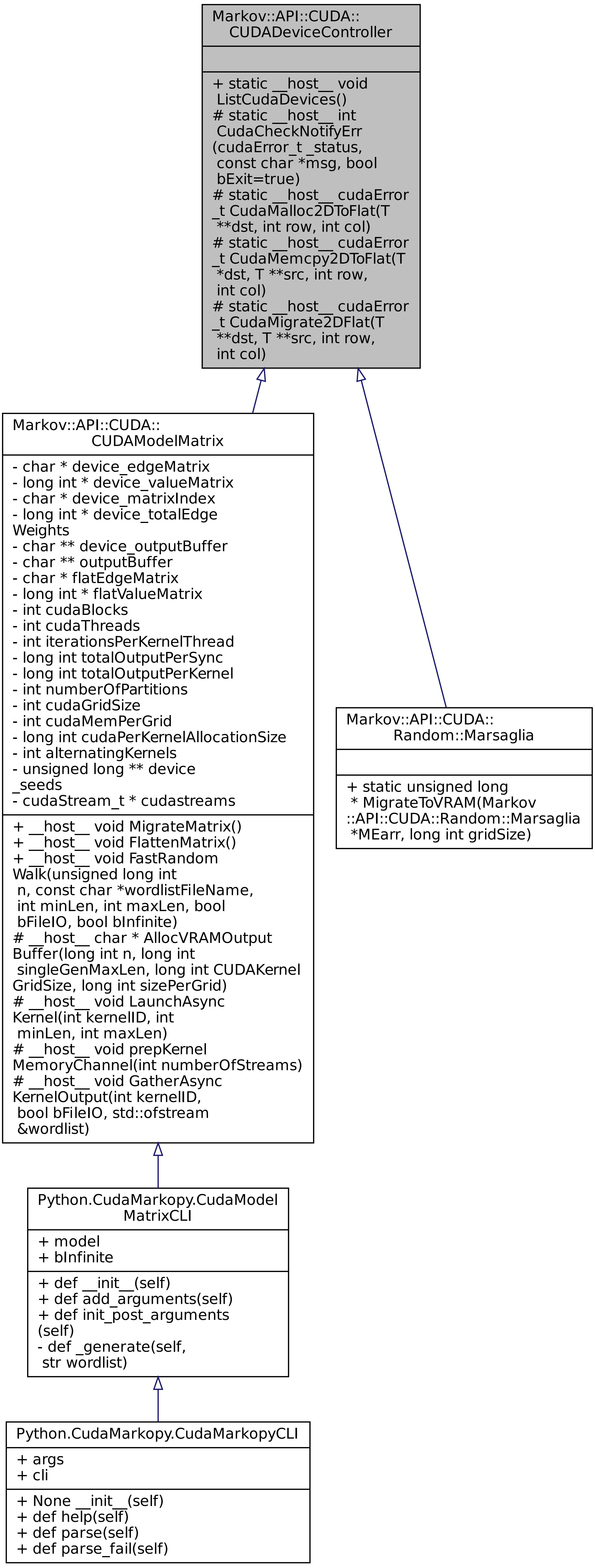
Markov::API::CUDA::CUDADeviceController
+ static __host__ void ListCudaDevices()
# static __host__ int CudaCheckNotifyErr(cudaError_t _status, const char *msg, bool bExit=true)
# static __host__ cudaError_t CudaMalloc2DToFlat(T **dst, int row, int col)
# static __host__ cudaError_t CudaMemcpy2DToFlat(T *dst, T **src, int row, int col)
# static __host__ cudaError_t CudaMigrate2DFlat(T **dst, T **src, int row, int col)
Markov::API::CUDA::CUDAModelMatrix
- char * device_edgeMatrix
- long int * device_valueMatrix
- char * device_matrixIndex
- long int * device_totalEdgeWeights
- char ** device_outputBuffer
- char ** outputBuffer
- char * flatEdgeMatrix
- long int * flatValueMatrix
- int cudaBlocks
- int cudaThreads
- int iterationsPerKernelThread
- long int totalOutputPerSync
- long int totalOutputPerKernel
- int numberOfPartitions
- int cudaGridSize
- int cudaMemPerGrid
- long int cudaPerKernelAllocationSize
- int alternatingKernels
- unsigned long ** device_seeds
- cudaStream_t * cudastreams
+ __host__ void MigrateMatrix()
+ __host__ void FlattenMatrix()
+ __host__ void FastRandomWalk(unsigned long int n, const char *wordlistFileName, int minLen, int maxLen, bool bFileIO, bool bInfinite)
# __host__ char * AllocVRAMOutputBuffer(long int n, long int singleGenMaxLen, long int CUDAKernelGridSize, long int sizePerGrid)
# __host__ void LaunchAsyncKernel(int kernelID, int minLen, int maxLen)
# __host__ void prepKernelMemoryChannel(int numberOfStreams)
# __host__ void GatherAsyncKernelOutput(int kernelID, bool bFileIO, std::ofstream &wordlist)
Markov::API::CUDA::Random::Marsaglia
+ static unsigned long * MigrateToVRAM(Markov::API::CUDA::Random::Marsaglia *MEarr, long int gridSize)
Python.CudaMarkopy.CudaModelMatrixCLI
+ model
+ bInfinite
+ def __init__(self)
+ def add_arguments(self)
+ def init_post_arguments(self)
- def _generate(self, str wordlist)
Python.CudaMarkopy.CudaMarkopyCLI
+ args
+ cli
+ None __init__(self)
+ def help(self)
+ def parse(self)
+ def parse_fail(self)