrop](#) (унаследован от [UIElement](#)), [OnPreviewGiveFeedback](#) (унаследован от [UIElement](#)),
[OnPreviewGotKeyboardFocus](#) (унаследован от [UIElement](#)), [OnPreviewKeyDown](#) (унаследован от [UIElement](#)), [OnPreviewKeyUp](#) (унаследован от [UIElement](#)), [OnPreviewLostKeyboardFocus](#) (унаследован от [UIElement](#)),
[OnPreviewMouseDoubleClick](#) (унаследован от [Control](#)), [OnPreviewMouseDown](#) (унаследован от [UIElement](#)), [OnPreviewMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewMouseLeftButtonUp](#) (унаследован от [UIElement](#)), [OnPreviewMouseMove](#) (унаследован от [UIElement](#)),
[OnPreviewMouseRightButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewMouseRightButtonUp](#) (унаследован от [UIElement](#)), [OnPreviewMouseUp](#) (унаследован от [UIElement](#)), [OnPreviewMouseWheel](#) (унаследован от [UIElement](#)), [OnPreviewQueryContinueDrag](#) (унаследован от [UIElement](#)),
[OnPreviewStylusButtonDown](#) (унаследован от [UIElement](#)), [OnPreviewStylusButtonUp](#) (унаследован от [UIElement](#)), [OnPreviewStylusDown](#) (унаследован от [UIElement](#)), [OnPreviewStylusInAirMove](#) (унаследован от [UIElement](#)), [OnPreviewStylusInRange](#) (унаследован от [UIElement](#)), [OnPreviewStylusMove](#) (унаследован от [UIElement](#)), [OnPreviewStylusOutOfRange](#) (унаследован от [UIElement](#)), [OnPreviewStylusSystemGesture](#) (унаследован от [UIElement](#)), [OnPreviewStylusUp](#) (унаследован от [UIElement](#)), [OnPreviewTextInput](#) (унаследован от [UIElement](#)), [OnPreviewTouchDown](#) (унаследован от [UIElement](#)), [OnPreviewTouchMove](#) (унаследован от [UIElement](#)), [OnPreviewTouchUp](#) (унаследован от [UIElement](#)), [OnPropertyChanged](#)²⁴¹,
[OnQueryContinueDrag](#) (унаследован от [UIElement](#)), [OnQueryCursor](#) (унаследован от [UIElement](#)), [OnRender](#) (унаследован от [UIElement](#)), [OnRenderSizeChanged](#) (унаследован от [FrameworkElement](#)), [OnStyleChanged](#) (унаследован от [FrameworkElement](#)), [OnStylusButtonDown](#) (унаследован от [UIElement](#)), [OnStylusButtonUp](#) (унаследован от [UIElement](#)), [OnStylusDown](#) (унаследован от [UIElement](#)), [OnStylusEnter](#) (унаследован от [UIElement](#)), [OnStylusInAirMove](#) (унаследован от [UIElement](#)), [OnStylusInRange](#) (унаследован от [UIElement](#)),
[OnStylusLeave](#) (унаследован от [UIElement](#)), [OnStylusMove](#) (унаследован от [UIElement](#)), [OnStylusOutOfRange](#) (унаследован от [UIElement](#)), [OnStylusSystemGesture](#) (унаследован от [UIElement](#)), [OnStylusUp](#) (унаследован от [UIElement](#)), [OnTemplateChanged](#) (унаследован от [Control](#)), [OnTextInput](#) (унаследован от [UIElement](#)),
[OnToolTipClosing](#) (унаследован от [FrameworkElement](#)), [OnToolTipOpening](#) (унаследован от [FrameworkElement](#)), [OnTouchDown](#) (унаследован от [UIElement](#)), [OnTouchEnter](#) (унаследован от [UIElement](#)), [OnTouchLeave](#) (унаследован от [UIElement](#)), [OnTouchMove](#) (унаследован от [UIElement](#)), [OnTouchUp](#) (унаследован от [UIElement](#)),
[OnVisualChildrenChanged](#) (унаследован от [Visual](#)), [OnVisualParentChanged](#) (унаследован от [FrameworkElement](#)), [ParentLayoutInvalidated](#) (унаследован от [FrameworkElement](#)), [PointFromScreen](#) (унаследован от [Visual](#)), [PointToScreen](#) (унаследован от [Visual](#)), [PopupPlacementCallback](#)²⁴¹, [PredictFocus](#) (унаследован от [FrameworkElement](#)), [RaiseEvent](#) (унаследован от [UIElement](#)), [ReadLocalValue](#) (унаследован от [DependencyObject](#)), [RegisterName](#) (унаследован от [FrameworkElement](#)), [ReleaseAllTouchCaptures](#)

(унаследован от [UIElement](#)), [ReleaseMouseCapture](#) (унаследован от [UIElement](#)), [ReleaseStylusCapture](#) (унаследован от [UIElement](#)), [ReleaseTouchCapture](#) (унаследован от [UIElement](#)), [RemoveHandler](#) (унаследован от [UIElement](#)), [RemoveLogicalChild](#) (унаследован от [FrameworkElement](#)), [RemoveVisualChild](#) (унаследован от [Visual](#)), [Rotate₂₄₂](#), [SetBinding](#) (унаследован от [FrameworkElement](#)), [SetCurrentValue](#) (унаследован от [DependencyObject](#)), [SetCursorPos₂₄₂](#), [SetResourceReference](#) (унаследован от [FrameworkElement](#)), [SetValue₂₄₃](#), [ShouldSerializeCommandBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeContent](#) (унаследован от [ContentControl](#)), [ShouldSerializeInputBindings](#) (унаследован от [UIElement](#)), [ShouldSerializeProperty](#) (унаследован от [DependencyObject](#)), [ShouldSerializeResources](#) (унаследован от [FrameworkElement](#)), [ShouldSerializeStyle](#) (унаследован от [FrameworkElement](#)), [ShouldSerializeTriggers](#) (унаследован от [FrameworkElement](#)), [ToString](#) (унаследован от [Control](#)), [TransformToAncestor](#) (унаследован от [Visual](#)), [TransformToDescendant](#) (унаследован от [Visual](#)), [TransformToVisual](#) (унаследован от [Visual](#)), [TranslatePoint](#) (унаследован от [UIElement](#)), [TryFindResource](#) (унаследован от [FrameworkElement](#)), [UnregisterName](#) (унаследован от [FrameworkElement](#)), [UpdateDefaultStyle](#) (унаследован от [FrameworkElement](#)), [UpdateDisplayStyle₂₄₃](#), [UpdateLayout](#) (унаследован от [UIElement](#)), [UpdateParameterValue₂₄₃](#), [VerifyAccess](#) (унаследован от [DispatcherObject](#))

События

[ContextMenuClosing](#) (унаследован от [FrameworkElement](#)), [ContextMenuOpening](#) (унаследован от [FrameworkElement](#)), [DataContextChanged](#) (унаследован от [FrameworkElement](#)), [DragEnter](#) (унаследован от [UIElement](#)), [DragLeave](#) (унаследован от [UIElement](#)), [DragOver](#) (унаследован от [UIElement](#)), [Drop](#) (унаследован от [UIElement](#)), [FocusableChanged](#) (унаследован от [UIElement](#)), [GiveFeedback](#) (унаследован от [UIElement](#)), [GotFocus](#) (унаследован от [UIElement](#)), [GotKeyboardFocus](#) (унаследован от [UIElement](#)), [GotMouseCapture](#) (унаследован от [UIElement](#)), [GotStylusCapture](#) (унаследован от [UIElement](#)), [GotTouchCapture](#) (унаследован от [UIElement](#)), [Initialized](#) (унаследован от [FrameworkElement](#)), [IsEnabledChanged](#) (унаследован от [UIElement](#)), [IsHitTestVisibleChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusWithinChanged](#) (унаследован от [UIElement](#)), [IsKeyboardFocusedChanged](#) (унаследован от [UIElement](#)), [IsMouseCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsMouseCapturedChanged](#) (унаследован от [UIElement](#)), [IsMouseDirectlyOverChanged](#) (унаследован от [UIElement](#)), [IsStylusCaptureWithinChanged](#) (унаследован от [UIElement](#)), [IsStylusCapturedChanged](#) (унаследован от [UIElement](#)), [IsStylusDirectlyOverChanged](#) (унаследован от [UIElement](#)), [IsVisibleChanged](#) (унаследован от [UIElement](#)), [KeyDown](#) (унаследован от [UIElement](#)), [KeyUp](#) (унаследован от [UIElement](#)), [LayoutUpdated](#) (унаследован от [UIElement](#)), [Loaded](#) (унаследован от [FrameworkElement](#)), [LostFocus](#) (унаследован от [UIElement](#)), [LostKeyboardFocus](#) (унаследован от [UIElement](#)), [LostMouseCapture](#) (унаследован от [UIElement](#)), [LostStylusCapture](#) (унаследован от [UIElement](#)), [LostTouchCapture](#) (унаследован от [UIElement](#)), [ManipulationBoundaryFeedback](#) (унаследован от [UIElement](#)), [ManipulationCompleted](#) (унаследован от [UIElement](#)), [ManipulationDelta](#) (унаследован от [UIElement](#)), [ManipulationInertiaStarting](#) (унаследован от [UIElement](#)), [ManipulationStarted](#) (унаследован от [UIElement](#)), [ManipulationStarting](#) (унаследован от [UIElement](#)), [MouseDoubleClick](#) (унаследован от [Control](#)), [MouseDown](#) (унаследован от [UIElement](#)), [MouseEnter](#) (унаследован от [UIElement](#)), [MouseLeave](#) (унаследован от [UIElement](#)), [MouseLeftButtonDown](#) (унаследован от [UIElement](#)), [MouseLeftButtonUp](#) (унаследован от [UIElement](#)), [MouseMove](#) (унаследован от [UIElement](#)), [MouseRightButtonDown](#) (унаследован от [UIElement](#)), [MouseRightButtonUp](#) (унаследован от [UIElement](#)), [MouseUp](#) (унаследован от [UIElement](#)), [MouseWheel](#) (унаследован от [UIElement](#)), [PreviewDragEnter](#) (унаследован от [UIElement](#)), [PreviewDragLeave](#) (унаследован от [UIElement](#)), [PreviewDragOver](#) (унаследован от [UIElement](#)), [PreviewDrop](#) (унаследован от [UIElement](#)), [PreviewGiveFeedback](#) (унаследован от [UIElement](#)), [PreviewGotKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewKeyDown](#) (унаследован от [UIElement](#)), [PreviewKeyUp](#) (унаследован от [UIElement](#)),

[PreviewLostKeyboardFocus](#) (унаследован от [UIElement](#)), [PreviewMouseDoubleClick](#) (унаследован от [Control](#)), [PreviewMouseDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseLeftButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseMove](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonDown](#) (унаследован от [UIElement](#)), [PreviewMouseRightButtonUp](#) (унаследован от [UIElement](#)), [PreviewMouseUp](#) (унаследован от [UIElement](#)), [PreviewMouseWheel](#) (унаследован от [UIElement](#)), [PreviewQueryContinueDrag](#) (унаследован от [UIElement](#)), [PreviewStylusButtonDown](#) (унаследован от [UIElement](#)), [PreviewStylusButtonUp](#) (унаследован от [UIElement](#)), [PreviewStylusDown](#) (унаследован от [UIElement](#)), [PreviewStylusInAirMove](#) (унаследован от [UIElement](#)), [PreviewStylusInRange](#) (унаследован от [UIElement](#)), [PreviewStylusMove](#) (унаследован от [UIElement](#)), [PreviewStylusOutOfRange](#) (унаследован от [UIElement](#)), [PreviewStylusSystemGesture](#) (унаследован от [UIElement](#)), [PreviewStylusUp](#) (унаследован от [UIElement](#)), [PreviewTextInput](#) (унаследован от [UIElement](#)), [PreviewTouchDown](#) (унаследован от [UIElement](#)), [PreviewTouchMove](#) (унаследован от [UIElement](#)), [PreviewTouchUp](#) (унаследован от [UIElement](#)), [PropertyChanged](#)²⁴³, [QueryContinueDrag](#) (унаследован от [UIElement](#)), [QueryCursor](#) (унаследован от [UIElement](#)), [RequestBringIntoView](#) (унаследован от [FrameworkElement](#)), [SizeChanged](#) (унаследован от [FrameworkElement](#)), [SourceUpdated](#) (унаследован от [FrameworkElement](#)), [StylusButtonDown](#) (унаследован от [UIElement](#)), [StylusButtonUp](#) (унаследован от [UIElement](#)), [StylusDown](#) (унаследован от [UIElement](#)), [StylusEnter](#) (унаследован от [UIElement](#)), [StylusInAirMove](#) (унаследован от [UIElement](#)), [StylusInRange](#) (унаследован от [UIElement](#)), [StylusLeave](#) (унаследован от [UIElement](#)), [StylusMove](#) (унаследован от [UIElement](#)), [StylusOutOfRange](#) (унаследован от [UIElement](#)), [StylusSystemGesture](#) (унаследован от [UIElement](#)), [StylusUp](#) (унаследован от [UIElement](#)), [TargetUpdated](#) (унаследован от [FrameworkElement](#)), [TextInput](#) (унаследован от [UIElement](#)), [ToolTipClosing](#) (унаследован от [FrameworkElement](#)), [ToolTipOpening](#) (унаследован от [FrameworkElement](#)), [TouchDown](#) (унаследован от [UIElement](#)), [TouchEnter](#) (унаследован от [UIElement](#)), [TouchLeave](#) (унаследован от [UIElement](#)), [TouchMove](#) (унаследован от [UIElement](#)), [TouchUp](#) (унаследован от [UIElement](#)), [Unloaded](#) (унаследован от [FrameworkElement](#))

Поля

[Delta](#)²⁴⁵, [accumulatedValue](#)²⁴⁴, [displayValue](#)²⁴⁴, [leftButtonPressed](#)²⁴⁴, [manager](#)²⁴⁴, [startMousePosition](#)²⁴⁵, [startScreenMousePosition](#)²⁴⁵, [stroke](#)²⁴⁵, [value](#)²⁴⁵

Knob - конструктор

Инициализирует новый объект типа Knob.

C#

```
public Knob()
```

См. также

Применяется: [Knob](#)²³³

Knob.DisplayValue - свойство

Значение, отображаемое на всплывающей подсказке.

C#

```
public string DisplayValue {get; set;}
```

См. такжеПрименяется: [Knob](#)₂₃₃**Knob.DisplayValueConverter - свойство**

Функция для преобразования текущего значения в строку.

C#

```
public Func<double, string> DisplayValueConverter {get; set;}
```

См. такжеПрименяется: [Knob](#)₂₃₃**Knob.Stroke - свойство**

Кисть, используемая для создания цветной метки на кнопке и цветной дуги.

C#

```
public Brush Stroke {get; set;}
```

См. такжеПрименяется: [Knob](#)₂₃₃**Knob.AttachTo - метод**

Метод, привязывающий ручку к объекту VstParameterManager.

C#

```
public void AttachTo(
    VstParameterManager manager,
    Brush stroke,
    Func<double, string> valueConverter = null
)
```

Параметры*manager**stroke**valueConverter*

См. также

Применяется: [Knob](#)₂₃₃

Knob.Ellipse_MouseDown - метод

Обработчик события нажатия на ручку.

C#

```
private void Ellipse_MouseDown(  
    object sender,  
    MouseButtonEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [Knob](#)₂₃₃

Knob.Ellipse_MouseMove - метод

Обработчик события движения мыши.

C#

```
private void Ellipse_MouseMove(  
    object sender,  
    MouseEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [Knob](#)₂₃₃

Knob.Ellipse_MouseUp - метод

Обработчик события отпускания кнопки мыши.

C#

```
private void Ellipse_MouseUp(  
    object sender,  
    MouseEventArgs e  
)
```

Параметры

sender

e

См. также

Применяется: [Knob](#)₂₃₃

Knob.OnPropertyChanged - метод

Метод, вызывающий событие PropertyChanged с переданными параметрами.

C#

```
private void OnPropertyChanged(  
    string v  
)
```

Параметры

v

См. также

Применяется: [Knob](#)₂₃₃

Knob.PopupPlacementCallback - метод

Метод, возвращающий позиции для всплывающей подсказки.

C#

```
private CustomPopupPlacement[] PopupPlacementCallback(  
    Size popupSize,  
    Size targetSize,  
    Point offset  
)
```

Параметры

popupSize

targetSize

offset

См. также

Применяется: [Knob](#)₂₃₃

Knob.Rotate - метод

Поворачивает ручку в соответствии с переданным значением.

C#

```
protected virtual void Rotate(  
    double normalizedValue  
)
```

Параметры

normalizedValue

См. также

Применяется: [Knob](#)₂₃₃

Knob.SetCursorPos - метод

Метод, передвигающий курсор мыши на указанные координаты.

C#

```
[DllImport("User32.dll")]  
private static extern bool SetCursorPos(  
    int X,  
    int Y  
)
```

Параметры

X

Y

См. также

Применяется: [Knob](#)₂₃₃

Knob.SetValue - метод

Устанавливает новое значение.

C#

```
public void SetValue(  
    double newValue  
)
```

Параметры

newValue

См. также

Применяется: [Knob](#)₂₃₃

Knob.UpdateDisplayValue - метод

Метод, обновляющий значение на всплывающей подсказке.

C#

```
public void UpdateDisplayValue()
```

См. также

Применяется: [Knob](#)₂₃₃

Knob.UpdateParameterValue - метод

Метод, обновляющий значение привязанного параметра.

C#

```
private void UpdateParameterValue()
```

См. также

Применяется: [Knob](#)₂₃₃

PropertyChanged - событие

Событие, показывающее, что значения некоторых свойств изменились.

C#

```
public event PropertyChangedEventHandler PropertyChanged
```

См. также

Применяется: [Knob](#)₂₃₃

accumulatedValue - поле

Расстояние по вертикали, которое прошла мышь с момента её захвата.

C#

```
private double accumulatedValue
```

См. также

Применяется: [Knob](#)₂₃₃

displayValue - поле

Текущее отображаемое значение.

C#

```
private string displayValue
```

См. также

Применяется: [Knob](#)₂₃₃

leftButtonPressed - поле

Показывает, нажата ли левая кнопка мыши.

C#

```
private bool leftButtonPressed
```

См. также

Применяется: [Knob](#)₂₃₃

manager - поле

Связанный с ручкой объект типа VstParameterManager.

C#

```
private VstParameterManager manager
```

См. также

Применяется: [Knob](#)₂₃₃

startMousePosition - поле

Начальная позиция мыши в момент её захвата относительно элемента.

C#

```
private Point startMousePosition
```

См. также

Применяется: [Knob₂₃₃](#)

startScreenMousePosition - поле

Начальная позиция мыши в момент её захвата относительно экрана.

C#

```
private Point startScreenMousePosition
```

См. также

Применяется: [Knob₂₃₃](#)

stroke - поле

Кисть, используемая для

C#

```
private Brush stroke
```

См. также

Применяется: [Knob₂₃₃](#)

value - поле

Текущее значение.

C#

```
private double value
```

См. также

Применяется: [Knob₂₃₃](#)

Delta - поле

Показывает, как сильно меняется значение при перемещении на один пиксель.

C#

```
private const double Delta = 0,0005
```

См. также

Применяется: [Knob](#)₂₃₃

Index

AbsClipper - класс 23
AdsrEnvelope - класс 24
AdsrEnvelope - конструктор 25
AmountManager - свойство 79
AmpManager - свойство 79
AsymmetryManager - свойство 80
AsymmetryToString - метод 42
AttachTo - метод 239
AttackCurveManager - свойство 101
AttackTimeManager - свойство 101
AudioComponent - класс 33
AudioComponentWithParameters - класс 35
AudioComponentWithParameters - конструктор 36
AudioProcessor - класс 37
AudioProcessor - конструктор 38
AudioProcessor - свойство 148
BaseCutoff - поле 113
BaseSampleRate - поле {MultimodSynth.DCBlocker} 59
BaseSampleRate - поле
 {MultimodSynth.ParameterFilter} 142
BindDelay - метод 229
BindDistortion - метод 229
BindFilter - метод 229
BindKeyboard - метод 230
BindMasterSettings - метод 230
BindOscA - метод 230
BindOscB - метод 231
BindParameters - метод 231
BindToPlugin - метод 232
BipolarKnob - класс 218
BipolarKnob - конструктор 223
BitCrusher - класс 40
Bounds - свойство 151
CalcCoef - метод 26
CalculateCutoff - метод 109
CalculateOutput - метод 60
CentsToString - метод 42
ChangeDelay - метод 67
ChangeDistortion - метод 81
ChannelCount - свойство 122
Clone - метод 207
Close - метод 152
Converters - класс 41
CreateAudioProcessor - метод 148
CreateEditor - метод 149
CreateMidiProcessor - метод 149
CreateNewEnvelope - метод 102
CreateNewFilter - метод 116
CreateNewOscillator - метод 132
CreateParameterManager - метод 136
CreatePluginInstance - метод 150
CreateProgramCollection - метод 156
CreatePrograms - метод 149
CreateVoice - метод 202
CubicClipper - класс 54
CurveManager - свойство 115
CutoffManager - свойство 115
DCBlocker - класс 56
DCBlocker - конструктор 56
DecayReleaseCurveManager - свойство 101
DecayTimeManager - свойство 101
DefaultExp - поле 58
DefaultExp - поле 142
DefaultSampleRate - поле {MultimodSynth.Utilities}
 188
DefaultSampleRate - поле
 {MultimodSynth.WaveTableOscillator} 213
DelayLfoRateToString - метод 43
DelayLine - класс 60
DelayManager - класс 64
DelayManager - конструктор 65
DelayManager - свойство 158
DelayModeToString - метод 43
DelayTimeToString - метод 43
Delta - поле 245
Deserialize - метод 207
DisplayValue - свойство 238
DisplayValueConverter - свойство 239
DistortionAmpToString - метод 44
DistortionLowpassCutoffToString - метод 44
DistortionManager - класс 78
DistortionManager - конструктор 79
DistortionManager - свойство 158
DistortionMode - перечисление 85
DistortionModeToString - метод 44
Downsampler - класс 89
Downsampler - свойство 158
EditorView - класс 223
Ellipse_MouseDown - метод 240
Ellipse_MouseMove - метод 240
Ellipse_MouseUp - метод 240
Envelope - класс 92
Envelope - конструктор 93
EnvelopeAmplitudeManager - свойство 102
EnvelopeCurveToString - метод 45
EnvelopeState - перечисление 29
EnvelopeTimeToString - метод 45
EnvelopesManager - класс 99
EnvelopesManager - конструктор 100
FeedbackManager - свойство 65
Filter - класс 108
Filter - конструктор 109
FilterCutoffEnvelopeManager - свойство 200

FilterCutoffMultiplierToString - метод 45
FilterManager - свойство 200
FilterModulationLookup - поле 112
FilterType - перечисление 180
FilterTypeManager - свойство 115
FilterTypeToString - метод 46
FiltersManager - класс 113
FiltersManager - конструктор 114
GeneratorFunction - делегат 211
GetElementPixelSize - метод 152
GetOrGenerateWaveTables - метод 184
HalfSquareGenerator - метод 184
IDelay - интерфейс 213
IDistortion - интерфейс 215
InitializeParameters - метод 36
InitializeParameters (ParameterFactory) - метод
 {MultimodSynth.AudioComponentWithParameters}
 37
InitializeParameters - метод
 {MultimodSynth.DelayManager} 68
InitializeParameters - метод
 {MultimodSynth.DistortionManager} 81
InitializeParameters - метод
 {MultimodSynth.EnvelopesManager} 103
InitializeParameters - метод
 {MultimodSynth.FiltersManager} 116
InitializeParameters - метод
 {MultimodSynth.OscillatorsManager} 132
InitializeParameters - метод {MultimodSynth.Routing}
 159
InitializeParameters - метод
 {MultimodSynth.VoicesManager} 202
InvertManager - свойство 66
InvertToString - метод 46
IsActive - свойство {MultimodSynth.Envelope} 93
IsActive - свойство {MultimodSynth.Voice} 195
KeyDown - метод 152
KeyUp - метод 153
Knob - класс 233
Knob - конструктор 238
KnobMode - свойство 152
Length - поле 210
LfoDepthManager - свойство 66
LfoRateManager - свойство 66
LowPassCutoffManager - свойство 80
MasterVolumeManager - свойство 159
MaxLfoDepth - поле 77
MaxTime - поле {MultimodSynth.DelayLine} 64
MaxTime - поле {MultimodSynth.DelayManager} 78
MaxVoicesCount - поле 206
MidiNote - структура 216
MidiNoteEventArgs - класс 120
MidiNoteEventArgs - конструктор 120
MidiNoteToFrequency - метод 185
MidiProcessor - класс 121
MidiProcessor - конструктор 122
MidiProcessor - свойство 148
MidiProcessor_NoteOff - метод 160
MidiProcessor_NoteOn - метод 160
MixManager - свойство {MultimodSynth.DelayManager}
 66
MixManager - свойство
 {MultimodSynth.DistortionManager} 80
ModeManager - свойство
 {MultimodSynth.DelayManager} 67
ModeManager - свойство
 {MultimodSynth.DistortionManager} 80
Modulation - свойство 195
ModulationType - перечисление 197
ModulationTypeManager - свойство 201
ModulationTypeToString - метод 47
MultimodSynth - пространство имен 23
MultimodSynth Reference 22
MultimodSynth.UI - пространство имен 218
Normalize - метод 208
Note - свойство {MultimodSynth.MidiNoteEventArgs}
 121
Note - свойство {MultimodSynth.Voice} 195
NoteNo - свойство 216
NoteOff - событие 124
NoteOn - событие 125
OnNoteOff - метод 122
OnNoteOn - метод 123
OnPropertyChanged - метод 241
OnSampleRateChanged - метод
 {MultimodSynth.AudioComponent} 34
OnSampleRateChanged - метод
 {MultimodSynth.DCBlocker} 57
OnSampleRateChanged - метод
 {MultimodSynth.DelayLine} 61
OnSampleRateChanged - метод
 {MultimodSynth.DelayManager} 68
OnSampleRateChanged - метод
 {MultimodSynth.DistortionManager} 81
OnSampleRateChanged - метод
 {MultimodSynth.Envelope} 94
OnSampleRateChanged - метод {MultimodSynth.Filter}
 109
OnSampleRateChanged - метод
 {MultimodSynth.Oscillator} 126
OnSampleRateChanged - метод
 {MultimodSynth.ParameterFilter} 139
OnSampleRateChanged - метод
 {MultimodSynth.PingPongDelay} 143
OnSampleRateChanged - метод
 {MultimodSynth.Routing} 161
OnSampleRateChanged - метод
 {MultimodSynth.SampleRateReductor} 164
OnSampleRateChanged - метод
 {MultimodSynth.SineLFO} 167

OnSampleRateChanged - метод
 {MultimodSynth.StereoOffsetDelay} 172

OnSampleRateChanged - метод
 {MultimodSynth.SvfFilter} 177

OnSampleRateChanged - метод
 {MultimodSynth.VariousTimeDelay} 190

OnSampleRateChanged - метод
 {MultimodSynth.VoicesManager} 202

OnSampleRateChanged - метод {MultimodSynth.Voice}
 195

OnSoundStop - метод 26

Open - метод 153

OpenButton_Click - метод 232

Order - свойство 90

OscAManager - свойство 201

OscAVolumeEnvelopeManager - свойство 201

OscBManager - свойство 201

OscBVolumeEnvelopeManager - свойство 201

Oscillator - класс 125

OscillatorsManager - класс 130

OscillatorsManager - конструктор 130

OversamplingOrderManager - свойство 159

OversamplingOrderToString - метод 47

ParameterCategories - свойство 156

ParameterFactory - класс 135

ParameterFactory - конструктор 136

ParameterFilter - класс 138

ParameterFilter - конструктор 139

ParameterInfos - свойство 156

PercentsToString - метод 47

PhaseIncrement - поле 211

PingPongDelay - класс 142

PingPongDelay - конструктор 143

PitchFineManager - свойство 131

PitchSemiManager - свойство 131

PlayNote - метод {MultimodSynth.VoicesManager} 203

PlayNote - метод {MultimodSynth.Voice} 196

Plugin - класс 147

Plugin - конструктор 147

Plugin - свойство 36

PluginCommandStub - класс 150

PluginEditor - класс 150

PluginEditor - конструктор 151

PluginPrograms - класс 155

PluginPrograms - конструктор 156

PopupPlacementCallback - метод 241

PrepareSamples - метод 208

PressNoteFromUI - метод 123

PressedNotesCount - свойство 121

Process - метод {MultimodSynth.AbsClipper} 23

Process - метод {MultimodSynth.AdsrEnvelope} 26

Process - метод {MultimodSynth.AudioProcessor} 39

Process - метод {MultimodSynth.BitCrusher} 40

Process - метод {MultimodSynth.CubicClipper} 55

Process - метод {MultimodSynth.DCBlocker} 57

Process - метод {MultimodSynth.DelayLine} 61

Process - метод {MultimodSynth.DelayManager} 68

Process - метод {MultimodSynth.DistortionManager}
 82

Process - метод {MultimodSynth.Downsampler} 90

Process - метод {MultimodSynth.EnvelopesManager}
 103

Process - метод {MultimodSynth.Envelope} 94

Process - метод {MultimodSynth.FiltersManager} 116

Process - метод {MultimodSynth.Filter} 109

Process - метод {MultimodSynth.IDelay} 213

Process - метод {MultimodSynth.IDistortion} 215

Process - метод {MultimodSynth.MidiProcessor} 124

Process - метод {MultimodSynth.OscillatorsManager}
 132

Process - метод {MultimodSynth.Oscillator} 126

Process - метод {MultimodSynth.ParameterFilter} 139

Process - метод {MultimodSynth.PingPongDelay} 144

Process - метод {MultimodSynth.Routing} 161

Process - метод {MultimodSynth.SampleRateReductor}
 164

Process - метод {MultimodSynth.SineLFO} 167

Process - метод {MultimodSynth.SoftClipper} 170

Process - метод {MultimodSynth.StereoOffsetDelay}
 173

Process - метод {MultimodSynth.SvfFilter} 177

Process - метод {MultimodSynth.VariousTimeDelay}
 190

Process - метод {MultimodSynth.VoicesManager} 203

Process - метод {MultimodSynth.Voice} 196

Process - метод {MultimodSynth.WaveTableOscillator}
 209

ProcessIdle - метод 154

ProcessingMutex - свойство 38

Programs - свойство 148

PropertyChanged - событие 243

QuarterSquareGenerator - метод 185

ReadParameters - метод 185

ReleaseNote - метод 203

ReleaseNoteFromUI - метод 124

ReleaseTimeManager - свойство 102

Reset - метод {MultimodSynth.DelayLine} 61

Reset - метод {MultimodSynth.Filter} 110

Reset - метод {MultimodSynth.IDelay} 214

Reset - метод {MultimodSynth.Oscillator} 127

Reset - метод {MultimodSynth.PingPongDelay} 144

Reset - метод {MultimodSynth.StereoOffsetDelay} 173

Reset - метод {MultimodSynth.SvfFilter} 178

Reset - метод {MultimodSynth.VariousTimeDelay} 190

Rotate - метод {MultimodSynth.UI.BipolarKnob} 223

Rotate - метод {MultimodSynth.UI.Knob} 242

Routing - класс 157

Routing - конструктор 158

Routing - свойство 39

SampleRate - свойство
 {MultimodSynth.AudioComponent} 34

SampleRate - свойство {MultimodSynth.AudioProcessor}
 39

SampleRateReductor - класс 163

Samples - поле 211

SaveButton_Click - метод 232

SawGenerator - метод 186

SemitonesToString - метод 48

Serialize - метод 209

SetAmount - метод {MultimodSynth.AbsClipper} 24

SetAmount - метод {MultimodSynth.BitCrusher} 41

SetAmount - метод {MultimodSynth.CubicClipper} 55

SetAmount - метод {MultimodSynth.DistortionManager}
 82

SetAmount - метод {MultimodSynth.IDistortion} 216

SetAmount - метод
 {MultimodSynth.SampleRateReductor} 165

SetAmount - метод {MultimodSynth.SoftClipper} 170

SetAmpTarget - метод 82

SetAmplitude - метод 94

SetAsymmetryTarget - метод 83

SetAttackCurve - метод
 {MultimodSynth.EnvelopesManager} 103

SetAttackCurve - метод {MultimodSynth.Envelope} 95

SetAttackRate - метод 27

SetAttackTargetRatio - метод 27

SetAttackTime - метод
 {MultimodSynth.EnvelopesManager} 104

SetAttackTime - метод {MultimodSynth.Envelope} 95

SetCursorPos - метод 242

SetCurve - метод {MultimodSynth.FiltersManager} 116

SetCurve - метод {MultimodSynth.Filter} 110

SetCutoff - метод 178

SetCutoffMultiplier - метод 110

SetCutoffMultiplierTarget - метод 117

SetDecayRate - метод 27

SetDecayReleaseCurve - метод
 {MultimodSynth.EnvelopesManager} 104

SetDecayReleaseCurve - метод
 {MultimodSynth.Envelope} 95

SetDecayReleaseTargetRatio - метод 28

SetDecayTime - метод
 {MultimodSynth.EnvelopesManager} 104

SetDecayTime - метод {MultimodSynth.Envelope} 96

SetDelay - метод {MultimodSynth.DelayLine} 62

SetDelay - метод {MultimodSynth.IDelay} 214

SetDelay - метод {MultimodSynth.PingPongDelay} 145

SetDelay - метод {MultimodSynth.StereoOffsetDelay}
 173

SetDelay - метод {MultimodSynth.VariousTimeDelay}
 191

SetFeedback - метод {MultimodSynth.DelayLine} 62

SetFeedback - метод {MultimodSynth.DelayManager}
 69

SetFeedback - метод {MultimodSynth.IDelay} 214

SetFeedback - метод {MultimodSynth.PingPongDelay}
 145

SetFeedback - метод
 {MultimodSynth.StereoOffsetDelay} 174

SetFeedback - метод
 {MultimodSynth.VariousTimeDelay} 191

SetFilterType - метод {MultimodSynth.FiltersManager}
 117

SetFilterType - метод {MultimodSynth.Filter} 111

SetFrequency - метод 168

SetGain - метод 178

SetInvert - метод 69

SetLfoDepth - метод 70

SetLfoRate - метод 70

SetLowPassCutoff - метод 83

SetMixTarget - метод 70

SetMode - метод {MultimodSynth.DelayManager} 71

SetMode - метод {MultimodSynth.DistortionManager}
 84

SetModulationType - метод 204

SetNoteFrequency - метод {MultimodSynth.Filter} 111

SetNoteFrequency - метод {MultimodSynth.Oscillator}
 127

SetOversamplingOrder - метод 161

SetPhaseIncrement - метод 209

SetPitchFine - метод 132

SetPitchMultiplier - метод 127

SetPitchSemi - метод 133

SetQ - метод 179

SetReleaseRate - метод 28

SetReleaseTime - метод
 {MultimodSynth.EnvelopesManager} 105

SetReleaseTime - метод {MultimodSynth.Envelope} 96

SetResponseTimeCoefficient - метод
 {MultimodSynth.DCBlocker} 58

SetResponseTimeCoefficient - метод
 {MultimodSynth.ParameterFilter} 140

SetStereo - метод {MultimodSynth.IDelay} 215

SetStereo - метод {MultimodSynth.PingPongDelay} 145

SetStereo - метод {MultimodSynth.StereoOffsetDelay}
 174

SetStereo - метод {MultimodSynth.VariousTimeDelay}
 192

SetStereoAmountTarget - метод 71

SetSustainLevel - метод {MultimodSynth.AdsrEnvelope}
 28

SetSustainLevel - метод
 {MultimodSynth.EnvelopesManager} 105

SetSustainLevel - метод {MultimodSynth.Envelope} 96

SetTarget - метод 140

SetTimeTarget - метод 72

SetTrackingCoeff - метод
 {MultimodSynth.FiltersManager} 118

SetTrackingCoeff - метод {MultimodSynth.Filter} 111

SetType - метод 179
SetValue - метод 243
SetWaveTable - метод
 {MultimodSynth.OscillatorsManager} 133
SetWaveTable - метод {MultimodSynth.Oscillator} 128
SineGenerator - метод 186
SineLFO - класс 166
SineLFO - конструктор 167
SoftClipper - класс 169
SoundStop - событие 29
SquareGenerator - метод 187
State - свойство 25
StereoAmountManager - свойство 67
StereoAmountToString - метод 48
StereoMode - перечисление 73
StereoOffsetDelay - класс 171
StereoOffsetDelay - конструктор 172
StopVoice - метод 204
Stroke - свойство 239
SustainLevelManager - свойство 102
SvfFilter - класс 176
SvfFilter - конструктор 176
TimeManager - свойство 67
ToAsymmetry - метод 48
ToCents - метод 49
ToDelayLfoRate - метод 49
ToDelayMode - метод 49
ToDelayTime - метод 50
ToDistortionAmp - метод 50
ToDistortionLowpassCutoff - метод 51
ToDistortionMode - метод 51
ToEnvelopeTime - метод 51
ToFilterCutoffMultiplier - метод 52
ToFilterType - метод 52
ToModulationType - метод 52
ToOversamplingOrder - метод 53
ToSemitones - метод 53
ToStereoAmount - метод 53
ToWaveTable - метод 54
TrackingCoeffManager - свойство 115
TriangleGenerator - метод 187
TriggerAttack - метод {MultimodSynth.AdsrEnvelope}
 29
TriggerAttack - метод {MultimodSynth.Envelope} 97
TriggerRelease - метод {MultimodSynth.AdsrEnvelope}
 29
TriggerRelease - метод {MultimodSynth.Envelope} 97
TriggerRelease - метод {MultimodSynth.Voice} 196
UpdateAmp - метод 84
UpdateAsymmetry - метод 84
UpdateCoefficients - метод {MultimodSynth.Oscillator}
 128
UpdateCoefficients - метод
 {MultimodSynth.SampleRateReductor} 165
UpdateCoefficients - метод {MultimodSynth.SineLFO}
 168
UpdateCoefficients - метод {MultimodSynth.SvfFilter}
 179
UpdateCutoffMultiplier - метод 118
UpdateDisplayValue - метод 243
UpdateEnvelopeAmplitude - метод 105
UpdateMasterVolume - метод 162
UpdateMix - метод {MultimodSynth.DelayManager} 72
UpdateMix - метод {MultimodSynth.DistortionManager}
 85
UpdateParameterValue - метод 243
UpdatePitchMultiplier - метод 134
UpdateSampleRates - метод 162
UpdateStereoAmount - метод 72
UpdateTime - метод 73
Utilities - класс 183
Utilities - конструктор 184
VariousTimeDelay - класс 189
VariousTimeDelay - конструктор 189
Velocity - свойство 217
Voice - класс 193
Voice - конструктор 194
VoicesManager - класс 199
VoicesManager - конструктор 200
VoicesManager - свойство 159
WaveTable - класс 210
WaveTableManager - свойство 131
WaveTableOscillator - конструктор 207
WaveTableOscillator (GeneratorFunction, Double,
 Double) - конструктор 206
WaveTableOscillator - класс 206
WaveTableToString - метод 54
WaveTables - поле 188
WriteParameters - метод 188
X2OversamplingCoefficients - поле 91
X4OversamplingCoefficients - поле 91
X8OversamplingCoefficients - поле 91
a - поле 140
a1 - поле 180
a2 - поле 180
a3 - поле 181
absClipper - поле 85
accumulatedValue - поле 244
activeVoices - поле 204
amount - поле 86
amp - поле 86
ampFilter - поле 86
amplitude - поле 97
asymmetryFilter - поле 86
attackBase - поле 30
attackCoef - поле 30
attackCurve - поле {MultimodSynth.EnvelopesManager}
 106
attackCurve - поле {MultimodSynth.Envelope} 97

attackRate - поле 30
attackTargetRatio - поле {MultimodSynth.AdsrEnvelope} 31
attackTargetRatio - поле {MultimodSynth.Envelope} 98
attackTime - поле {MultimodSynth.EnvelopesManager} 106
attackTime - поле {MultimodSynth.Envelope} 98
b - поле 141
bitCrusher - поле 87
bounds - поле 154
buffer - поле {MultimodSynth.DelayLine} 62
buffer - поле {MultimodSynth.Downsampler} 91
bufferLength - поле 62
coeff - поле 168
coefficients - поле 92
cos - поле 168
cubicClipper - поле 87
currBufferIndex - поле 92
currValue - поле 31
currentDelay - поле 73
currentDistortion - поле 87
curve - поле 118
cutoff - поле {MultimodSynth.Filter} 112
cutoff - поле {MultimodSynth.SvfFilter} 181
cutoffMultiplier - поле {MultimodSynth.FiltersManager} 118
cutoffMultiplier - поле {MultimodSynth.Filter} 112
cutoffMultiplierFilter - поле 119
dcBlocker - поле 87
dcOffset - поле 88
decayBase - поле 31
decayCoef - поле 31
decayRate - поле 32
decayReleaseCurve - поле {MultimodSynth.EnvelopesManager} 106
decayReleaseCurve - поле {MultimodSynth.Envelope} 98
decayReleaseTargetRatio - поле {MultimodSynth.AdsrEnvelope} 32
decayReleaseTargetRatio - поле {MultimodSynth.Envelope} 98
decayTime - поле {MultimodSynth.EnvelopesManager} 106
decayTime - поле {MultimodSynth.Envelope} 99
delay - поле {MultimodSynth.DelayLine} 63
delay - поле {MultimodSynth.DelayManager} 74
delay - поле {MultimodSynth.VariousTimeDelay} 192
delayL - поле {MultimodSynth.PingPongDelay} 146
delayL - поле {MultimodSynth.StereoOffsetDelay} 175
delayL - поле {MultimodSynth.VariousTimeDelay} 192
delayR - поле {MultimodSynth.PingPongDelay} 146
delayR - поле {MultimodSynth.StereoOffsetDelay} 175
delayR - поле {MultimodSynth.VariousTimeDelay} 192
denominator - поле 171
displayValue - поле 244
dryCoeff - поле {MultimodSynth.DelayManager} 74
dryCoeff - поле {MultimodSynth.DistortionManager} 88
envA - поле 197
envB - поле 197
envFilter - поле 197
envelope - поле 99
envelopeAmplitude - поле 107
envelopeAmplitudeFilter - поле 107
envelopes - поле 107
feedback - поле {MultimodSynth.DelayLine} 63
feedback - поле {MultimodSynth.DelayManager} 74
feedback - поле {MultimodSynth.PingPongDelay} 146
filter - поле {MultimodSynth.Filter} 113
filter - поле {MultimodSynth.Voice} 198
filterType - поле 119
filters - поле 119
fmAmountMultiplier - поле 198
freeVoicesIndices - поле 205
frequency - поле {MultimodSynth.Oscillator} 128
frequency - поле {MultimodSynth.SineLFO} 169
gain - поле 181
holdTime - поле 165
hwndSource - поле 154
ic1eq - поле 181
ic2eq - поле 182
instance - поле 155
isActive - поле 141
leftButtonPressed - поле 244
leftCoeff - поле 146
leftDelayCoeff - поле 193
lfo - поле 74
lfoDepth - поле 74
lowPass - поле 88
m0 - поле 182
m1 - поле 182
m2 - поле 182
manager - поле 244
masterVolume - поле 162
masterVolumeFilter - поле 163
maxDelay - поле 75
mixFilter - поле {MultimodSynth.DelayManager} 75
mixFilter - поле {MultimodSynth.DistortionManager} 88
mode - поле {MultimodSynth.DelayManager} 75
mode - поле {MultimodSynth.DistortionManager} 89
modulationType - поле 205
namePrefix - поле 137
normalizationCoeff - поле {MultimodSynth.DCBlocker} 58
normalizationCoeff - поле {MultimodSynth.SoftClipper} 171
noteFrequency - поле {MultimodSynth.Filter} 113
noteFrequency - поле {MultimodSynth.Oscillator} 129
noteToVoicesMapping - поле 205
noteVelocity - поле 198
offsetBufferL - поле 175

offsetBufferR - поле 175
order - поле 92
osCA - поле 198
osCB - поле 199
oscillators - поле 134
parameterCategory - поле
 {MultimodSynth.AudioComponentWithParameters} 37
parameterCategory - поле
 {MultimodSynth.ParameterFactory} 137
parameterPrefix - поле 37
phaseIncrement - поле {MultimodSynth.Oscillator} 129
phaseIncrement - поле
 {MultimodSynth.SampleRateReductor} 165
phaseIncrement - поле
 {MultimodSynth.WaveTableOscillator} 212
phaser - поле {MultimodSynth.Oscillator} 129
phaser - поле {MultimodSynth.SampleRateReductor}
 166
pingPongDelay - поле 75
pitchFine - поле 134
pitchMultiplier - поле
 {MultimodSynth.OscillatorsManager} 134
pitchMultiplier - поле {MultimodSynth.Oscillator} 129
pitchMultiplierFilter - поле 135
pitchSemi - поле 135
plugin - поле {MultimodSynth.AudioProcessor} 39
plugin - поле {MultimodSynth.MidiProcessor} 125
plugin - поле {MultimodSynth.ParameterFactory} 137
plugin - поле {MultimodSynth.PluginEditor} 155
plugin - поле {MultimodSynth.PluginPrograms} 157
plugin - поле {MultimodSynth.UI.EditorView} 233
pressedNotesCount - поле 125
programs - поле 138
q - поле 183
r - поле 58
readOffset - поле 63
readPoint - поле 63
releaseBase - поле 32
releaseCoef - поле 32
releaseRate - поле 33
releaseTime - поле {MultimodSynth.EnvelopesManager}
 107
releaseTime - поле {MultimodSynth.Envelope} 99
responseTimeCoefficient - поле
 {MultimodSynth.DCBlocker} 59
responseTimeCoefficient - поле
 {MultimodSynth.ParameterFilter} 141
rightCoeff - поле 147
rightDelayCoeff - поле 193
sample - поле 166
sampleRate - поле 35
sampleRateReductor - поле 89
samplesForOversampling - поле 163
sin - поле 169
softClipper - поле 89
start.mousePosition - поле 245
start.screen.mousePosition - поле 245
state - поле 33
steps - поле 41
stereoAmount - поле 76
stereoAmountFilter - поле 76
stereoOffsetDelay - поле 76
stroke - поле 245
sustainLevel - поле {MultimodSynth.AdsrEnvelope} 33
sustainLevel - поле {MultimodSynth.EnvelopesManager}
 108
sustainLevel - поле {MultimodSynth.Envelope} 99
target - поле 141
timeFilter - поле 76
trackingCoeff - поле {MultimodSynth.FiltersManager}
 119
trackingCoeff - поле {MultimodSynth.Filter} 113
threshold - поле 171
type - поле 183
value - поле {MultimodSynth.ParameterFilter} 141
value - поле {MultimodSynth.UI.Knob} 245
valueChangedAction - поле 142
variousTimeDelay - поле 77
voicesPool - поле 205
waveTable - поле {MultimodSynth.OscillatorsManager}
 135
waveTable - поле {MultimodSynth.Oscillator} 129
waveTable - поле {MultimodSynth.WaveTableOscillator}
 212
waveTables - поле 212
waveTablesAmount - поле 212
wetCoeff - поле {MultimodSynth.DelayManager} 77
wetCoeff - поле {MultimodSynth.DistortionManager}
 89
wetSign - поле 77
writePoint - поле 64
xm1 - поле 59
ym1 - поле 59