

## ОГЛАВЛЕНИЕ

РЕФЕРАТ .....	12
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ С ИХ ОПРЕДЕЛЕНИЕМ .....	13
ВВЕДЕНИЕ .....	14
1 ПРЕДДИПЛОМНОЕ ИССЛЕДОВАНИЕ.....	16
1.1 Назначение программного комплекса РДО .....	16
1.2 Функции программного комплекса .....	18
1.3 Постановка задачи .....	20
1.4 Выводы по преддипломному этапу .....	22
2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	23
2.1 Введение .....	23
2.2 Основания для разработки .....	23
2.3 Назначение разработки .....	24
2.4 Требования к программе или программному изделию .....	24
2.4.1 Требования к функциональным характеристикам .....	24
2.4.2 Требования к надежности.....	24
2.4.3 Условия эксплуатации .....	25
2.4.4 Требования к составу и параметрам технических средств ---	25
2.4.5 Требования к информационной и программной совместимости.....	25
2.4.6 Требования к маркировке и упаковке .....	26
2.4.7 Требования к транспортированию и хранению .....	26
2.4.8 Требования к программной документации.....	2
2.5 Техничко-экономические показатели .....	26
2.6 Стадии и этапы разработки .....	26
2.7 Порядок контроля и приемки .....	27

2.8	Приложения	27
3	КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ	28
3.1	Цели разработки системы	28
3.2	Компоненты РДО	28
3.3	База данных	30
3.3.1	Структуры баз данных	30
3.3.1.1	Требования к структуре	30
3.3.1.2	Описание структуры	32
3.3.2	Выбор СУБД	35
3.3.3	Выбор драйвера работы с базой данных	
3.3.4	Сериализация данных в базу данных	36
3.3.5	Работа модели с данными из БД	37
4	ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	38
4.1	Инициализация структуры базы данных	38
4.2	Сериализация данных в базу данных	38
4.2.1	Создание инструментов для работы с базой данных	3
4.2.1.1	Необходимые классы	39
4.2.1.2	Описание интерфейсов компонентов классов	3
4.2.2	Использование инструментов для работы с базой данных	4
4.3	Работа модели с данными из базы данных	43
4.3.1	Выбор способа проверки работы	43
4.3.2	Синтаксические конструкции	44
5	РАБОЧЕЕ ПРОЕКТИРОВАНИЕ	45
5.1	Реализация инструментов работы с базами данных	45
5.1.1	Класс GeneralDB	45
5.1.1.1	Метод queryExec	45
5.1.1.2	Метод insertRow	46
5.1.1.3	Метод insertRowInd	46

5.1.1.4	Метод queryListPushBack	47
5.1.1.5	Метод queryListExec	47
5.1.1.6	Метод init	47
5.1.2	Класс InitStructDB	48
5.1.2.1	Метод dropDB	48
5.1.2.2	Метод createDB	49
5.1.2.3	Конструктор класса InitStructDB	49
5.2	Использование созданных инструментов для сериализации модели в базу данных	50
5.2.1	Сериализация типов ресурсов	50
5.2.2	Сериализация ресурса	50
5.2.3	Сериализация значений пара метров ресурсов в базу данных	51
6	ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ	53
6.1	Планирование исследования	53
6.1.1	Контроль версий РДО студии	5
6.1.2	Контроль этапа инициализации модели	54
6.1.3	Контроль работы модели с данными из базы данных	54
6.1.4	Планирование экспериментов	54
6.1.5	Бланк журнала экспериментов	56
6.2	Анализ проведенных экспериментов	57
6.2.1	Общие выводы о работе системы	57
6.2.2	Выводы о работе модели с данными из базы данных	59
6.2.3	Анализ причин обнаруженного поведения системы	59
7	ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ	61
7.1	Организация и планирование процесса разработки программного продукта	61
7.1.1	Расчет трудоемкости разработки технического задания	64

7.1.2	Расчет трудоемкости выполнения эскизного проекта -----	65
7.1.3	Расчет трудоемкости выполнения технического проекта ---	66
7.1.4	Расчет трудоемкости разработки рабочего проекта -----	67
7.1.5	Расчет трудоемкости выполнения внедрения -----	68
7.1.6	Расчет суммарной трудоемкости-----	6
7.2	Определение стоимости разработки ПП-----	70
7.2.1	Расчет основной заработной платы-----	-
7.2.2	Расчет дополнительной заработной платы-----	-
7.2.3	Отчисления в социальные фонды -----	72
7.2.4	Накладные расходы -----	72
7.2.5	Расходы на амортизацию оборудования-----	73
7.3	Определение стоимости разработки системы -----	74
8	МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ-----	75
8.1	Опасные и вредные факторы -----	75
8.1.1	Опасные факторы:-----	75
8.1.2	Вредные факторы:-----	75
8.2	Требования к помещениям для работы с ПЭВМ-----	76
8.2.1	Требования к микроклимату, содержанию вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ -----	78
8.2.2	Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ -----	80
8.2.3	Требования к освещению на рабочих местах, оборудованных ПЭВМ -----	81
8.2.4	Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ-----	84
8.2.5	Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах -----	85

8.2.6	Общие требования к организации рабочих мест пользователей ПЭВМ-----	8
8.2.7	Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей -----	8
8.2.8	Электробезопасность рабочего помещения -----	89
8.2.9	Требования пожарной безопасности-----	9
8.2.10	Требования к организации медицинского обслуживания пользователей ПЭВМ-----	9
8.2.11	Требования к проведению государственного санитарно- эпидемиологического надзора и производственного контроля -----	92
8.2.12	Расчет системы защитного заземления компьютера -----	92
8.3	Утилизация ПЭВМ -----	96
8.3.1	Разборка изделий -----	9
8.3.1.1	Разборка персональных компьютеров (ПЭВМ), рабочих станций и серверов-----	9
8.3.1.2	Обеспечение комплексности технологии разборки-----	100
8.3.1.3	Извлечение вторичных черных металлов-----	10
8.3.1.4	Извлечение вторичных цветных металлов-----	102
8.3.2	Реализация партий -----	10
8.3.2.1	Классификация отходов -----	103
8.3.2.2	Сертификация партий -----	10
8.3.2.3	Основные требования к партиям электронного лома и упаковка -----	108
8.3.2.4	Сдача на склад -----	110
8.3.2.5	Заключение договора на реализацию -----	110
8.3.2.6	Транспортировка -----	110

8.3.2.7 Соблюдение требований безопасности при работе с вторичными драгоценными металлами -----	112
9 РЕЗУЛЬТАТЫ -----	116
9.1 Процесс работы со сторонней системой -----	1
ЗАКЛЮЧЕНИЕ -----	118
СПИСОК ЛИТЕРАТУРЫ -----	119
ПРИЛОЖЕНИЕ 1. БЛАНК ЖУРНАЛА ЭКСПЕРИМЕНТОВ -----	120
ПРИЛОЖЕНИЕ 2. ЖУРНАЛ ЭКСПЕРИМЕНТОВ-----	121

## РЕФЕРАТ

Отчет 121 с., 5 рис., 13 табл., 8 источников, 2 прил.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, РЕСУРС-ДЕЙСТВИЕ-ОПЕРАЦИЯ, МОДЕЛЬ, АКТИВНОСТЬ, ТИП РЕСУРСА, РЕСУРС.

Объектом разработки является подсистема хранения состояния модели на основе реляционной базы данных в среде моделирования РДО. Ресурс, действие, операция (РДО) – программный комплекс, предназначенный для имитационного моделирования сложных дискретных систем с целью проведения их анализа и синтеза.

Цель работы – вывод состояния модели в реляционную базу данных, а так же исследование возможности увеличения быстродействия работы среды моделирования при работе алгоритма подбора релевантного ресурса через базу данных.

При создании подсистемы проведены исследования, целью которых являлось установить данные, которые должны храниться в базе данных для того, чтобы сделать систему более открытой.

В результате работы разработана структура баз данных, которая обеспечит организованное и однозначное хранение ресурсов, а также разработана подсистема, сериализующая состояние модели в эти базы данных.

Эффективность созданной системы заключается в повышении открытости системы, что расширяет возможности для работы с РДО, а также снижает минимально необходимый уровень квалификации для пользователя, которому необходимо строить графики на основе результатов моделирования с помощью сторонних систем.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ С ИХ ОПРЕДЕЛЕНИЕМ

UML	Universal Modeling Language (Универсальный язык моделирования)
БД	База данных
БЗ	База знаний
ИМ	Имитационная модель
ОС	Операционная система
ПП	Программный продукт
ПО	Программное обеспечение
ПЭВМ	Персональная электронно-вычислительная машина
РДО	Ресурс Действие Операция – система имитационного моделирования
СДС	Сложная дискретная система
ТЗ	Техническое задание
ЭП	Эскизный проект
ВДТ	Видео дисплейный терминал

## ВВЕДЕНИЕ

Математическое моделирование представляет собой неотъемлемую часть современного производства и не только. Все более сложные и объемные задачи перекладываются с фантазии экспертов в различных областях на плечи сред моделирования, в том числе и имитационного. РДО, как не последний представитель этого семейства программного обеспечения, должен предоставить не только принципиальную возможность выполнения поставленной задачи но и адекватную реализацию этой возможности. Под этими словами подразумеваю удобный интерфейс, возможность работы на разных платформах, быстрое получение результатов в удобном виде и прочие требования, которые в общем случае можно применить практически к любому программному обеспечению.

Хочу отметить, что язык РДО и среда моделирования РДО-студия активно развиваются. Но и прогресс не стоит на месте. Когда-то очень удачное решение сей час может выглядеть не очень удобным. Алгоритмы, которые когда-то вполне удовлетворяли потребностям пользователей, теперь могут вынудить их ждать окончания работы слишком долго. Функциональные возможности, которых хватало с лихвой, теперь могут печалить пользователя.

Развитие системы происходит на глазах: например, РДО-студия стала прекрасно запускаться и работать не только на операционных системах семейства Windows но и семейства Linux. Язык РДО обзавелся возможностью работать с треугольным законом распределения (который, кстати я и внедрил в язык). Разработчики обзавелись

системой автоматического тестирования. Произошла миграция на более прогрессивные библиотеки. Прогресс на лицо!

Но, увы (или при более лояльном взгляде – вполне нормально для развивающегося продукта), RDO-студия не идеальна. Например, она содержит алгоритмы, которые в больших моделях (по-настоящему больших) становятся бутылочным горлышком для скорости работы среды. К тому же RDO-студия предоставляет довольно сильный инструмент для обработки результатов моделирования и построения отчетов, но, увы, не безгранично гибкий.

Именно об этих проблемах и пойдет речь на остальных страницах этой кладези ученой мысли

# 1 ПРЕДДИПЛОМНОЕ ИССЛЕДОВАНИЕ

## 1.1 Назначение программного комплекса РДО

Задачи системного анализа и синтеза объектов различной природы и назначения часто решаются с использованием имитационных моделей. Эти модели позволяют исследовать динамические аспекты поведения сложных дискретных систем и процессов. Имитация, в частности, позволяет выполнить анализ функционирования объекта, прогнозирование, организационное управление, поддержать принятие решений при проектировании и управлении.

RAO-studio является средством имитационного моделирования, позволяющим воспроизводить на ПЭВМ динамику объекта, принятие решений сложной системой управления, и даже моделировать деятельность человека при принятии решений. В основе имитатора лежит РДО-метод формализации знаний о дискретных системах и процессах. Знания представляются в форме модифицированных продукционных правил, событий и процессов. При этом сохраняются такие достоинства продукционных систем, как универсальность, гибкость и наличие формальных механизмов логического вывода. Традиционные продукционные правила являются частным случаем модифицированных, поэтому в имитационную модель легко могут быть включены, например, экспертные системы.

Язык описания объектов, алгоритмов управления и задач в RAO-studio – это по существу язык представления знаний. Он требует от пользователя лишь знаний в предметной области, а не программирования. Пользователь описывает ресурсы, правила

функционирования, требуемые показатели и анимационные кадры непосредственно в терминах предметной области не прибегая при этом к представлению своей системы в терминах какого-либо известного метода (системы очередей, сети Петри автоматъ) или языка типа SLAM II, ARENA, SIMPLE++ и других. Это резко повышает гибкость, мощность и наглядность модели. РДО – язык высокого уровня, использующий символические имена, арифметические и логические выражения и функции, генераторы псевдослучайных чисел модифицированные и простые продукции.

Основные элементы RAO-studio – это модифицированная продукционная система, аппарат событий, с помощью которых в системе реализованы событийный, процессный и подход сканирован активностей. События начала действий базы знаний и процессы инициируются механизмом логического вывода, а события, запланированные пользователем (в явном виде) или системой (в неявном виде – события окончания действий) – специальным блоком. При имитации состояние системы изменяется в соответствии с описанием события, происходящим в данный момент модельного времени. После любого изменения состояния, т.е. при каждом событии, вызывается система вывода. Она просматривает в базе знаний и процессах продукционные правила и проверяет по условиям, могут ли начаться какие-либо действия. При нахождении таких действий инициируются события их начала.

Продукционная система, блок имитации событий совместно осуществляют построение имитационной модели системы. На основании анализа результатов имитации вычисляются требуемые показатели функционирования системы.

Система трассировки выводит подробную информацию о событиях в специальный файл, который затем обрабатывается для детального анализа работы модели. Система анимации отображает на экране во время имитации поведение моделируемого объекта.

RAO-studio может быть применен для создания имитационных моделей, систем планирования игр и тренажеров экспертных систем реального времени и гибридных систем, включающих экспертные системы, имитационные модели и алгоритмы оптимизации.

## 1.2 Функции программного комплекса

При выполнении работ, связанных с созданием и использованием ИМ в среде РДО, пользователь оперирует следующими основными понятиями [1]:

*Модель* – совокупность объектов языка РДО, описывающих какой-то реальный объект, собираемые в процессе имитации показатели кадры анимации и графические элементы, используемые при анимации, результаты трассировки.

*Прогон* – это единая неделимая точка имитационного эксперимента. Он характеризуется совокупностью объектов, представляющих собой исходные данные и результаты полученные при запуске имитатора с этими исходными данными.

*Проект* – один или более прогонов, объединенных какой-либо общей целью. Например это может быть совокупность прогонов которые направлены на исследование одного конкретного объекта или выполнение одного контракта на имитационные исследования по одному или нескольким объектам.

*Объект* – совокупность информации, предназначенной для определенных целей и имеющая смысл для имитационной программы. Состав объектов обусловлен РДО-методом, определяющим парадигму представления СДС на языке РДО.

Объектами исходных данных являются:

- типы ресурсов (с расширением .rtp);
- ресурсы (с расширением .rss);
- событий (с расширением .evn);
- образцы активностей (с расширением .pat);
- точки принятия решений (с расширением .dpt);
- процессы (с расширением .prc);
- константы, функции и последовательности (с расширением .fun);
- кадры анимации (с расширением .frm, .bmp);
- требуемая статистика (с расширением .pmd);
- прогон (с расширением .smr);
- проект (с расширением .rdox).

Объекты, создаваемые РДО-имитатором при выполнении прогона:

- результаты (с расширением .pmv);
- трассировка (с расширением .trc).

На данный момент РДО реализует следующие основные функции

1) Создание модели на языке РДО:

- создание основных объектов (\*.rtp, \*.rss, \*.evn, \*.pat, \*.dpt, \*.prc, \*.smr);

- создание объектов данных и функций (\*.fun);
- создание объектов вывода (\*.frm, \*.pmd, \*.bmp).

2) Проведение экспериментов:

- изменение параметров системы в процессе моделирования;
- генерация случайных чисел

3) Вывод результатов моделирования:

- анимация объектов модели;
- вывод необходимых показателей ;
- построение графиков;
- трассировка изменений объектов модели

### **1.3 Постановка задачи**

Во введении были озвучены две основные проблемы, которые будут рассмотрены в рамках этого дипломного проекта.

Во-первых, проблема гибкости отчетов по работе моделей упирается в набор предложенных инструментов. Он сейчас довольно широк, но предугадать всех пожеланий пользователя практически невозможно. Зато можно предоставить пользователю возможность реализовывать свои пожелания теми средствами, которые он сам решит задействовать. Сейчас результаты выводятся в текстовом виде консоли РДО-студии. Вывод состояния модели в базу данных предоставит пользователю интерфейс с автоматизированного получения данных. Пользователь сможет строить отчеты, которые удовлетворят

его потребности с помощью систем-построителей отчетов в том числе и в режиме реального времени, а также других систем, которые пользователь может разработать самостоятельно.

Во-вторых, проблема быстродействия для больших моделей РД заключена в алгоритме выбора релевантного ресурса.

В шаблонах (Pattern) языка РДО для подхода сканирования активностей есть условие выбора ресурса по типу ресурса которое должно быть выполнено, чтобы шаблон смог отработать заложенные пользователем в него действия с использованием этого ресурса. Суть алгоритма заключается в том, что перебираются все ресурсы, существующие в модели, вообще все. При этом над каждым ресурсом-кандидатом выполняются относительно длительные операции для определения его пригодности для шаблона.

Этот алгоритм очень непросто изменить: для этого потребуется внести большие изменения в логику работы языка. Альтернативным решением может стать делегирование логики работы этого алгоритма сторонней системе Базе данных например

Несмотря на потери быстродействия при вводе состояния модели в базу данных, есть предположение, что в конечном итоге удастся выиграть за счет более оптимальных алгоритмов работы СУБД с данными.

Таким образом, на данном этапе нет задачи «внедрить в систему». Есть задача создать работоспособный прототип чтобы проверить будет ли выигрыш в быстродействии. Подробнее вопрос будет рассмотрен в Исследовательской части

## 1.4 Выводы по преддипломному этапу

Рассмотренные выше проблемы представляют собой серьезное препятствие для пользователя, который захочет получить *специфический* отчет от среды моделирования. Он столкнется со сложностями: даже если пользователь разработает модель, он не сможет получить нужный показатель и будет вынужден либо делать выводы на основе промежуточных результатов, либо выбрать другую среду моделирования.

Пользователь, который захочет разработать модель с большим количеством ресурсов, столкнется с некомфортной работой: каждый тестовый запуск модели приводит к длительному ожиданию. Предполагаю, что эту проблему удастся сделать менее острой.

## 2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Техническое задание разработано в соответствии с ГОСТ 19.201-78 (Единая система программной документации Техническое задание Требования к содержанию и оформлению) [4].

### 2.1 Введение

Хранение состояния модели в базе данных позволяет расширить возможности пользователя для получения более детального описания состояния системы в каждый момент модельного времени. Такой интерфейс с предоставления данных позволяет использовать автоматизированные средства для обработки данных, что в свою очередь повышает эффективность работы и открывает возможность разработки новых подходов для получения выводов и количественных результатов.

### 2.2 Основания для разработки

- 1) Разработка ведется на основании следующих документов:
  - Задание на выполнение дипломного проекта.
  - Календарный план на выполнение дипломного проекта
- 2) Тема дипломного проекта:

Подсистема хранения состояния модели в РДО на основе реляционной базы данных

## **2.3 Назначение разработки**

Функциональное и эксплуатационное назначение разработки подсистемы хранения состояния модели в базе данных состоит в создании открытого интерфейса предоставляющего пользователю возможность воспользоваться автоматизированными системами обработки данных.

## **2.4 Требования к программе или программному изделию**

### ***2.4.1 Требования к функциональным характеристикам***

Требования к составу выполняемых функций реализуемой системы заключаются:

- 1) в выводе данных о состоянии модели в реляционную базу данных;
- 2) в предоставлении открытого доступа к данным этой базы данных.

### ***2.4.2 Требования к надежности***

Действия пользователя не должны приводить к сбоям в работе программы.

Система должна обладать средствами самодиагностики и автоматической индикации типов ошибок

### ***2.4.3 Условия эксплуатации***

Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В ±1 0% 50 Гц с защитным заземлением при следующих климатических условиях:

- температура окружающей среды – от 15 до 30 градусов С
- относительная влажность воздуха - от 30% до 80%;
- атмосферное давление - от 630 мм. р.с. до 800 мм. р.с.

### ***2.4.4 Требования к составу и параметрам технических средств***

Программный продукт должен работать на компьютерах со следующими характеристиками:

- объем ОЗУ не менее 1Гб;
- объем жесткого диска не менее 20 Гб;
- микропроцессор с тактовой частотой не менее 2ГГц
- монитор не менее 15” с разрешением от 1024\*768 и выше.

### ***2.4.5 Требования к информационной и программной совместимости***

Данная система должна работать под управлением операционных систем:

- Windows 7,
- Windows 8;
- Ubuntu 12.10.

#### **2.4.6 Требования к маркировке и упаковке**

Не предъявляются.

#### **2.4.7 Требования к транспортированию и хранению**

Не предъявляются.

#### **2.4.8 Требования к программной документации**

Не предъявляются.

### **2.5 Технико-экономические показатели**

Расчет экономической эффективности разработанного приложения не является целью дипломного проектирования, однако возможный экономический эффект может быть достигнут за счет следующих преимуществ системы:

- 1) Сокращение времени и средств на автоматизацию обработки данных.
- 2) Повышение гибкости работы с результатами моделирования.

### **2.6 Стадии и этапы разработки**

Состав, содержание и сроки выполнения работ по созданию системы в соответствии с календарным планом на выполнение дипломного проекта.

## 2.7 Порядок контроля и приемки

Контроль и приемка приложения должны состоять из следующих этапов:

- 1) запуск РДО-студии и проверка неизменности работы на тестовых моделях;
- 2) подключение к консоли управления СУБД и проверка адекватности данных трассировке модели.

## 2.8 Приложения

Документы, используемые при разработке, приведены в списке использованных источников.

Используемое при разработке программное обеспечение:

- Операционная система Microsoft Windows 8 Professional;
- Среда разработки Microsoft Visual Studio 2008 Professional;
- Библиотека Qt;
- Генератор синтаксических анализаторов Bison;
- СУБД PostgreSQL 9.2.4;
- Централизованная система управления версиями Subversion;
- Система документирования исходных текстов Doxygen;
- Исходные коды системы РДО;
- Инструмент управления СУБД – Navicat.

## 3 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

### 3.1 Цели разработки системы

Основной целью данного дипломного проекта является повышение эффективности моделирования в среде РДО методом повышения уровня открытости системы. Цель состоит из двух подцелей :

1) сделать информацию о работе модели более открытой .

Данная подцель так же декомпозируется на две:

- разработать структуру базы данных, которая позволит достаточно полно хранить информацию о состоянии модели;

- реализовать вывод состояния модели в базу данных.

2) проверить работоспособность алгоритма подбора релевантных ресурсов через базу данных.

Цель декомпозируется на три, причем первые две дублируются в первой подцели

- разработать структуру базы данных, которая позволит достаточно полно хранить информацию о состоянии модели;

- реализовать вывод состояния модели в базу данных;

- реализация возможности работы РДО-студии через базу данных.

### 3.2 Компоненты РДО

Система имитационного моделирования РДО безусловно является сложной и статически и динамически. На это указывает сложная

иерархическая структура системы со множеством различных связей между компонентами и ее сложное поведение во времени.

Ярко выраженная иерархическая структура и модульность системы определяют направление изучения системы сверху вниз. Т.е. принцип декомпозиции применяется до тех пор, пока не будет достигнут уровень абстракции, представление на котором нужных объектов не нуждается в дальнейшей детализации для решения данной задачи.

Для отображения зависимости между компонентами системы РДО и выделения среди них модернизируемых служит соответствующая диаграмма в нотации UML.

Базовый функционал представлен на диаграмме компонентов `rdo_kernel` реализует ядровые функции системы. Не изменяется при разработке системы.

`RAO-studio.exe` реализует графический интерфейс с пользователем. Не изменяется при разработке системы.

`rdo_repository` реализует управление потоками данных внутри системы и отвечает за хранение и получение информации о модели. Не изменяется при разработке системы.

`rdo_mbuilder` реализует функционал, используемый для программного управления типами ресурсов и ресурсами модели. Не изменяется при разработке системы.

`rdo_converter` конвертирует модели созданные в старой версии РДО, производя резервное копирование файлов оригинальной модели

Благодаря ему обеспечивается обратная совместимость версий системы. Не изменяется при разработке системы.

rdo\_simulator управляет процессом моделирования на всех его этапах. Он осуществляет координацию и управление компонентами rdo\_runtime и rdo\_parser. Не изменяется при разработке системы.

rdo\_parser производит лексический и синтаксический разбор исходных текстов модели, написанной на языке РДО Модернизируется при разработке системы.

rdo\_runtime отвечает за непосредственное выполнение модели, управление внутренней базой данных, базой знаний, планирование выполнения событий, и работу процессов Модернизируется при разработке системы.

Таким образом основные изменения должны затронуть модули rdo\_parser и rdo\_runtime.

### **3.3 База данных**

#### **3.3.1 Структуры баз данных**

##### **3.3.1.1 Требования к структуре**

Структура базы данных должна обеспечить возможность разработчику и РДО-студии хранить состояние модели, а также хранить дополнительную информацию об изменениях в системе (модели), которые перевели систему из одного состояния в другое.

Непосредственно состояние представляет собой состояние всех ресурсов и их параметров. Но для того, чтобы сформировать запрос для получения конкретного ресурса, этой информации мало Поэтому

структура базы данных должна также содержать информацию и о типах ресурсов.

Информация об изменениях в системе представляют собой трассировку этих самых изменений. На момент постановки задачи трассировка выводилась в один общий текстовый файл. Это сковывало пользователя перед автоматизацией обработки данных перед этим ему нужно разработать ПО, которое будет разбирать (парсить) этот текстовый файл, а потом уже обрабатывать данные. Правильно организованная структура БД позволит группировать изменения в системе, что избавит пользователя от необходимости разрабатывать ПО для работы с текстовым файлом трассировки

Каждый из блоков данных требует довольно обширную структуру

- 1) состояние ресурсов, их параметров, типов (далее – работа с ресурсами);
- 2) трассировка.

Поэтому каждый из блоков данных будет храниться в отдельной базе данных.

База данных должна хранить необходимую и достаточную информацию о состоянии модели. Дефицит информации не позволит работать с имеющимися данными: не будет общей картины. Профицит информации увеличит объем обрабатываемых данных, что приведет к снижению производительности работы СУБД.

В этой простой концепции кроется дилемма, что необходимо для одной системы может быть избыточно для другой. Вот пример

РДО-студия знает о типах всех параметров всех ресурсов. Для получения значения для конкретного параметра ей достаточно обратиться к этому параметру в памяти, принадлежащей процессу RAO studio.exe, и узнать его тип, а после этого обратиться через SQL-запрос к соответствующей таблице через драйвер СУБД

В свою очередь сторонняя система, например, система построения отчетов не имеет сведений о типах параметров поэтому для нее нужно ввести необходимую для нее (но избыточную для РДО-студии) информацию о типах.

Эта дилемма решается просто: вся текущая работа ведется ради сторонних систем, а, значит, они должны иметь всю необходимую информацию.

### **3.3.1.2 Описание структуры**

#### **3.3.1.2.1 База данных для хранения состояния ресурсов**

Отношения, которые должны содержаться в структуре базы данных, удовлетворяющей описанной в пункте 4.1.1

trp – содержит перечень типов ресурсов, но не их параметров и типов параметров;

rss – содержит перечень всех ресурсов, существующих в модели в каждый момент времени

rss\_param – содержит перечень всех параметров всех существующих ресурсов;

real\_rv – перечень всех значений в модели типа real;

int\_rv – перечень всех значений в модели типа int;

identifier\_rv – перечень всех значений в модели типа identifier;

bool\_rv – перечень всех значений в модели типа bool;

string\_rv – перечень всех значений в модели типа string

enum\_rv – перечень всех значений в модели типа enum

array\_rv – перечень всех массивов в модели;

array\_value – перечень всех идентификаторов значений массивов в модели;

rdo\_value – перечень всех величин, которые могут быть присвоены как значение параметра. Это отношение представляет собой излишек данных, но оно необходимо для сторонних систем: иначе они не смогут, основываясь только на структуре базы данных, определить, в каком отношении (какого типа) каждый параметр каждого ресурса

### 3.3.1.2 База данных для хранения трассировки

Некоторая информация дублируется – уже была в базе данных для хранения состояния ресурсов - но это плата за правильно организованное хранение. Отношения для хранения трассировки:

trc\_resource\_type – хранит перечень всех типов ресурсов в модели;

trc\_param – перечень всех параметров всех типов ресурсов;

trc\_enum\_vv – перечень всех возможных значений типа enum

trc\_resources – перечень всех ресурсов;

trc\_patterns – перечень всех шаблонов модели;

trc\_pattern\_relres – перечень всех типов ресурсов, которые нужны для всех шаблонов модели;

trc\_activities – перечень всех активностей, которые могут быть выполнены в модели;

trc\_watches – перечень всех точек сбора статистики показателей в модели.

Все предыдущие отношения в базе данных для хранения трассировки необходимы для контроля целостности данных. Сама трассировка представлена списком изменений, которые происходят в модели.

trc\_time – перечень всех моментов модельного времени, в которых происходят изменения в модели;

trc\_rows – перечень всех изменений в системе – обобщенный

Далее приведен набор отношений для описания событий в модели. Целый набор необходим из-за большой разнородности информации, которая описывает эти типы событий:

trc\_es – перечень системных событий;

trc\_eb\_ef – перечень событий начала и конца действий в модели;

trc\_e\_res – перечень ресурсов, принимающих участие в действии в модели;

trc\_ei – перечень нерегулярных событий в модели;

trc\_er – перечень продукционных правил в модели;

Набор отношений для описания изменений состояния ресурса

trc\_r – перечень всех изменений состояния ресурса

trc\_value – перечень всех идентификаторов значений параметров ресурсов;

trc\_r\_param\_value – перечень всех параметров ресурсов, которые подверглись изменению;

trc\_value\_int – перечень всех значений типа int;

trc\_value\_real – перечень всех значений типа real;

trc\_value\_bool – перечень всех значений типа bool;

trc\_value\_string – перечень всех значений типа string

trc\_value\_array – перечень всех массивов в трассировке;

trc\_value\_array\_vv – перечень всех идентификаторов значений в массивах.

Набор отношений для описания поиска по графу

trc\_sb – информация о начале поиска точки;

trc\_so – информация о раскрываемой вершине

trc\_st – информация о приемнике раскрываемой вершины

trc\_sd – информация о полученном решении поиска по графу;

trc\_se – информация о задействованных ресурсах для поиска по графу.

Отношения, которые не требуют дополнительных отношений для описания своих сущностей :

trc\_dps – перечень всей статистики по поиску на графе

trc\_v – перечень изменений точек сбора статистики показателей модели.

### **3.3.2 Выбор СУБД**

В роли СУБД была выбрана PostgreSQL. Данная СУБД интенсивно развивается и обладает множеством достоинств [3]. Перечислю из них

те, что представляют интерес для меня как для разработчика в рамках данного проекта:

- 1) полная совместимость с версией языка SQL 2008
- 2) кроссплатформенность;
- 3) поддержка кодировки UNICODE;
- 4) наличие определенных служебных полей в описаниях баз данных;
- 5) имеет интерфейс прикладного программирования для языка C++;
- 6) лицензия BSD.

### **3.3.3 Выбор драйвера работы с базой данных**

Принято решение использовать в качестве драйвера работы с базой данных – драйвер Qt. Выбор обусловлен тем что Qt обеспечивает работу приложения с СУБД PostgreSQL, к тому же РДО активно использует библиотеку Qt – чем меньше «швов» между компонентами, тем надежнее работает система.

### **3.3.4 Сериализация данных в базу данных**

Данные представлены двумя блоками:

- 1) Данные о типах ресурсов содержатся в rdo\_parser. Поэтому и программный код сериализации этих данных должен быть расположен в этой компоненте. Данные о самих ресурсах и их параметрах содержатся в компоненте rdo\_runtime. Здесь должен быть реализован механизм первичной сериализации данных а также механизм обновления данных в базе.

- 2) Данные трассировки формируются в обоих компонентах: rdo\_runtime и rdo\_parser.

### ***3.3.5 Работа модели с данными из БД***

Алгоритм поиска релевантного ресурса, удовлетворяющего условиям выбора, реализован через перебор всех ресурсов, существующих в модели. Ожидается, что СУБД работает более оптимально, и запрос к базе данных типа "SELECT" продемонстрирует преимущество в скорости работы, не смотря на потерю быстроты взаимодействия на меж процессное взаимодействие

## 4 ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

### 4.1 Инициализация структуры базы данных

Базы данных должны быть созданы в момент между окончанием компиляции модели и началом работы модели. Выбор этого момента обусловлен тем, что после компиляции модели в памяти, принадлежащей РДО студии существует структурированная модель представленная экземплярами классов C++. А именно: тип ресурса, ресурсы.

Следом должны быть созданы структуры баз данных. В ходе работы модели отношения базы данных наполняются – на этом этапе могут работать системы построения отчетов реального времени. После окончания работы модели работа с базами данных прекращается. Таким образом, системы построения отчетов, работающие не в режиме реального времени имеют возможность нормальной работы

Этот сценарий подойдет только для первого запуска какой-ли модели РДО. В остальных случаях (в большинстве) надо перед выше описанной последовательностью выполнить очистку базы данных предыдущих запусков моделей.

Таким образом, первым этапом работы с базами данных станет их удаление, если они существуют, и только затем их создание и т.д.

### 4.2 Сериализация данных в базу данных

#### 4.2.1 Создание инструментов для работы с базой данных

Для работы с базой данных необходим целый набор инструментов которые помогут работать с ней удобно и быстро. Необходимый минимальный набор инструментов (атомарный) предоставляет драйв

Qt. На их основе и должны появиться инструменты, необходимые в рамках данной задачи

#### 4.2.1.1 Необходимые классы

Для создания этих инструментов создан новый проект в рамках исходного кода РДО. В нем создано три новых класса:

- IDB - класс-интерфейс, через который происходят обращения созданному набору инструментов, нужен для ограничения доступа к методам из остальных классов исходного кода РДО;
- GeneralDB – наследник IDB - класс, который реализует методы этого интерфейса и предоставляет набор инструментов для работы с существующей базой данных
- InitStructDB – наследник GeneralDB – класс, который уничтожает старые базы данных, создает новые и инициализирует структуры новых баз данных.

#### 4.2.1.2 Описание интерфейсов компонентов классов

В разделе 4.2.1 были перечислены классы, которые разработаны для работы с СУБД. Ниже представлено краткое описание методов и атрибутов этих классов.

Класс IDB не имеет атрибутов. Его методы:

- 1) `virtual void insertRow (const QString& tableName, const QString& qRow) = 0;`  
метод вносит новые данные(qRow) в существующее отношение (tableName).

- 2) `virtual int insertRowInd (const QString& tableName, const QString& qRow) = 0;`  
метод вносит новые данные (qRow) в существующее отношение (tableName) и возвращает индексный номер новых данных (it);
- 3) `virtual void queryExec (const QString& query) = 0;`  
метод выполняет произвольный запрос (query) к СУБД
- 4) `virtual void queryListPushBack(const QString& query) = 0;`  
метод добавляет запрос (query) в список отложенных запросов;
- 5) `virtual void queryListExec () = 0;`  
метод выполняет список отложенных запросов;
- 6) `virtual int queryListExecInd () = 0;`  
метод выполняет список отложенных запросов и возвращает индексный номер (it) внесенных данных из последнего запроса;
- 7) `virtual void pushContxt (CREF(bany) context) = 0;`  
метод хранит произвольное значение (context), сохраненное для дальнейших действий
- 8) `template <class T> void pushContext (T context);`  
то же, что и 7 – другая сигнатура метода, более удобная;
- 9) `virtual bany popContxt () = 0;`  
метод, возвращающий значение сохраненное для дальнейших действий
- 10) `template <class T> T popContext ();`  
то же, что и 9 – другая сигнатура метода, более удобная;
- 11) `virtual bool isEmptyContext () = 0;`  
метод проверяет, существует ли значение, отложенное для дальнейших действий
- 12) `virtual QSqlDatabase getQtDB () = 0;`

метод возвращает объект подключения Qt (QSqlDatabase).

Класс GeneralDB реализует все методы класса IDB, а также содержит атрибуты:

- 1) QString m\_hostName; - имя узла, на котором развернута СУБД;
- 2) QString m\_databaseName; - имя базы данных, к которой будут адресованы запросы;
- 3) QString m\_userName; - имя пользователя, от имени которого отправляются запросы;
- 4) QString m\_password; - пароль пользователя, от имени которого отправляются запросы;
- 5) int m\_port; - TCP порт, через который инициализируется подключение к СУБД;
- 6) QSqlDatabase m\_db; - объект подключения Qt;
- 7) QString m\_queryList; - список отложенных запросов;
- 8) bany m\_context; - значение, отложенное для дальнейших действий

Также класс GeneralDB содержит дополнительные методы:

- 1) GeneralDB(const QString& hostName, const QString& databaseName, const QString& userName, const QString& password, const int& port);  
конструктор класса, его параметры инициализируют значения атрибутов класса;
- 2) ~GeneralDB();  
деструктор класса;
- 3) void initDB();  
метод создает подключение к указанной базе данных

Класс InitStructDB не имеет атрибутов. Его методы:

- 1) `InitStructDB(QString nameDB);`  
конструктор класса, создает разные структуры базы данных в зависимости от имени базы данных (`nameDB`);
- 2) `static void dropDB(QString db);`  
метод, удаляющий базу данных с именем `db`
- 3) `static void createDB(QString db);`  
метод, создающий базу данных с именем `db`
- 4) `void rdoValueTable (QString tableName, QString dataType);`  
метод создает отношение (`tableName`), хранящее значение параметров ресурсов РДО (`dataType`);
- 5) `void trigger (QString tableName, QString functionName);`  
метод, создает триггер для отношения (`tableName`), вызывающий функцию (`functionName`);
- 6) `void generateCreateDBQuery();`  
метод, создает структуру базы данных для хранения типов ресурсов, ресурсов и их параметров;
- 7) `void generateCreateTrcDBQuery();`  
метод, создает структуру базы данных для хранения трассировки.

#### ***4.2.2 Использование инструментов для работы с базой данных***

В свете того, что наиболее полную информацию об объекте хранит сам объект, наиболее логично и рационально выполнять экспорт данных из этих объектов. Так сериализация в базу данных информации о типах ресурсов должна быть реализована как метод класса `RDORTPResType`, а сериализация данных о ресурсах должна быть расположена в `RDOResource`.

Исключение из этой концепции составят параметры ресурсов для них в исходном коде РДО не заведено отдельных классов.

## **4.3 Работа модели с данными из базы данных**

### ***4.3.1 Выбор способа проверки работы***

Для того, чтобы проверить полноценную замену работы текущего алгоритма подбора релевантного ресурса на делегирование этого процесса подбора СУБД, необходимо разработать механизм генерации SQL-запросов из синтаксических конструкций РДО. Это трудоемкая задача, которая может не иметь смысла: необходимо сначала проверить эффективность такой замены. Эту проверку можно провести в частных случаях. Т.е. ввести в синтаксические конструкции РДО возможность написать в теле модели конкретный SQL-запрос вместо обычного (на данный момент) синтаксического выражения.

Такое нововведение имеет временный характер, введено исключительно для проверки.

Нельзя не заметить, что исключение из цепочки обработки модели генератора SQL-запросов исказит картину, т.е. сделает работу модели немного быстрее полноценной замены. Но если даже такая вариация продемонстрирует падение быстродействия, то разработка генератора SQL-запросов оказалась бы бессмысленной. В этом и заключается Исследовательская часть.

### 4.3.2 Синтаксические конструкции

Синтаксические конструкции РДО обрабатываются программным модулем Bison.

На момент постановки задачи синтаксическая конструкция, выбирающая ресурс для шаблона моделирования, выглядела следующим образом:

**\$Body**

\_Релевантный \_ ресурс

**Choice from** <логическое выражение>

[<способ выбора>]

Для решения поставленной задачи использована следующая синтаксическая конструкция:

**\$Body**

\_Релевантный \_ ресурс

**sql\_select** '<SQL-запрос>'

Для обеспечения работоспособности данной конструкции были внесены соответствующие изменения в исходный код РДО

## 5 РАБОЧЕЕ ПРОЕКТИРОВАНИЕ

### 5.1 Реализация инструментов работы с базами данных

#### 5.1.1 Класс *GeneralDB*

##### 5.1.1.1 Метод `queryExec`

```
void GeneralDB::queryExec(const QString& query)
{
    1. std::cout << query.toStdString() << "... " << std::endl;

    2. m_db.exec(query);

    3. QSqlError lastError = m_db.lastError();
    4. switch (lastError.type())
    5. {
    6. case QSqlError::NoError:
        std::cout << query.toStdString() << "... ok"
        << std::endl;
        break;

    7. default:
        std::cerr << query.toStdString() << "...
        failed: " << lastError.text().toStdString()
        << std::endl;
        break;

    8. }
}
```

В этом методе происходит выполнение произвольного запроса (строка 2), а также происходит вывод в консоль работы с запросами (все остальные строки тела метода). Очень полезная информация при поиске ошибок в исходном коде.

### 5.1.1.2 Метод insertRow

```
void GeneralDB::insertRow(const QString& tableName, const
QString& qRow)
{
    1. queryExec(QString("INSERT INTO %1 VALUES(%2);")
                .arg(tableName)
                .arg(qRow));
}
```

В этом методе происходит выполнение запроса типа "INSERT". Метод упрощает работу разработчика, делает исходный код лучше читаемым.

### 5.1.1.3 Метод insertRowInd

```
int GeneralDB::insertRowInd(const QString& tableName, const
QString& qRow)
{
    1. QSqlQuery* query = new QSqlQuery(m_db);
    2. query->exec(QString("INSERT INTO %1 VALUES(%2) RETURNING
        id;")
                .arg(tableName)
                .arg(qRow));
    3. query->next();
    4. return query->value(query->record().indexOf("id")).toInt();
}
```

Метод аналогичен методу insertRow из раздела 5.1.1.2. Отличие в том, что этот метод возвращает индексный номер вновь внесенных в отношение данных.

В строке 1 происходит создание указателя на объект запроса, адресованного указанной базе данных

Строка 2 формирует запрос и отправляет его СУБД.

Строка 3 получает от СУБД результат, возвращенный в выполненном запросом.

В строке 4 происходит извлечение данных из поля id, приведение этого значения к целочисленному и возвращение итогового значения из метода.

#### 5.1.1.4 Метод queryListPushBack

```
void GeneralDB::queryListPushBack(const QString& query)
{
    1. m_queryList.append(query);
}
```

В конец строки (атрибута класса) добавляется новый запрос или его часть.

#### 5.1.1.5 Метод queryListExec

```
void GeneralDB::queryListExec()
{
    1. queryExec(m_queryList);
    2. m_queryList.clear();
}
```

Строка 1 отправляет СУБД запрос или несколько запросов, представленных одной строкой

Строка 2 очищает атрибут класса от имеющего в нем значения. Только что отправленный запрос не имеет ценности для истории

#### 5.1.1.6 Метод init

```
void GeneralDB::initDB()
{
    1. m_db = QSqlDatabase::addDatabase("QPSQL", m_databaseName);

    2. m_db.setHostName(m_hostName);
    3. m_db.setDatabaseName(m_databaseName);
    4. m_db.setUserName(m_userName);
    5. m_db.setPassword(m_password);
}
```

```
6. m_db.setPort          (m_port);

7. if (!m_db.open())
8. {
           std::cout << "Connection to datebase failed! :("
           << std::endl;
9. }
}
```

Строка 1 указывает библиотеке Qt каким именно драйвером подключаться к базе данных. QPSQL - драйвер для работы с СУБД PostgreSQL.

Строки 2 – 6 задают параметры подключения.

Строки 7 – 9 выполняют подключение к базе данных. Если подключение не удалось выполнить, в консоль выводится сообщение об ошибке.

## 5.1.2 Класс *InitStructDB*

### 5.1.2.1 Метод *dropDB*

```
void InitStructDB::dropDB(QString db)
{
    1. {
        2. GeneralDB dbTemp("localhost", "postgres", "postgres",
           "rdo", 5432);
        3. dbTemp.queryExec(QString("DROP DATABASE IF EXISTS %1;")
           .arg(db));
    4. }
    5. QSqlDatabase::removeDatabase("postgres");
}
```

Строка 2 создает подключение к технологической базе данных

Строка 3 адресует запрос удаления базы данных с указанным названием, если она существует, технологической базе данных

Строка 5 удаляет сущность Qt, созданную для контакта с технологической базой данных

### 5.1.2.2 Метод createDB

```
void InitStructDB::createDB(QString db)
{
    1. {
        2. GeneralDB dbTemp("localhost", "postgres", "postgres",
            "rdo", 5432);
        3. dbTemp.queryExec(QString("CREATE DATABASE %1;")
            .arg(db));
        4. }
    5. QSqlDatabase::removeDatabase("postgres");
}
```

Строка 2 создает подключение к технологической базе данных

Строка 3 адресует запрос создания базы данных с указанным названием технологической базы данных

Строка 5 удаляет сущность Qt, созданную для контакта с технологической базой данных

### 5.1.2.3 Конструктор класса InitStructDB

```
InitStructDB::InitStructDB(QString nameDB)
    : GeneralDB("localhost", nameDB, "postgres", "rdo", 5432)
{
    1. if(nameDB == QString("rdo"))
        2. generateCreateDBQuery ();
    3. if(nameDB == QString("trc"))
        4. generateCreateTrcDBQuery();

    5. queryListExec();
}
```

Строки 1 и 2 иницируют создание очереди отложенных запросов для создания структуры базы данных для работы с ресурсами.

Строки 3 и 4 иницируют создание очереди отложенных запросов для создания структуры базы данных для работы с трассировкой.

Строка 5 выполняет отложенную очередь запросов.

## 5.2 Использование созданных инструментов для сериализации модели в базу данных

### 5.2.1 Сериализация типов ресурсов

```
void RDORTPResType::serializeInDB(REF(IDB) db) const
{
    db.insertRowInd("rtp",QString("DEFAULT,'%1',%2")
        .arg(QString::fromStdString(name()))
        .arg(m_permanent ? "true" : "false"));
}
```

В этом методе используется инструмент 5.1.1.3. Таким образом, в отношении rtp отправляется запрос на внесение данных об очередном типе данных: его имя и признак перманентности.

### 5.2.2 Сериализация ресурса

```
void RDOResource::serializeInDB() const
{
    1. int rss_id = m_db->insertRowInd("rss",QString("%1,%2,%3")
        .arg(getTraceID())
        .arg(boost::lexical_cast<int>(getTypeId()))
        .arg(traceable() ? "true" : "false"));
    2. int param_id = -1;

    3. BOOST_FOREACH(const RDOValue& param, m_paramList)
    4. {
        5. param.serializeInDB(*m_db);
        6. m_db->insertRow("rss_param",QString("%1,%2,%3")
            .arg(rss_id)
            .arg(++param_id)
            .arg(m_db->popContext<int>()));
    7. }
}
```

Строка 1 иницирует запись в отношение rss данных об очередном ресурсе: его порядком номере, номере его типа и признаке трассировки.

Строка 2 создает целочисленный объект который используется для задания порядковых номеров параметров.

Строка 5 иницирует метод записи объекта в базу данных.

Строка 6 записывает в отношении rss\_param сведения о параметре ресурса.

### 5.2.3 Сериализация значений параметров ресурсов в базу данных

```
void RDOValue::serializeInDB(REF(IDB) db) const
{
    1. LPRDOArrayType pThisArrayType;

    2. QString connectionName = db.getQtDB().connectionName();
    3. if (connectionName == "rdo")
    4. {
        5.     switch (typeID())
        6.     {
            7.     case RDOType::t_unknow      : break;
            8.     case RDOType::t_int        :
DEFINE_SERIALIZE_RDO_VALUE("int_rv", getInt()); break;
            9.     case RDOType::t_real      :
DEFINE_SERIALIZE_RDO_VALUE("real_rv", getDouble()); break;
            10.    case RDOType::t_enum       :
DEFINE_SERIALIZE_RDO_VALUE("enum_rv",QString("%1").arg(
QString::fromStdString(getAsString()))); break;
            11.    case RDOType::t_bool      :
DEFINE_SERIALIZE_RDO_VALUE("bool_rv",QString("%1").arg(
QString::fromStdString(getAsString())));break;
            12.    case RDOType::t_string    :
DEFINE_SERIALIZE_RDO_VALUE("string_rv",QString("%1").arg(
QString::fromStdString(getString ( ))));break;
            13.    case RDOType::t_identificator :
DEFINE_SERIALIZE_RDO_VALUE("identificator_rv",QString("%1").
.arg(QString::fromStdString(getAsString()))); break;
            14.    case RDOType::t_pointer    : pThisArrayType =
m_pType.object_dynamic_cast<RDOArrayType>(); if
(pThisArrayType) getPointer<RDOArrayValue>()->
serializeInDB(db); break;
            15.    default                    : throw
RDOValueException("Данная величина не может быть записана в
базу данных");
        16.    }
        17.    }
        18.    else if (connectionName == "trc")
        19.    {
            20.    switch (typeID())
            21.    {
                22.    case RDOType::t_int        :

```

```

    23. case RDOType::t_enum           :
db.pushContext<int>(db.insertRowInd("trc_value_int",QString(
"DEFAULT, %1 ").arg(getInt())));break;
    24. case RDOType::t_real          :
db.pushContext<int>(db.insertRowInd("trc_value_real" ,
QString("DEFAULT, %1 ").arg(getDouble())));break;
    25. case RDOType::t_bool          :
db.pushContext<int>(db.insertRowInd("trc_value_bool",QString(
"DEFAULT, %1 ").arg(QString::fromStdString(getAsString()))));
break;
    26. case RDOType::t_string        :
db.pushContext<int>(db.insertRowInd("trc_value_string",QString(
"DEFAULT, '%1' ").arg(QString::fromStdString(getAsString()))));
break;
    27. case RDOType::t_pointer       : pThisArrayType =
m_pType.object_dynamic_cast<RDOArrayType>(); if
(pThisArrayType) getPointer<RDOArrayValue>()->
serializeInDB(db); break;
    28. case RDOType::t_identificator :
    29. case RDOType::t_unknow        :
    30. default                       : throw
RDOValueException("Трассировка данной величины не может быть
записана в базу данных");
    31. }
    32. }
}

```

Строка 1 создает пустой указатель на объект типа величины которая может храниться в массиве. Указатель может быть не использован в методе.

Строка 2 определяет имя базы данных, в которую нужно внести данные.

Строки 3 – 16 иницируют запись в базу данных для хранения ресурсов.

Строки 18 – 31 иницируют запись в базу данных для хранения трассировки.

## 6 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

### 6.1 Планирование исследования

Исследование изменения быстродействия работы программного обеспечения можно проверить статистически – сравнить время работы программного обеспечения в разных ситуациях. Меня, как разработчика, интересует влияние на быстродействие каждого из модулей разработанной мною подсистемы а также влияние на быстродействие РДО-студии на определенных этапах ее работы.

#### 6.1.1 Контроль версий РДО-студии

Для того чтобы выделить влияние каждого модуля, можно использовать набор версий РДО-студии которые не будут предоставлены пользователям, но для исследования они могут быть использованы.

В данном разделе дипломного проекта рассмотрено сравнение быстродействия РДО-студии нескольких версий

- 1) основная версия на текущий момент (без внедрения в исходный код каких-либо разработок данного дипломного проекта);
- 2) версия, в исходный код которой внесены все приведенные выше разработки, но работа РДО-студии не включает какой-либо работы с СУБД;
- 3) версия, которая хранит в базе данных только ресурсы;
- 4) версия, которая хранит в базе данных только трассировку событий;
- 5) версия, которая хранит в базах данных и ресурсы, и трассировку.

### **6.1.2 Контроль этапа инициализации модели**

Важен этап инициализации модели. До внедрения данной разработки он состоял только из подэтапа компиляции модели. Теперь к нему добавился подэтап сериализации структуры модели в базу данных.

Предлагается собирать статистику длительности работы модели с этапом инициализации и без него. Это необходимо сделать для каждой версии РДО-студии.

### **6.1.3 Контроль работы модели с данными из базы данных**

Отнюдь не все версии РДО-студии, приведенные в пункте 6.1.1 могут работать с данными из базы данных. Для этой работы нужна сериализация ресурсов. Как видно, данную проверку можно провести только на версиях 3) и 5).

Значит, для версий 3) и 5) будет по две экспериментальные площадки.

### **6.1.4 Планирование экспериментов**

Таким образом, в рамках исследовательской части имеется 14 экспериментальных площадок: (одна для версий 1) 2) 4) + две для версий 3) и 5) x 2 варианта определения времени работы модели.

В рамках каждой площадки нужно обеспечить идентичные условия выполнения эксперимента для того, чтобы результаты были показательны. На каждой площадке должна отработать три модели с разным количеством ресурсов (именно они определяют объем обрабатываемых данных, а, значит, и ресурсоемкость каждой модели).

Малое количество ресурсов (около десятка или сотни) не станет причиной длительной работы РД-студии: отработает модель за секунды (версия 1) или за 13 (версия 5) – не так уж важно для пользователя, который запускает работу модели вручную и после этого разбирает результаты моделирования. Поэтому тесты должны быть нагрузочными: модели будут содержать 1000, 5000 и 10000 ресурсов.

Будет получено 42 длительности работы модели: (3 модели) x (14 площадок).

Каждый запуск модели сопровождается в некоторой степени случайными событиями запуском других приложений или запуском каких-то процессов внутри этих приложений. В современном мире количество одновременно запущенных процессов на персональном компьютере может достигать сотни. Уследить за каждым из них невозможно, а зачастую невозможно контролировать их поведение.

Для того, чтобы исключить случайность нужно для каждой из 4 целей провести бесконечно большее число экспериментов и усреднить полученные значения. Этого, увы, сделать не представляется возможным. Пробные эксперименты показали высокую «кучность» результатов, поэтому я, как исследователь, решил ограничить количество запусков модели для определения каждой длительности работы модели пятью запусками.

Итого, будет проведено 210 запусков: (5 запусков) x (42 длительности работы модели).

### 6.1.5 Бланк журнала экспериментов

Бланк журнала экспериментов представлен в Приложении 1.

В этом разделе представлено его короткое описание.

Таблица 1. Пояснение к бланку журнала экспериментов.

№	Условные обозначения	Пояснение
1	trunk	версия РДО-студии 1
2	like a trunk	версия РДО-студии 2
3	dev_db (mainstream)	версия РДО-студии 3 – без работы модели с данными из базы данных
4	dev_db(3+sql_query)	версия РДО-студии 3 – с работой модели с данными из базы данных
5	dev_db(trc)	версия РДО-студии 4
6	dev_db(3+trc)	версия РДО-студии 5 - без работы модели с данными из базы данных
7	dev_db(4+trc)	версия РДО-студии 5 - с работой модели с данными из базы данных
8	default	время работы модели с этапом инициализации модели
9	only_runtime	время работы модели без этапа инициализации модели
10	init_time	время инициализации модели

## 6.2 Анализ проведенных экспериментов

Журнал проведенных экспериментов в рамках исследовательской части дипломного проекта приведен в приложении 2.

### 6.2.1 Общие выводы о работе системы

На основе полученных данных можно сделать следующие выводы:

1. Версии РДО-студии 1) и 2) продемонстрировали ожидаемую идентичность результатов (в некотором приближении).  
Этот факт говорит о том, что внесенные изменения в систему не навредили работе всей системы в целом.
2. Модуль работы с базой данных для хранения ресурсов создает гораздо большую нагрузку при инициализации модели, чем модуль работы с базой данных для хранения трассировки.  
Это выражается в более длительном ожидании начала работы модели.
3. Как ни странно, но время инициализации модели почти не увеличивается при подключении к работе модуля работы с ресурсами еще и модуля работы с трассировкой.  
Это неожиданный, но положительный факт. Время ожидания пользователем не увеличивается.
4. Модуль работы с трассировкой создает больше нагрузки во время работы модели, чем модуль работы с ресурсами.  
Это неочевидный факт. Тест проводился с максимальной детализированной трассировкой изменений в модели. При уменьшении детализации, негативное влияние на производительность РДО-студии так же уменьшится.

5. Обнаружена большая потеря производительности работы РДО-студии при обращении к СУБД по сравнению с работой текущей версии РДО студии. Потери большие составляют от 35% до 180% для 10 000 ресурсов в модели и от 270% до 570% для 1000 ресурсов.

Несомненно, это недостаток подсистемы. В защиту разработанной системы можно озвучить несколько немаловажных фактов:

- 1) Модели, содержащие 10 000 ресурсов – настоящая редкость. На памяти разработчиков РДО таких моделей еще не было. Разработка модели системы, для описания которой понадобилось бы столько ресурсов будет и так сопряжена с большими временными затратами – время работы модели не будет «узким» местом процесса разработки.
- 2) Модели, содержащие 1 000 ресурсов, так же представляют нечастое явление. Но, несмотря на большие относительные потери производительности, общее время работы не превышает 30 секунд. Это время не станет проблемой в процессе разработки модели.

### **6.2.2 Выводы о работе модели с данными из базы данных**

К сожалению, испытания показали, что работа алгоритма по поиску релевантного ресурса, реализованная в текущей версии РДО студии гораздо более эффективна, чем работа алгоритма, реализованная через делегирование этой функции СУБД. Потеря производительности составляет около 25% при сравнении с работой РДО студии с СУБД в версии 3) или 5).

Это говорит о том, что предложенный способ оптимизации работы алгоритма поиска релевантного ресурса неприемлем.

### **6.2.3 Анализ причин обнаруженного поведения системы**

Такие большие потери производительности требуют подробного рассмотрения. Причины такого поведения системы:

- 1) Межпроцессное взаимодействие – на момент постановки задачи все данные РДО-студии содержались в области оперативной памяти, принадлежащей процессу `RAOstudio.exe`. Когда в работе РДО-студии появилась СУБД, то появилась необходимость в передаче информации между процессом `RAO-studio.exe` и процессом `postgres.exe`. Для этого требуется обращение к операционной системе. Таким образом процесс моделирования вовлекавший ранее только один процесс в пределах ОС теперь вовлек в работу два процесса и ОС.
- 2) Техническая необходимость отправки множества запросов вместо нескольких агрегированных. В процессе моделирования постоянно появляются новые объекты и для контроля целостности процесс контроля индексный номеров делегирован

СУБД, что приводит к необходимости частой отправки запросов для получения этих номеров.

- 3) Стандарт SQL был разработан, прежде всего, для обеспечения возможности организованного и безопасного хранения данных, а так же удобства работы с ними. Это большие плюсы стандарта, которые не могли не вылиться в некоторые минусы. Известная проблема работы с СУБД – их производительность. Например, запросы к базам данных в рамках разработанной системы выполняются за 45-85 мс (время отличается на ПЭВМ разной мощности). По меркам СУБД – это прекрасное время обработки запросов. Но для данной задачи этого мало

## 7 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ

В организационно-экономической части дипломного проекта выполнен расчет стоимости разработки подсистемы хранения состояния модели РДО на основе реляционной базы данных.

Т.к. РДО-студия представляет собой программное обеспечение, которое можно использовать независимо от разработчиков на разных персональных компьютерах, удовлетворяющих требованиям, то РДО-студию можно определить как программный продукт (ПП).

В организационно-экономическую часть данного дипломного проекта входят:

- 1) организационная часть – расчет трудоемкости разработки ПП;
- 2) экономическая часть – расчет себестоимости разработки ПП.

### 7.1 Организация и планирование процесса разработки программного продукта

Разработка подсистемы хранения состояния модели РДО на основе реляционной базы данных – сложный и длительный процесс

Организация и планирование процесса разработки программного продукта или программного комплекса при традиционном методе планирования предусматривает выполнение следующих работ:

1. формирование состава выполняемых работ и группировка их по стадиям разработки;
2. расчет трудоемкости выполнения работ;
3. установление профессионального состава и расчет количества исполнителей ;
4. определение продолжительности выполнения отдельных этапов разработки;
5. построение календарного графика выполнения разработки;

6. контроль выполнения календарного графика.

Ниже приведен перечень стадий и состава работ Таблица 2) при создании программного продукта.

Таблица 2. Перечень стадий и состава работ

Стадия разработки программного продукта	Состав выполняемых работ
Техническое задание	<p>Постановка задач, выбор критериев эффективности. Разработка технико-экономического обоснования разработки.</p> <p>Определение состава пакета прикладных программ, состава и структуры информационной базы Предварительный выбор методов выполнения работы. Разработка календарного плана выполнения работ.</p>
Эскизный проект	<p>Предварительная разработка структуры входных и выходных данных. Разработка общего описания алгоритмов реализации решения задач. Разработка пояснительной записки</p> <p>Консультации разработчиков постановки задач. Согласование и утверждение эскизного проекта.</p>
Технический проект	<p>Разработка алгоритмов решения задач. Разработка пояснительной записки Согласование и утверждение технического проекта. Разработка структуры программы.</p> <p>Разработка программной документации и передача ее для включения в технический проект Уточнение структуры анализа и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.</p>
Рабочий проект	<p>Комплексная отладка задач и сдача в опытную эксплуатацию.</p> <p>Разработка проектной документации Программирование и отладка программ. Описание контрольного примера.</p> <p>Разработка программной документации Разработка согласование программы и методики испытаний.</p> <p>Предварительное проведение всех видов испытаний.</p>

Внедрение	Подготовка и передача программной документации для сопровождения с оформлением соответствующего Акта приема-сдачи работ. Проверка алгоритмов и программ решения задач, корректировка документации после опытной эксплуатации программного продукта.
-----------	---

Трудоемкость разработки программной продукции зависит от ряда факторов, основными из которых являются следующие: степень новизны разрабатываемого программного комплекса, сложность алгоритма его функционирования, объем используемой информации вид ее представления и способ обработки, а также уровень используемого алгоритмического языка программирования.

По степени новизны разрабатываемый программный продукт может быть отнесен к группе новизны «Б» (разработка ПП, не имеющей аналогов, в том числе разработка пакетов прикладных программ).

По степени сложности алгоритма функционирования – к 1 группе сложности (реализующие оптимизационные и моделирующие алгоритмы).

По виду представления исходной информации ПП относится к группе 11 – исходная информация представлена в форме документов, имеющих различный формат и структуру. Требуется учитывать взаимовлияние показателей в различных документах.

По структуре выходной информации относим программный продукт к группе 22 – требуется вывод на печать одинаковых документов, вывод информационных носителей на машинные носители.

$t_{пп}$  - трудоемкость разработки программного продукта определяется как сумма величин трудоемкости выполнения отдельных стадий разработки. Формула для определения трудоемкости разработки ПП имеет вид:

$$t_{\text{ПП}} = t_{\text{ТЗ}} + t_{\text{ЭП}} + t_{\text{ТП}} + t_{\text{РП}} + t_{\text{В}},$$

где  $t_{\text{ТЗ}}$  - трудоемкость разработки технического задания на ПП;

$t_{\text{ЭП}}$  - трудоемкость разработки эскизного проекта ПП;

$t_{\text{ТП}}$  - трудоемкость разработки технического проекта ПП;

$t_{\text{РП}}$  - трудоемкость разработки рабочего проекта ПП;

$t_{\text{В}}$  - трудоемкость внедрения разработанного ПП.

### 7.1.1 Расчет трудоемкости разработки технического задания

Трудоемкость разработки технического задания рассчитывается по формуле:  $t_{\text{ТЗ}} = t_{\text{РЗ}} + t_{\text{РП}}$ ,

где  $t_{\text{РЗ}}$  - затраты времени разработчика постановки задачи на разработку ТЗ, чел.-дн.;

$t_{\text{РП}}$  - затраты времени разработчика ПП на разработку ТЗ, чел.-дн.

$$t_{\text{РЗ}} = t_3 \cdot K_{\text{РЗ}},$$

$$t_{\text{РП}} = t_3 \cdot K_{\text{РП}},$$

где  $t_3$  - норма времени на разработку ТЗ на ПП, чел.-дн.

Исходя из рекомендаций (3) в данном случае  $z = 47$  чел.-дн. (Группа новизны ПП - Б; функциональное назначение ПП – техническая подготовка производства);

$K_{\text{РЗ}}$  - коэффициент, учитывающий удельный вес трудоемкостей работ, выполняемых разработчиком постановки задачи на стадии ТЗ. В случае совместной с разработчиком ПО разработки  $K_{\text{РЗ}} = 0,65$ ;

$K_{PP}$  - коэффициент, учитывающий удельный вес трудоемкост работ, выполняемых разработчиком ПП на стадии ТЗ. В случае совместной с разработчиком ПО разработки  $K_{PP} = 0,35$ .

$$t_3 = 47 \cdot (0,65 + 0,35) = 47 \text{ чел.-дн.}$$

### 7.1.2 Расчет трудоемкости выполнения эскизного проекта

Трудоемкость разработки эскизного проекта  $t_{ЭП}$  рассчитывают по формуле:  $t_{ЭП} = t_{P3} + t_{PP}$ ,

где  $t_{P3}$  - затраты времени разработчика постановки задач на разработку ЭП, чел.-дн.;

$t_{PP}$  - затраты времени разработчика ПП на разработку ЭП, чел.-дн.;

$$t_{P3} = t_{Э} \cdot K_{P3},$$

$$t_{PP} = t_{Э} \cdot K_{PP},$$

где  $t_{Э}$  - норма времени на разработку ЭП программного продукта, чел.-дн.

$$t_{Э} = 99 \text{ чел.-дн.};$$

$K_{P3}$  - коэффициент, учитывающий удельный вес трудоемкост работ, выполняемых разработчиком постановки задачи на стадии ЭП. В случае совместной с разработчиком ПО разработки ЭП  $K_{P3} = 0,7$ ;

$K_{PP}$  - коэффициент, учитывающий удельный вес трудоемкост работ, выполняемых разработчиком ПП на стадии ЭП. В случае совместной с разработчиком постановки задач разработки ЭП  $K_{PP} = 0,3$ ;

$$t_{ЭП} = 99 \cdot (0,7 + 0,3) = 99 \text{ чел.-дн}$$

### 7.1.3 Расчет трудоемкости выполнения технического проекта

Трудоемкость разработки технического проекта зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации и определяется как сумма времени затраченного разработчиком постановки задач и разработчиком ПП, и определяется по формуле:  $t_{ТП} = (t_{рз} + t_{рп}) \cdot K_B \cdot K_P$ ,

где  $t_{рз}$ ,  $t_{рп}$  – норма времени, затрачиваемого на разработку ТП разработчиком постановки задач и разработчиком ПО (одна форма входной информации и две формы выходной)

$$t_{рз} = 43 \text{ ч.},$$

$$t_{рп} = 13 \text{ ч.}$$

$K_P$  - коэффициент учета режима обработки информации

$$K_P = 1.45$$

$K_B$  - коэффициент учета вида используемой информации определяется из выражения:

$$K_B = \frac{K_{П} \cdot N_{П} + K_{НС} \cdot N_{НС} + K_B \cdot N_B}{N_{П} + N_{НС} + N_B},$$

где  $K_{П}$ ,  $K_{НС}$ ,  $K_B$  - значения коэффициентов учета вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

$N_{П}$ ,  $N_{НС}$ ,  $N_B$  – количество наборов данных для переменной, нормативно-справочной информации и баз данных соответственно

Для группы новизны Б:

$$K_{П} = 1,2 \quad N_{П} = 2$$

$$K_{НС} = 1,08 \quad N_{НС} = 0$$

$$K_B = 3,12 \quad N_B = 1$$

$$\text{Таким образом, } K_B = \frac{1,2 \cdot 2 + 1,08 \cdot 0 + 3,12 \cdot 1}{2 + 0 + 1} = 1,84$$

$$t_{ТП} = (43 + 13) \cdot 1,84 \cdot 1,45 = 149 \text{ чел.-дн.}$$

#### 7.1.4 Расчет трудоемкости разработки рабочего проекта

Трудоемкость разработки рабочего проекта зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации, сложности и алгоритм функционирования, сложности контроля информации, степени использования готовых программных модулей, уровня алгоритмического языка программирования и определяется по формуле:

$$t_{ПП} = K_K \cdot K_P \cdot K_Y \cdot K_3 \cdot K_{ИА} \cdot (t_{ПЗ} + t_{ПП}),$$

где  $K_P$  - коэффициент учета режима обработки информации = 1,45

$K_K$  - коэффициент учета сложности контроля информации = 1,07

$K_Y$  - коэффициент учета уровня алгоритмического языка программирования = 1,0 (алгоритмический язык высокого уровня);

$K_3$  - коэффициент учета степени использования готовых программных модулей - 0,8

(на 20 - 25% использования готовых прогр. Модулей);

$K_{ИА}$  - коэффициент учета вида используемой информации сложности алгоритма ПП.

Значение коэффициента  $K_{ИА}$  определяют из выражения:

$$K_{ИА} = \frac{K_{II}^I \cdot N_{II} + K_{HC}^I \cdot N_{HC} + K_B^I \cdot N_B}{N_{II} + N_{HC} + N_B},$$

где  $K'_П, K'_{НС}, K'_Б$  - значения коэффициентов учёта сложности алгоритма ПП и вида используемой информации для переменной нормативно-справочной информации и баз данных соответственно

В нашем случае присутствует только один вид используемой информации - переменной; соответствующее группе новизны Б значение коэффициента  $K'_П = 1,62$   $K'_{НС} = 0,97$   $K'_Б = 0,81$

(группа новизны - Б)

$$K'_Б = \frac{1,62 \cdot 2 + 0,97 \cdot 0 + 0,81 \cdot 1}{2 + 0 + 1} = 1,35$$

$t_{РЗ}, t_{РП}$  - норма времени, затраченного на разработку РП на алгоритмическом языке высокого уровня разработчиком постановки задач и разработчиком ПП.

$$t_{РЗ} = 17 \text{ ч.},$$

$$t_{РП} = 80 \text{ ч.}$$

$$t_{РП} = (17 + 80) \cdot 1,07 \cdot 1,45 \cdot 1,0 \cdot 0,8 = 121 \text{ чел.-дн.}$$

### 7.1.5 Расчет трудоемкости выполнения внедрения

Трудоёмкость выполнения стадии внедрения может быть рассчитана по формуле:

$$t_В = (t_{РЗ} + t_{РП}) \cdot K_К \cdot K_Р \cdot K_З,$$

где  $t_{РЗ}, t_{РП}$  - норма времени, затраченного разработчиком постановки задач и разработчиком ПО на выполнение процедур внедрения ПП.

$$t_{РЗ} = 10 \text{ ч.},$$

$$t_{РП} = 16 \text{ ч.}$$

$$t_{РП} = (10 + 16) \cdot 1,07 \cdot 1,45 \cdot 0,8 = 33 \text{ чел.-дн.}$$

### 7.1.6 Расчет суммарной трудоемкости

Таким образом, суммарная трудоемкость разработки программной продукции:

$$t_{\text{рп}} = 47 + 99 + 149 + 121 + 33 = 449 \text{ чел.-дн.}$$

Трудоемкость всех этапов разработки программного продукта представлена ниже (см. Таблица 3).

Таблица 3. Трудоемкость этапов разработки ПП.

Стадия разработки	Трудоемкость чел.-дн.
Техническое задание	47
Эскизный проект	99
Технический проект	149
Рабочий проект	121
Внедрение	33

Для целей контроля и планирования в выполнении работ в данном случае используем ленточный график потому что разработку осуществляет небольшой, стабильный по составу коллектив исполнителей. Для построения ленточного графика необходимо знать срок начала работ, срок окончания работ и количество работников, участвующих на каждом этапе разработки.

Продолжительность выполнения всех работ по этапам разработки программного продукта определяется из выражения

$$T_i = \frac{\tau_i + Q}{n_i},$$

где  $\tau_i$  - трудоемкость  $i$ -й работы чел.-дн;

$Q$  - трудоемкость дополнительных работ, выполняемых исполнителем, чел.-дн.;

$n_i$  - количество исполнителей, выполняющих работу чел

Пусть разработка системы ведется одним специалистом, не привлекаемым к дополнительным работам, то продолжительность разработки программного продукта:

$$T=449/2 = 224,5 \approx 225 \text{ раб.дней}.$$

Для построения календарного плана необходимо перевести рабочие дни в календарные. Для этого длительность каждого этапа нужно разделить на поправочный коэффициент  $K = 0,7$ . В результате получим:

$$T = \frac{225}{0,7} = 322 \text{ кал.дн}$$

$$T_{ТЗ} = \frac{23,5}{0,7} = 34 \text{ кал.дн}$$

$$T_{ЭП} = \frac{49,5}{0,7} = 71 \text{ кал.дн}$$

$$T_{ТП} = \frac{74,5}{0,7} = 107 \text{ кал.дн}$$

$$T_{РП} = \frac{60,5}{0,7} = 87 \text{ кал.дн}$$

$$T_{В} = \frac{16,5}{0,7} = 24 \text{ кал.дн}$$

## 7.2 Определение стоимости разработки ПП

Для определения стоимости работ, необходимо на основании плановых сроков выполнения работ и численности исполнителей, рассчитать общую сумму затрат на разработку программного продукта.

Себестоимость ПП представляет собой стоимостную оценку используемых в процессе производства продукции работ, услуг, природных ресурсов, сырья материалов, топлива, энергии, основных фондов, трудовых ресурсов, а также других затрат на ее производство и реализацию.

Затраты, образующие себестоимость ПП, группируются в соответствии с их экономическим содержанием по следующим элементам:

- Расчёт основной заработной платы;
- Расчёт дополнительной заработной платы;
- Отчисления в социальные фонды;
- Накладные расходы;
- Расчет расходов на амортизацию оборудования.

### **7.2.1 Расчёт основной заработной платы**

В статью включается основная заработная плата всех исполнителей, непосредственно занятых разработкой данного ПП, учётом их должностного оклада и времени участия в разработке. Расчёты

ведутся по формуле 
$$C_{30} = \sum_i \frac{3_i}{d} \cdot t_i,$$

где  $3_i$  - среднемесячный оклад  $i$ -го исполнителя руб;

$d$  - среднее количество рабочих дней в месяце

$t_i$  - трудоемкость работ, выполняемых  $i$ -м исполнителем, чел.-дн.

(определяется из календарного плана-графика).

$$3_i = 25000 \text{ руб};$$

$$d = 24;$$

$$t_i = 203,55;$$

$$C_{30} = 2 \cdot \frac{25000}{24} \cdot 225 = 468750 \text{ руб.}$$

### **7.2.2 Расчёт дополнительной заработной платы**

В статье учитываются все выплаты непосредственным исполнителям за время (установленное законодательством), непроработанное на производстве, в том числе: оплата очередных отпусков, компенсация за неиспользованный отпуск, оплата льготных часов подросткам и др. Расчет ведется по формуле

$$C_{зд} = C_{зо} \cdot a_{\partial},$$

где  $a_{\partial}$  - коэффициент на дополнительную заработную плату.

$$a_{\partial} = 0,2$$

$$C_{зд} = 468750 \cdot 0,2 = 93750 \text{ руб.}$$

### **7.2.3 Отчисления в социальные фонды**

В статье учитываются отчисления в бюджет социального страхования по установленному законодательному тарифу от суммы основной и дополнительной заработной платы

$$C_{cc} = a_{cc} \cdot (C_{зо} + C_{зд});$$

$$a_{cc} = 0,3 + 0,01;$$

где

0,3 – отчисления в социальные фонды;

0,01 – страхование от несчастных случаев.

$$C_{cc} = (468750 + 93750) \cdot 0,31 = 174375 \text{ руб.}$$

### **7.2.4 Накладные расходы**

В статье учитываются другие затраты, входящие в состав себестоимости продукции, но не относящиеся к ранее перечисленным элементам затрат. Сюда можно отнести такие виды затрат: ежемесячная

арендная плата за помещение, оплата электроэнергии, оплата отопления, оплата телефона и других услуг связи и т.д.

$$C_n = C_{зо} \cdot a_n;$$

$\alpha_n = 2$  – коэффициент накладных расходов;

$$C_n = 468750 \cdot 2 = 937500 \text{ руб.}$$

### 7.2.5 Расходы на амортизацию оборудования

Расчет определяет суммарные затраты на приобретение специального оборудования:

$$C_{CO} = \frac{\sum_{i=1}^n C_{Bi} \cdot \alpha_i \cdot t_i}{F_d \cdot 100}$$

где:

$C_{Bi}$  – балансовая цена  $i$ -го вида оборудования, руб;

$\alpha_i$  – норма годовых амортизационных отчислений для оборудования  $i$ -го вида, %;

$t_i$  – время использования  $i$ -го вида оборудования при выполнении данной разработки ч

$F_d$  – действительный годово й фонд рабочего време сотрудника, ч (пятидневная неделя, восьмичасовой рабочий день 2080 ч.).

Таблица 4. Специальное оборудование и ПО, используемое при разработке

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу, руб	Сумма, руб
1	ПЭВМ	шт	2	25000	50000

Затраты на амортизацию оборудования (ПЭВМ):

$$C_{CO} = \frac{50000 \cdot 20 \cdot 225 \cdot 8}{2080 \cdot 100} = 8653,85 \text{ руб.}$$

### 7.3 Определение стоимости разработки системы

Результаты расчетов затрат на разработку программного продукта.

Результаты расчетов приведены ниже (см. Таблица 5).

Таблица 5. Стоимость работ на каждом этапе разработки ПП.

Наименование статьи	Сметная стоимость, руб
Основная заработная плата	468750
Дополнительная заработная плата	93750
Отчисления в социальные фонды	174375
Накладные расходы	937500
Расходы на амортизацию оборудования	8653,85
Итого	1683028,85

Произведенный расчет показал

- Суммарная трудоемкость продукта составляет 449 чел.дней. Поэтому на каждом этапе необходимо привлечение нескольких специалистов.
- Длительность разработки программного продукта составляет 322 календарный день
- Себестоимость программного продукта составляет 1683028,85 руб.

## 8 МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ

### 8.1 Опасные и вредные факторы

#### 8.1.1 Опасные факторы:

- пожарная опасность, обусловленная наличием на рабочем месте мощного источника энергии;
- повышенное значение напряжения в электрической цепи замыкание которой может произойти через тело человека

#### 8.1.2 Вредные факторы:

##### *Физические*

- повышенные уровни электромагнитного излучения;
- повышенный уровень статического электричества
- повышенные уровни запыленности воздуха рабочей зоны
- пониженная или повышенная влажность воздуха рабочей зоны;
- пониженная или повышенная подвижность воздуха рабочей зоны;
- повышенный или пониженный уровень освещенности
- повышенный уровень прямой блескости
- повышенный уровень отраженной блескости
- неравномерность распределения яркости в поле зрения;
- повышенная яркость светового изображения;
- повышенный уровень пульсации светового потока

### *Химические*

- повышенное содержание в воздухе рабочей зоны двуокиси углерода, вредных выхлопов;

### *Психофизиологические*

- напряжение зрения;
- напряжение внимания;
- интеллектуальные нагрузки;
- эмоциональные нагрузки;
- длительные статические нагрузки;
- монотонность труда;
- большой объем информации обрабатываемой в единицу времени;
- нерациональная организация рабочего места.

### *Биологические*

- повышенное содержание в воздухе рабочей зоны микроорганизмов.

## **8.2 Требования к помещениям для работы с ПЭВМ**

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке.

Естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемы́ми устройствами типа жалюзи и занавесей, внешних козырьков и др.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электроннолучевой трубки (ЭЛТ) должна составлять не менее 6 м<sup>2</sup> в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м<sup>2</sup>. При использовании ПЭВМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м<sup>2</sup> на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка - 0,7 - 0,8; для стен - 0,5 - 0,6; для пола - 0,3 - 0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов в высоковольтных трансформаторов технологического оборудования, создающего помехи в работе ПЭВМ.

### **8.2.1 Требования к микроклимату, содержанию вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ**

В производственных помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям указанных выше нормативов.

В производственных помещениях, в которых работа с использованием ПЭВМ является основной, величины показатели микроклимата на рабочих местах должны соответствовать действующим санитарным нормам (СанПиН 2 2 4 548-96 Гигиенические требования к микроклимату производственных помещений). При использовании данной подсистемы характер труда оператора ПЭВМ относится к категории 1а, т.к. работы производятся сидя и не требуют физического напряжения, расход энергии составляет до 120 ккал/ч, следовательно, величины показателей микроклимата на рабочих местах в ПЭО должны соответствовать оптимальным величинам показателей для категории работ 1а (см. Таблица 6).

Таблица 6. Оптимальные величины показателей микроклимата

Период года	Категория работ	Температура воздуха, °С не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – Ia	22-24	40-60	0,1
Теплый	Легкая – Ia	23-25	40-60	0,1

Содержание вредных химических веществ в воздухе производственных помещений, в которых работа с использованием ПЭВМ является основной, не должно превышать предельно допустимых концентраций вредных веществ в воздухе рабочей зоны в соответствии с действующими гигиеническими нормативами (Гигиенические нормативы ГН 2.1.6.13 1338-03 Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест). ПДК среднесуточная: окись углерода – 3,0 мг/м<sup>3</sup>, окислы азота – 0,04 мг/м<sup>3</sup>.

Данный микроклимат может обеспечиваться системой общеобменной приточно-вытяжной вентиляции. В приточной вет нужно предусмотреть использование фильтров очистки воздуха, для удаления из воздуха вредных веществ перед его подачей на рабочие места. Для повышения влажности воздуха в помещениях с ВДТ и ПЭВМ следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной или прокипяченной питьевой водой. Также в помещениях, оборудованных ПЭВМ, должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

### 8.2.2 Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарно-эпидемиологическими нормативами СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы) (см. Таблица 7).

Таблица 7. Допустимые значения уровней шума

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука в дБА
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

Снизить уровень шума в помещениях с ВДТ и ПЭВМ можно использованием подвесных акустических звукопоглощающих панелей с максимальными коэффициентами звукопоглощения в области частот 250-1000 Гц (источниками шума являются операторы ПЭВМ). Дополнительным звукопоглощением служат однотонные занавеси из плотной ткани гармонирующие с окраской стен и подвешенные складку на расстоянии 15 - 20 см от ограждения. Ширина занавеси должна быть в 2 раза больше ширины окна.

Шумящее оборудование (печатающие устройства серверы и т.п.) уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ

В помещениях, в которых работа с ПЭВМ является основной, уровень вибрации на рабочих местах не должен превышать допустимых значений для жилых и общественных зданий в соответствии действующими санитарно-эпидемиологическими нормативами (Санитарные нормы СН 2.2.4/2.1.8.566-96 Производственная вибрация, вибрация в помещениях жилых и общественных зданий) (см Таблица 8)

Таблица 8. Допустимые значения уровней вибрации.

Среднегеометрические частоты октавных полос, Гц	Допустимые значения оси X, Y			
	по виброускорению		по виброскорости	
	м/с <sup>2</sup> × 10 <sup>-3</sup>	дБ	м/с × 10 <sup>-3</sup>	дБ
2	10	80	0,79	84
4	11	81	0,45	79
8	14	83	0,28	75
16	28	89	0,28	75
31,5	56	95	0,28	75
63	110	101	0,28	75
Корректированные значения и их уровни	10	80	0,28	75

Для снижения вибрации в помещениях оборудование, аппараты и приборы, являющиеся источниками вибрации, необходимо устанавливать на амортизирующие прокладки. Также могут быть использованы средства индивидуальной защиты

### **8.2.3 Требования к освещению на рабочих местах, оборудованных ПЭВМ**

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной световым проемам, и естественный свет падал преимущественно слева

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения В

помещениях, в случаях преимущественной работы с документами следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>.

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м<sup>2</sup> и яркость потолка не должна превышать 200 кд/м<sup>2</sup>.

Показатель дискомфорта в помещениях – не более 15.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м<sup>2</sup>, защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между

рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования – 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей.

Применение светильников без рассеивателей и экранируемых решеток не допускается.

При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников расположенных сбоку от рабочих мест параллельно линии зрения пользователя при рядном расположении видеодисплеев и терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1,4.

Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

#### **8.2.4 Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ**

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей, представлены ниже (см. Таблица 9)

Таблица 9. Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах.

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

Основным источником электромагнитного излучения на рабочем месте оператора компьютера является монитор. Величина излучения монитора во-многом зависит от его модели, среди наиболее безопасных необходимо выделить мониторы с маркировкой "Low Radiation", компьютеры с жидко-кристаллическим экраном и самые безопасные – мониторы с установленной защитой по методу замкнутого металлического экрана. Рекомендуется использовать подобные видеотерминалы.

В случае использования видеотерминалов с ЭЛТ для уменьшения воздействия электромагнитных полей на человека устанавливаются поглощающие или отражающие экраны перед видеомонитором. Современный качественный защитный экран позволяет уменьшить уровень электромагнитного излучения и электростатического поля в 10-15 раз и более. Кроме того, существуют специальные очки для защиты глаз от электромагнитного излучения.

### **8.2.5 Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах**

Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах, представлены в Таблица 10.

Таблица 10. Допустимые параметры ВДТ

N	Параметры	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/кв.м
2	Неравномерность яркости рабочего поля	Не более +-20%
3	Контрастность (для монохромного режима)	Не менее 3:1
4	Временная нестабильность изображения (мелькания)	Не должна фиксироваться
5	Пространственная нестабильность изображения (дрожание)	Не более $2 \times 10^{-4}L$ , где L - проектное расстояние наблюдения, мм

Для дисплеев на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией на конкретный т

дисплея, и не менее 60 Гц для дисплеев на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.).

Обеспечить выполнение этих требований можно соответствующим образом выбирая компьютерную технику, а также используя различные защитные экраны.

### ***8.2.6 Общие требования к организации рабочих мест пользователей ПЭВМ***

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ при выполнении творческой работы требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5- 2,0 м

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 - 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

### ***8.2.7 Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей***

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Модульными размерами рабочей поверхности стола для ПЭВМ на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте равной 725 мм

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400 - 550 мм и углам наклона вперед до 15 град, и назад до 5 град.;
- высоту опорной поверхности спинки 300 + 20 мм ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах + 30 градусов;
- регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50- 70 мм
- регулировку подлокотников по высоте над сиденьем в пределах 230 +-30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 1 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю или на

специальной, регулируемой по высоте рабочей поверхностью, отделенной от основной столешницы

### **8.2.8 Электробезопасность рабочего помещения**

Работа с разработанным ПО должна проводиться в офисных помещениях, относящихся, согласно ПУЭ (правилам устройства электроустановок), к классу помещений «без повышенной опасности» т.к. в помещении нет условий повышенной и особой опасности

Питание устройств ПЭВМ осуществляется от однофазной сети переменного тока с частотой 50 Гц и напряжением 220 В

В целях обеспечения необходимой электробезопасности при проведении работ в помещениях с ПЭВМ, необходимо выполнять следующие требования:

- для обеспечения работы операторов ПЭВМ необходимо исключить возможность случайного соприкосновения людей с токонесущими частями оборудования. Это достигается путем изоляции токоведущих частей ЭВМ и приборов и размещения их в недоступных зонах;
- не оставлять ЭВМ и другое оборудование под напряжением без наблюдения;
- не подключать разъемы кабелей ЭВМ при включении напряжения в сети;
- сеть электропитания технических средств должна обеспечивать защитное заземление технических средств, предусмотренное их технической документацией

### **8.2.9 Требования пожарной безопасности**

Помещение, где функционирует разработанная подсистема, относится к категории В — пожароопасные помещения, в котором имеются твердые сгораемые вещества, способные только гореть, но не взрываться при контакте с кислородом воздуха. Наиболее вероятной причиной пожара является неисправность электрооборудования и электросетей. При эксплуатации ЭВМ возможны возникновения следующих аварийных ситуаций: короткие замыкания, перегрузки, повышение переходных сопротивлений в электрических контактах, перенапряжение, возникновение токов утечки. При возникновении аварийных ситуаций происходит резкое выделение тепловой энергии, которая может явиться причиной возникновения пожара. Требования к пожаробезопасности зданий и сооружений определяются согласно СНИ 21.01-97[5].

Для снижения вероятности возникновения пожара необходимо проводить различные профилактические мероприятия:

- § Организационные – правильная эксплуатация электрооборудования, правильное содержание зданий и помещений ;
- § Технические — соблюдение противопожарных правил и норм, норм при проектировании зданий, при устройстве отопления, вентиляции, освещения, правильное размещение оборудования;
- § Мероприятия режимного характера – запрещение курения в неустановленных местах и т.д.;
- § Эксплуатационные — своевременные профилактические осмотры и ремонт неисправного электрооборудования.

Для снижения вероятности возникновения и распространения пожара на ранней стадии необходимо

- § установить пожарную сигнализацию с системой оповещения работников, дежурного по объекту и, желательно, автоматическое оповещение противопожарных служб;
- § иметь в наличии несколько ручных углекислотных огнетушителей (например огнетушитель марки ОУЗ);
- § помещение должно быть планом эвакуации при пожаре и аптечкой первой помощи. В помещении должен быть ответственный за пожарную безопасность.

#### **8.2.10 Требования к организации медицинского обслуживания пользователей ПЭВМ**

Лица, работающие с ПЭВМ более 50% рабочего времени (профессионально связанные с эксплуатацией ПЭВМ) должны проходить обязательные предварительные при поступлении на работу и периодические медицинские осмотры в установленном порядке.

Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену) при условии соблюдения гигиенических требований, установленных СанПиН 2.2.2/2.4.1340-03 **Гигиенические требования к персональным электронно-вычислительным машинам и организации работы**. Трудоустройство беременных женщин следует осуществлять в соответствии с законодательством Российской Федерации.

### **8.2.11 Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля**

Государственный санитарно-эпидемиологический надзор за производством и эксплуатацией ПЭВМ осуществляется в соответствии с **СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.**

Не допускается реализация и эксплуатация на территории Российской Федерации и типов ПЭВМ, не имеющих санитарно-эпидемиологического заключения.

Инструментальный контроль за соблюдением требований **СанПиН 2.2.2/2.4.1340-03** осуществляется в соответствии с действующей нормативной документацией

Производственный контроль за соблюдением санитарных правил осуществляется производителем и поставщиком ПЭВМ, а также предприятиями и организациями, эксплуатирующими ПЭВМ в установленном порядке, в соответствии с действующими санитарными правилами и другими нормативными документами.

### **8.2.12 Расчет системы защитного заземления компьютера**

Данное помещение является помещением с повышенной электроопасностью, т.к. имеется возможность касания токоприемников (ВДТ и ПЭВМ) с проводящими системами, связанными с землей (система централизованного отопления), поэтому заземление необходимо.

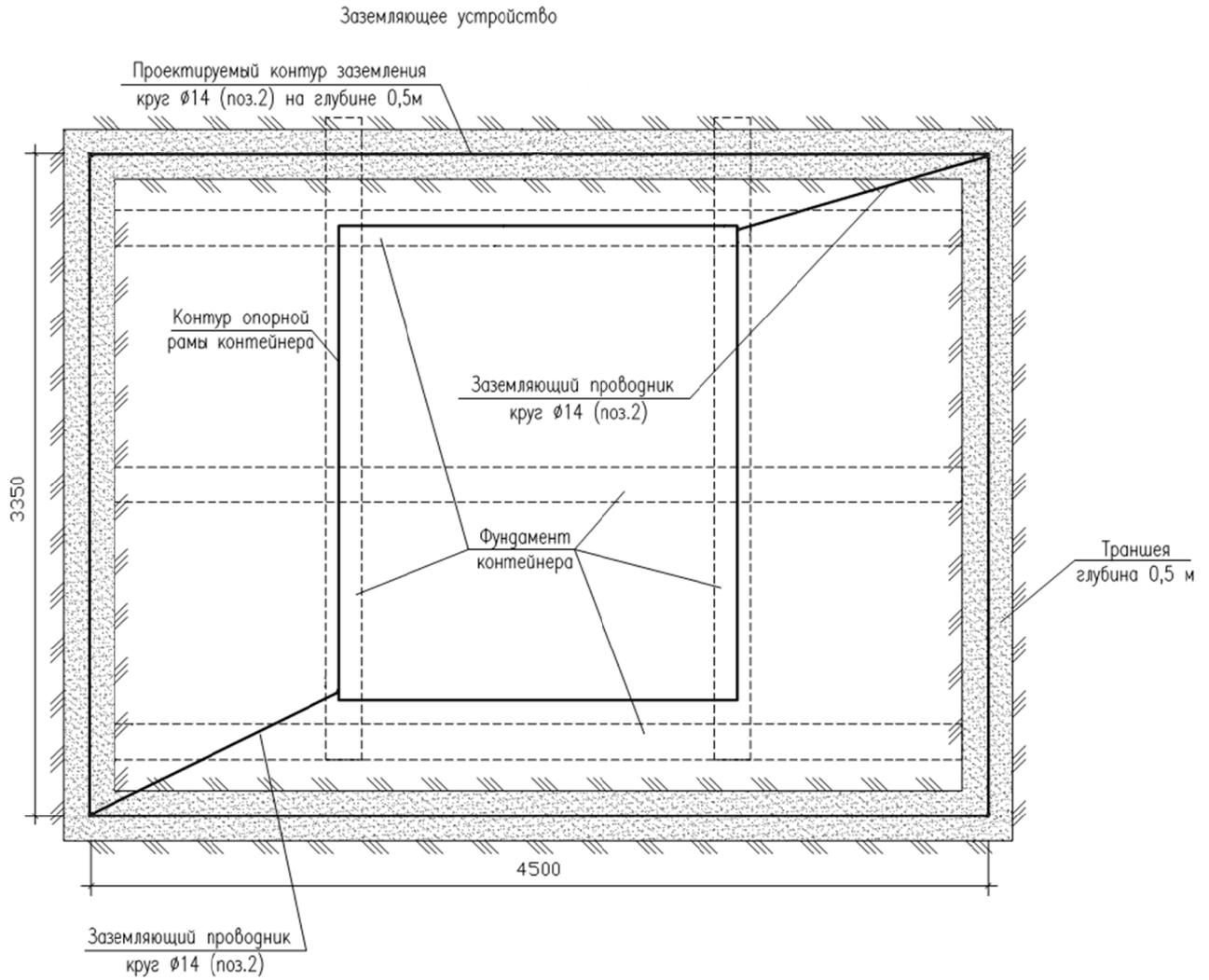


Рис.1. Схема заземляющего устройства (вид сверху)

### Расчет заземления

В качестве заземляющего устройства выбран вариант на базе модульно-стержневого заземлителя.

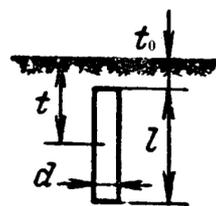


Рис.2. Схема одного заземлителя.

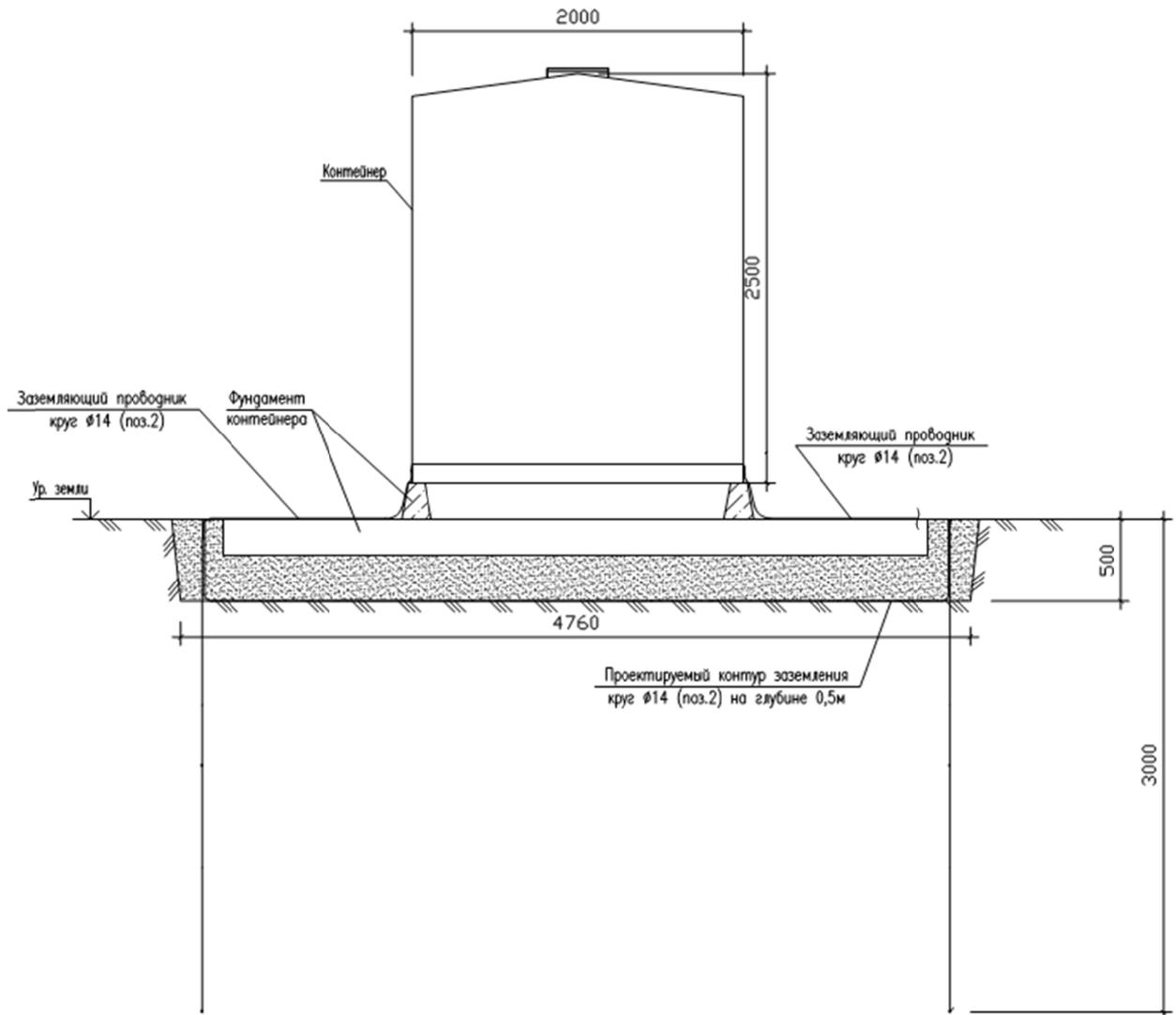


Рис.3. Схема заземляющего устройства (вид сбоку)

Заземляющее устройство представляет собой сварную конструкцию и состоит из следующих элементов:

- заземлитель модульно-стержневого типа (стержень  $L=1,5\text{ м}$ ,  $\text{Ø}16\text{ мм}$ , количество 15 шт.) собирается с помощью втулок переходных,
- контур заземления (круг  $\text{Ø}14\text{ мм}$ ).

Грунт в месте установки контейнера – суглинок с удельным сопротивлением ( $\rho$ )  $60\text{ Ом}\cdot\text{м}$ .

Формула для расчета сопротивления заземления одиночного вертикального заземлителя:

$$R_B = \frac{\rho}{2\pi L} * \left[ \ln\left(\frac{2L}{d}\right) + 0,5 * \ln\left(\frac{4T + L}{4T - L}\right) \right] =$$
$$= \frac{60}{2\pi * 3} * \left[ \ln\left(\frac{2 * 3}{0,016}\right) + 0,5 * \ln\left(\frac{4 * 1,5 + 3}{4 * 1,5 - 3}\right) \right] = 20,6 \text{ Ом}$$

где:

$\rho$  – удельное сопротивление грунта (Ом\*м),

$L$  – длина заземлителя (м),

$d$  – диаметр заземлителя (м),

$T$  – заглубление заземлителя (расстояние от поверхности земли до середины заземлителя) (м),

$\pi$  – математическая константа Пи.

Формула для расчета сопротивления заземления горизонтального кольцевого заземлителя:

$$R_T = \frac{\rho}{2\pi^2 D} * \ln\left(\frac{4\pi D^2}{d * t}\right) = \frac{60}{2\pi^2 * 5} * \ln\left(\frac{4\pi * 5^2}{0,014 * 0,5}\right) = 10,85 \text{ Ом}$$

где:

$\rho$  – удельное сопротивление грунта (Ом\*м),

$D$  – диаметр контура заземлителя (м),

$d$  – диаметр заземлителя (м),

$t$  – глубина залегания заземлителя (расстояние от поверхности земли до заземлителя) (м),

$\pi$  – математическая константа Пи.

Формула для расчета общего сопротивления устройства заземления:

$$R_{\text{общ}} = \frac{R_{\Gamma} R_{\text{В}}}{R_{\text{В}} K_{\text{ип}} + R_{\Gamma} K_{\text{из}} N} = \frac{10,85 * 20,6}{20,6 * 0,74 + 10,85 * 0,65 * 5} = 3,68 \text{ Ом}$$

где:

$K_{\text{из}}$  – коэффициент использования (учет взаимного влияния электродов) для 5 электродов 0,65,

$K_{\text{ип}}$  – коэффициент использования полосы для 5 электродов 0,74,

$N$  – количество электродов в заземлителе.

Устройство заземления удовлетворяет требованиям ПУЭ расчетное заземление на более 4 Ом.

### 8.3 Утилизация ПЭВМ

Извлечение драгоценных металлов из вторичного сырья является частью проблемы использования возвратных ресурсов, которая включает в себя следующие аспекты: нормативно-правовой, организационный, сертификационный, технологический, экологический, экономико-финансовый. Проблема использования вторичного сырья, содержащего драгоценные материалы из компьютеров, периферийного оборудования и иных средств вычислительной техники (СВТ) актуальна в связи с техническим перевооружением отраслей промышленности

К драгоценным металлам относятся: золото, серебро, платина, палладий, родий, иридий, рутений, осмий, а также любые химические соединения и сплавы каждого из этих металлов. Статья 2 п. 4 "Федерального закона о драгоценных металлах и драгоценных камнях" от 26 марта 1998 года №1463 гласит: "Лом и отходы драгоценных металлов подлежат сбору во всех организациях, в которых образуются указанные лом и отходы. Собранные лом и отходы подлежат

обязательному учёту и могут перерабатываться собирающими их организациями для вторичного использования или реализовываться организациям, имеющим лицензии на данный вид деятельности для дальнейшего производства драгоценных металлов".

Порядок учёта хранения транспортировки инвентаризации сбора и сдачи отходов драгоценных металлов из СВТ, деталей и узлов содержащих в своём составе драгоценные металлы для предприятия учреждения и организации (далее - предприятие), независимо от форм собственности, установлен инструкцией Министерства финансов Российской Федерации от 4 августа 1992 года №

### **8.3.1 Разборка изделий**

Последовательность разборки определяется типом изделия СВТ, его конструктивными особенностями и комплектацией.

Как правило, процесс разборки должен выполняться в последовательности, обратной процессу сборки изделия. Основное направление деятельности на этапе «Разборка»

#### **8.3.1.1 Разборка персональных компьютеров (ПЭВМ), рабочих станций и серверов**

Технологии разборки ПЭВМ, рабочих станций, серверов и информационно-вычислительных систем едины поскольку состав их модулей стандартный. Он содержит системный блок и комплект периферийных устройств.

Разборку ПЭВМ и составных модулей целесообразно осуществлять по технологической схеме представленной на рисунке (см. Рис. 4)

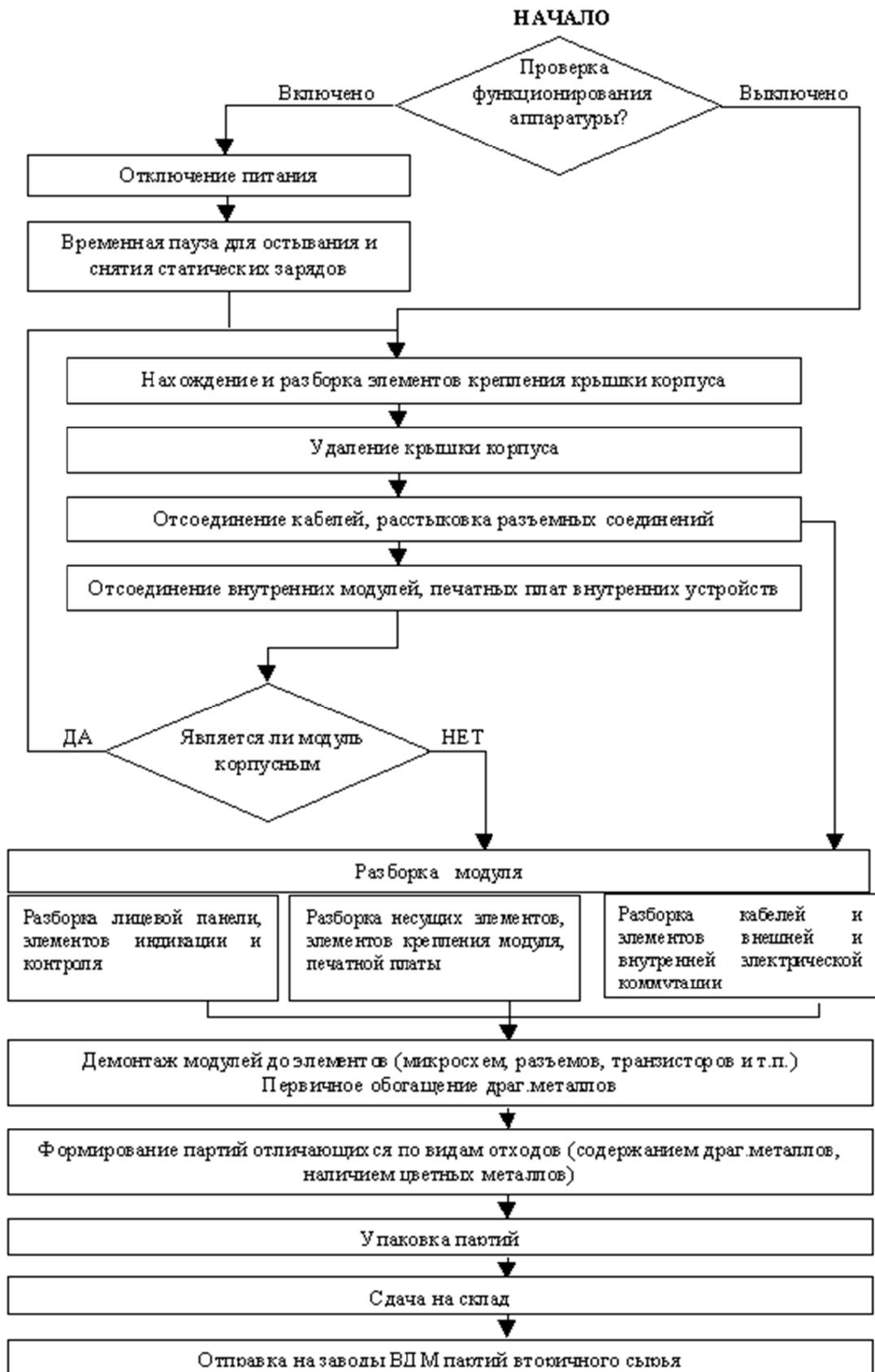


Рис.4. Технологическая схема разборки ПЭВМ.

Порядок разборки системного блока:

- Выключить компьютер и отсоединить шнур питания от розетки и системного блока. Отсоединить переходной шнур питания от системного блока к монитору.
- Отсоединить от компьютера клавиатуру, монитор, манипулятор "мышь", принтер, сканер и иные внешние устройства
- Найти элементы крепления крышки корпуса (винты шурупы, пружинные защелки и т.д.). Освободить крышку от элемента крепления.
- Снять крышку.
- Отсоединить внутренние кабели и плоские шлейфы
- Найти элементы крепления дисководов (НМД НГМД) в отсеке для дисководов (винты, шурупы, саморезные винты, пружинные защелки и др.). Освободить дисководы и извлечь их из дискового отсека.
- Освободить от крепёжных элементов периферийные платы. Извлечь из разъёмов непосредственного контактирования все периферийные платы
- Найти элементы крепления системной платы к корпусу (винты, шурупы). Освободить элементы крепления и извлечь системную плату из корпуса.
- Извлечь модули памяти из разъёмов системной платы
- Найти элемент крепления блока питания к корпусу (винты шурупы, саморезные винты, пружинные защелки и пр.). Освободить элементы крепления и извлечь блок питания.
- Разобрать блок питания и извлечь высоковольтные конденсаторы содержащие тантал.

- Разобрать ПП и модули памяти до компонентов (микросхем, транзисторов, разъемов и т.п.)
- Произвести сортировку компонентов и сформировать партии электронного лома.
- Упаковать партии, составить опись, произвести расчет (анализ) драгметаллов и передать их на склад.
- Провести сортировку цветных и черных металлов, пластмасс, сформировать партии и передать их на склад или на переработку.

При оценке содержания драгоценных металлов в партии электронного лома отечественных ПЭВМ необходимо руководствоваться паспортными данными. При оценке ПЭВМ импортного производства необходимо провести ориентировочные расчеты по отечественным налогам

### 8.3.1.2 Обеспечение комплексности технологии разборки

При разборке изделий СВТ образуются материалы и изделия, которые имеют материальную ценность и подлежат реализации.

Примерный перечень материалов представлен ниже с м. Таблица 11.

Таблица 11. Перечень материалов, подлежащих утилизации.

Вид материалов или изделий	Характеристика
Печатные платы, разъемы и соединители, микросхемы	вторичные драгоценные металлы
Электрические провода и кабели, соединители	вторичная медь и ее сплавы
Свинец и олово из печатных плат	вторичные припойные пасты (олово и свинец)

Вид материалов или изделий	Характеристика
Танталовые конденсаторы К-53-1	вторичный тантал
Некоторые корпуса компьютеров, дисковод и т.д.	алюминиевые сплавы
Корпуса стоек, ячеек, шкафов, компьютеров	сталь
Крепежные изделия	болты, гайки и винты
Вентиляторы и электромоторы	по паспорту СВТ
Пластиковая "фракция"	стеклотекстолит, пластмасса разъемов и соединителей
Экраны компьютеров	стеклофаза, содержащая Pb, Cd, CdS, редкоземельные металлы

Таким образом, можно сделать вывод о целесообразности извлечения вторичных черных и цветных металлов, пластмасс, стекла, крепежных изделий, вентиляторов и электромоторов.

### 8.3.1.3 Извлечение вторичных черных металлов

Отечественная практика показывает, что на 1 г извлекаемого золота приходится около 1 кг лома черных металлов. В связи с высокой стоимостью транспортно-погрузочных работ рекомендуется производить отгрузку предприятиям-покупателям партий лома черных металлов весом не менее 10 тонн. Блоки, панели, съемные кожухи рамы, каркасы шкафов и стоек стационарных ЭВМ, изготовленные из стального нормализованного профиля или листа, подвергаются сортировке, набираются в партии и реализуются.

Предпочтительно заключение договоров при условии, когда предприятие-покупатель своим транспортом вывозит вторичные металлы с территории предприятия-продавца.

Крепёжные изделия заготовки стального профиля листов вентиляторы, электропускатели, кнопки, электрический кабель направляются на реализацию непосредственно в торговую сеть.

Опыт показывает, что денежные средства от реализации этих изделий не превышают 0,6 % от общей суммы

#### **8.3.1.4 Извлечение вторичных цветных металлов**

В процессе разборки изделий СВТ образуется лом (содержащий медь) классификация которого должна проводиться по ГОСТ 1639.

В соответствии с ГОСТ 1639 медные шины целесообразно относить к классу А, группам I и II; латунь - к группам IV-VIII; бронзу - к группам XI-XII; отходы кабеля и проводов ПП следует относить к классу Г, группа XIII.

Все виды ломов необходимо сортировать по классам и группам, формировать в партии и реализовывать.

В процессе разборки изделий СВТ алюминий и его сплавы обычно содержатся в типовых конструкциях изделий. По ГОСТ 1639 их следует относить к классам АЗ и Б5.

Все виды отходов необходимо сортировать, формировать в партии и реализовывать.

Свинцово-оловянные припои содержатся в печатных платах и их количество превышает количество золота в десятки раз.

Припои регенерируются при переработке печатных плат.

При разборке СВТ танталовые конденсаторы необходимо складировать отдельно для последующей реализации

Переработка изделий из пластмасс

Пластмассы следует сортировать по видам.

Переработке подлежат термопласты: поливинилхлорид, полиэтилен, полистирол и т.п.

Стекла люминесцентных экранов электронно-лучевых трубок следует использовать в производстве керамики и в качестве сырья при производстве новых люминесцентных трубок.

### **8.3.2 Реализация партий**

На рисунке ниже (см. Рис.5) представлены основные направления деятельности на этапе "Реализация партий".

Основные действия на этапе "Реализация партий" представляют собой последовательность действий, создающих основу для успешного выполнения процедур завершающего этапа утилизации СВТ.

#### **8.3.2.1 Классификация отходов**

В настоящее время в России и за рубежом не существует единой классификации вторичного сырья, содержащего драгоценные металлы. Поэтому, возможно разделение вторичного сырья по следующим признакам.

По содержанию драгоценных металлов:

- бедное (менее 1 % золота, 5 % серебра и 1 % металлов платиновой группы);
- богатое (более 1 % золота, 5 % серебра и 1 % металлов платиновой группы)

По составу материала основания:

- на металлической основе
- на органической (пластиковой) основе
- на керамической основе

- на комбинированной основе

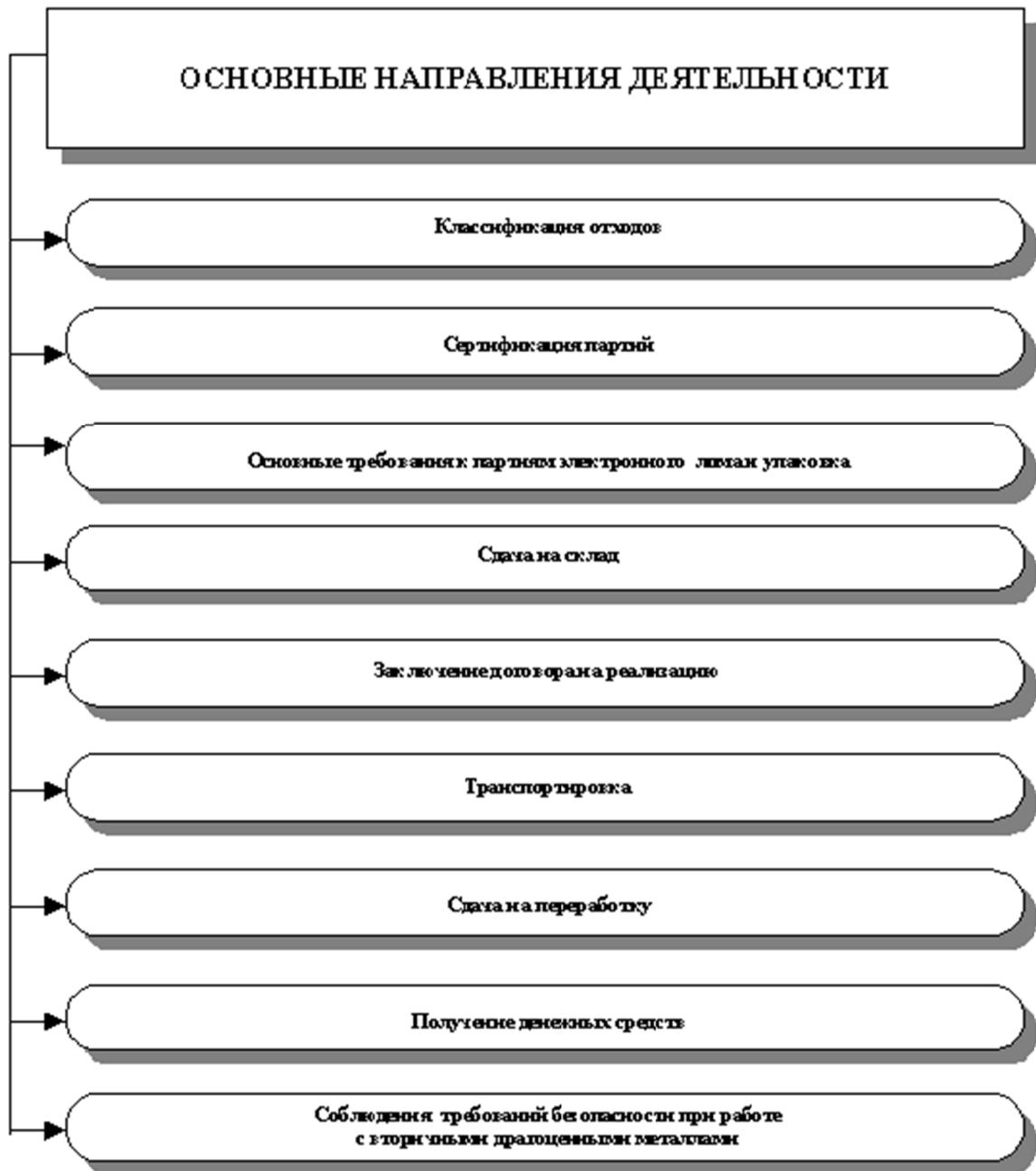


Рис.5. Основные направления деятельности на этапе «Реализации партий»

По физическим признакам:

- твердые компактные отходы
- сыпучие (порошки);
- жидкие.

Возможна классификация вторичного сырья в зависимости от сферы производства в:

- ювелирной промышленности
- химической промышленности
- электронной, электрохимической, оборонной радиопромышленности (радиолампы, разъёмы, контакты, контактные устройства, платы на органической основе, микросхемы, радиодетали, кабели и провода, лента, высечка, вырубка, аккумуляторы, элементы питания, прочие отходы);
- бытовых отходах (лом бытовой радиоэлектронной аппаратуры, бытовой стеклянный и фарфоровый, ювелирных украшений и т.д.)

Отходы классифицируются по элементному составу.

При этом электронный лом отличается особым многообразием состава. Например, современный компьютерный лом содержит несколько десятков видов деталей, содержащих благородные, цветные и чёрные металлы.

#### 8.3.2.1.1 Классификация сырья вторичных драгоценных металлов

Согласно методике опробования электронного лома и отходов, содержащих драгоценные металлы, разработанной ИАСЦ ГНЦ "ГИРЕДМЕТ", сырьё вторичных драгоценных металлов следует рассортировывать на классы, приведённые в Таблица 12

Таблица 12. Классификация электронного лома и отходов, содержащих драгоценные металлы (по видам)

Номер класса	Вид сырья
1.	Микросхемы, транзисторы, диоды россыпью
2.	Конденсаторы россыпью
3.	Ножки разъёмов позолоченные и посеребренные
4.	Контакты разделанные
5.	Платы, содержащие элементы классов 1 и 2
6.	Разъемы позолоченными и посеребренными ножками
7.	Реле, переключатели, тумблеры
8.	Радиолампы
9.	Платы, содержащие элементы классов 1, 2, 6, 7 и 8
10.	Крупногабаритные детали (волноводы и т.п.)с покрытием из
11.	драгметаллами
12.	Элементы питания, аккумуляторы, ампульные батареи Сыпучие материалы (шихта катализаторов, зола фотоматериалов, шлак фиксирующий и т.п.)

### 8.3.2.2 Сертификация партий

В целях обеспечения строгого учёта сохранности сокращения потерь и эффективности использования драгоценных металлов, содержащихся в электронном ломе и отходах, а также для обеспечения единства и требуемой точности измерений при опробовании и проведении анализов химического состава, необходимо руководствоваться нормативными документами утверждёнными Комитетом Российской Федерации по драгоценным металлам драгоценным камням, Комитетом Российской Федерации по стандартизации, метрологии и сертификации: "Порядком выдачи

сертификатов химического состава на партии электронного лома и отходов, содержащих драгоценные металлы", "Временной методикой опробования электронного лома и отходов, содержащих драгоценные металлы".

Указанные документы определяют порядок проведения работ и требования по опробованию и сертификации химического состава партий электронного лома и отходов

#### 8.3.2.2.1 Сертификация химического состава электронного лома и отходов, содержащих драгоценные металлы, включает следующие работы:

- оформление и представление заявки в соответствующий орган по сертификации;
- создание комиссии по опробованию;
- опробование, оформление документов по результатам опробования, передача пробы на анализ;
- собственно анализ пробы и оформление количественного химического анализа;
- оформление сертификата.

Выполнение измерений химического состава проб (анализ) электронного лома и отходов, содержащих драгоценные металлы, следует производить методами количественного химического анализа по аттестованным методикам, Приложение 2.

Анализ проб осуществляется аналитическими лабораториями, которые аккредитованы Комитетом Российской Федерации по стандартизации, метрологии и сертификации, а также рекомендованными организациями и органами по сертификации, Приложение 3.

По результатам анализа аккредитованная лаборатория оформляет протокол измерений химического состава пробы электронного лома или отходов по установленной форме

По результатам процедур опробования и анализа химического состава электронного лома или отходов, орган по сертификации оформляет и выдает заявителю Сертификат химического состава на содержание драгоценных металлов установленной формы

Сертификат химического состава и комплекс документов по опробованию сертифицируемой партии электронного лома включается в состав сопроводительной документации при передаче партии вторичного сырья от сдатчика заготовителю или переработчику, а также при вывозе за границу для переработки.

При возникновении разногласий, в процессе передачи сертифицированных партий электронного лома и отходов от сдатчика заготовителю или переработчику, производится арбитраж. В этом случае партия не может быть передана переработчику до получения заключения арбитражной лаборатории аккредитованной Комитетом Российской Федерации по стандартизации метрологии и сертификации

### **8.3.2.3 Основные требования к партиям электронного лома и упаковка**

Утвержденные технические требования к партиям электронного лома, в частности его упаковке, до настоящего времени отсутствуют.

В связи с этим рекомендуется придерживаться следующих правил.

Не допускается смешивание различных классов лома.

В ломе и отходах драгоценных металлов не допускаются посторонние предметы, не относящиеся к естественной засоренности

Лом и отходы драгоценных металлов должны храниться в специально предназначенной для этого таре в пакетах из полиэтиленовой плёнки по ГОСТ 10354, в фанерных ящиках по ГОСТ 9396, в металлических ящиках с замками собственного изготовления или мешках из мешочной бумаги по ГОСТ 2226

Ящики внутри должны быть выложены упаковочной бумагой по ГОСТ 515 или каким-либо плёночным материалом. Пакеты и мешки предназначенные для хранения и отправки лома и отходов, должны изготавливаться с вывернутыми внутрь двумя боковыми швами без нижнего шва.

Мешки или пакеты, уложенные в деревянные или металлические ящики, допускается клеивать какой-либо клеевой лентой по ГОСТ 18351.

Опечатывание отходов в таре производится после прошивания мешков и пакетов шпагатом по ГОСТ 17308 или заваривания открытых краев полиэтиленовых пакетов с последующим складыванием краёв в "гармошкой" и прошивание её двумя концами шпагата

Партия лома и отходов должна состоять из одного или нескольких мест в каждом из которых вложена одна или несколько позиций различающихся по составу компонентов, конфигураций, габаритным размерам и другими признаками, не меняющих принципиально сущности последующего опробования на перерабатывающих заводах.

Взвешивание и упаковка отходов производится с участием материально ответственных лиц подразделений предприятия с датчика

После контрольного взвешивания каждое место должно быть опломбировано или опечатано сургучной печатью датчика

В сопроводительных документах указывается описание пломбы или печати.

#### **8.3.2.4 Сдача на склад**

Передача партии электронного лома из производственного подразделения на склад драгоценных металлов предприятия осуществляется на основе приёмно-передаточного акта акта изъятия узлов и изделий техники и других нормативных документов

#### **8.3.2.5 Заключение договора на реализацию**

Между продавцом и покупателем заключается типовой договор предметом которого является полученная партия электронного лома.

В договоре указываются обязанности сторон, порядок подготовки, отправки лома и отходов драгоценных металлов, порядок приёмки опробования лома и отходов, порядок расчётов и ответственность сторон.

#### **8.3.2.6 Транспортировка**

Транспортирование лома и отходов драгоценных металлов с содержанием золота и металлов платиновой группы более 5 % должно производиться через специальную связь в соответствии с инструкцией Министерства связи Российской Федерации о перевозке ценностей

Лом и отходы с содержанием золота и металлов платиновой группы менее 5 %, а также отходы серебра отправляются на перерабатывающие заводы почтовыми посылками, багажом по железной дороге или другим видом транспорта с оценочной стоимостью отгружаемого груза.

Сдача на переработку.

Порядок сдачи партии на переработку определяется условиями договора. Типичные условия договора для завода ВДМ следующие.

Вскрытие посылок (мест) производится на заводе приё мной комиссией, которая взвешивает и сверяет фактическое наличие сырья и его качественный состав по каждому виду сырья с данными продавца. По результатам приёмки сырья покупатель высылает продавцу приёмный акт в течение 5 дней от даты поступления сырья.

При доставке сырья транспортом продавца, приёмный акт и количество мест выдаётся на руки уполномоченному представителю продавца в день сдачи сырья.

Представителю продавца необходимо иметь копию описи сдаваемого сырья.

В случае нарушения упаковки или целостности печати материально ответственный работник покупателя в акте на приём отходов отмечает все нарушения.

При расхождении фактически установленных данных при приёмке сырья с данными, значившимися в сопроводительных документах продавца, а также при отсутствии сопроводительных документов, окончательными результатами приёмки является масса брутто нетто сырья, установленные приёмной комиссией покупателя.

Взвешивание, опробование, пробоотбор и химический анализ проб каждой партии производится в соответствии с нормативно-технической документацией и по технологии применяемой покупателем.

Однотипные позиции партии подлежат объединению и опробованию по единой технологической схеме.

По результатам опробования сырья на содержание драгоценных металлов, которое производится в течение 60 дней со дня его поступления, покупатель высылает продавцу паспорт с указанием количества драгоценного металла с учетом процента выхода чистого металла в готовую продукцию.

Паспорт подписывается руководителем предприятия-покупателя, главным бухгалтером или их заместителями и скрепляется печатью покупателя.

Порядок получения денежных средств зависит от условия договора. Типичные условия завода ВДМ следующие.

Стоимость поставленного продавцом сырья определяется паспортом покупателя, составленным на основании прейскуранта завода, по мировым ценам на продукцию, получаемую из сырья на день, предшествующий вписке паспорта и пересчитанным в рубли по курсу установленному Центральным Банком Российской Федерации на день выписки паспорта.

Денежные средства перечисляются на расчетный счет продавца течение трех банковских дней со дня получения подтверждающего документа о поступлении денежных средств на расчетный счет покупателя.

Продавец производит оплату с каждой поставленной партии за опробование.

Все платежи по договору должны производиться в валюте Российской Федерации в безналичной форме.

### **8.3.2.7 Соблюдение требований безопасности при работе с вторичными драгоценными металлами**

Выполнение работ по разборке списанных СВТ предполагает соблюдение общих правил, изложенных в инструкциях по охране труда для слесаря механо-сборочных работ и лиц, работающих с ручным электроинструментом.

Специальные требования техники безопасности при работе с вторичными драгоценными металлами следующие.

Не допускается сбор, заготовка и переработка радиоактивного лома и отходов драгоценных металлов.

Степень воздействия на организм человека вредных веществ выделяющихся в процессе заготовки и переработки лома и отходов драгоценных металлов, класс опасности и их предельно-допустимая концентрация (ПДК) в воздухе рабочей зоны установлены по ГОСТ 12.1.005 и ГОСТ 12.1.007, Таблица 13.

Таблица 13. Степень воздействия драгоценных металлов на организм человека

Наименование металла	Характер действия на организм человека	Пути проникновения	Класс опасности	ПДК вредных веществ в воздухе рабочей зоны, мг/м <sup>3</sup>
Серебро и его соединения	Отходы могут оказывать раздражающее действие на слизистую оболочку носа и дыхательных путей.	Органы дыхания	11	0,5-1
Золото, платина и его соединения	При длительном контакте могут вызывать аллергические дерматиты и экземы.	Открытые участки кожи	-	-
Рутений		Органы дыхания	11	-
Родий	Оказывает раздражающее действие на слизистую оболочку носа и дыхательных путей. У рабочих, занятых очисткой родия иногда развивается сверхчувствительность	-	-	-

Контроль за содержанием вредных веществ в воздухе рабочей зоны проводится в соответствии с требованиями ГОСТ 12.1.005 и ГОСТ 12.1.007. Анализ проб воздуха проводится в соответствии с нормативно-технической документацией, утвержденной Минздравом Российской Федерации.

Производственные помещения в местах образования вредных веществ и пыли должны быть оборудованы вентиляцией согласно ГОСТ 12.4.021 с обеспечением санитарно-гигиенических требований к воздуху рабочей зоны

Для снятия статического электричества пылеприёмники воздухопроводы вентиляционных установок должны иметь заземление, выполненное и обозначенное в соответствии с ГОСТ 12.2.007.0, ГОСТ 12.2.007.14 и ГОСТ 21130.

Для предотвращения попадания пыли, твердых веществ на слизистую оболочку глаз необходимо пользоваться защитными очками типа ПО-2, ПО-3 согласно ГОСТ 12.4.013.

При работе с пылящими отходами необходимо пользоваться фильтрующими респираторами РУ-60 и РУ-60му по ГОСТ 17269 и респираторами "Лепесток" по ГОСТ 12.4.028.

При этом респираторы должны периодически подвергаться промывке.

Средства индивидуальной защиты работающих с ломом и отходами драгоценных металлов и сплавов должны соответствовать типовым отраслевым нормам бесплатной выдачи рабочим и служащим металлургических производств. Спецодежда должна соответствовать ГОСТ 29057 и ГОСТ 29058.

Помещения в местах выгрузки и загрузки лома и отходов, оказывающих вредное воздействие на организм человека должны быть оборудованы местными отсосами согласно ГОСТ 12.4.021.

Производственные помещения должны соответствовать требованиям "Санитарных норм проектирования промышленных предприятий СН245-71".

Метеорологические условия производственных помещений должны соответствовать санитарным нормам проектирования промышленных предприятий по ГОСТ 12 1 005

Требования безопасности при погрузочно-разгрузочных работах лома и отходов драгоценных металлов и сплавов должны соответствовать ГОСТ 12.3.009.

Требования по обеспечению взрывобезопасности.

Предприятия и организации, заготавливающие и перерабатывающие лом и отходы драгоценных металлов сплавов, должны проверять весь лом и отходы драгоценных металлов на взрывобезопасность.

Из лома необходимо отобрать и удалить взрывоопасные предметы, материалы, в том числе электронно-вакуумные трубки дисплеев.

Замкнутые сосуды, резервуары и другие полые предметы (баллоны, цилиндры, сосуды, электровакуумные изделия и т.д.) разгерметизируются и освобождаются от содержимого (газов или жидкостей )

Разгерметизацию должны производить рабочие, прошедшие специальное обучение, которые перед началом работы инструктируются в установленном порядке о мерах предосторожности.

## 9 РЕЗУЛЬТАТЫ

В ходе выполнения данного дипломного проекта были получены результаты:

- 1) Состояние модели записывается в БД.
- 2) Пользователь имеет несколько версий РДО для работы на выбор без подключения к БД, с подключением к БД ресурсов, с подключением к БД трассировки, с подключением к БД ресурсов и БД трассировки.
- 3) Потери быстродействия составляют от 35 % до 570 %
- 4) Ускорить работу алгоритма поиска релевантного ресурса с помощью делегирования задачи СУБД не удалось.

Адекватность работы результата 1) проверена на одной из сторонних систем, которая строит отчеты на основе данных из БД – Navicat.

### 9.1 Процесс работы со сторонней системой

Сторонние системы требуют схожих действий для получения требуемого результата (при условии, что они обеспечивают возможность его получения). Выбран Navicat.

Navicat представляет собой платное программное обеспечение с 30 дневным бесплатным периодом. Оно предоставляет пользователю множество инструментов для работы по созданию и управлению БД, а также для разработки модели базы данных с использованием реверсивного инжиниринга. Также система предоставляет возможность строить отчеты на основе данных из БД.

Этапы работы по получению графика на основе данных из БД:

- 1) Получить дистрибутив Navicat. Он доступен на сайте

- 2) Установить ПО на ПК, удовлетворяющий ее требованиям
- 3) Запустить ПО, установить подключение к СУБД.
- 4) Выбрать БД, по данным которой будет построен график
- 5) Запустить построитель отчетов.
- 6) Сформировать условия для отбора данных для построения отчета.
- 7) Проверить полученные данные, убедиться в правильности заданных условий отбора
- 8) Добавить в отчет график.
- 9) Выбрать столбцы для осей графика
- 10) Привести график к желаемому виду.

## ЗАКЛЮЧЕНИЕ

В рамках данного дипломного проекта были получены следующие результаты.

Проведено предпроектное исследование системы имитационного моделирования РДО и сформулированы предпосылки создания в системе модулей для записи данных в БД.

На этапе концептуального проектирования системы была разработана модель данных с помощью нотации IDEF1X. Также укрупненно было рассмотрено внутреннее устройство РДО с целью определения компонент, которые должны подвергнуться изменению.

На этапе технического проектирования с помощью диаграммы классов разработана архитектура новой системы. С помощью диаграммы последовательностей продемонстрирован пример схемы взаимодействия компонент РДО при записи состояния модели в базу данных.

На этапе рабочего проектирования написан программный код для реализации спроектированных ранее алгоритмов работы и архитектуры компонентов `rdo_parser`, `rdo_runtime` и `rdo_sql_general_db` системы РДО. Проведены отладка и тестирование новой системы в ходе которых исправлялись найденные ошибки. Анализ быстродействия приведен в Исследовательской части. Работоспособность системы была проверена с помощью сторонней системы `Navicat`.

Таким образом, поставленная цель дипломного проекта достигнута в полном объеме.

## СПИСОК ЛИТЕРАТУРЫ

1. Материалы по системе имитационного моделирования РДО. [В Интернете] <http://rdo.rk9.bmstu.ru/forum/viewforum.php?f=4>.
2. **Емельянов В.В., Ясиновский С.И.** *Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО.* Москва : Анвик, 1998.
3. Информация о функциональных возможностях СУБД PostgreSQL. [В Интернете] <http://www.postgresql.org/about/>
4. ГОСТ 19.201-78. *Единая система программной документации. Техническое задание.*
5. **Сажин Ю.Б., Самохин С.В.** *Выполнение организационно - экономической части дипломного проекта по разработке и использованию программного продукта : Методическое пособие.* – М .: Изд-во МГТУ им. Н.Э . Баумана, 2006. – 60 с .: ил.
6. СНиП 21.01-97. *Нормы пожарной безопасности «Определение категорий.»*
7. ГОСТ Р 12.1.019-2009. *Система стандартов безопасности труда. Электробезопасность.*
8. *Сборник типовых расчетов по курсу «Охрана труда» для студентов.* МВТУ : б.н., 1984 г.