**Функциональная спецификация**

Библиотека для создания и управления контейнером, который позволяет хранить в себе папки и файлы в зашифрованном виде.

**Задание**

Написать библиотеку, реализующую зашифрованный контейнер с предоставлением API, которое позволяет читать/записывать туда папки и файлы.

**Требования и возможности**

Библиотека должна работать с файлами на перезаписываемом носителе. Целевым контейнером для хранения информации, для работы с которым предназначена библиотека, представляет из себя 2 файла: файл базы данных и бинарный файл с данными файлов контейнера.

Встраиваемая БД SQLite позволяет добиться целостности данных благодаря механизму фиксации транзакций, который гарантирует только полную запись данных в базу даже при отключении питания. Безопасность для конфиденциальности хранимых данных гарантируется применением передовой системы шифрования AES.

Для поддержки автономности базы данных используется встраиваемая СУБД SQLite версии 3. Для осуществления шифрации используется библиотека Crypto++ версии 5.6. Эти библиотеки должны входить в комплект сборки целевого приложения вместе с настоящей библиотекой.

**Действия при работе с библиотекой (Use Cases)**

Данная библиотека может быть подключена к системе, для которой одной из целей является локальное хранение данных в одном файле, а также обеспечение безопасности этих данных.

Для использования функций библиотеки предоставляется лаконичный API, основой которого являются интерфейсы классов контейнера, файла контейнера и папки контейнера, а также вспомогательные интерфейсы и типы. Управление файловой системой осуществляется с помощью методов соответствующих интерфейсов (т.е. для управления папками используется интерфейс папки, для управления файлами и их данными используется интерфейс файла).

**API и поведение функций библиотеки**

Библиотека предоставляет API для создания и изменения содержимого контейнера с папками и файлами. Основой реализации API является интерфейс контейнера. Объект с интерфейсом контейнера можно получить только с помощью внешних методов API (конкретно - метода Connect).

API библиотеки содержит функции создания/разрыва соединения с БД, классы файлов и папок файловой системы контейнера, методы для записи/чтения данных файлов контейнера, создания/удаления и других модификаций элементов его файловой системы.

**Возможности**

Благодаря описанному выше API возможно создавать множество контейнеров, которые не будут зависеть друг от друга, и управлять ими одновременно. Все методы интерфейсов при невозможности выполнения своей задачи бросают исключение строго определенного типа. Все типы данных аргументов функций API библиотеки определены в .h-файлах, предоставляемых вместе с библиотекой.

**Дополнительная информация по использованию библиотеки**

Все методы интерфейсов при невозможности выполнения своей задачи бросают исключение строго определенного типа. Если исключение произошло из-за нехватки памяти или проблем с носителем, на котором располагаются контейнеры (или по другой «физической» причине), это исключение не будет перехвачено библиотекой, чтобы дать возможность вызывающей стороне самостоятельно решить проблему.

Каждый возвращенный указатель на интерфейс – валидный. Функции API не возвращают не валидных указателей – вместо этого будет выброшено соответствующее исключение, в котором описана причина невозможности возвращения валидного указателя. Каждый возвращенный указатель должен быть обработан вызывающей стороной, включая его удаление: библиотека не является владельцем созданных объектов!

Имя файла считается валидным, если в нем отсутствует нуль-символ, а также следующие символы: \ / : \* ? \ " < > |

Символом разделителя пути может быть один из вышеуказанных символов, не включая нуль-символ.

Методам чтения/записи данных из/в файлы контейнера нужно передавать только те файловые потоки, которые были открыты в двоичном режиме.

Подробности использования конкретных функций API описаны в комментариях в заголовочных файлах библиотеки.