ГЛАВА 3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

Реализация интерфейса COM-сервера

В главе 2 была проведена разработка интерфейса COM-сервера и подготовка к его реализации. Перед непосредственным написанием кода данного модуля следует уточнить алгоритм его работы как обособленной программы.

Согласно ранее упомянутым спецификациям Component Object Model, вначале необходимо запустить сам COM-сервер. В соответствии со спецификациями, приведенными в Приложении 1, StabMed при запуске автоматически регистрируется в качестве COM-сервера, однако может быть запущен и по запросу клиента. Далее, согласно описанным в [6] приемам, мы должны инициализировать программный модуль, который обеспечит корректную работу Python с Component Object Model. После при помощи метода Dispatch() необходимо создать объект, через который в дальнейшем и будут происходить обращения к методам, которые поддерживает COM-интерфейс. Таким образом, получаем алгоритм, схема которого представлена на рис. :.

Программная реализация данного алгоритма находится в файле com.py (Приложение 2). В переменной COM\_SERVER\_NAME хранится ProgID необходимого интерфейса.

Реализация

Разработка тестов для отладки программных модулей

Тестирование является важнейшим элементов процесса реализации

Для отладки программных модулей следует разработать тестовый чек-лист для каждого из них в соответствии с требованиями и реализацией. Ниже приведены чек-листы для всех частей автоматизированной системы.

Чек-лист для COM-интерфейса

|  |  |
| --- | --- |
| Тестируемый функционал | Ожидаемый результат |
| Соединение с COM-сервером | Переход окна StabMed в статус «активное», отсутствие программных исключений |
| Метод GetTestCount | Возвращает количество проведенных обследований |
| Метод GetTestID | Возвращает TestID проведенного обследования по порядку |
| Метод GetLastTestID | Возвращает TestID последнего проведенного обследования |
| Метод GetActiveTestID | Возвращает TestID обследования, обрабатываемого в настоящий момент |
| Метод GetOpenedTestCount | Возвращает количество открытых в настоящий момент обследований |
| Метод GetOpenedTestID | Возвращает TestID проведенного обследования по порядку открытия |
| Метод GetProbesCount | Возвращает количество проб в обследовании или COM\_NOT\_FOUND, если обследования TestID нет |
| Метод GetProbeID | Возвращает ProbeID пробы в обследовании по TestID и порядковому номеру пробы |
| Метод GetChannelCount | Возвращает количество каналов, записанных в пробе ProbeID или COM\_NOT\_FOUND, если пробы ProbeID нет |
| Метод GetChannelID | Возвращает ChannelID канала в пробе ProbeID и порядковому номеру канала |
| Метод GetChannelType | Возвращает тип канала по его идентификатору |
| Метод GetSignal | Возвращает сигнал по идентификатору пробы и канала |
| Метод GetPatientName | Возвращает ФИО пациента по идентификатору обследования |
| Метод GetPatientBirthday | Возвращает дату рождения пациента по идентификатору обследования |
| Метод GetTestName | Возвращает название методики по идентификатору обследования |
| Метод GetTestDateTime | Возвращает дату и время проведения обследования по идентификатору обследования |
| Метод GetProbeName | Возвращает название пробы по идентификатору пробы |

Чек-лист для проверки хранилища данных

|  |  |
| --- | --- |
| Тестируемый функционал | Ожидаемый результат |
| Начальное создание базы данных (CREATE) | Создание таблиц со связями и атрибутами в соответствии со спецификациями |
| Добавление в базу новой записи (INSERT) | Появление в базе данных новой записи |
| Экспорт записей из базы (SELECT) | Генерирование текстового файла с результатом запроса в базу данных |

Чек-лист для проверки модуля математической обработки сигналов

|  |  |
| --- | --- |
| Тестируемый функционал | Ожидаемый результат |
| Осуществление быстрого преобразования Фурье над массивом стабилографических данных | Массив значений, отображающих значения функции после преобразования Фурье |
| Вычисление косвенных стабилографических параметров | Переменным присвоены соответствующие значения, вычисленные по формулам в соответствии со спецификациями |
| Нахождение в спектре дыхательной и сердечной частот | Значения совпавших частот |
| Выделение частот с аномально высокой мощностью | Диапазон частот, записанный в соответствующую переменную |

Чек-лист для проверки графического пользовательского интерфейса

|  |  |
| --- | --- |
| Тестируемый функционал | Ожидаемый результат |
| Отображение рабочего окна | Появление окна приложения на экране |
| Кнопка «Загрузить сигнал» | Загрузка сигнала и его отображение в соответствующей зоне окна |
| Кнопка «Удалить сигнал» | Исчезновение сигнала из соответствующих окон |
| Панель выбора источника сигнала | Отображение в соответствующих зонах окна сигнала, выбранного на панели |
| Маркировка сигнала как патологического | Запись информации о наличии признаков патологии в сигнале в базу данных |

Компонентное тестирование автоматизированной системы

В процессе отладки системы целесообразно начать процесс тестирования как можно раньше, поскольку в рамках относительно небольшого программного модуля найти и исправить ошибку проще, чем при изучении системы в целом.

Для проведения полноценного компонентного тестирования следует составить тест-кейсы для каждой требуемой проверки в соответствии с требованиями и представленными выше чек-листами. В ряде случаев тест-кейсы для нескольких проверок однотипны, поэтому конкретные описания тест-кейсов для них не приведены.

Тестирование интерфейса Component Object Model:

Интеграционное тестирование автоматизированной системы

Выводы и результаты