КОРПОРАЦИЯ Галактика

# Защищенный программный комплекс «Платформа Галактика»

## Описание программы

RU.ГАНМ.00032-01 13 01

galaktika.ru

galaktika.ru/zpk/

2020

# АННОТАЦИЯ

Настоящий документ является описанием программного обеспечения (ПО) защищенного программного комплекса «Платформа Галактика» (далее – ЗПК).

В документе представлены общие сведения о программе, функциональное назначение, описание логической структуры, используемые технические средства, входные и выходные данные программы.

Документ подготовлен в соответствии с «ГОСТ 19.402-78 Единая система программной документации (ЕСПД). Описание программы».

# СОДЕРЖАНИЕ

# 1 ОБЩИЕ СВЕДЕНИЯ

Полное наименование: Программное обеспечение защищенного программного комплекса «Платформа Галактика».

Сокращенное наименование: ПО ЗПК.

Для полноценного функционирования ЗПК необходима операционная система (ОС) специального назначения Astra Linux Special Edition релиз 1.6 с входящими в ее состав компонентами:

– средство организации единого пространства пользователей Astra Linux Directory;

– защищенный Web-сервер Apache 2;

– защищенная СУБД PostgreSQL 9,6.

Для реализации интерфейса пользователя и функционала ЗПК используются языки программирования JavaScript, Python, PL/PgSQL.

Для очистки оперативной памяти используется программа стороннего производителя – «FreeMemory».

# 2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

ПО ЗПК представляет собой технологическую платформу с возможностью создания его базе типовых модулей бизнес-приложений класса ERP, AMM, EAM (не ограничиваясь перечисленными), с учетом требований конфиденциальности, в том числе, предназначенное для функционирования в ЗСПД.

Для возможности дальнейшего развития ПО ЗПК при его создании предусмотрено:

– интероперабельность ПО ЗПК;

– универсальность ПО ЗПК для минимизации конфигурирования при создании на его базе типовых модулей бизнес-приложений класса ERP, AMM, EAM (не ограничиваясь перечисленными).

ПО ЗПК обеспечивает следующие функции, перечень которых приведен в таблице 1.

Таблица 1 – Перечень процессов и функций

| Процесс | Функции ПО ЗПК |
|---|---|
| Администрирование | 1 Возможность просмотра списка зарегистрированных организаций и пользователей;<br>2 Добавление пользователей и организаций;<br>3 Управление ролями пользователей;<br>4 Изменение реквизитов пользователей и организаций;<br>5 Возможность просмотра списка установленных модулей и предусмотренных ими функциях |
| Аудит | 1 Регистрация действий пользователей при создании, изменении или удалении записей;<br>2 Хранение истории выполненных операций;<br>3 Возможность просмотра журнала действий |
| Интеграция | Предоставляет набор API методов для выполнения функций, предусмотренных установленными модулями |

| Процесс | Функции ПО ЗПК |
|---|---|
| Масштабирование | 1 Модульная архитектура;<br>2 Возможность развертывания разных функциональных модулей на обособленных серверах приложений в том числе в пределах отдельных контуров |
| Формирование общего и частного наборов данных | 1 Обеспечение мультитенантности;<br>2 Возможность работы пользователей организации в пределах изолированной области видимости данных;<br>3 Возможность просмотра общих сведений, публикуемых участниками других организаций |
| Форматно-логический контроль | Обеспечивает проверку входящей информации с учетом требований к формату, обеспечению целостности и непротиворечивости данных |
| Развертывание и обновление | 1 Предоставляет утилиты в составе дистрибутива, которые используются для развертывания системы и/или ее отдельных функциональных модулей;<br>2 Предоставляет средство для подготовки обновления базы данных при добавлении нового функционального модуля, либо при обновлении модуля и/или системы |

ПО ЗПК обеспечивает возможность выполнения перечисленных ниже функций:

– возможность интеграции по API;

– функции выгрузки данных в файлы текстового формата;

– функции загрузки данных из файлов текстового формата;

– функции отображения названия программы, версии программы, копирайта.

# 3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

## 3.1 Алгоритмы программы

Каждый модуль системы, включая модуль ядра, предоставляет классы и реализует входящие в их состав методы, которые обеспечивают выполнение функций, предусмотренных соответствующим модулем.

Состав (набор) методов каждого класса определяется требованиями к функциональности модуля и разграничению области их применения согласно прикладным действиям.

Состав классов, предусмотренных отдельным модулем, является фиксированным и не подлежит изменению.

Состав методов, принадлежащих каждому из классов, предусмотренных отдельным модулем, также является фиксированным и не подлежит изменению.

В системе предусмотрены следующие классы ролей:

1 роль модуля;

2 роль класса;

3 роль организации;

4 роль пользователя.

Классы ролей и их связи представлены на рис 1.

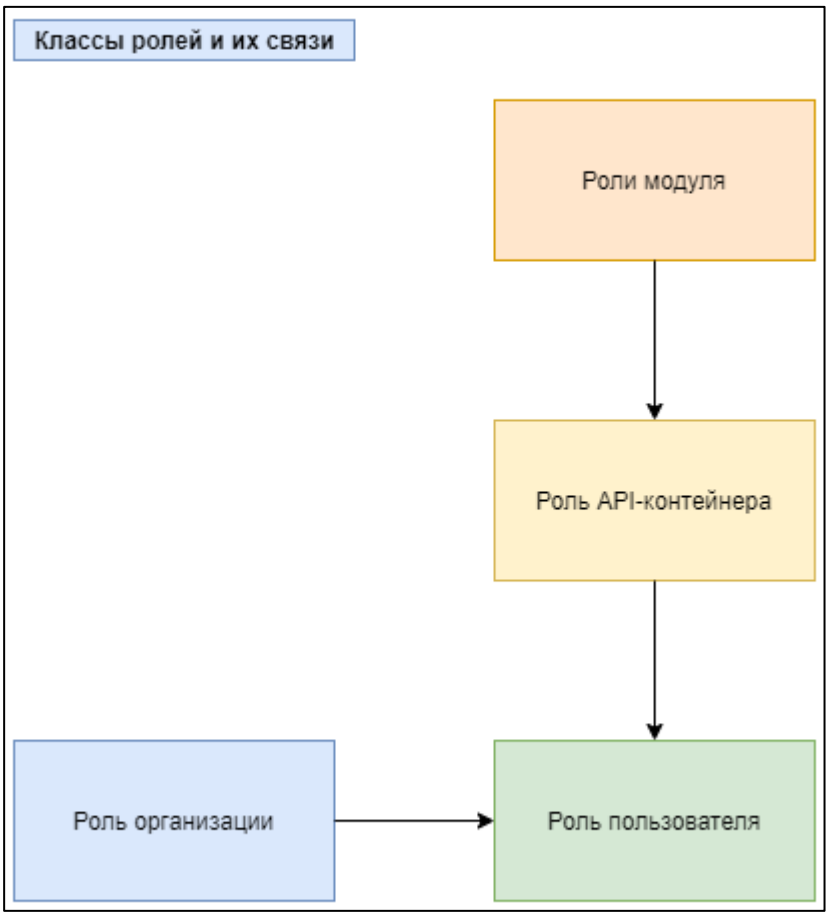


Рисунок 1 – Классы ролей и их связи

Алгоритм – это точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время.

Схема алгоритма – это графический способ его представления с элементами словесной записи.

Ниже приводятся описания последовательности правил, указывающих исполнителю действия, которые он должен выполнить, чтобы за конечное время перейти от (варьируемых) исходных данных к искомому результату в рамках исполнения комплекса задач ПО ЗПК.

### 3.1.1 Последовательность выполняемых задач

#### 3.1.1.1 Постановка задачи

Автоматизация выполняемых задач и процессов ПО ЗПК.

#### 3.1.1.2 Текстовый алгоритм решения задачи

1) Задачи выполняемые Firefox SPA JavaScript Application

– Выходы:

а) На Apache Web Server – Вызов Web API в контексте безопасности пользователя.

– Входы:

а) От WSGI Python Application – Результат обработки запроса.

2) Задачи выполняемые Apache Web Server

– Выходы:

а) Аутентификация в ЕПП по протоколу Kerberos;

б) На WSGI Python Application – Переключение обработчика в контексте безопасности пользователя.

– Входы:

а) Аутентификация в ЕПП по протоколу Kerberos.

3) Задачи выполняемые WSGI Python Application

– Выходы:

а) На PostgreSQL Database PL – Вызов Database API в контексте безопасности пользователя;

б) На Firefox SPA JavaScript Application – Результат обработки запроса.

– Входы:

а) От Apache Web Server – Переключение обработчика в контексте безопасности пользователя;

б) От PL/pgSQL Application – Результат обработки запроса.

4) Задачи выполняемые PostgreSQL Database PL

– Выходы:

а) Аутентификация в ЕПП по протоколу GSSAPI;

б) На PL/pgSQL Application – Переключение обработчика в контексте безопасности пользователя.

– Входы:

а) От WSGI Python Application – Вызов Database API в контексте безопасности пользователя;

б) Аутентификация в ЕПП по протоколу GSSAPI.

5) Задачи выполняемые PL/pgSQL Application

– Выходы:

а) Обработка запроса;

б) На WSGI Python Application – Результат обработки запроса.

– Входы:

а) От PostgreSQL Database PL – Переключение обработчика в контексте безопасности пользователя;

б) Обработка запроса.

#### 3.1.1.3 Схема алгоритма решения задачи

Схема алгоритма последовательности выполняемых задач приведена на рис. 2.

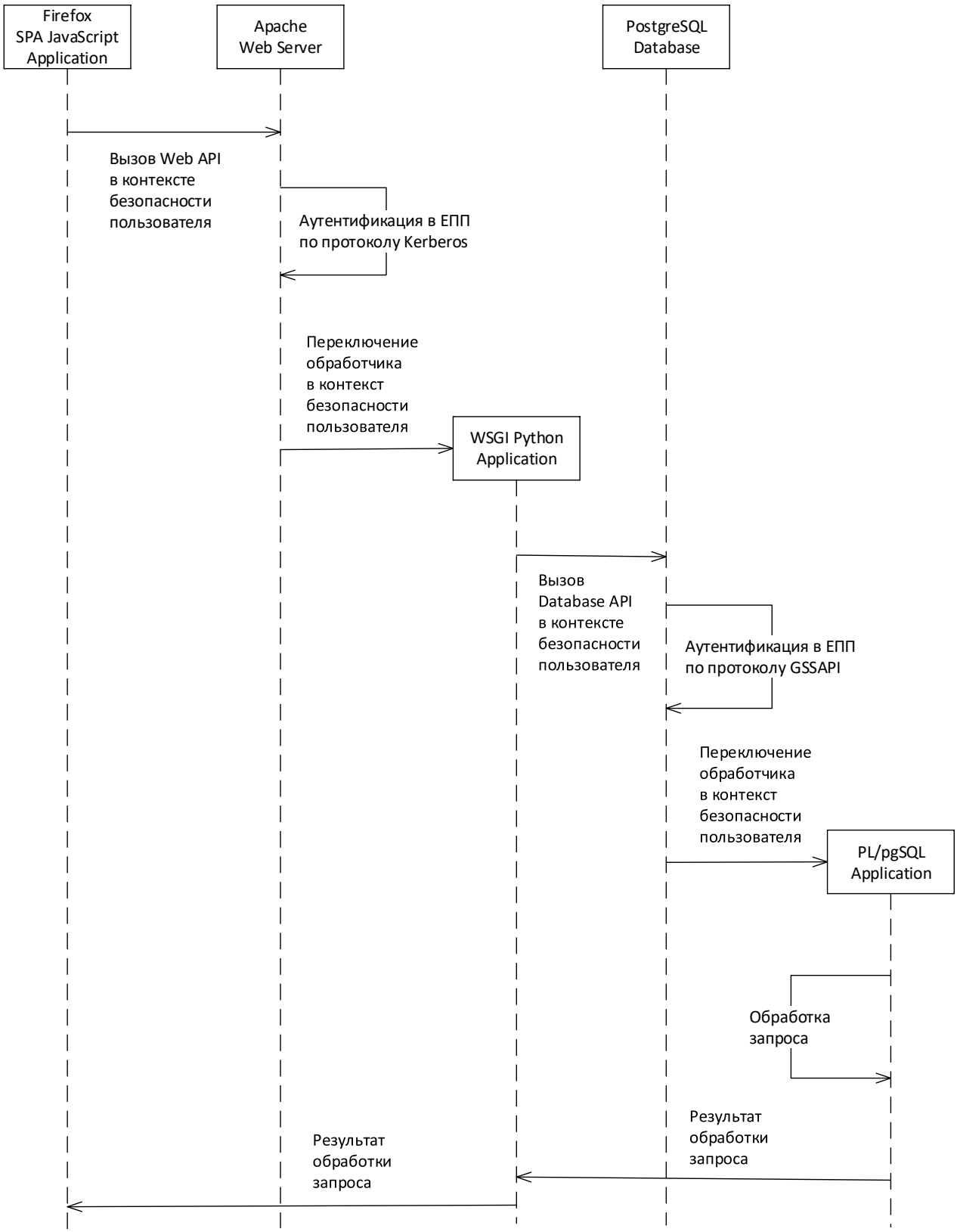


Рисунок 2 – Схема алгоритма последовательности выполняемых задач

### 3.1.2 Запрос и обработка данных

#### 3.1.2.1 Постановка задачи

Автоматизация процессов обработки данных ПО ЗПК.

#### 3.1.2.2 Текстовый алгоритм решения задачи

1) Определение модуля и хранимых данных:
   - определение модуля доступного пользователю;
   - выбор хранимых данных, доступных пользователю.
2) Вызов хранимой процедуры с параметрами из json и обработка запроса;
3) Сохранение информации о вызове в журнал;
4) Возврат в json результатов обработки запроса.

#### 3.1.2.3 Схема алгоритма решения задачи

Схема алгоритма выполнения запроса и обработки данных приведена на рис. 3.

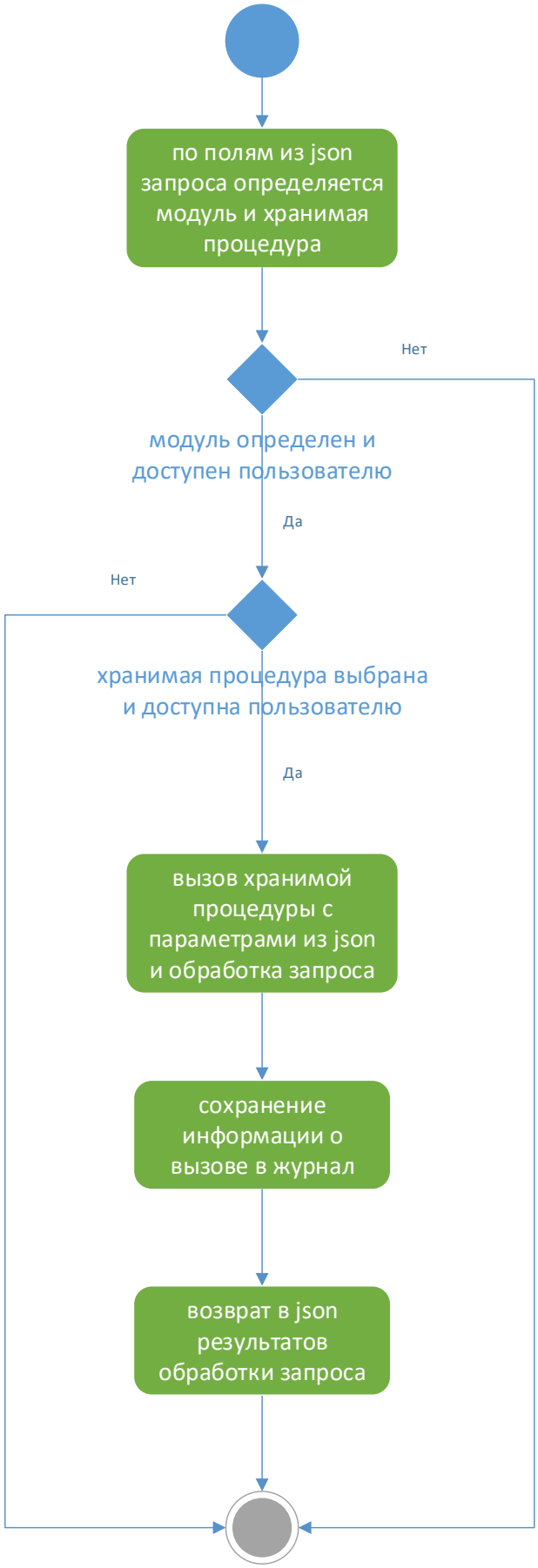


Рисунок 3 – Схема алгоритма выполнения запроса и обработки данных

## 3.2 Структура программы

В состав компонентов ПО ЗПК входят:

1) База данных;
2) Сервер приложений;
3) Модуль ядра;
4) Клиентские модули;
5) Вспомогательная утилита администрирования.

### 3.2.1 База данных

База данных содержит объекты, обеспечивающие:

– выполнение функций унифицированного ядра системы и прикладных модулей платформы;

– хранение, выбор данных, их фильтрацию, а также модификацию путем реализации алгоритмов выполнения действий API на основании запросов, поступающих со стороны сервера приложений;

– первичную обработку ошибок, возникающих в процессе выполнения алгоритмов.

### 3.2.2 Сервер приложений

В состав сервера приложений входят следующие компоненты:

1) WEB-сервер, обеспечивающий:
   – установку соединения при обработке запроса на выполнение API-функции, поступающего от клиента сервера приложений;
   – создание обработчика для данного запроса;
   – передачу обработчику данных поступившего запроса для его выполнения;
   – возврат результатов обработки запроса клиенту.
2) WSGI-приложение, которое:
   – реализует выполнение API-метода на основании запроса,

поступившего от web-сервера;

– осуществляет запуск алгоритма обработки требуемого API-метода в базе данных;

– выполняет вторичную обработку ошибок, возникающих при выполнении запроса;

– возвращает результаты обработки запроса web-серверу для их последующей передачи клиенту сервера приложений.

### 3.2.3 Модуль ядра

Компоненты модуля ядра размещаются в базе данных системы. Данный модуль не предусматривает пользовательского интерфейса в виде клиентского приложения.

В состав модуля ядра входят:

– объекты для хранения дополнительных сведений о зарегистрированных организациях и их пользователях;

– инфраструктура для регистрации классов и методов API, используемых платформенными модулями при их установке или обновлении;

– общие функции, используемые прикладными платформенными модулями для формирование пользовательского интерфейса клиентской части;

– регистрации действий пользователей, выполняемых посредством методов API соответствующих платформенных модулей в журнале действий пользователей;

– дополнительные механизмы фильтрации данных текущего пользователя и соответствующей ему организации;

– хранилище состояния настроек пользовательского интерфейса клиентских приложений модулей платформы;

– хранилище содержания пользовательских файлов, загружаемых в систему и требующими хранения в СУБД и дополнительные механизмы фильтрации списка доступных файлов согласно настроенным правилам

разграничения доступа;

– функции, реализующие механизм блокировки записей при их одновременном редактировании различными пользователями.

### 3.2.4 Клиентские модули

В состав клиентских модулей входят:

– Модуль «Аудит»;

– Модуль «Справочники и НСИ».

#### 3.2.4.1 Модуль «Аудит»

Клиентская часть модуля предоставляет графический интерфейс пользователя для выполнения следующих функций:

– просмотр журнала действий пользователей системы;

– просмотр сведений об установленных модулях платформы;

– просмотр сведений о зарегистрированных классах и методах установленных модулей;

– просмотр списка зарегистрированных пользователей и организаций;

– просмотр списка эффективных разрешений пользователей согласно настройкам разграничения доступа.

Компонент базы данных предоставляет и реализует хранимые процедуры, обеспечивающие возможность извлечения данных для их представления в графическом интерфейсе клиентской части.

#### 3.2.4.2 Модуль «Справочники и НСИ»

Клиентская часть модуля предоставляет графический интерфейс пользователя для выполнения следующих функций:

– просмотр и администрирование общесистемных справочников и НСИ;

– просмотр и администрирование справочников и НСИ организации;

– просмотр и изменение реквизитов организации.

Компонент базы данных предоставляет и реализует:

– хранилище записей справочников и НСИ;

– дополнительные механизмы фильтрации данных согласно настройкам прав разграничения доступа;

– хранимые процедуры для извлечения и модификации данных, обработка которых выполняется данным модулем.

### 3.2.5 Вспомогательная утилита администрирования

Вспомогательная утилита администрирования предназначена для частичной автоматизации действий администратора системы при выполнении им базовых функций управления системой, а именно:

– регистрация организаций для работы в системе;

– регистрация сотрудников организаций для работы в системе;

– назначение прав доступа к функциям системы для зарегистрированных пользователей;

– отзыв прав доступа к функциям системы для зарегистрированных пользователей.

Утилита представляет собой скрипт, запускаемый из консоли терминала ОС, и не требует использования графического интерфейса пользователя.

Утилита не является обязательной для использования в процессе администрирования системы.

Все действия, выполняемые утилитой, могут быть выполнены администратором самостоятельно.

Примечание – Программа «FreeMemory» состоит из одной запускаемой формы и не имеет других составных частей.

# 4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

В состав используемых технических средств входят сервера и рабочие станции архитектуры Intel X86 с характеристиками не ниже следующих:

1) Сервер веб-приложений: не менее 2-х процессоров (количество ядер – 16, тактовая частота – 2 ГГц), оперативная память – не менее 32 ГБ (DDR3, 1600 МГц), ёмкость жестких дисков – не менее 200 ГБ, подключение к сети 1 Гбит/с;

2) Сервер базы данных: не менее 2-х процессоров (количество ядер – 8, тактовая частота – 2 ГГц), оперативная память – не менее 64 ГБ (DDR3, 1600 МГц), ёмкость жестких дисков – не менее 400 ГБ; подключение к сети 1 Гбит/с;

3) Сервер ЕПП: не менее 2-х процессоров (количество ядер – 8, тактовая частота – 2 ГГц), оперативная память – не менее 16 ГБ (DDR3, 1600 МГц), ёмкость жестких дисков – не менее 200 ГБ, подключение к сети 1 Гбит/с;

4) Рабочая станция: не менее 2-х процессоров (количество ядер – 4, тактовая частота – 2 ГГц), оперативная память – не менее 16 ГБ (DDR3, 1600 МГц), ёмкость жестких дисков – не менее 40 ГБ, подключение к сети 1 Гбит/с.

# 5 ВЫЗОВ И ЗАГРУЗКА

Вызов и загрузка ПО ЗПК описаны в документе RU.ГАНМ.00032-01 95 01 Руководство администратора.

# 6 ВХОДНЫЕ ДАННЫЕ

Входными данными являются:

– данные, вводимые пользователями в интерактивном режиме;

– автоматически регистрируемые данные;

– данные внешних и смежных систем, поступающие через информационные сервисы;

– данные иных сторонних источников.

Данные, вводимые пользователями в интерактивном режиме, включают:

– нормативно-справочную информацию, ведущуюся непосредственно в системе;

– данные о пользователях системы и правах их доступа.

Автоматически регистрируемые данные включают:

– данные, полученные системой при создании документов;

– информацию о действиях пользователей и системы;

– информацию, фиксируемую компонентами системы, о результатах выполнения автоматических операций.

Примечание – Ввод и корректировка данных осуществляется только через модули Системы. Прямой доступ пользователей к БД не предполагается.

# 7 ВЫХОДНЫЕ ДАННЫЕ

Выходные данные представлены содержимым БД, организованным в формате HTML, а также интерактивным содержимым (аудио и видео поток).

Выходными данными являются формируемые в среде исполнения выходные данные алгоритмов решения функциональных задач, предназначенные для представления в виде:

– таблиц;

– графиков;

– информации с картографической привязкой;

– файлов формата xml, json, pdf, формируемые при экспорте записей БД и выводе их на печать;

– анимационных эффектов и т.п.

– сообщений и оповещений, генерируемых Системой при возникновении ошибок или определенных действиях.

Выходные данные передаются на АРМ пользователей для отображения или реплицируются во внешние информационные системы.

# ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

| | | |
|---|---|---|
| АРМ | – | Автоматизированное рабочее место |
| БД | – | База данных |
| ГОСТ | – | Государственный стандарт |
| ЕСПД | – | Единая система программной документации |
| ЗПК | – | Защищенный программный комплекс |
| ЗСПД | – | Закрытый сегмент передачи данных |
| НСИ | – | Нормативно-справочная информация |
| ОС | – | Операционная система |
| ПО | – | Программное обеспечение |
| СУБД | – | Система управления базами данных |