Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**MigrateIMS**

**Конвертер для миграции абонентов АТС**

Пояснительная записка к курсовому проекту по дисциплине

«Технология разработки программного обеспечения»

Выполнил студент

специальности 09.03.01

Бубенщиков Олег Юрьевич

01.05.2019

г. Среднеуральск 2019

**Реферат**

Курсовой проект состоит из 44 страниц, содержит 3 таблицы и один рисунок.

В пояснительной записке описаны основные этапы разработки программного обеспечения, реализованные автором.

Целью проекта являлось получение навыков в области разработки программного обеспечения, полученных в процессе изучения методологии разработки в рамках учебного курса.

В результате проведённой работы были достигнуты следующие цели:

- Разработана техническая документация к программному продукту.

- Выполнена разработка программного обеспечения.

- Проведены тестовые испытания.

Разработанное ПО - конвертер «MigrateIMS» позволяет автоматизировать процесс переноса абонентской ёмкости с исходной телефонной станции на целевую систему, с минимальными временными затратами. В процессе переноса, для каждого номера выполняется обработка и преобразование его атрибутов (Услуг и другой служебной информации) по заданным правилам.

Разработка выполнена на языке программирования Python 3.x. С учётом кросс-платформености языка, С конвертером можно работать в операционных системах типа ОС Windows и ОС Linux. Для работы конвертера не требуется установка и использование СУБД. Вся обработка и хранение данных производится в оперативной памяти компьютера. Соответственно, на целевой ОС достаточно установить интерпретатор языка Python 3.x и требуемые библиотеки.

Использование разработанного ПО предназначено для отрасли связи.

Применение конвертера позволяет оператору связи на порядок снизить временные и трудовые затраты персонала в процессе переноса абонентских данных с исходной АТС на целевую систему. Оператор связи получает возможность оперативно создавать поэтапные графики переключения, когда переключается не вся абонентская ёмкость сразу, а частями.

Пояснительная записка к курсовому проекту выполнена в редакторе MS Word.

Оглавление

[1 Техническое задание 4](#_Toc8278175)

[1.1 Введение 4](#_Toc8278176)

[1.2 Основание для разработки 4](#_Toc8278177)

[1.3 Назначение разработки 4](#_Toc8278178)

[1.4 Технические требование к программе или программному изделию 5](#_Toc8278179)

[1.5 Требования к программной документации 8](#_Toc8278180)

[1.6 Технико-экономические показатели 8](#_Toc8278181)

[1.7 Стадии и этапы разработки 9](#_Toc8278182)

[2 Соглашение о требованиях 10](#_Toc8278183)

[2.1 Описание программного изделия 10](#_Toc8278184)

[2.2. Цели 14](#_Toc8278185)

[2.3 Стратегия 15](#_Toc8278186)

[2.4 Используемые материалы 23](#_Toc8278187)

[2.5 Передача заказчику и ввод в действие 23](#_Toc8278188)

[2.6 Тактика 24](#_Toc8278189)

[2.7 Извещение об изменении календарных сроков 25](#_Toc8278190)

[3 Спецификации 26](#_Toc8278191)

[3.1 Внешняя спецификация 26](#_Toc8278192)

[3.2 Внутренняя спецификация 29](#_Toc8278193)

[4 Тестирование 34](#_Toc8278194)

[4.1 Общие принципы тестирования 34](#_Toc8278195)

[4.2 Организация испытаний программных изделий 34](#_Toc8278196)

[4.3 Виды испытаний программного изделия. Стадии испытаний 34](#_Toc8278197)

[4.4 Режимы испытаний программ 34](#_Toc8278198)

[4.5 Категории испытания программного изделия 34](#_Toc8278199)

[4.6 Технология тестирования, классы эквивалентности 35](#_Toc8278200)

[2.7 Построение тестов 39](#_Toc8278201)

[5 Руководство системного программиста 41](#_Toc8278202)

[5.1 Общие сведения о программе 41](#_Toc8278203)

[5.2 Структура программы 43](#_Toc8278204)

[5.3 Настройка программы 43](#_Toc8278205)

[5.4. Проверка программы 43](#_Toc8278206)

[3.5 Дополнительные возможности 43](#_Toc8278207)

[3.6 Сообщения системному программисту 44](#_Toc8278208)

[6 Заключение 44](#_Toc8278209)

# 1 Техническое задание

## 1.1 Введение

Разрабатываемое программное обеспечение – конвертор IMS (далее – MigrateIMS), предназначен для конвертирования исходного формата данных в новый формат данных, с использованием заданных правил преобразования.

Конвертор MigrateIMS предназначен для помощи техническому персоналу телефонных станций (далее – АТС), для выполнения автоматического конвертирования исходной базы абонентских номеров, существующего телефонного коммутационного оборудования (далее - АТС) в формат, используемый на виртуальной платформе коммутации (далее - vIMS), в процессе подготовки к переносу (далее – миграции).

## 1.2 Основание для разработки

Разработка ведётся на основании следующих документов:

Данное техническое задание.

## 1.3 Назначение разработки

Функциональное и эксплуатационное назначение программы:

Данная программа предназначена для преобразования исходной базы данных, в новый формат, пригодный для импортирования в БД программного коммутатора vIMS.

## 1.4 Технические требование к программе или программному изделию

**1.4.1 Требования к функциональным характеристикам**

- Программа может работать как в консольном, так и в графическом режимах.

- В графическом режиме должны поддерживаться работа с клавиатурой и манипулятором типа «мышь».

- Программа должна генерировать данные, в формате, пригодном для последующего импорта в БД vIMS.

- Программа должна иметь возможность поддерживать работу с разными типами и форматами входных данных в зависимости от типа АТС.

- Правила интерпретации атрибутов из исходных структур данных в новые структуры, должны описываться в конфигурационных файлах (далее - шаблонах).

- Программа должна иметь возможность работать с разными наборами шаблонов.

- Программа должна считывать настройки и правила обработки (шаблоны) из конфигурационного файла.

- Программа должна иметь возможность выполнять конфигурирование шаблонов и формирование конфигурационного файла в графическом режиме.

- Для каждого сеанса работы должен формироваться журнал, в который записываются все служебные сообщения, а также информация об аварийных ситуациях, связанная с выполнением процесса обработки или возникающих проблемах в обработке данных. Для каждой записи генерируется штамп времени и уровень логирования.

- Программа должна иметь возможность просмотра журнала в графическом режиме

**1.4.2 Требования к надёжности**

- надежность выдачи правильных результатов проверяется выполнением интеграционных тестов.

- Контроль выходных данных проверяется на предмет вхождения каждого информационного атрибута в разрешённый диапазон.

- Ошибочные действия пользователя, неверные значения конфигурационных данных, неверный формат или тип входных данных записываются в лог-файл, а также выводятся в консоль или отображаются в графическом интерфейсе для информирования пользователя в процессе работы.

**1.4.3 Требования к эксплуатации**

- Пользователь должен иметь уверенные навыки работы с компьютером.

- Персонал, выполняющий работы по конвертированию данных должен иметь достаточную квалификацию и знания по работе с продуктом vIMS.

- Уметь выполнять групповой импорт абонентов в БД vIMS, знать тип и формат данных формируемых для импорта в БД vIMS.

- Уметь разбираться в структуре входных данных, подаваемых на вход конвертера.

- Уметь составлять правила конвертирования.

**1.4.4 Требования к составу и параметрам технических средств**

Программное обеспечение разрабатывается для персональной ЭВМ (IBM PC - совместимой) с характеристиками не ниже:

- процессор с частотой 1ГГц;

- объем ОЗУ 4Гб;

- графический адаптер SVGA;

- клавиатура и манипулятор типа «мышь»;

- сетевой адаптер Ethernet;

- не менее 1Мб свободного места на жестком диске для самой программы, а также место для файла(-лов) обрабатываемых исходных данных, размер которых определяется самим пользователем.

**1.4.5 Требования к информационной и программной совмести**

- Операционная система (Windows-10, Linux).

- Язык программирования - Python с версией не ниже 3.6.х.

## 1.5 Требования к программной документации

Комплект документации поставляется в составе дистрибутива. Документация оформляется в структуру файлов и каталогов в формате *reStructuredText* и должна содержать следующие пункты:

- Описание процедуры установки необходимой версии интерпретатора Python.

- Описание процедуры установки программы.

- Общее описание процесса конвертирования.

- Описание синтаксиса командной строки.

- Описание подготовки исходных данных в зависимости от типа конвертируемой АТС.

- Описание структуры шаблонов с правилами преобразования и заполнения.

- Описание вариантов использования конвертера.

- Формат использования документации - статические HTML страницы.

## 1.6 Технико-экономические показатели

Программа является узкоспециализированной и не имеет аналогов.

Использование данной программы для конвертирования больших массивов данных в разы сокращает временной интервал процедуры подготовки и миграции абонентов, что уменьшает временные и ресурсные затраты персонала, а также позволяет максимально уменьшить время простоя связи для абонентов переключаемой АТС.

## 1.7 Стадии и этапы разработки

а) Ознакомление со стандартами и протоколами VIMS, используемыми

для миграции абонентов в БД, а также структурой исходных данных переключаемой АТС:

- Срок выполнения — Ноябрь 2018г.

б) Разработка концептуальной модели функционирования будущей

программы:

- Срок выполнения — Ноябрь 2018г.

в) Непосредственная разработка законченного программного

средства (рабочий проект):

- Срок выполнения — Декабрь 2018г. - Январь 2019г.

г) Отладка и тестирование:

- Срок выполнения — Январь 2019г.

д) Внедрение:

- Срок выполнения — не определён.

# 2 Соглашение о требованиях

## 2.1 Описание программного изделия

**2.1.1 Наименование и шифры изделия**

2.1.1.1 Полное наименование изделия

Переносимая программа конвертирования абонентской базы разных типов АТС в формат для импорта в виртуальный узел коммутации vIMS.

2.1.1.2 Сокращенные наименования

MigrateIMS.

2.1.1.3 Шифры изделия

Отсутствуют.

2.1.1.4 Шифры проекта

Отсутствуют.

**2.1.2 Краткое описание изделия**

Конвертор MigrateIMS предназначен для помощи техническому персоналу телефонных станций (далее – АТС), для выполнения автоматического конвертирования исходной базы абонентских номеров, существующего телефонного коммутационного оборудования (далее - АТС) в формат, используемый на виртуальной платформе коммутации (далее - vIMS), в процессе подготовки к переносу (далее – миграции).

**2.1.3 Сведения об авторском праве**

Не требуется.

**2.1.4 Результирующие компоненты изделия**

Результирующие компоненты изделия перечислены в таблице 2.1.

Таблица 2.1 - Результирующие компоненты изделия

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Обозначения:  Основное изделие – не используется для создания других изделий  Вспомогательное изделие – используется для создания других изделий  Уровень поддержки 1: удовлетворяются заявки на исправление дефектов; возможно сообщение об изменениях; принимаются заявки на расширение функциональных возможностей изделия  Уровень поддержки 2: удовлетворяются заявки на исправление дефектов; возможно сообщение об изменениях; заявки на расширение не принимаются  Уровень поддержки 3: удовлетворяются заявки на исправление дефектов  Р – группа разработки |  | Формируется целиком | Модифицируется | Распространяется | Не распространяется | Ответственная группа |
| Спецификации |  |  |  |  |  |
| Внешняя спецификация | Х |  |  | Х | Р |
| Внутренняя спецификация | Х |  |  | Х | Р |
| Спецификация испытаний (не надо) |  |  |  |  |  |
| Спецификация сопровождения (не надо) |  |  |  |  |  |
| Другие спецификации |  |  |  |  |  |
| Документация |  |  |  |  |  |
| Техническое описание системы |  |  |  |  |  |

Продолжение таблицы 2.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | Справочное руководство | Х |  | Х |  | Р |
| Справочный буклет |  |  |  |  |  |
| Тип  Изде-  лия | Основ-ное | Х | Началь-ный  Уровень  под-держки | | Руководство оператора |  |  |  |  |  |
| Указатель системных сообщений |  |  |  |  |  |
| Вспо-мога-тельное |  | Информационный листок выпуска |  |  |  |  |  |
|  | | | 1 | Х | Другие печатные издания |  |  |  |  |  |
| 2 |  | Рекламные материалы |  |  |  |  |  |
|  |  | Программное обеспечение |  |  | Х |  |  |
|  |  | Листинги |  |  |  |  |  |
|  |  | Исходные модули | Х |  | Х | Х | Р |
|  |  | Объектные модули |  |  |  |  |  |

Продолжение таблицы 2.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Контрольные примеры | Х |  |  | Х | Р |
|  |  | Средства разработки |  |  |  |  |  |
|  |  | Прочие средства |  |  |  |  |  |

## 2.2. Цели

Поставленной задачей было написание программы, представляющей собой систему по обработке и преобразованию исходных абонентских данных АТС разного типа в заранее определённый формат выходных данных, используемый для последующего добавления в БД vIMS.

**2.2.1 Согласование заявок на проверку**

2.2.1.1 Отклоненные заявки

Отсутствуют.

2.2.1.2 Принятые заявки

Отсутствуют.

**2.2.2 Согласование заявок на расширение функциональных**

возможностей изделия

2.2.2.1 Отклонённые заявки

Отсутствуют.

2.2.2.2 Принятые заявки

Отсутствуют.

**2.2.3 Согласование заявок на внесение исправлений**

2.2.3.1 Отклонённые заявки

Отсутствуют.

**2.2.4 Согласование планов**

2.2.4.1 Исключённые пункты плана

Отсутствуют.

2.2.4.2 Включенные пункты плана

Отсутствуют.

**2.2.5 Перечень требований пользователя**

Заказчику требуется:

- Программа должна работать под любой операционной системой семейства Windows или Linux.

- Программа должна поддерживать гибкое формирование исходных конфигурационных данных, в том числе, для разных типов АТС, с учетом доступных в системе обработчиков исходных данных для данного типа АТС.

**2.2.6 Рассмотренные альтернативы**

Альтернатив нет. Приложение является узкоспециализированным.

**2.2.7 Окупаемость капиталовложений**

Капиталовложений нет.

## 2.3 Стратегия

**2.3.1 Соглашения относительно представления материала**

2.3.1.1 Обозначения

В данном документе не используется никаких специальных обозначений.

2.3.1.2 Терминология

Вся специальная терминология определяется в контексте данного документа.

**2.3.2 Генерируемое программное обеспечение**

Не используется.

**2.3.3 Системное программное обеспечение**

Программа состоит из следующих функциональных блоков (рис. 2.1)

2.3.3.4

Блок обработки входных данных АТС

2.3.3.3

Блок интерфейса пользователя

2.3.3.5

Блок представления выходных данных, отчётов

2.3.3.2

Блок конфигураций

2.3.3.1

Блок

вариантов использования

2.3.3.6

Блок чтения и записи данных

Рисунок 2.1

2.3.3.1 Общие характеристики «Блок вариантов использования»

2.3.3.1.1 Внешние ограничения

2.3.3.1.1.1 Действующие стандарты

Не используются.

2.3.3.1.1.2 Ограничения на совместимость

Не существует программных изделий, совместимых с данным программным продуктом.

2.3.3.1.1.3 Программные ограничения

Данная программа работает с любыми операционными системами семейства Windows или Linux, поддерживающих работу языка программирования Python не ниже версии 3.х.

2.3.3.1.1.4 Аппаратные ограничения

Для эксплуатации разрабатываемого программного обеспечения IBM-PC совместимый компьютер, поддерживающий архитектуру х86.

2.3.3.1.2 Внешние характеристики

2.3.3.1.2.1 Результаты работы блока вариантов использования

Обработанные данные абонентов формируются в список и передаются в блок представления выходных данных.

2.3.3.1.2.2 Процессы блока вариантов использования

Блок вариантов использования, выполняет обработку принятого списка номеров исходного формата, в новый формат, с учетом использования правил преобразования.

2.3.3.1.2.3 Входы блока вариантов использования

- Набор правил использования.

- Cписок обрабатываемых номеров.

2.3.3.1.3 Эргономические характеристики

2.3.3.1.3.1 Безопасность и секретность

Программное обеспечение не предназначено для распространения в открытом виде. За сохранение конфиденциальной информации и распространение обрабатываемых данных несет ответственность пользователь.

2.3.3.1.3.2 Надёжность

Некорректность входных данных или содержание некорректных данных в конфигурации может приводить к сбою системы. В случае появления сбоя, для восстановления работоспособности выполняется устранением ошибок во входных данных или в конфигурации, на основании анализа сбоя. В программе реализуется контроль типа и значений в вводимых данных и конфигурации.

2.3.3.1.3.3 Рестарт

В программе отсутствует информация, которую периодически сохранять для возобновления работы после отказа блока питания компьютера. Все необходимые для работы данные содержатся в конфигурационных файлах, которые из самой программы не изменяются.

2.3.3.1.3.4 Соответствие требованиям заказчика

Система должна соответствовать требованиям технического задания.

2.3.3.1.3.5 Рабочие характеристики

Программа не накладывает никаких ограничений на конфигурацию, помимо ограничений, определяемых оборудованием.

2.3.3.1.3.6 Обеспечение эксплуатации

Никаких требований к условиям эксплуатации не выдвигается. Для выполнения работы требуется квалифицированный персонал.

Работа программного изделия начинается с запуска исполняемого файла. Вся информация о ходе работы записывается в лог-файл *migrate.log.* Завершение работы программы происходит автоматически при завершении обработки всех исходных данных в консольном режиме работы или действиями оператора в графическом режиме.

2.3.3.1.3.7 Мобильность

Данный программный продукт распространяется с открытым исходным кодом.

2.3.3.1.4 Внутренние характеристики

2.3.3.1.4.1 Удобство сопровождения

Программы не компилируется. Команды выполняются непосредственно интерпретатором языка Python, что дает разработчику или человеку, занимающемуся сопровождением данного ПО, достаточно полную информацию о работе программы.

2.3.3.1.4.2 Алгоритмы

Подлежат описанию во внутренней спецификации.

2.3.3.2 Общие характеристики функции «Блок конфигураций»

Все пропущенные пункты см. в п.2.3.3.1.

2.3.3.2.2 Внешние характеристики

2.3.3.2.2.1 Результаты работы блока конфигураций

Формируется объект, содержащий все конфигурационные данные.

2.3.3.2.2.2 Процессы блока конфигураций

Блок конфигураций считывает о обрабатывает основной конфигурационный файл, в котором указаны ссылки на загрузку исходных данных и необходимых шаблонов конфигураций. Считанные данные формируются как атрибуты объекта конфигурации.

2.3.3.2.2.3 Входы блока конфигураций

В качестве входных данных принимается словарь с исходными конфигурационными данными.

2.3.3.3 Общие характеристики «Блок интерфейса пользователя»

Все пропущенные пункты см. в п.2.3.3.1.

2.3.3.3.2 Внешние характеристики

2.3.3.3.2.1 Результаты работы блока интерфейса пользователя

Результатом работы блока «Интерфейс пользователя» являются:

- отображение на экране правил конфигурации и флагов(режимов) обработки подготавливаемых данных.

- формирование и редактирование правил обработки в графическом режиме.

- отображение процесса обработки(конвертирования) исходных данных.

- просмотр лог-файла.

2.3.3.3.2.2 Процессы блока интерфейса пользователя

Блок «Интерфейс пользователя» выполняет следующие процессы:

- Преобразование в удобное для понимания представление конфигурационных данных и управления ими.

- Обработка клавиш навигации.

2.3.3.2.2.3 Входы блока интерфейса пользователя

- Значения полей и флагов управления, введённые пользователем в графическом окне.

- выходные данные потока обработки.

2.3.3.4 Общие характеристики «Блок обработки входных данных АТС»

Все пропущенные пункты см. в п.2.3.3.1.

2.3.3.4.2 Внешние характеристики

2.3.3.4.2.1 Результаты работы блока обработки входных данных АТС

Формируется список объектов исходной базы данных.

2.3.3.4.2.2 Процессы блока обработки входных данных АТС

Блок «Обработки входных данных АТС» выполняет следующие процессы:

- Обработка исходной базы номеров и вычисление необходимых свойств и атрибутов каждого номера.

- Перенос обработанных данных абонентов в новую модель, использующуюся для дальнейшей обработки в конвертере.

2.3.3.4.2.3 Входы блока обработки входных данных АТС

В качестве входных данных принимается исходная база номеров АТС и конфигурационные данные.

2.3.3.5 Общие характеристики «Блок представления выходных данных»

Все пропущенные пункты см. в п.2.3.3.1.

2.3.3.5.2 Внешние характеристики

2.3.3.5.2.1 Результаты работы блока представления выходных данных

На основании обработанной базы номеров, формируются списки объектов, с требуемым представлением данных, для последующего использования в импорте на vIMS.

2.3.3.5.2.2 Процессы блока представления выходных данных

Используются следующие процессы:

- формирование объектов из обработанной базы номеров с использованием шаблонов конфигурации, в формат, необходимый для последующего импорта в БД vIMS.

2.3.3.5.2.3 Входы блока представления выходных данных

В качестве входных данных принимается обработанный список номеров и шаблоны конфигурации.

2.3.3.6 Общие характеристики функции «Блок чтения и записи данных»

Все пропущенные пункты см. в п.2.3.3.1.

2.3.3.6.2 Внешние характеристики

2.3.3.6.2.1 Результаты работы блока чтения и записи данных

Результатом работы «Блока чтения и записи данных» является запись или чтение указанных файлов в нужном формате.

2.3.3.6.2.2 Процессы блока чтения и записи данных

В блоке чтение и записи данных выполняются следующие процессы:

- чтение указанных файлов в заданном формате.

- запись данных в указанном формате.

2.3.3.6.2.3 Входы блока чтения и записи данных

Информация, подающаяся на входы блока чтения и записи данных, зависит от направления движения этих данных:

- Для чтения, на входы подаются данные о расположении запрашиваемого файла.

- Для записи – имя файла и данные.

2.3.4 Внутренние ограничения

Единственные ограничения, которые накладываются на данное ПО, связаны с ограничениями операционной системы (частота процессора, количество оперативной памяти и т.д.), влияющих на время обработки больших массивов данных.

## 2.4 Используемые материалы

**2.4.1 Справочные документы**

Отсутствуют.

## 2.5 Передача заказчику и ввод в действие

**2.5.1 Средства защиты права собственности на изделие**

Соблюдение прав собственности не требуется.

**2.5.2 Ресурсы, обеспечивающие ввод в действие**

Для выполнения работы требуется квалифицированный персонал, знающий структуру данных переносимой АТС, а также, обладающий навыками работы на vIMS.

ПО разворачивается и работает на любой ОС семейства Windows или Linux, поддерживающих работу языка программирования Python не ниже версии 3.х.

**2.5.3 Носители информации**

В качестве носителя информации может использоваться любое доступное устройство хранения с возможность чтения и записи, а также разрешённых прав для доступа к данному носителю от имени пользователя запускающего ПО.

## 2.6 Тактика

**2.6.1 Взаимосвязи**

2.6.1.1 Требуемые взаимосвязи

Не выдвигаются требования к другим изделиям.

2.6.1.2 Обеспечение взаимосвязи

Требования других изделий к программе отсутствуют.

**2.6.2 Техническая ревизионная комиссия**

Создание ревизионной комиссии не требуется.

**2.6.3 Проверка изделия**

2.6.3.1 Уровни испытаний

Уровни испытаний приведены в таблице 2.2.

Таблица 2.2 - Уровни испытаний

|  |  |  |  |
| --- | --- | --- | --- |
| Категория испытаний | Класс испытаний | | |
| А | В | С |
| Демонстрация в действии |  |  | / |
| Аттестация |  | Р | / |
| Полная функциональная проверка |  |  | / |
| Проверка новых свойств |  |  | / |
| Эксплуатационные испытания |  |  | / |
| Испытания надежности |  |  | / |
| Проверка устойчивости |  |  | / |
| Возвратная проверка |  |  | / |
| Пусковые испытания |  |  | / |
| Испытания конфигураций |  | Р | / |

Продолжение таблицы 2.2

|  |
| --- |
| Режимы испытаний  I – проводятся группой испытаний ( )  II – контролируются группой испытаний ( )  III – группа испытаний не участвует ( Х ) |
| Подразделения, проводящие испытания  Р – группа разработки  И – группа испытаний  О – группа обслуживания  / - испытания исключены |

2.6.3.2 Эталоны для сравнения

Отсутствуют.

**2.6.4 Обеспечение поддержки**

2.6.4.1 Мероприятия, обеспечивающие продвижение программного обеспечения на рынок

– не производятся.

2.6.4.2 Мероприятия, связанные с обучением

Персонал, выполняющий миграцию номеров с использованием конвертера, требуется обучить приемам работы с ним и вариантам использования.

2.6.4.3 Средства, обеспечивающие модернизацию программного изделия

Не предусматриваются.

## 2.7 Извещение об изменении календарных сроков

Не требуется

# 3 Спецификации

## 3.1 Внешняя спецификация

Входные данные:

- Тип АТС — выбор нужного типа обрабатываемой станции из выпадающего меню.

- Тип Номера — выбор нужного типа номера обработки из выпадающего меню.

- Папка выгрузки — поле для указания пути с указанием директории в ОС до места выгрузки обработанных данных. Тип поля — строковый.

- База данных АТС — поле для указания пути до обрабатываемого файла исходных данных АТС. Тип поля — строковый

- Шаблон категорий АОН — поле для указания пути до расположения файла с правилами преобразования категорий АОН для указанной Базы данных АТС. Тип поля — строковый

- Шаблон Услуг - поле для указания пути до расположения файла с правилами преобразования услуг номера для указанной Базы данных АТС. Тип поля — строковый

- Шаблон IMS - поле для указания пути до расположения файла с правилами конфигурации конкретного узла IMS для конечных данных абонентов. Тип поля — строковый

Выходные данные:

- Файл web\_portal.txt — содержит список сконвертированных номеров в формате, определённом для последующего импорта на узел VIMS. Тип файла — текстовый формат.

- Лог- файлы: migrate\_debug.log и migrate\_info.log — содержат информацию о процессе работы программы. Тип файлов — текстовый формат.

main\_win: procedure;

// объявить структуру, содержащую конфигурационные данные

declare 1 config; // словарь атрибутов конфигурации

2 node: string [250]; // тип АТС

2 type\_dn: string [250]; // тип обрабатываемых номеров

2 dest\_dir: string [250]; // директория выгрузки данных

2 sourse\_file\_db: string [250]; // имя файла с БД

2 mapping\_category: string [250]; // имя файла с шаблоном категорий

2 mapping\_service: string [250]; // имя файла с шаблоном услуг

2 mapping\_ims: string [250]; // имя файла с шаблоним IMS

// Запустить вывод логирования работы на экран

call show\_log;

// обработка кнопок формы

do case (кнопка)

//do выполнить конвертирование;

"кнопка RUN": call run\_migrate (config); // запустить скрипт миграции

//do получить конфигурационные данные;

"кнопка Config": call config\_window(config); // открыть графическую форму настройки конфигурации

"кнопка Quit": call Exit; // завершить работу приложения

end do;

end main\_win;

## 3.2 Внутренняя спецификация

// Вывод логирования действий программы на экран

show\_log: procedure;

begin

declare queue; // структура-очередь

// запустить чтение лог-файл в бесконечном цикле

call read\_log;

// запустить отображение данных из очереди на экран в бесконечном цикле

call pull\_display\_from\_log;

end;

end show\_log;

// Процедура чтения лог-файл в бесконечном цикле

read\_log: procedure;

// Процедура забирает из очереди данные и выводит на экран в бесконечном цикле

pull\_display\_from\_log: procedure;

// Процедура запускает скрипт миграции абонентов

run\_migrate (config): procedure

begin

// создается объект-хранилище для исходной базы номеров АТС

declare node\_repo: type (BaseRepo);

// создаётся объект-хранилище для хранения обработанной базы номеров IMS

declare main\_repo: type(ImsSubsRepo);

// выполнить обработку исходной БД

call node\_repo(config);

// выполнить конвертирование номеров по заданному алгоритму

call main\_interactor(main\_repo, node\_repo, config);

// выполнить запись результатов в файл

call write\_wp(main\_repo, config);

end;

end run\_migrate;

// Процедура выполняет запись обработанных данных о номерах в файл

write\_wp: procedure (main\_repo, config);

begin

declare wp\_list; // массив для хранения списка номеров для обработки

declare view\_wp; // массив для хранения обработанного списка номеров

// получить из репозитория список номеров по критерию

wp\_list = get(получить из main\_repo список номеров по критерию config.type\_dn)

// сформировать формат представления данных для записи в файл

view\_wp = get(call функция\_представления\_данных\_для\_записи\_в\_текстовый\_файла (wp\_list))

// записать конечные данные в файл

call write\_file (config.dest\_dir, 'имя\_файла', view\_wp);

end;

end write\_wp

// Процедура графического окна формы конфигурации

config\_window: procedure (config);

begin

declare field\_select\_node; // графический элемент-селектор

declare field\_select\_type\_number; // графический элемент-селектор

declare field\_1; // графический элемент-поле ввода

declare field\_2; // графический элемент-поле ввода

declare field\_3; // графический элемент-поле ввода

declare field\_4; // графический элемент-поле ввода

declare field\_5; // графический элемент-поле ввода

do case (кнопка)

"Кнопка OK": call return\_config(config); // возврать конфиг и закрыть окно

"Кнопка Х": call exit; // закрыть окно конфигурации

end do;

// возврать конфиг и закрыть окно

return\_config: procedure (config);

begin

if check\_fields() есть истинна; // проверка полей ввода

then

call add\_to\_config(config); // добавить атрибуты в словарь конфигурации

call Exit; // закрыть окно конфигурации

end;

end return\_config;

// Процедура проверки полей ввода

check\_fields: procedure RETURNS (BOOL);

begin

// поле директория выгрузки данных

get (field\_1)

if (field\_1 < 0 or field\_1 > 250 or field\_1 не найдена указанная директория);

then

call show\_alarm (field\_1);

return False; // неуспешная проверка

// имя файла с БД

get (field\_2)

if (field\_2 < 0 or field\_2 > 250 or field\_2 не найден указанный файл);

then

call show\_alarm (field\_2);

return False; // неуспешная проверка

// имя файла с шаблоном категорий

get (field\_3)

if (field\_3 < 0 or field\_3 > 250 or field\_3 не найден указанный файл);

then

call show\_alarm (field\_3);

return False; // неуспешная проверка

// имя файла с шаблоном услуг

get (field\_4)

if (field\_4 < 0 or field\_4 > 250 or field\_4 не найден указанный файл);

then

call show\_alarm (field\_4);

return False; // неуспешная проверка

// имя файла с шаблоном IMS

get (field\_5)

if (field\_5 < 0 or field\_5 > 250 or field\_5 не найден указанный файл);

then

call show\_alarm (field\_5);

return False; // неуспешная проверка

return True; // возвратить истину

end;

end check\_fields;

// Добавить атрибуты в словарь конфигурации

add\_to\_config: procedure;

begin

config.node = get(field\_select\_node);

config.type\_dn = get(field\_select\_type\_number);

config.dest\_dir = get(field\_1);

config.sourse\_file\_db = get(field\_2);

config.mapping\_category = get(field\_3);

config.mapping\_service = get(field\_4);

config.mapping\_ims = get(field\_5);

end;

end add\_to\_config;

// отображает графическое окно с информацией об ошибке

show\_alarm: procedure (description);

degin

показать графическое окно с информацией об ошибке

end;

end show\_alarm;

end;

end config\_window;

// Процедура завершения работы приложения

Exit: procedure;

# 4 Тестирование

## 4.1 Общие принципы тестирования

Для проведения тестирования программы «MigrateIMS» выполняется системное тестирование

## 4.2 Организация испытаний программных изделий

Организация испытаний программного изделия преследует цель установления факта наличия/отсутствия ошибок и расхождения между истинными свойствами программного изделия и его спецификациями.

## 4.3 Виды испытаний программного изделия. Стадии испытаний

Для проверки правильности работы программы выбран класс B (тестирование после разработки).

## 4.4 Режимы испытаний программ

Тестирование будет выполняться в режиме III (выполняется разработчиком).

## 4.5 Категории испытания программного изделия

Категория испытаний программного изделия – Аттестация.

## 4.6 Технология тестирования, классы эквивалентности

**4.6.1 Варианты использования**

Тестовый сценарий использования: Выполнить конвертирование.

Основное действующее лицо: Оператор, Конвертер

Цель: Получить исходные данные номеров в новом формате.

Уровень: Обобщенный.

Область действия: Программа MigrateIMS.

Основной сценарий:

1. Оператор запускает главный скрипт исполнения.
2. Конвертер инициирует открытие главного графического окна.
3. Оператор инициирует настройку конфигурации
4. Конвертер инициирует открытие модального окна с настройками конфигурации.
5. Оператор определяет необходимые настройки и данные и подтверждает ввод.
6. Конвертер проводит валидацию введённых данных в поля ввода, закрывает модальное окно и делает активным основное.
7. Оператор инициирует запуск процесса конвертирования.
8. Конвертер выполняет обработку данных по выбранному сценарию.
9. Конвертер формирует выходные данные.
10. Конвертер информирует оператора о завершении обработки данных.

Расширения:

5.1 Оператор выполняет отмену настройки конфигурации и закрывает окно без подтверждения кнопкой ОК.

5.1.1 Конвертер применяет текущие значения

5.1.2 Конвертер закрывает окно конфигурации и возвращается в главное окно.

5.2 Оператор применяет текущую конфигурацию без изменений.

5.2.1 Конвертер применяет текущие значения

5.2.2 Конвертер закрывает окно конфигурации и возвращается в главное окно.

6.1 Валидация входных данных не прошла

Конвертер отображает модальное окно с указанием ошибки валидации

Пользователь подтверждает прочтение

Конвертер закрывает окно с сообщением об ошибке и возвращается в

окно конфигурации.

8.1 Исходная БД не содержит указанный для обработки тип номера

8.1.1 Конвертер выводит сообщение об отсутствии номеров для запрашиваемого типа в исходной БД.

8.2 Конфигурационные файлы имеют неверный формат json

8.2.1 Конвертер сообщает об ошибке и прекращает обработку данных

8.3 Исходный файл базы номеров имеет неожидаемый формат для указанного типа АТС.

8.3.1 Конвертор сообщает об ошибке и прекращает обработку данных

9.1 Конвертер не может сохранить выходные данные в файл.

9.1.1 ошибка перехватывается встроенными обработчиками исключений и пишется в лог с указанием уровня срочности.

Классы эквивалентности для проверки форм ввода конфигурационных данных в тестовом сценарии п.п. 6, 6.1 описаны в таблице 3.1.

Таблица 3.1 - Классы эквивалентности формы ввода

|  |  |  |
| --- | --- | --- |
| Входные условия | Классы эквивалентности | |
| Правильные | Неправильные |
| Папка выгрузки файла | Строка длинной от 1 до 250 знаков (1) | Пустая строка (2)  Больше 250 знаков (3) |
| Папка выгрузки файла | Папка по указанному пути существует (4) | Папка по указанному пути не существует (5) |
| База данных АТС | Строка длинной от 1 до 250 знаков (6) | Пустое поле (7)  Больше 250 знаков (9) |
| База данных АТС | файл по указанному пути существует(10) | файл по указанному пути не существует(11) |
| Шаблон категорий АОН | Строка длинной от 1 до 250 знаков (12) | Пустое поле (13)  Больше 250 знаков (14) |
| Шаблон категорий АОН | файл по указанному пути существует(15) | файл по указанному пути не существует(16) |
| Шаблон Услуг | Строка длинной от 1 до 250 знаков (17) | Пустое поле (18)  Больше 250 знаков (19) |
| Шаблон Услуг | файл по указанному пути существует(20) | файл по указанному пути не существует(21) |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| Шаблон IMS | Строка длинной от 1 до 250 знаков (22) | Пустое поле (23)  Больше 250 знаков (24) |
| Шаблон IMS | файл по указанному пути существует (25) | файл по указанному пути не существует (26) |

## 

## 2.7 Построение тестов

На основе тестового сценария определены следующие тесты:

1. Основной тестовый сценарий: Тип АТС: DEMO; Тип номера: SIP; положительные классы эквивалентности Таблица 1.1 подклассы: 1,4,6,10,12, 15, 17, 20

2. Альтернативный тестовый сценарий п.5.1: без изменения текущей конфигурации.

3. Альтернативный тестовый сценарий п.5.2: без изменения текущей конфигурации.

4. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 классы: 2

6. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 классы: 3

7. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 5

8. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 7

9. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 9

10. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 11

11. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 13

12. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 14

13. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 16

14. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 18

15. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 19

16. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 21

17. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 23

18. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 24

19. Альтернативный тестовый сценарий п.6.1: отрицательные классы эквивалентности Таблица 1.1 подклассы: 26

20. Альтернативный тестовый сценарий п.8.1: В исходной БД АТС отсутствуют номера указанного типа.

21. Альтернативный тестовый сценарий п.8.2: неверный формат json.

22. Альтернативный тестовый сценарий п.8.3: неверный формат БД для указанного типа АТС.

23. Альтернативный тестовый сценарий п.9.1: Недостаточно прав для записи файла в указанную директорию.

# 5 Руководство системного программиста

## 5.1 Общие сведения о программе

Конвертер MigrateIMS предназначен для конвертирования исходной базы номеров телефонных станций в новый формат представления, совместимый для импорта (миграции) на новый узел коммутации vIMS. Другими словами, конвертер выполняет обработку каждого номера, его атрибутов: услуги, категория АОН и т.д, которыми он пользовался, а также служебные настройки: статус отключения за неуплату, административное ограничения исходящей/входящей связи и т.д. Далее, на основании шаблонов конвертирования, все вычисленные атрибуты номера преобразуются в новые значения и структуры, которые понятны для vIMS.

Учитывая специфику хранения данных в разных типах АТС, исходная база номеров может содержать разные форматы и структуры данных, в том числе это может быть не один файл и несколько. Для каждого типа АТС разрабатывается индивидуальный программный модуль, способный преобразовывать исходные «сырые» данные к нужному внутреннему формату представления. Затем к этим данным применяются правила конвертирования.

Правила конвертирования (правила преобразования исходного типа к типу, применяемому на vIMS) описываются в шаблонах конфигурации в формате «json».

Создаются следующие шаблоны:

- Шаблон для переноса услуг (ДВО – дополнительные виды обслуживания).

- Шаблон для переноса категорий АОН.

- Шаблон vIMS который содержит общие (глобальные) правила переноса и формирования конечных данных для каждого конвертируемого номера.

Дополнительные флаги для управления процессом конвертирования, позволяют оператору указать, по каким правилам вести обработку исходных данных. Часть флагов, управляющая потоком выполнения, указывается в качестве аргументов командной строки, в случае запуска программы через консольное приложение. Другая часть флагов относится к более тонкой настройке для управления бизнес-правилами и находится в качестве атрибутов в шаблоне IMS.

В качестве результата обработки, конвертер формирует следующие данные:

1. Файл со списком абонентов в формате для последующего импорта в БД VIMS.

2. Файл со списком активных услуг абонентов в формате для последующего импорта в БД vIMS.

3. Лог-файл.

## 

## 5.2 Структура программы

Программа MigrateIMS состоит из следующих компонентов:

1) migrate.py – главный исполняемый скрипт.

2) requirements.txt – файл со списком зависимостей.

3) requirements\_doc.txt – файл со списком зависимостей для генерации документации из дистрибутива.

4) sources – пакет с кодовой базой.

5) templates – папка с дефолтными шаблонами.

6) tests – папка с тестами.

7) docs – папка с документацией.

Перед начала использования требуется установка интерпретатора Python требуемой версии и установка зависимостей (внешних библиотек).

## 5.3 Настройка программы

Не нуждается в настройке.

## 5.4. Проверка программы

Проверка программы производится в следующем порядке:

Формируются все необходимые исходные данные

1. В командной строке выполняется запуск главного исполняемого файла. Пример: python migrate.py

2. Результат: Программа запускается, выполняет обработку исходных данных, формирует набор результирующих файлов, завершает работу.

## 3.5 Дополнительные возможности

Программа не обладает дополнительными возможностями.

## 3.6 Сообщения системному программисту

В процессе обработки данных, конвертер выводит все диагностические сообщения в консоль (терминал), а также в лог-файл. Всем сообщения присваивается категория срочности:

1) INFO – информационное.

2) WARNING – следует обратить внимание, есть проблемы с обработкой данных.

3)ERROR - Возникшее исключение на уровне интерпретатора.

# 6 Заключение

В результате выполнения курсового проекта, мной были получены навыки разработки и составления программной документации для разрабатываемого программного обеспечения.