

Библиотека ByteProcess

Класс ByteProcessor

Header-файл: ByteProcessor.h

Описание:

Класс ByteProcessor представляет из себя обработчик байт последовательностей, где каждый байт обрабатывается по алгоритму:

Байт состоит из битов type (6-ой и 7-ой) и битов data (0-5), хранящие данные:

- Если type = 0b00, то data – беззнаковое целое;
- Если type = 0b01, то data – знаковое целое;
- Если type = 0b10, то data – одна из букв латинского алфавита
 - В случае, если data > 25 записывает в буфер OutChars "UND"
- Если type = 0b11, то в буфер OutChars записывается "NULL"

Класс использует интерфейсы Source и Sink, которые представляют устройства ввода и вывода соответственно, для чтения и записи данных.

Данный обработчик побайтово считывается данные из Source, обрабатывает в соответствии с упомянутым алгоритмом и записывает данные в Sink.

Публичные функции:

thread::id ByteProcessor::StartNewWork(Source* InSource, Sink* InSink)

Запускает новый поток по обработке байт последовательностей с устройствами ввода/вывода *InSource* и *InSink*.

Возвращает уникальный идентификатор созданного потока.

void ByteProcessor::JoinWorkIfExists(thread::id InThreadID)

Останавливает вызывающий поток до окончания выполнения потока по идентификатору InThreadID.

Если потока не существует, то не делает ничего.

void ByteProcessor::StopExistingWork(thread::id InThreadID)

Останавливает потока по идентификатору InThreadID.

Если потока не существует, то не делает ничего.

void ByteProcessor::StopAllExisitingWorks()

Останавливает все работающие на момент вызова потоки.

Класс Source

Header-файл: Source.h

Описание:

Данный класс является интерфейсом для всех устройств ввода, его функции должны быть реализованы в наследуемом классе. Используется для работы ByteProcessor.

Публичные функции:

```
int64_t Source::Read(char* InBuffer, int64_t InSize);
```

Чистая виртуальная функция чтения данных в размере InSize из буфера InBuffer.

Возвращает количество считанных байтов или -1, если произошла ошибка.

Класс Sink

Header-файл: Sink.h

Описание:

Данный класс является интерфейсом для всех устройств ввода, его функции должны быть реализованы в наследуемом классе. Используется для работы ByteProcessor

Публичные функции:

```
int64_t Sink::Write(char* InBuffer, int64_t InSize);
```

Чистая виртуальная функция записи данных в размере InSize в буфер InBuffer.

Возвращает количество записанных байтов или -1, если произошла ошибка.