

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
Кафедра прикладной информатики и информационных технологий

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ
ПОДСИСТЕМЫ УЧЁТА ПРОИЗВОДСТВЕННОГО ПРОЦЕССА
ФАСАДНОЙ КОМПАНИИ ООО «ВИД»**

Выпускная квалификационная работа студента

очной формы обучения
направления подготовки 09.03.03. Прикладная информатика
4 курса группы 07001205
Нефёдова Сергея Николаевича

Научный руководитель
к.г.н. Петина М.А.

БЕЛГОРОД 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ	6
1.1 Техничко-экономическая характеристика предметной области.....	6
1.1.1 Характеристика предприятия	6
1.1.2 Организационная структура управления предприятием и её характеристика	7
1.1.3 Краткая характеристика подразделения или видов его деятельности .	10
1.2 Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	12
1.3 Постановка задачи	18
1.3.1 Цель и назначение автоматизированного варианта решения задачи...	18
1.3.2 Общая характеристика организации решения задачи на ЭВМ	18
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования	21
2 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ.....	23
2.1 Обоснование проектных решений по техническому обеспечению.....	24
2.2 Обоснование проектных решений по информационному обеспечению. ..	28
2.3 Обоснование проектных решений по программному обеспечению	28
2.4 Обоснование проектных решений по технологическому обеспечению	34
2.5 Обоснование выбора программных средств	35
3 ПРОЕКТНАЯ ЧАСТЬ.....	37
3.1 Информационное обеспечение задачи.....	37
3.1.1 Информационная модель и ее описание	37
3.1.2 Используемые классификаторы и системы кодирования	42
3.1.3 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией	44
3.1.4 Характеристика базы данных	47
3.1.4.1 Характеристика модели БД.....	47
3.1.4.2 Характеристика даталогической модели БД	49

3.1.5 Характеристика результатной информации	50
3.1.5.1 Характеристика таблиц с результатной информацией.....	50
3.1.5.1 Характеристика результатных документов	53
3.2 Программное обеспечение задачи.....	54
3.3 Технологическое обеспечение задачи.....	55
3.3.1 Организация технологии сбора, передачи, обработки и выдачи информации.....	55
3.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации	57
3.4 Описание контрольного примера реализации проекта	58
3.5 Расчет экономической эффективности	67
3.6 SWOT-анализ разработки.....	74
ЗАКЛЮЧЕНИЕ	76
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	77
ПРИЛОЖЕНИЯ.....	82

ВВЕДЕНИЕ

Эффективность функционирования предприятия или организации любой отрасли и сферы деятельности напрямую зависит от скорости, точности и своевременности обмена данными как внутри этого предприятия между его составляющими частями (отделами, подсистемами и т.д.), так и вне его, то есть взаимодействие и обмен данными этой организации с другими (конкурирующими, предприятиями-партнерами и т.д.). И чем больше, масштабнее предприятие, тем серьезнее перед его управляющими встает проблема организации и контроля потоков огромного количества информации предприятия.

Для качественного решения таких проблем на предприятиях используются автоматизированные системы учета. Автоматизированная система учета – это человеко-машинная система, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления в различных сферах человеческой деятельности[29].

Таким образом, тема выпускной квалификационной работы: «Разработка автоматизированной информационной подсистемы учёта производственного процесса фасадной компании ООО «Вид»» является актуальной.

Внедрение АСУ позволяет систематизировать обмен данными, регламентировать состав и формы представления данных, а также структуру информационных потоков в системе, значительно повысить точность и четкость их ведения, гарантировать их сохранность, предоставлять полную взаимоувязанную информацию по всем аспектам проектного управления. Все это приводит к слаженной работе сотрудников организации и во много раз увеличивает эффективность функционирования предприятия в целом.

Объектом исследования является – компания ООО «Вид».

Предмет исследования – автоматизация учёта рабочего процесса производственного цеха фасадной компании «Вид».

Целью выпускной квалификационной работы является разработка

автоматизированной информационной подсистемы учёта производственного процесса.

К числу задач, решаемых в выпускной квалификационной работе можно отнести:

- исследовать деятельность производственного цеха организации ООО «Вид»;
- построить функциональную модель производственного цеха организации ООО «Вид»;
- проанализировать существующие автоматизированные информационные подсистемы учёта производственного процесса;
- описать технологии проектирования автоматизированной информационной подсистемы учёта производственного процесса;
- дать техническое, технологическое, информационное и программное обоснование проектных решений;
- дать обоснование экономической эффективности проекта.

Работа состоит из трёх частей, в которых рассматриваются все аспекты разработки подсистемы учета производственного процесса.

В первой части будет проведен анализ предметной области. Будут разработаны модели деятельности организации («КАК ЕСТЬ»). На основании данных моделей будут сформулированы задачи проекта.

Вторая часть работы посвящена выбору средств и методов для решения поставленных задач.

Третья часть посвящена проектированию, разработке экономической эффективности проекта.

Выпускная квалификационная работа представлена на 81 странице, содержит 41 рисунок, 12 таблиц и 2 приложения.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия

Общество с ограниченной ответственностью "Вид" зарегистрировано 30 мая 2012 года. Компания ООО "Вид" расположена по адресу: Российская Федерация, Белгородская область, г. Белгород, пр-т Славы, 131.

Общество имеет в собственности обособленное имущество, учитываемое на его самостоятельном балансе, может от своего имени приобретать и осуществлять имущественные и личные неимущественные права, нести обязанности, быть истцом и ответчиком в суде.

Общество в своей деятельности руководствуется принятым Уставом и законодательством Российской Федерации.

Основной вид деятельности общества – производство строительных металлических конструкций и изделий.

Фасадная компания «Вид» – это команда специалистов, осуществляющая полный комплекс услуг по оформлению фасадов:

- Навесные вентилируемые фасады;
 - 1) из композитных материалов
 - 2) из керамогранита
 - 3) из фиброцементных плит и т.д.
- входные группы, козырьки, навесы;
- наружная реклама;
- архитектурная подсветка зданий;
- дополнительные услуги компании
 - 1) проектирование фасадов
 - 2) создание дизайн-проекта
 - 3) комплектация фасадных систем и облицовочных материалов

- инженерное сопровождение
- изготовление и монтаж;
- фрезерно-гравировальные работы;
- раскрой листовых материалов.

Организация постоянно выступает подрядчиком на таких объектах как торгово-развлекательные комплексы «МегаГРИНН», «Сити-Молл Белгородский», торговый центр «Европа», компания «Айсберг» и менее крупные организации.

1.1.2 Организационная структура управления предприятием и её характеристика

В организации ООО «Вид» используется линейная структура управления. Это одна из самых простых организаций управленческой структуры. Во главе любого отдела, связанного как с управленческой, так и с производственной деятельностью, стоит руководитель, обладающий всеми предусмотренными полномочиями. Руководитель осуществляет единоличный контроль за работниками, находящимися у него в подчинении, а также централизует все управленческие функции. Принятые руководителем решения по цепочке передаются соответствующим отделам. Руководитель отдела подчиняется руководителю вышестоящего в рамках организационной иерархии.

Организационная структура ООО «Вид» представлена на рисунке 1.1:



Рисунок 1.1 - Организационная структура ООО «Вид»

Директор организации руководит в соответствии с действующим законодательством производственно-хозяйственной и финансово-экономической деятельностью предприятия, неся всю полноту ответственности за последствия принимаемых решений, сохранность и эффективное использование имущества предприятия, а также финансово-хозяйственные результаты его деятельности.

Директор, избранный сроком на 5 лет, без доверенности действует от имени общества, представляет его интересы и совершает сделки, выдает доверенности, издает приказы о назначении на должности работников общества, об их переводе и увольнении, а также осуществляет иные полномочия, не отнесенные законом к компетенции собрания.

Порядок деятельности директора и принятие им решений устанавливают внутренние документы Общества и договор, заключенный между Обществом и лицом, осуществляющим функции директора.

Директору организации подчиняется начальник отдела продаж.

Основной задачей начальника отдела продаж является осуществлять организацию сбыта продукции. В своей деятельности начальник отдела продаж руководствуется Уставом организации, приказами и распоряжениями руководства организации, правилами внутреннего трудового распорядка, а

также законодательными и нормативными актами, касающимися выполняемой им работы.

Начальник отдела продаж обязан:

- осуществлять организацию сбыта продукции;
- анализировать и систематизировать клиентскую базу организации;
- разрабатывать стратегии для отдела продаж;
- составлять итоговые отчеты по продажам, вести рабочую и отчетную документацию;
- проводить переговоры с клиентами и компаниями и контролировать их задолженности;
- заключать договора на реализацию продукции;
- разрешать конфликтные ситуации, возникающими между менеджерами и клиентами организации;
- координировать и организовывать работу отдела продаж;
- устанавливать отпускные цены на товары, разрабатывать ценовую политику;
- осуществлять подбор сотрудников отдела;
- обеспечивать безопасные условия труда для подчиненных сотрудников.

Начальнику отдела продаж подчиняется менеджер отдела продаж.

Начальник производственного отдела подчиняется непосредственно директору организации.

Начальник производственного отдела руководствуется в своей деятельности законодательными актами, Уставом организации, правилами внутреннего трудового распорядка, другими нормативными актами организации, приказами и распоряжениями директора.

Начальник производственного отдела выполняет такие обязанности, как:

- осуществлять организацию производства продукции;
- составлять итоговые отчеты по производству, вести рабочую и отчетную документацию;

- координировать и организовывать работу производственного отдела;
- осуществлять подбор сотрудников отдела;
- обеспечивать безопасные условия труда для подчиненных сотрудников.

Начальнику производственного отдела подчиняется сотрудник производственного отдела.

Начальник монтажного отдела руководствуется в своей деятельности законодательными актами, Уставом организации, правилами внутреннего трудового распорядка, другими нормативными актами организации, приказами и распоряжениями директора.

Начальник монтажного отдела выполняет такие обязанности, как:

- осуществлять организацию монтажа продукции;
- составлять итоговые отчеты по монтажу, вести рабочую и отчетную документацию;
- координировать и организовывать работу монтажного отдела;
- осуществлять подбор сотрудников отдела;
- обеспечивать безопасные условия труда для подчиненных сотрудников.

Начальнику монтажного отдела подчиняется сотрудник монтажного отдела.

1.1.3 Краткая характеристика подразделения или видов его деятельности

Производственный отдел является ключевым подразделением данной организации ООО «Вид». В фасадной компании этот отдел представлен производственным цехом, в котором расположено два станка с числовым программным управлением, один станок для вальцевания деталей, один

плоттер для печатной продукции и три организованных рабочих места, с которых производится управление оборудованием.

Основной производственного отдела в данной организации является изготовление деталей из различных материалов для последующего монтажа соответствующим отделом на объекте заказчика.

В производственном отделе основной исполнительской должностью является работник отдела, который непосредственно подчиняется начальнику отдела (см. рисунок 1.1).

Работник производственного отдела в своей деятельности руководствуется нормативными документами, Уставом организации, правилами внутреннего трудового распорядка организации, приказами и распоряжениями директора и начальника производственного отдела.

В обязанности работника производственного отдела входит:

- разработка плана работ;
 - 1) составление чертежей
 - 2) распределение чертежей на партии
- фрезеровка деталей из;
 - 1) металлов
 - 2) дерева
 - 3) ПВХ
 - 4) ДСП
 - 5) алюмокompозитных материалов
- вальцевание готовых деталей;
- печать продукции;
- Передача готовых деталей монтажному отделу.

Именно для работника производственного отдела и будет разрабатываться автоматизированная система учёта производственного процесса.

1.2 Обоснование необходимости и цели использования вычислительной техники для решения задачи

В настоящее время коммерческим предприятием приходится функционировать в сложных финансово-экономических условиях. В связи с этим организациям необходимо максимизировать продуктивность своей деятельности. Главной целью процесса увеличения результативности является максимально эффективная организация деятельности предприятия на всех уровнях. Для решения этой задачи задействуются знания нескольких дисциплин, таких как логистика и управление, бухгалтерский учет, менеджмент и т.д.

Данный эффект достигается, в том числе, и автоматизацией бизнес процессов компании – неотъемлемой частью всех современных организаций, необходимой для их успешного функционирования. На данный момент, большинство предприятий используют автоматизированный документооборот, например, бухгалтерский учёт, включающий в себя работу с большим количеством различных документов. Исходя из этого, становится ясно, что автоматизация деятельности становится ключевой задачей, которую необходимо решить руководству для того, чтобы предприятие работало максимально эффективно.

В данном проекте рассматривается автоматизация учёта производственного процесса производственного цеха фасадной компании «Вид». Данная задача стала актуальна благодаря увеличению количества и объёмов заказов продукции клиентами, в связи с чем возникла необходимость вести чёткую, регламентируемую документацию учёта работ в производственном цеху предприятия, таких как:

- мониторинг прогресса выполнения определённого заказа;
- автоматически составляемый отчёт о выполненной работе;
- очередь заказов на выполнение;
- список выполненных заказов.

Использование вычислительной техники дает удобство, быстроту поиска информации и оформления документов, поднимает эффективность работы на принципиально новый уровень, предоставляя ранее недоступные функции. Прежде всего, это касается подсистемы аналитической информации. Ранее, при использовании системы бумажного учета получить информацию о совершенных сделках, проанализировать наиболее востребованные услуги можно было только подняв всю документацию и договора, то теперь это станет гораздо более быстрым и доступным, что позволит лучше оценивать востребованность разных услуг на рынке.

Для определения цели и назначения автоматизированного варианта решения задачи автоматизации, необходимо провести анализ процессов, поддающихся автоматизации в рамках решения задачи. Для этого необходимо рассмотреть контекстную диаграмму деятельности производственного цеха фасадной компании «Вид». Контекстная модель представлена на рисунке 1.2.

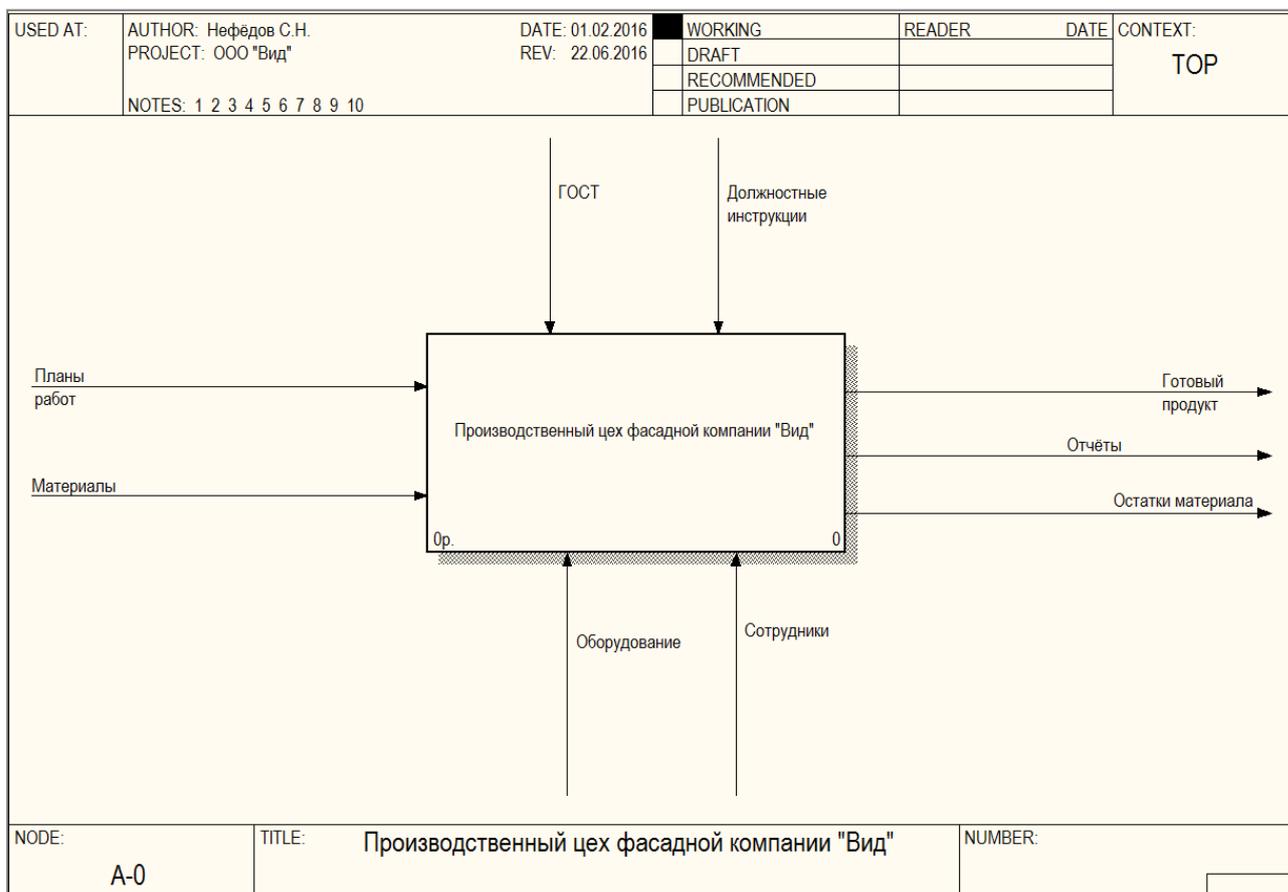


Рисунок 1.2 - Контекстная диаграмма (IDEF0)

Для построения модели бизнес-процессов организации была выбрана методология IDEF0. Методология предполагает наличие следующих видов связей:

- вход;
- выход;
- механизм;
- управление.

В нашем случае входом являются материальные ресурсы, которые перерабатывает данная система. К ним относятся:

- планы работ;
- материалы.

Данные ресурсы поступают в организацию от заказчика, после чего обрабатываются и на выходе системы преобразуются в следующие выходные связи:

- готовый продукт;
- отчёты;
- остатки материала.

Деятельность организации можно разделить на четыре части:

- проектирование;
- изготовление продукта;
- упаковка продукта;
- формирование отчёта.

Первая часть – проектирование включает следующие процессы:

- анализ задачи;
- выполнение чертежа;
- формирование проекта.

Декомпозиция деятельности производственного цеха представлена на рисунке 1.3.

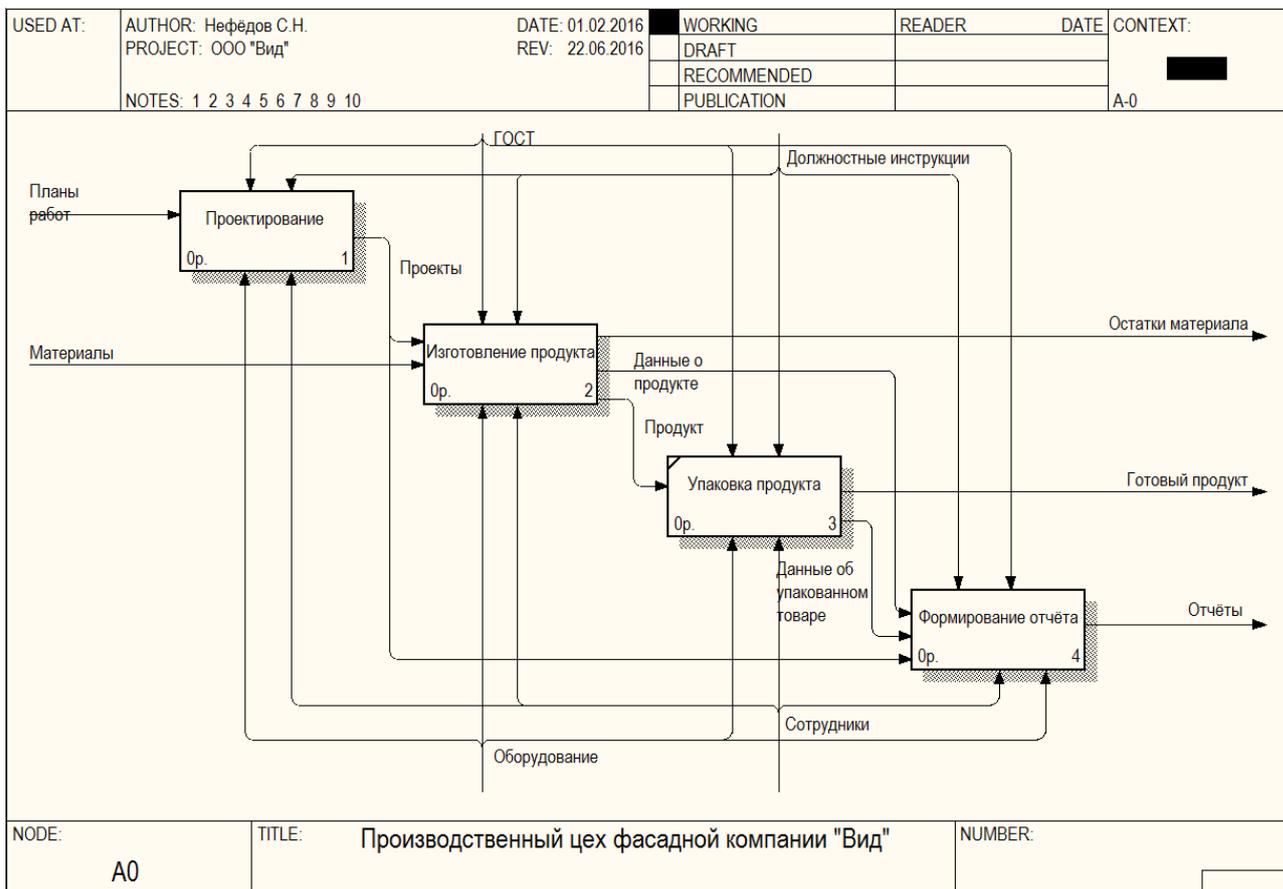


Рисунок 1.3 – Декомпозиция контекстной диаграммы (IDEF0)

Входами являются документы «Планы работ». На основании этих документов формируется задание на изготовление. Результатом работы данного процесса является проект конечного продукта, содержащий необходимые данные для выполнения заказа. Более подробно подпроцессы данного процесса можно рассмотреть на рисунке 1.4:

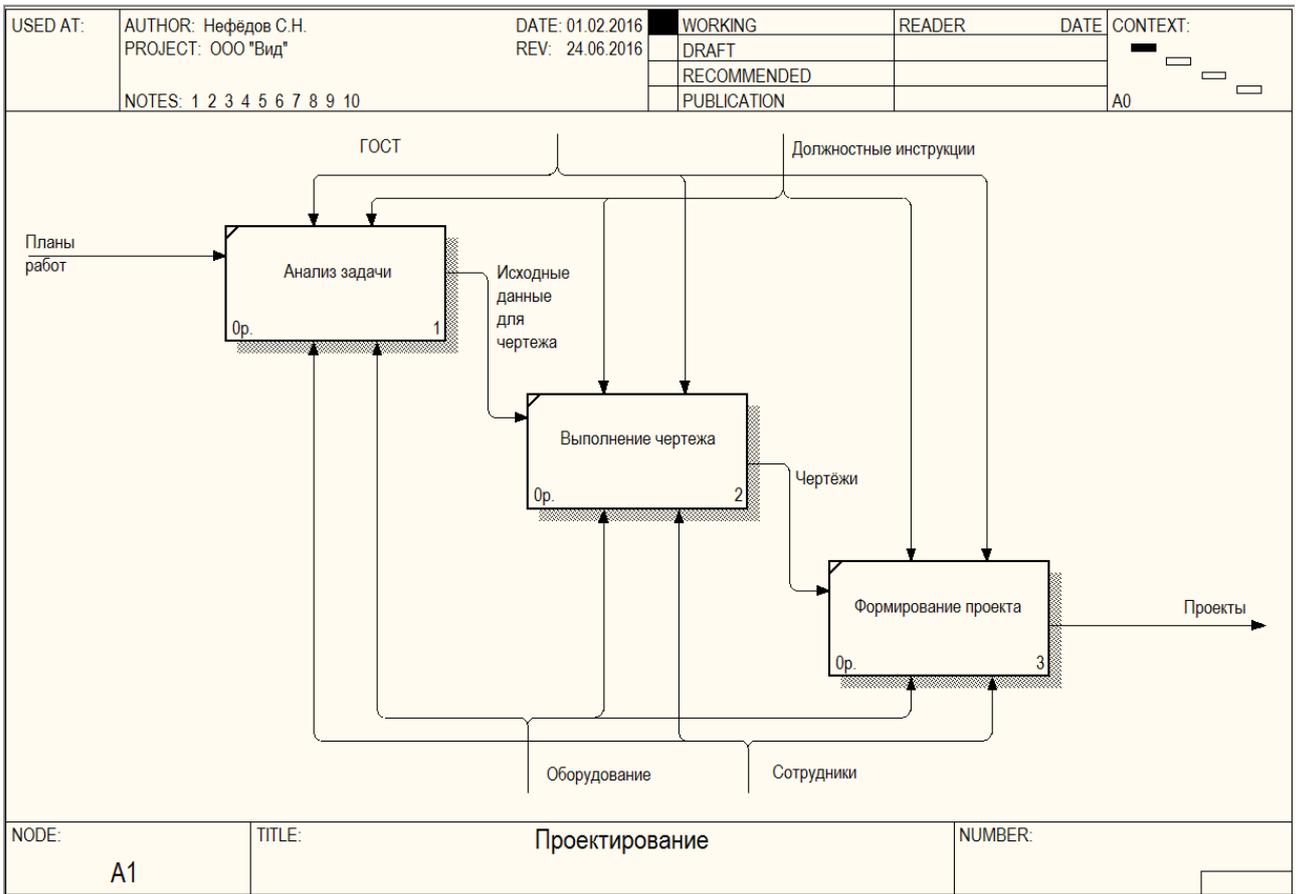


Рисунок 1.4 – Диаграмма декомпозиции блока «Проектирование продукта»
(IDEF0)

Следующим процессом, который необходимо рассмотреть, является блок «Изготовление продукта», описывающий процесс производства данного предприятия. Декомпозиция бизнес процесса показана на рисунке 1.5.

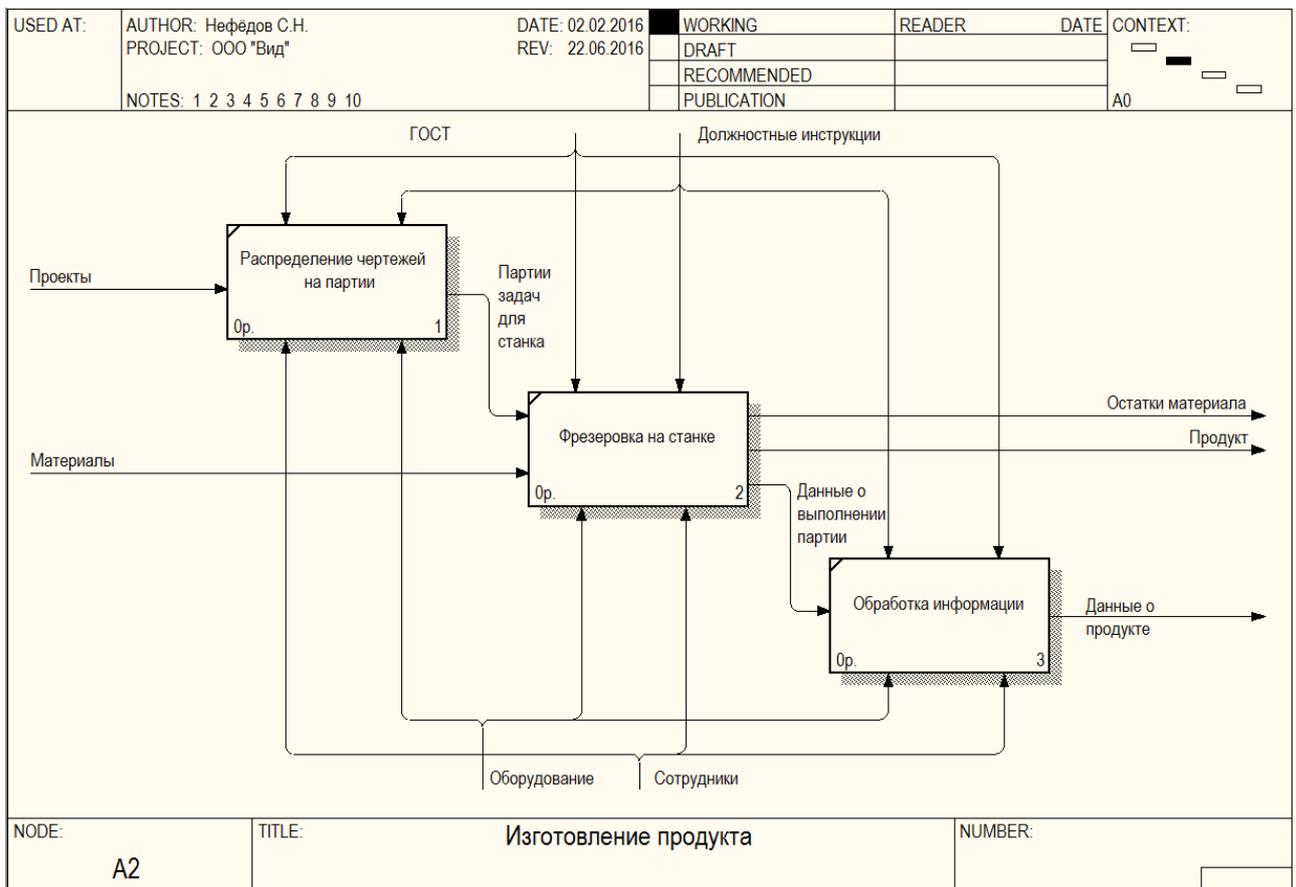


Рисунок 1.5 – Диаграмма декомпозиции блока «Изготовление продукта»
(IDEF0)

Входной информацией для данного процесса является такой ресурс как материал, из которого производится изделие, а так же проект. Результатом выполнения этого процесса является продукт, отправляемый на упаковку, неиспользованный материал и данные о завершённом заказе, составляемый исходя из данных о выполненных партиях, заполняющихся вручную сотрудниками цеха, без возможности отслеживания прогресса выполнения заказа кем-либо, помимо рабочих в производственном цеху.

1.3 Постановка задачи

1.3.1 Цель и назначение автоматизированного варианта решения задачи

На основании проведённого анализа были выделены слабые стороны существующей системы учёта производственного процесса:

- некорректное распределение обязанностей сотрудника по заполнению отчётной документации;
- недостаточная информативность существующей документации;
- отсутствие автоматизации документооборота учёта производственного процесса фасадной компании «Вид».

Основываясь на перечисленных проблемах была сформирована задача автоматизации учёта производственного процесса путём проектирования, разработки и внедрения автоматизированной информационной подсистемы учёта производственного процесса фасадной компании «Вид».

1.3.2 Общая характеристика организации решения задачи на ЭВМ

Для реализации проектируемой системы был выбран вариант «клиент-серверной» архитектуры, так как она отвечает всем необходимым условиям проектируемой автоматизированной системы. Клиент-серверная архитектура позволит распределить нагрузку на сеть, объединить рабочие места пользователей в локальную сеть, все пользователи будут использовать централизованное хранилище данных, что позволит избежать устаревания и неточности информации. Клиент-серверная архитектура будет реализована на базе SQL-сервера. SQL-сервер будет выступать в роли координатора клиентских приложений, который сможет обрабатывать сложные запросы, в том числе вызов хранимых процедур и триггеров, что позволит снизить объем

передаваемого трафика по сети и блокировку таблиц базы данных во время выборки информации.

Клиентская часть будет реализована в виде набора модулей и исполнимых файлов, реализованных на языке высокого уровня. Пользователь, используя клиентскую часть автоматизированной системы, сможет производить различные манипуляции с данными, находясь сколь угодно далеко от центрального серверного компьютера, при этом возможен вариант установки данной системы на одну пользовательскую машину, что будет отвечать потребностям пользователей, не имеющих в наличии локальной сети.

Входные данные будут вводиться при помощи экранных форм системы, которые предназначены для облегчения пользователю ввода данных. В справочниках базы данных хранится условно-постоянная информация. Далее представлены данные, выступающие в качестве условно-постоянной информации:

- список товаров;
- список видов товаров;
- список производителей товаров;
- список стран-производителей товаров;
- список характеристик товаров;
- список мест хранения.
- список клиентов;
- список поставщиков;

К источникам поступления оперативной и условно-постоянной информации относятся:

- сведения о поставщиках;
- документы от поставщика;
- информация о товарах;
- данные о клиентах;
- заказы клиентов.

Условно-постоянная информация поступает с разной периодичностью.

В основном информация поступает ежедневно, однако она может поступать и раз в месяц, и раз в квартал, и раз в год.

Первичная информация также вводится в базу данных. Первичные данные могут быть получены из различных источников устных или письменных, а также из печатных документов.

Для решения задачи автоматизации учёта производственного процесса можно выделить следующие этапы:

- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о заказах;
- создание форм для ввода, редактирования и удаления информации о материалах;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о клиентах;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о цветах материала;
- создание различных отчетов.

Решение задачи автоматизации учёта производственного процесса реализовано при помощи диалогового режима, при котором существует возможность пользователя взаимодействовать с вычислительной системой в процессе работы.

Пользовательский интерфейс представлен в виде меню. Главное меню выводится на экран при открытии программы и представляет собой список команд, которые позволяют управлять работой программы.

Результатными документами являются различные отчеты. Для каждого отчета есть возможность вывода на экран и печати на принтере.

1.4 Анализ существующих разработок и обоснование выбора технологии проектирования

На сегодняшний день существует множество автоматизированных систем для ведения учёта производственного процесса на предприятии, однако не все могут подходить под специфику данного предприятия. Рассмотрим наиболее известные и распространённые программные продукты.

Система «1С:Управление производственным предприятием 8» - это комплексное прикладное решение, охватывающее основные контуры управления и учёта на производственном предприятии. Прикладное решение создаёт единое информационное пространство для отображения финансово-хозяйственной деятельности предприятия, охватывая основные бизнес-процессы. В то же время чётко разграничивается доступ к хранимым сведениям, а также возможности тех или иных действий в зависимости от статуса работников.

Перечислим ключевые особенности системы:

- независимость данных управленческого, бухгалтерского и налогового учёта;
- сопоставимость данных управленческого, бухгалтерского и налогового учёта;
- совпадение суммовых и количественных оценок активов и обязательств по данным управленческого, бухгалтерского и налогового учёта, при отсутствии объективных причин их расхождения [37].

Галактика ERP - ядро комплекса бизнес-решений Галактика BusinessSuite, главное назначение которого – выполнение в едином информационном пространстве типовых и специализированных задач управления предприятием, холдингом, группой компаний в условиях современной экономики. Система адресована средним и крупным предприятиям и обладает широкой функциональностью для информационной

поддержки всего спектра задач стратегического планирования и оперативного управления

Преимущества Галактика ERP:

- универсальность использования на самых разнообразных предприятиях;
- система позволяет планировать затраты на производство и хозяйственные нужды, вести их фактический учет;
- система Галактика ERP дает возможность контролировать рентабельность как предприятия в целом, так и по отдельным заказам, группам товаров, направлениям деятельности;
- поддержка принятия управленческих решений;
- автоматизировать процессов управления производством как с изменением бизнес-процессов на предприятии, так и при сохранении существующих бизнес-процессов;
- масштабируемость [41].

Описанные выше системы учёта производственного процесса обладают широкими функциональными возможностями, однако они имеют высокую стоимость и не полностью учитывают специфику данной организации. Поэтому было принято разработать собственную систему.

Разрабатываемая система будет направлена на автоматизацию учета производственного процесса на предприятии ООО«Вид».

Отличия разрабатываемой автоматизированной системы от готового программного продукта:

- удобство и простота использования;
- интуитивно понятный интерфейс;
- не требует сложной установки и настройки;
- система полностью учитывает специфику организации и максимально полно отвечает ее нуждам;
- возможность модернизации системы и ее развитие;
- быстрое освоение разработанной системы персоналом.

2 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение автоматизированного рабочего места - комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы[39].

Комплекс технических средств автоматизированного рабочего места составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно - справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная. Централизованное техническое обеспечение базируется на использовании в

информационной системе больших ЭВМ и вычислительных центров. Децентрализация технических средств, предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Ниже приведено рекомендуемой техническое обеспечение решения поставленной задачи.

Таблица 2.1 - Рекомендуемое техническое обеспечение.

Наименование	Рекомендуемые характеристики
Процессор	Intel(R) Atom(TM) 1.66GHz +
Оперативная память	2048Mb
Видеокарта	NVidiaVGA совместимая с объемом оперативной памяти не менее 256 mb
Жесткий диск	Объем не менее 2Gb
Монитор	VGA-совместимый дисплей с разрешением не менее 1024x600
Манипуляторы	клавиатура, мышь
Дополнительная орг. техника	Многофункциональное устройство

Перечисленный в таблице 2.1 перечень технических средств является средним по стоимости и позволяет вести оперативную и эффективную обработку информации.

2.2 Обоснование проектных решений по информационному обеспечению

Под информационным обеспечением понимается совокупность информации, циркулирующей в данной информационной системы, также взаимосвязь потоков данных, определение входящей и исходящей информации.

Информационное обеспечение подразделяется на немашинное и

внутримашинное. Под немашинным информационным обеспечением понимается совокупность информации, которая воспринимается непосредственно человеком, без применения средств вычислительной техники (классификаторы технико-экономической информации, документы, методические инструктивные материалы) [39].

Внутримашинное информационное обеспечение – это совокупность данных, формирующих информационную базу на электронном носителе (экранные формы для ввода первичных данных в ЭВМ или вывода результатной информации, структуры информационной базы: входных, выходных файлов, базы данных) [39].

Разрабатываемая автоматизированная система будет использоваться в организации, в которой, на данный момент, существует сформированный документооборот, где все входные и выходные документы имеют установленные формы.

Работа с информацией будет происходить посредством экранных форм, то есть будут спроектированы экранные формы для хранения переменной, условно-постоянной информации (данные о товарах, поставщиках, клиентов) и реквизитов документов, хранящихся в бумажном виде.

Форма – это главный визуальный компонент, который представляет собой видимое окно Windows и являющийся неотъемлемой частью практически любого приложения. На форму помещаются визуальные компоненты, которые образуют интерфейсную часть приложения, и системные (невизуальные) компоненты, предназначенные для осуществления доступа к системным ресурсам. Для работы с формой предназначен компонент Form класса TForm. С создания формы начинается конструирование приложения. Таким образом, форма является компонентом, который служит контейнером для всех других компонентов. Посредством форм происходит диалог между пользователем и системой[1].

Для разработки экранных форм приложения использовались разные инструментальные средства, которые позволяют представить информацию в

удобном виде, ускорить ввод данных и вывести результатную информацию, например, таблицы, кнопки, выпадающие списки и так далее.

Результатную информацию можно будет вывести двумя способами:

- в форме отчета на печать;
- на экран.

Далее будет обоснован способ организации информационной базы.

Информационная база – это совокупность данных, организованная определенным образом, которые хранятся в виде файлов в памяти вычислительной системы, при помощи которых удовлетворяются информационные потребности решаемых задач и управленческих процессов.

Основными способами организации баз данных является создание централизованных и распределенных баз данных. Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к базе. Такой способ использования баз данных часто применяют в локальных сетях. Распределенные базы данных представляют определенным образом связанные между собой базы данных, рассредоточенные на какой-либо территории (локально или регионально), обеспечивающие свободный обмен информацией и поиск данных в них. Для работы с распределенными данными создаются системы управления распределенными базами данных (СУРБД).

По способу доступа к базе данных они подразделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом. Системы централизованных баз данных с сетевым доступом предполагают архитектуры: файл-сервер и клиент-сервер[14].

Файл-сервер. В данном случае сервер, на котором лежит база данных, является исключительно хранилищем и не обладает каким-либо функционалом, позволяющим производить математические и/или логические вычисления. Файл-серверная архитектура применима исключительно при работе с небольшими объемами данных, поскольку если объем данных будет велик, то

это грозит существенными задержками работы сети и непосредственно пользовательских компьютеров, которые, как известно, изначально не предполагают больших нагрузок, и имеют довольно таки невысокую производительность. В результате компьютеры пользователей будут банально виснуть, общая производительность труда упадет.

Клиент-сервер. При использовании данной архитектуры, на самом сервере, содержащем базу данных, функционирует некоторое программное обеспечение, которое называется сервером баз данных. Таким образом, архитектура «клиент-сервер» адаптирована для работы с большими объемами данных - сеть нагружается меньше, требования к пользовательским компьютерам, с точки зрения производительности, минимизируются. Однако возрастают требования к серверу, содержащему базу данных, поскольку теперь он один тянет нагрузку всех пользователей[12].

Наиболее широкое распространение получила реляционная модель данных. При такой организации вся информация представлена в виде таблиц и отношений. Таблицы являются совокупностью записей. Между отношениями (таблицами) существуют связи типа один-ко-многим, многие-ко-многим. Каждое отношение имеет ключ - это поле записи (атрибут) однозначно идентифицирующее ее. Данное свойство реляционной модели данных исключает дублирование информации, ускоряет поиск и доступ к конкретным данным.

Принятый в реляционной модели подход к структурированию и целостности данных позволяет удобно организовать и упорядочить процесс проектирования и реализации сложных баз данных, а реляционные операции обладают мощными возможностями управления данными и их обработки [38].

В результате будет разработана реляционная модель централизованной базы данных с использованием СУБД FireBird с архитектурой «клиент-сервер».

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – это набор программных средств, обеспечивающих функционирование информационной подсистемы [1].

Программное обеспечение, можно условно разделить на три категории:

- системное программное обеспечение (программы общего пользования) необходимо для функционирования компьютера, работы с файлами, защиты программ и данных, а также для разработки прикладного программного обеспечения;
- прикладное программное обеспечение - совокупность программ для решения общих универсальных задач;
- инструментальное программное обеспечение (системы программирования), обеспечивающее разработку новых программ для компьютера на языке программирования[8].

Операционная система предоставляет пользователю интерфейс по взаимодействию с ресурсами машины, а также обеспечивает оптимальное распределение ресурсов между различными процессами.

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами[8].

Операционные системы классифицируют по следующим основным признакам.

По количеству одновременно работающих пользователей операционные системы делятся:

- однопользовательские;
- многопользовательские.

Многопользовательские операционные системы, в отличие от однопользовательских, поддерживают одновременную работу на ЭВМ

нескольких пользователей за различными терминалами.

По числу задач, одновременно выполняемых под управлением системы, операционные системы подразделяются на:

- однозадачные ОС;
- многозадачные ОС.

Понятие многозадачности означает поддержку параллельного выполнения нескольких программ, существующих в рамках одной вычислительной системы, в один момент времени. Однозадачные ОС поддерживают режим выполнения только одной программы в отдельный момент времени.

По количеству поддерживаемых процессоров операционные системы бывают:

- однопроцессорные;
- многопроцессорные.

В соответствии с этим признаком многопроцессорные ОС, в отличие от однопроцессорных, поддерживают режим распределения ресурсов нескольких процессоров для решения той или иной задачи.

По типу интерфейса (способа взаимодействия с пользователем):

- текстовый интерфейс командной строки;
- графический интерфейс.

По типу доступа пользователя к ЭВМ операционные системы делятся:

- системы с пакетной обработкой, в которых из программ, подлежащих выполнению, формируется пакет (набор) заданий, вводимых в ЭВМ и выполняемых в порядке очередности с возможным учетом приоритетности;

- системы с разделением времени, обеспечивающие одновременный диалоговый (интерактивный) режим доступа к ЭВМ нескольких пользователей на разных терминалах, которым по очереди выделяются ресурсы машины, что координируется операционной системой в соответствии с заданной дисциплиной обслуживания;

– системы реального времени, обеспечивающие определенное гарантированное время ответа машины на запрос пользователя с управлением им какими-либо внешними по отношению к ЭВМ событиями, процессами или объектами [8].

В настоящее время имеется множество операционных систем, и перед пользователем встает задача выбора операционной системы, которая будет отличаться от других теми или иным критериями. Любой операционной системе присущи свои плюсы и минусы. При выборе операционной системы, пользователь должен понимать, насколько та или иная операционная система обеспечит решение необходимых ему задач.

На всех компьютерах ООО «Вид» установлена многопользовательская, многозадачная 32-разрядная операционная система с графическим пользовательским интерфейсом Windows 7 Ultimate. Эта версия раскрывает максимальный потенциал Windows 7 и является наиболее универсальным решением. В неё включены все возможности всех других версий, включая запуск многих приложений Windows XP в режиме Windows XP, шифрование данных с помощью функций BitLocker и BitLockerToGo. Дополнительную гибкость версии придаёт возможность работы на 35 языках.

Функциональные возможности Windows 7 Ultimate:

- значительное повышение производительности: быстрый запуск и завершение работы, быстрое переключение режимов и сеансов;
- модули управления ресурсами системы, оптимизация распределения нагрузки, защита от непредвиденных сбоев и зависаний, незаметная для пользователя автоматическая диагностика системы;
- интеллектуальная технология ускорения поиска необходимых файлов и программ и удобная система просмотра результатов поиска;
- система родительского контроля для предотвращения несанкционированного использования и блокировки нежелательных данных;
- гибкая система настройки Windows 7 Ultimate;

- обеспечение совместимости устройств, предоставление всех необходимых драйверов через центр обновлений Windows;
- обеспечение полной совместимости приложений в рамках системы, встроенные средства совместимости;
- возможность легко переписать данные и параметры со старого ПК на новый.
- фундаментальные функции безопасности: защита ядра от изменений, усиление защиты служб, предотвращение несанкционированного исполнения данных, предохранение от случайных изменений структуры адресного пространства, сохранение обязательных уровней целостности;
- увеличение времени работы от аккумулятора за счёт снижения общего числа фоновых задач, диагностики причин перерасхода энергии, автоматическое сохранение питания во время простоя, снижение яркости дисплея, спящий режим и так далее;
- для IT-специалистов в Microsoft Windows 7 Ultimate предусмотрены работа через командную строку, поддержка WMI-сценариев для просмотра данных средства анализа стабильности системы (RAS), мониторинг стабильности системы[40].

В организации используется прикладное программное обеспечение Microsoft Office 2007, которое содержит всё необходимое для работы с текстом, таблицами, диаграммами, презентациями, базами данных, электронной почтой и многим другим.

На компьютерах установлен антивирусный продукт KasperskyAnti-Virus, необходимый для обеспечения безопасности операционной системы.

Для разработки автоматизированной системы учёта производственного процесса будут использоваться система управления базами данных FireBird, утилита IBExpert и среда визуального программирования Borland C++ Builder 6.0.

FireBird - это мощная, компактная, реляционная СУБД, основанная на ядре Borland InterBase и поддерживающая архитектуру «клиент-сервер». Она

предназначена для хранения и обработки больших объемов информации, в условиях работы нескольких пользователей. Для управления базой данных сервер FireBird использует домены, просмотры, хранимые процедуры, триггеры, генераторы, транзакции, а также пользовательские функции.

Firebird (InterBase) обладает такими преимуществами, как:

- многоверсионная архитектура, что обеспечивает параллельную обработку оперативных и аналитических запросов;
- компактность (дистрибутив 5Mb);
- высочайшая эффективность;
- самая полная языковая поддержка для хранимых процедур и триггеров.

Firebird (InterBase) является сервером баз данных (SQL сервер). Один SQL сервер Firebird сможет обрабатывать сразу несколько независимых баз данных, с множеством пользовательских соединений на каждой[5].

Firebird широко используется с 2001 года. Это коммерчески независимый проект C и C++ программистов, технических советников и разработчиков мультиплатформенных систем управления базами данных, основанный на исходном тексте, выпущенном корпорацией Borland 25 июля 2000 года в виде свободной версии Interbase 6.0. Выгодно отличается от MS SQL компактностью, кросплатформенностью (поддерживает Windows, GNU/Linux, FreeBSD, Solaris, OS X, HP-UX). Удобен и прост в использовании. Любой начинающий, знающий SQL на уровне оператора Select, легко в нем разберется. И, что немаловажно, за него не нужно платить. Допускает подключение пользовательских функций (UDF) в виде dll-библиотек, разработанных на любом языке программирования с использованием любой среды разработки (Delphi, C++Builder, MS VisualStudio, C++, Pascal). Имеется достаточно большое количество средств доступа к базам данных Firebird (Interbase) из-под ODBC, ADO, ADO.NET, BDE, php, perl, python. Этот sql сервер практически не имеет ограничений к применению.

Firebird (InterBase) обладает несомненными преимуществами перед

другими СУБД этого уровня, такими как: mysql, msql, postgresql. От таких СУБД (sqlserver) как MS SQL и Oracle его выгодно отличает:

- компактный размер;
- простота установки и администрирования;
- бесплатное распространение.

Для работы с FireBird используют утилиту IVExpert, которая позволяет не только полностью управлять структурами баз данных, но также создавать механизмы управления базой данных и отлаживать их [22].

IVExpert – инструмент для разработки FireBird баз данных на основе технологии InterBase 6.0. IVExpert позволяет осуществлять проектирование с заметной легкостью, быстротой, надежностью и удобством для разработчика. IVExpert включает много инструментов и особенностей кодирования: визуальные редакторы для всех типов базы данных, SQL- редакторы и сценарии, отладчик для хранимых процедур, генераторов и триггеров, исключения, домены и многое другое [22].

Borland C++ Builder 6.0 - выпущенное компанией Borland средство быстрой разработки приложений, позволяющее создавать приложения на языке C++, используя при этом среду разработки и библиотеку компонентов Delphi. C++Builder включает обширный набор средств, которые повышают производительность труда программистов и сокращают продолжительность цикла разработки, а также поставляется ряд компонентов InterBaseeXpress (IBX), позволяющих работать с сервером Firebird [28].

Borland C++ Builder 6.0 обеспечивает непревзойденную производительность и все преимущества визуальной разработки на основе легконастраиваемой среды AppBrowser IDE, предоставляет средства параллельной разработки, позволяющие параллельно визуально редактировать текст программы и изменять внешний вид используемых форм, специальные средства для повышения скорости программирования, такие как CodeInsight, CodeBrowser, ClassExplorer, ClassCompletion и ParameterCompletion [1].

2.4 Обоснование проектных решений по технологическому обеспечению

Технологическое обеспечение определяет последовательность процедур и задач по работе с информационной системой. Технологическое обеспечение включает в себя процедуры, реализуемые конкретной информационной технологией [34].

Весь технологический процесс можно подразделить на процессы сбора и ввода исходных данных в вычислительную систему, процессы размещения и хранения данных в памяти системы, процессы обработки данных с целью получения результатов и, процессы выдачи данных в виде, удобном для восприятия пользователем [34].

Операции сбора и регистрации данных осуществляются с помощью различных средств. Различают механизированный, автоматизированный и автоматический способы сбора и регистрации информации и данных.

- механизированный способ представляет собой сбор и регистрацию информации непосредственно человеком с использованием простейших приборов (весы, счетчики и так далее);

- автоматизированный способ предполагает использование машиночитаемых документов, регистрирующих автоматов, универсальных систем сбора и регистрации, обеспечивающих совмещение операций формирования первичных документов и получения машинных носителей;

- автоматический способ используется в основном при обработке данных в режиме реального времени. Информация с датчиков, учитывающих ход производства: выпуск продукции, затраты сырья, простои оборудования и так далее – поступает непосредственно в ЭВМ [34].

На данный момент времени в ООО «ВИД» используется механизированный способ сбора и регистрации информации. С помощью персонального компьютера вся поступившая информация приводится к машиночитаемому виду.

В разрабатываемой автоматизированной системе технологический процесс сбора информации будет представлять собой ввод информации с первичных документов в базу данных, при котором будет контролироваться допустимость введенных значений и обеспечиваться ввод данных путем выбора из списка. Все внесенные данные будут храниться в базе данных. Поступившая информация будет подвержена обработке и на ее основе можно будет получить результатную информацию в виде отчетов и выходных документов.

Информационная база (база данных) хранится в файле VID.FDB, который управляется с помощью клиент-серверной СУБД FireBird и утилиты IVExpert.

Для исключения возникающих в ходе работы ошибок будет проведено обучение персонала функциональным возможностям разработанной системы. Таким образом, будет усовершенствован учет производственного процесса в данной организации.

2.5 Обоснование выбора программных средств

Для проектирования информационных систем широко используются CASE-технологии. В настоящее время существует достаточное количество CASE-средств, например: OracleDesigner, BPwin, ERwin, RationalRose, Silverrun и другие. Однако проанализировав все преимущества и недостатки данных программных средств, были выбраны CASE-средства BPwin и Erwin, так как эти программные продукты имеют широкие функциональные возможности для проектирования информационной системы, низкую стоимость, а также ранее был получен опыт работы с ними.

Для разработки функциональных моделей автоматизированной информационной подсистемы учёта производственного процесса будем использовать программное средство AllFusionProcessModeler 7. Данный продукт представляет собой CASE-средство для моделирования, анализа, документирования и оптимизации бизнес-процессов. Поддерживается

использование следующих видов диаграмм:

- BusinessProcess (методология IDEF0). Данный вид диаграмм предназначен для описания бизнес процессов путем разбиения его на более мелкие процессы, которые взаимодействуют друг с другом. Каждый процесс осуществляет преобразование входящей информации в исходящую путем использования какого-то механизма и под каким-то управляющим воздействием;

- ProcessFlow (методология IDEF3) предполагает представление процесса как упорядоченной последовательности событий;

- DataFlow (методология DFD), или диаграммы потоков данных. В данной методологии предполагается разбиение процесса на более мелкие подпроцессы, которые путем взаимодействия друг с другом образуют желаемый результат[20].

Для проектирования структуры базы данных автоматизированной информационной подсистемы учёта производственного процесса используется программное средство AllFusionErwinDataModeler 7, который позволяет визуализировать структуру информационной базы, а также взаимосвязи между различными блоками данных. Данный продукт позволяет не только «нарисовать» модель базы данных, но и на основании построенной модели предусмотренное автоматическое создание структуры базы данных, то есть создание таблиц базы, их полей и установление связей между различными таблицами.

AllFusionERwinDataModeler позволяет осуществлять автоматизированное проектирование реляционных баз данных в визуальном редакторе. Для этого используется три основных типа объектов:

- сущности, которые представляют собой таблицу базы данных. Сущности обычно состоят из ряда атрибутов;

- атрибуты, представляющие конкретное поле таблицы базы данных. Атрибут представляет конкретное свойство, которое может принимать различные значения;

- связи, которые отражают взаимосвязи между таблицами базы[20].

3 ПРОЕКТНАЯ ЧАСТЬ

3.1 Информационное обеспечение задачи

3.1.1 Информационная модель и ее описание

Важным этапом в проектировании автоматизированной системы является построение информационной модели.

Информационная модель — модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об изменениях входных величин моделировать возможные состояния объекта [6].

В работе информационная модель будет создана структурно – функциональная модель, построенная с использованием CASE-средства AllFusionProcessModeler.

Новая информационная модель «КАК ДОЛЖНО БЫТЬ» претерпит некоторые изменения, то есть будут выделены и автоматизированы задачи, непосредственно относящиеся к оператору станка с ЧПУ. Вся поступившая информация будет фиксироваться, храниться и обрабатываться в разработанной автоматизированной системе, а также появиться возможность получать различную отчетность и анализировать ее.

Рассмотрим структурно – функциональную модель учёта производственного процесса в организации в нотации IDEF0. Контекстная диаграмма представлена на рисунке 3.1:

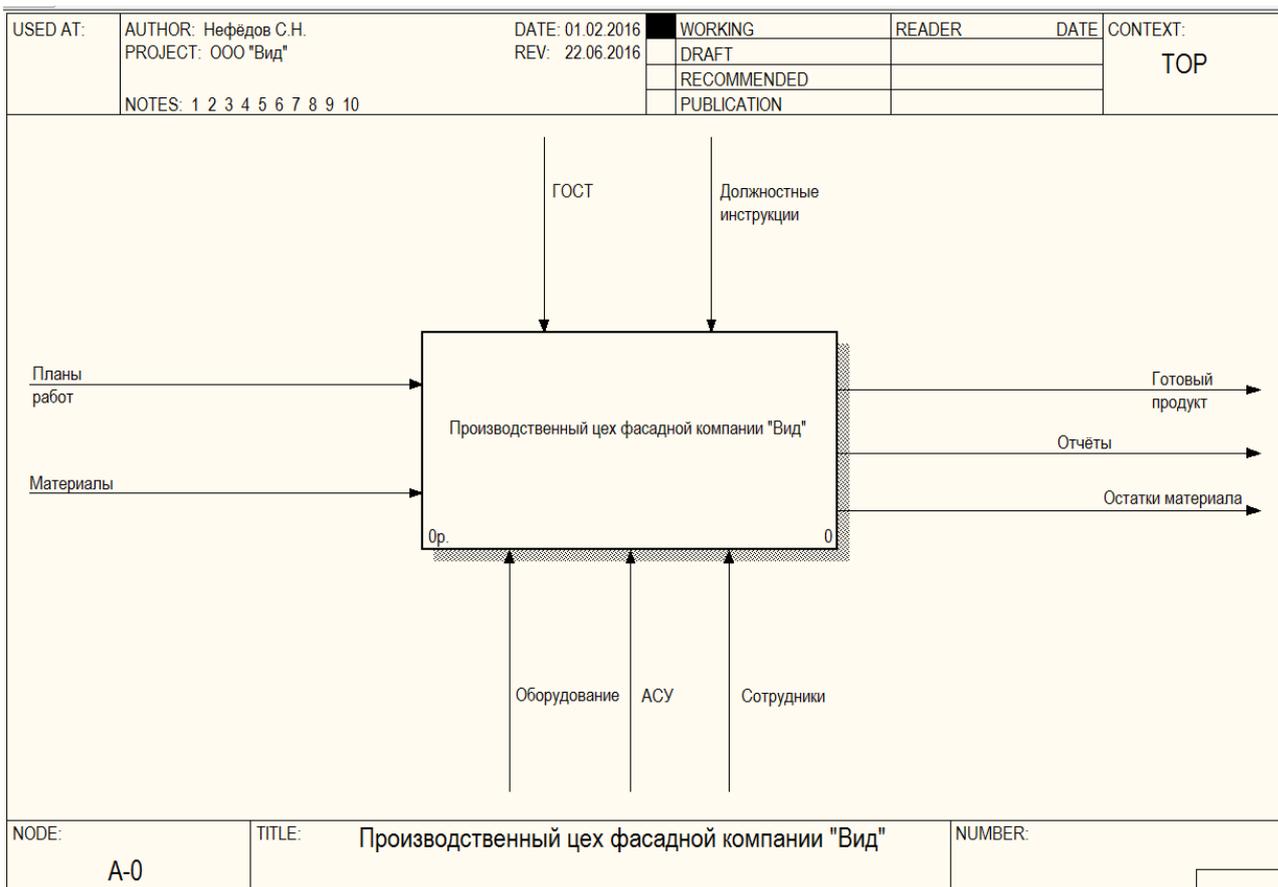


Рисунок 3.1 - Контекстная диаграмма (IDEF0)

Из контекстной диаграммы видно, что входной информацией являются план работ, материал и требования к продукту.

Выходная информация представляет собой готовый продукт, отчёт по заказам и остаток материалов.

В качестве управляющих объектов выступают требования качества и должностные инструкции.

К механизмам исполнения относятся оборудование и сотрудники.

После построения контекстная диаграмма детализируется с помощью диаграммы декомпозиции первого уровня. На этой диаграмме отображаются функции системы, которые должны быть реализованы в рамках основной функции. Декомпозиция контекстной диаграммы представлена на рисунке 3.2:

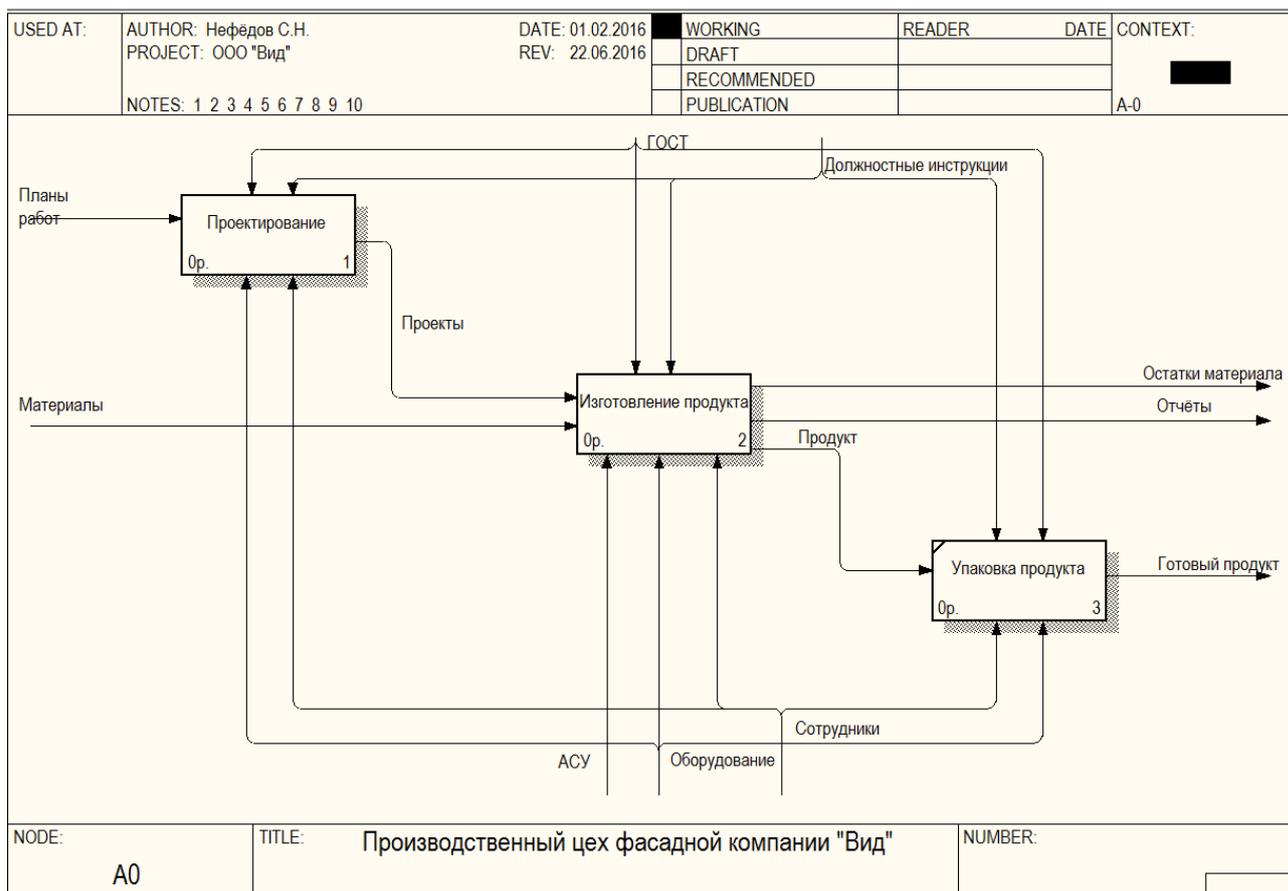


Рисунок 3.2 – Декомпозиция контекстной диаграммы (IDEF0)

На рисунке видно, что диаграмма декомпозиции содержит три функциональных блока: «Проектирование продукта», «Изготовление продукта», «Упаковка продукта»:

- блок «Проектирование», в котором происходит проектирование заказа, заключающееся в составлении чертежей деталей. В результате выполнения данного блока будет получен проект заказа;

- блок «Изготовление продукта», в котором осуществляется производство деталей на основании проекта и дополнительных требований заказчика из поставленных материалов. В результате преобразования этого блока будет получена готовая продукция.

- блок «Упаковка продукта», в котором осуществляется упаковывание деталей по партиям и подготовка к отправке на место сбора фасада. В результате преобразования данного блока будет получен упакованные детали заказа, переданный заведующему складу.

Диаграмма декомпозиции функционального блока «Изготовление продукта» показана на рисунке 3.3:

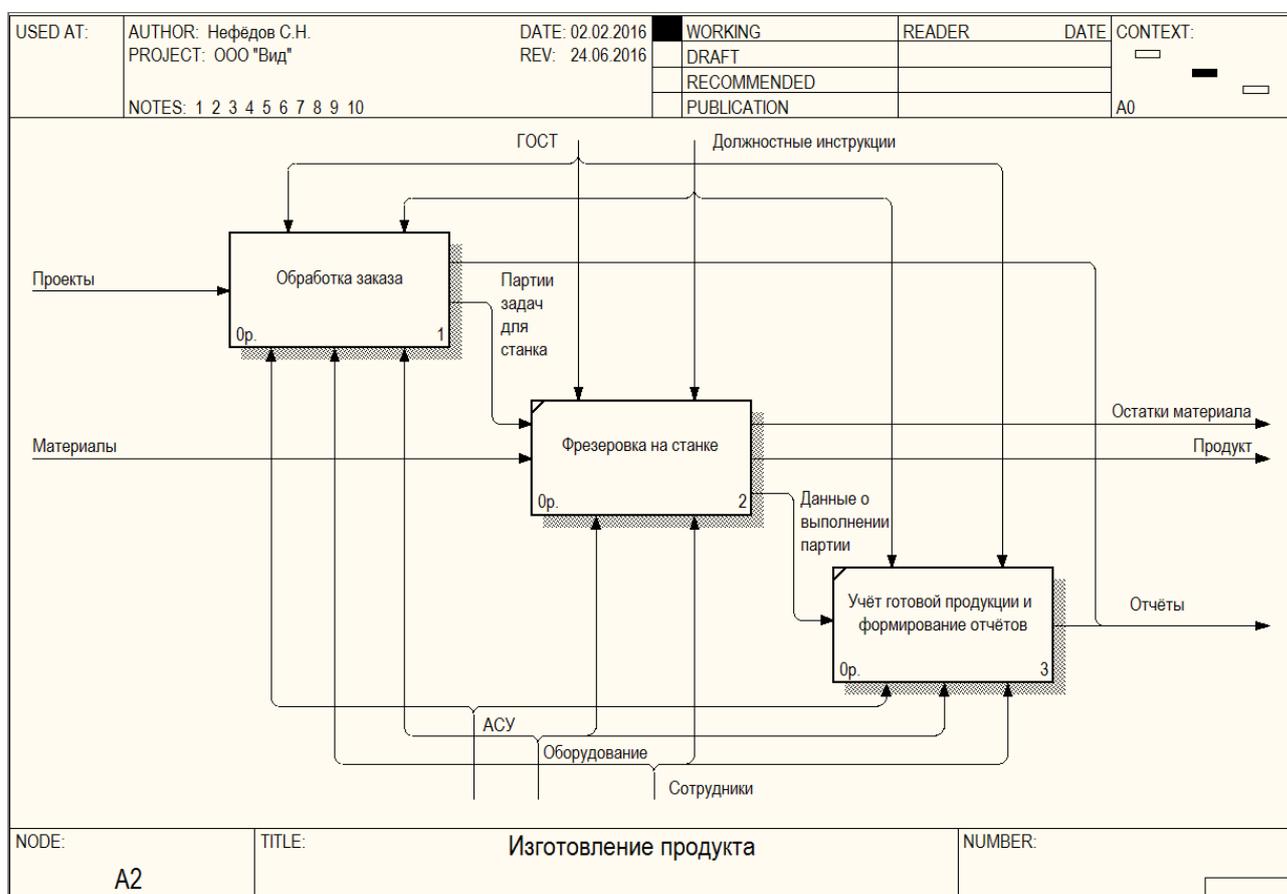


Рисунок 3.3 – Диаграмма декомпозиции блока «Изготовление продукта» (IDEF0)

Диаграмма декомпозиции блока «Изготовление продукта» содержит три функциональных блока:

- блок «Обработка заказа», в котором осуществляется реализация функций разработанной системы учёта. В результате выполнения данного блока будут получены партии задач для станка с ЧПУ и отчёты по существующим заказам;
- блок «Фрезеровка на станке», в котором происходит изготовление продуктов из материалов по составленным партиям. Результатом выполнения блока будут получены готовый продукт, остаток материала и отчёт по выполненной партии;

– блок «Учёт готовой продукции и формирование отчётов», в котором осуществляется обновление дополнительных данных о заказе в системе после выполнения очередной партии деталей и составление отчётов. В результате преобразования данного блока будет отчёт о выполненной партии заказа.

Диаграмма декомпозиции функционального блока «Изготовление продукта» показана на рисунке 3.4:

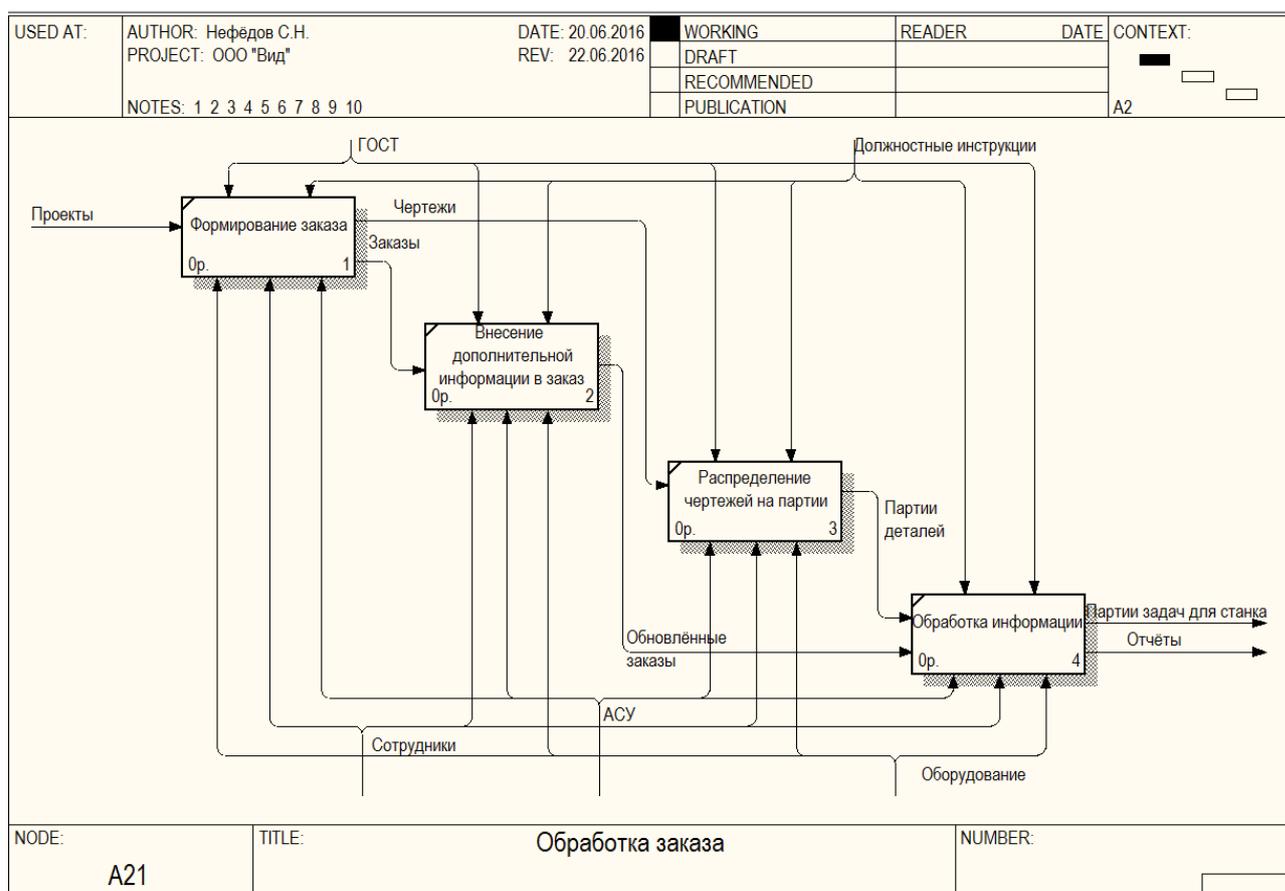


Рисунок 3.4 – Диаграмма декомпозиции блока «Обработка заказа в системе учёта» (IDEF0)

На рисунке видно, что диаграмма декомпозиции блока «Обработка заказа в системе» содержит четыре функциональных блока: «Формирование заказа», «Внесение дополнительной информации в заказ», «Распределение чертежей на партии» и «Обработка информации»:

- блок «Обработка заказа в системе», в котором осуществляется заполнение первичных данных о заказе в систему учёта на основании проекта. В результате выполнения данного блока будут чертежи и заказ в системе;
- блок «Внесение дополнительной информации в заказ», в котором заполняется дополнительная информация о заказе в систему для реализации её функций. Результатом выполнения блока будет обновлённый заказ;
- блок «Распределение чертежей на партии», в котором осуществляется распределение чертежей деталей на материале, исходя из его размеров. В результате преобразования данного блока будет получен план партий, содержащий файлы каждой партии отдельно и общее количество партий;
- блок «Обработка информации», в котором в котором происходит подготовка заказа к выполнению на станке с ЧПУ. В результате преобразования этого блока будет отправка партии на фрезеровку и сформированные отчёты о заказах.

3.1.2 Используемые классификаторы и системы кодирования

Классификатором называется систематизированный перечень наименованных объектов, каждому из которых в соответствие дан уникальный код. Классификация объектов производится согласно правилам распределения заданного множества объектов на подмножества (классификационные группировки) в соответствии с установленными признаками их различия или сходства.

Кодирование представляет собой присвоение кодов классификационной группировке или объекту классификации. Кодирование предназначено для формализованного описания наименований объектов, характеристик, группировок. Обычно кодирование представляет собой процесс обозначения исходного множества объектов или сообщений набором символов заданного алфавита на основе совокупности определенных правил [26].

Существуют порядковый, серийно-порядковый, последовательный и параллельный методы кодирования.

При порядковом методе кодирования кодовым обозначением служат числа натурального ряда и каждый объект кодируется с помощью текущего номера, который однозначно идентифицирует этот объект.

При серийно-порядковом методе кодами служат числа натурального ряда с закрепленной отдельной серией этих чисел за объектами классификации с одинаковыми признаками.

При последовательном методе в кодовом обозначении знаки на каждой ступени деления зависят от результатов разбиения на предыдущих ступенях. В результате кодовое обозначение группировки дает информацию о последовательности признаков, характеризующих эту группировку.

При параллельном методе признаки классификации кодируются независимо друг от друга определенными разрядами или группой разрядов кодового обозначения [26].

В таблице 3.1 представлены используемые классификаторы.

Таблица 3.1 – Используемые классификаторы.

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Система классификации	Вид классификатора
Код цвета	3	Порядковая	Отсутствует	Локальный
Код клиента	3	Порядковая	Отсутствует	Локальный
Код материала	2	Порядковая	Отсутствует	Локальный
Код заказа	3	Порядковая	Отсутствует	Локальный
Код статуса	1	Порядковая	Отсутствует	Локальный
Код толщины	2	Порядковая	Отсутствует	Локальный
Код вида	2	Порядковая	Отсутствует	Локальный

Данные классификаторы будут использованы при разработке базы данных автоматизированной информационной подсистемы учёта производственного процесса.

3.1.3 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией

Входная информация представляет собой всю информацию, необходимую для решения задачи и расположенную на различных носителях, например, в первичных документах, на машинных носителях, в памяти персонального компьютера.

Вся нормативно-справочная информация хранится в справочниках, содержащих сведения о заказах, клиентах, материалах, видах, цветах и толщине материалов. В справочниках хранится информация многократного использования, предусмотрена возможность добавления, удаления и редактирования содержащихся данных.

Справочник «Заказы» содержит всю необходимую информацию о заказе и заполняется в два этапа:

- первичная информация, полученная при составлении проекта, в которую входит заказчик, объём работ в метрах квадратных, номер материала для изготовления деталей и дата, к которой этот заказ должен быть выполнен;
- вторичная информация, необходимая для работы функции отображения статуса и прогресса выполнения заказа по партиям. Такой информацией является общее количество партий деталей, а так же количество выполненных партий из общего количества и статус заказа, заполняющиеся автоматически после добавления общего количества партий со значениями «0» и «В очереди» соответственно.

Пример заполнения справочника представлен на рисунках 3.5-3.6:

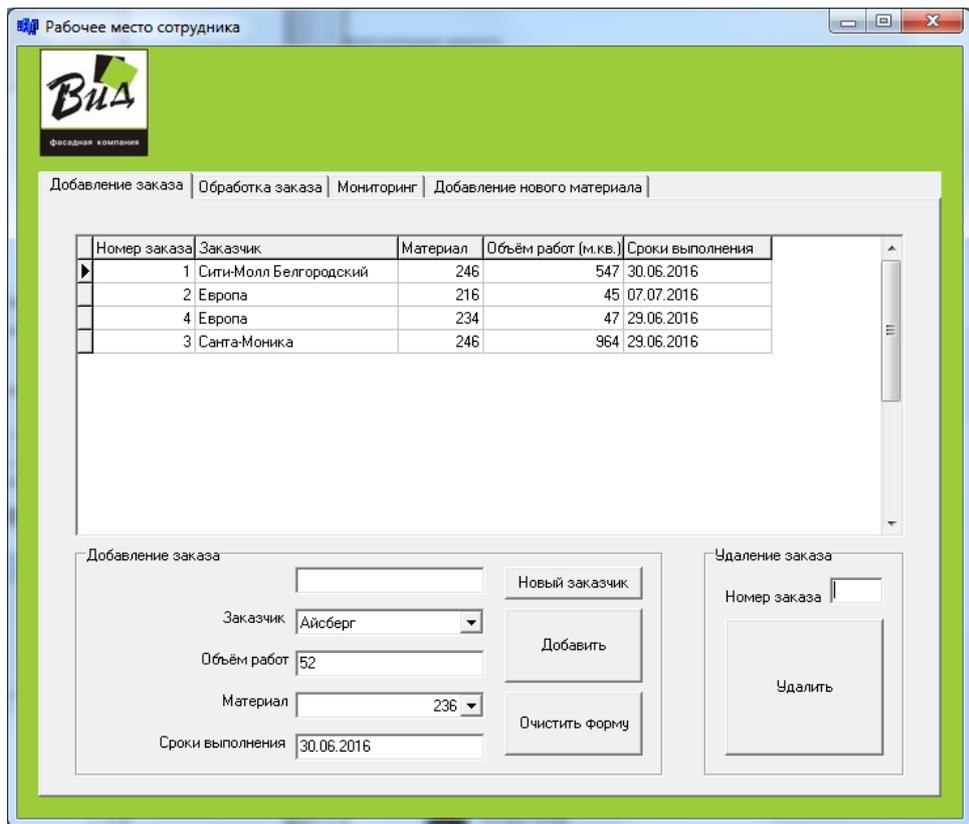


Рисунок 3.5 – Экранная форма первичной информации справочника «Заказы»

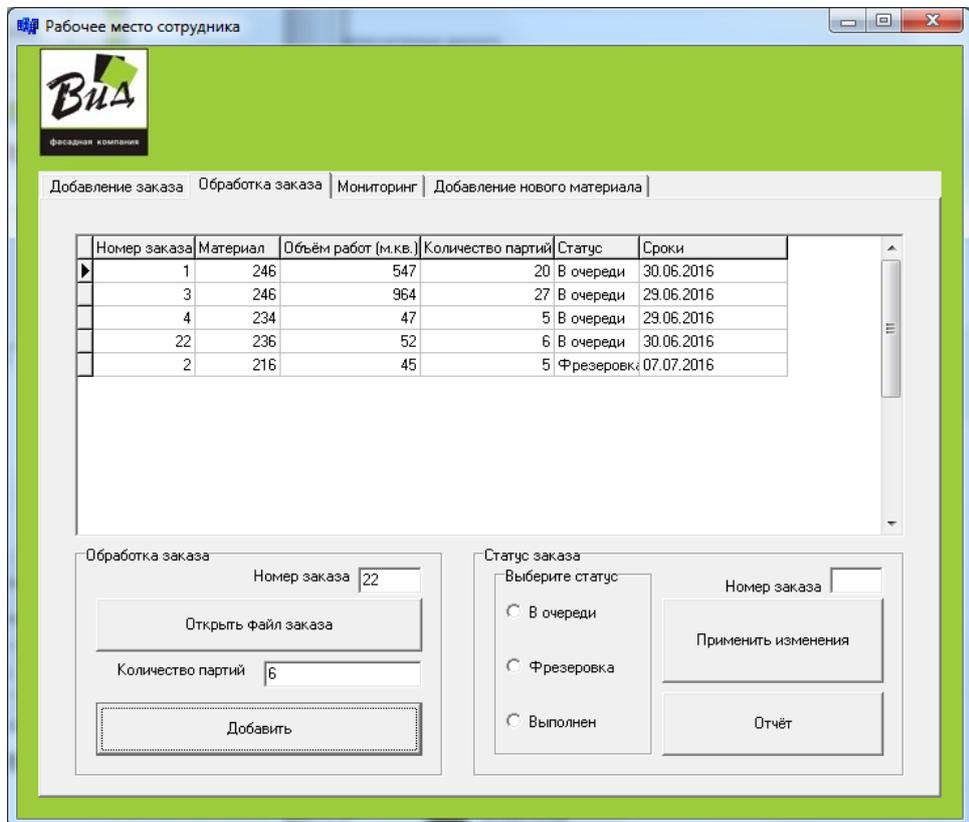


Рисунок 3.6 – Экранная форма вторичной информации справочника «Заказы»

Справочник «Клиент» хранит информацию о клиентах фасадной компании «Вид» и, по причине постоянного наращивания клиентской базы, функция добавления нового заказчика используется довольно часто, поэтому данный справочник заполняется непосредственно на странице добавления нового заказа. На рисунке 3.7 представлена форма заполнения справочника:

Номер заказа	Заказчик	Материал	Объем работ (м.кв.)	Сроки выполнения
1	Сити-Молл Белгородский	246	547	30.06.2016
22	Айсберг	236	52	30.06.2016
2	Европа	216	45	07.07.2016
4	Европа	234	47	29.06.2016
3	Санта-Моника	246	964	29.06.2016

Добавление заказа

Заказчик: БелГУ

Объем работ: Айсберг

Материал: Европа

Сроки выполнения: Санта-Моника

Новый заказчик

Добавить

Очистить форму

Удаление заказа

Номер заказа

Удалить

Рисунок 3.7 – Экранная форма первичной информации справочника «Заказы»

В справочнике «Материал» хранятся данные о материалах, доступных для фрезеровки. На рисунке 3.8 представлена форма заполнения справочника:

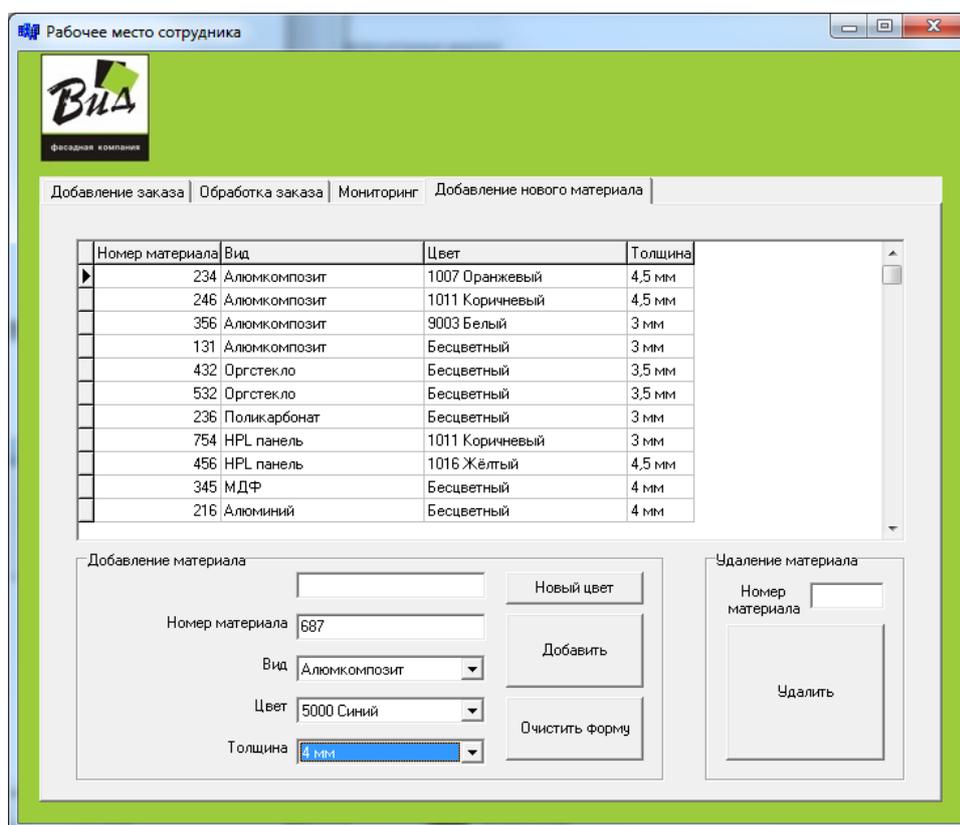


Рисунок 3.8 - Экранная форма справочника «Материал»

Справочник «Цвет материала», «Вид материала» и «Толщина материала» содержит сведения о классификации материалов.

Таким образом, были рассмотрены все справочники разработанной автоматизированной системы учета производственного процесса.

3.1.4 Характеристика базы данных

3.1.4.1 Характеристика модели БД

Инфологическая модель представляет собой модель предметной области, определяющая совокупности информационных объектов, их атрибутов и отношений между объектами, динамику изменений предметной области, а также характер информационных потребностей пользователей. Инфологическая модель предметной области описана моделью "сущность—связь", в основе которой лежит деление реального мира на отдельные различимые сущности, находящиеся в определенных связях друг с другом, с

использованием программного средства AllFusionErwinDataModeler 7.

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый - логический (точка зрения пользователя) - описывает данные, задействованные в бизнесе предприятия. Второй - физический - определяет представление информации в базе данных. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления [14].

Логическая модель базы данных представлена на рисунке 3.9:



Рисунок 3.9 – Логическая модель базы данных

На основе логической модели базы данных была спроектирована физическая модель, показанная на рисунке 3.10:

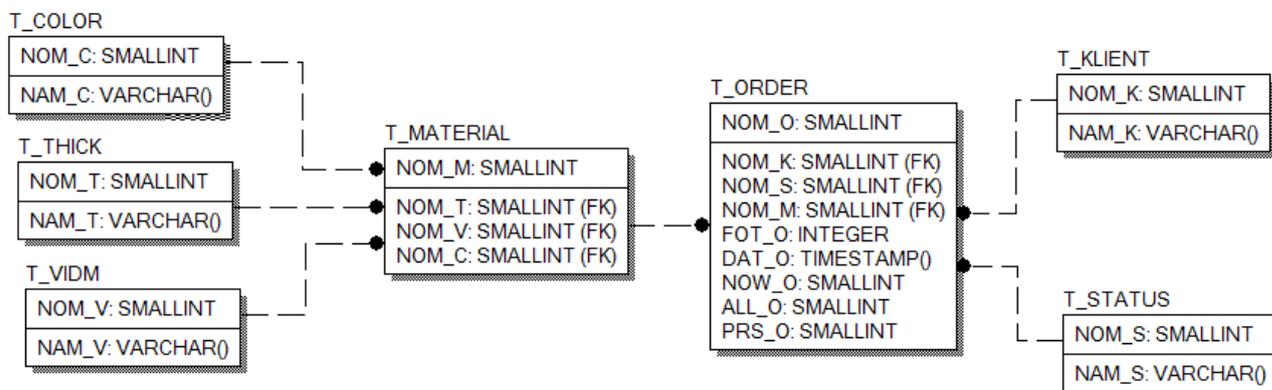


Рисунок 3.10 – Физическая модель базы данных

На основе физической модели была разработана база данных.

3.1.4.2 Характеристика даталогической модели БД

Даталогическая модель - это модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учётом конкретной реализации СУБД, а также с учётом специфики конкретной предметной области на основе ее инфологической модели [14].

В таблице 3.2 представлено описание даталогической модели базы данных.

Таблица 3.2 – Описание даталогической модели базы данных

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
Заказ	T_ORDER	Номер заказа	NOM_O	Smallint
		Номер клиента	NOM_K	Smallint
		Номер материала	NOM_M	Smallint
		Метраж заказа	FOT_O	Smallint
		Срок выполнения заказа	DAT_O	Timestamp
		Выполнено партий	NOW_O	Smallint
		Всего партий	ALL_O	Smallint
		Прогресс заказа	PRS_O	Numeric
		Номер статуса	NOM_S	Smallint
Клиент	T_KLIENT	Номер клиента	NOM_K	Smallint
		Имя клиента	NAM_K	Varchar(20)
Материал	T_MATERIAL	Номер материала	NOM_M	Smallint
		Номер вида	NOM_V	Smallint
		Номер цвета	NOM_C	Smallint
		Номер толщины	NOM_T	Smallint
Вид материала	T_VIDM	Номер вида	NOM_V	Smallint
		Название вида	NAM_V	Varchar(15)
Цвет материала	T_COLOR	Номер цвета	NOM_C	Smallint
		Название цвета	NAM_C	Varchar(15)
Толщина материала	T_THICK	Номер толщины	NOM_T	Smallint
		Значение толщины	NAM_T	Varchar(6)
Статус	T_STATUS	Номер статуса	NOM_S	Smallint
		Название статуса	NAM_S	Varchar(10)

Реализованная база данных будет интегрирована в автоматизированную информационную подсистему учёта производственного процесса.

3.1.5 Характеристика результатной информации

3.1.5.1 Характеристика таблиц с результатной информацией

Таблицы с результатной информацией формируются в результате запросов к объектам, хранящим входные данные - справочникам с условно-постоянными сведениями [28].

В разработанной автоматизированной системе были созданы таблицы и представления. Например, на рисунке 3.11 показана работа представления, позволяющего просмотреть необходимую для отслеживания производственного процесса информацию о заказах на основе таблиц «Заказ», «Клиент» и «Статус», используемое в документе «Прогресс заказа». Информация, полученная при помощи представления, подлежит дальнейшему хранению в базе данных и при необходимости ее можно редактировать, удалять и обрабатывать, а также добавлять новую информацию.

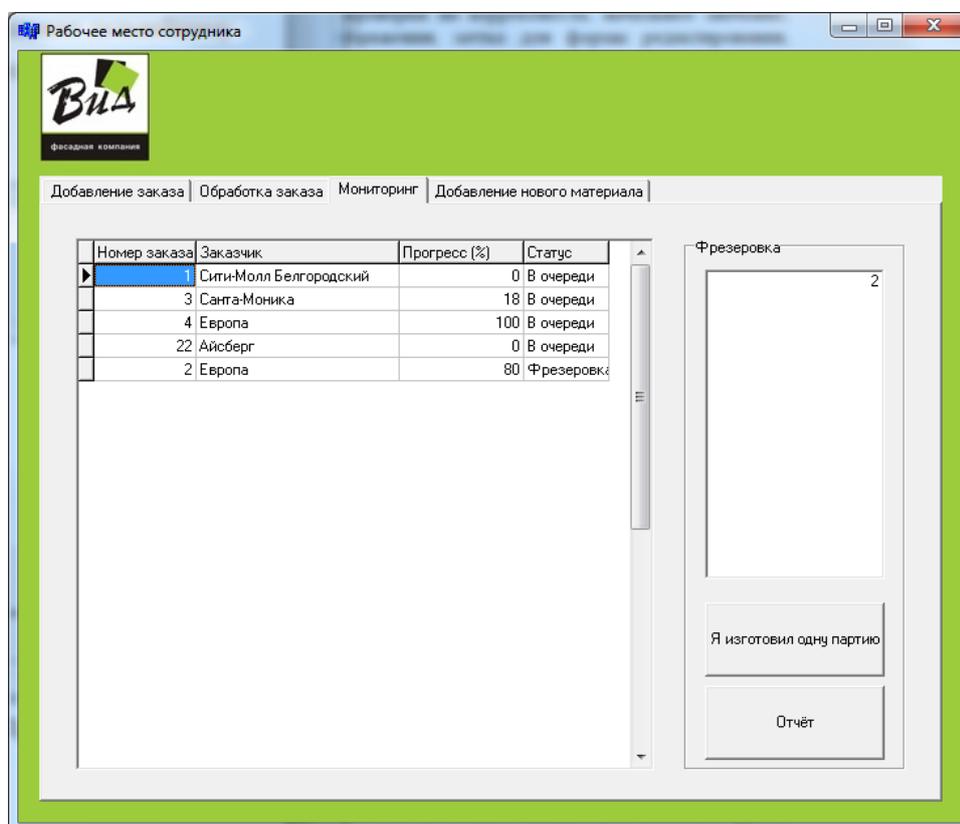


Рисунок 3.11 – Работа представления в документе «Прогресс заказа»

Программный SQL-код этого представления приведен рисунке 3.12:

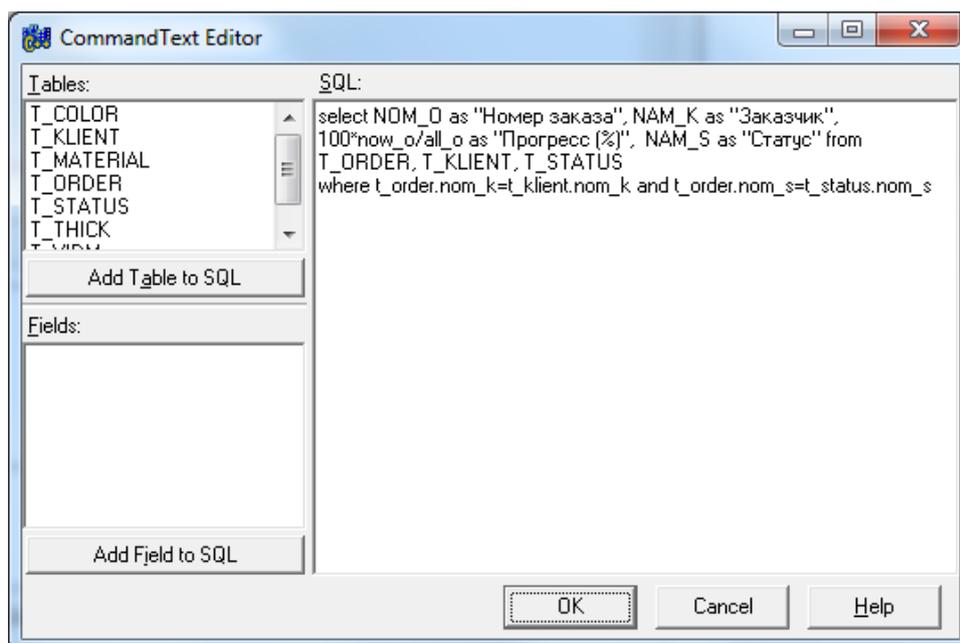


Рисунок 3.12 – Программный SQL-код представления

Аналогичным образом с помощью представлений были созданы все остальные документы.

Документ «Заказ» предназначен для регистрации новых заказов в системе, используя первичные данные. Форма и списки документов представлены на рисунке 3.13.

Далее был создан документ «Обработанный заказ», который предназначен для последующей обработки в документе «Прогресс заказа», представленный на рисунке 3.14.

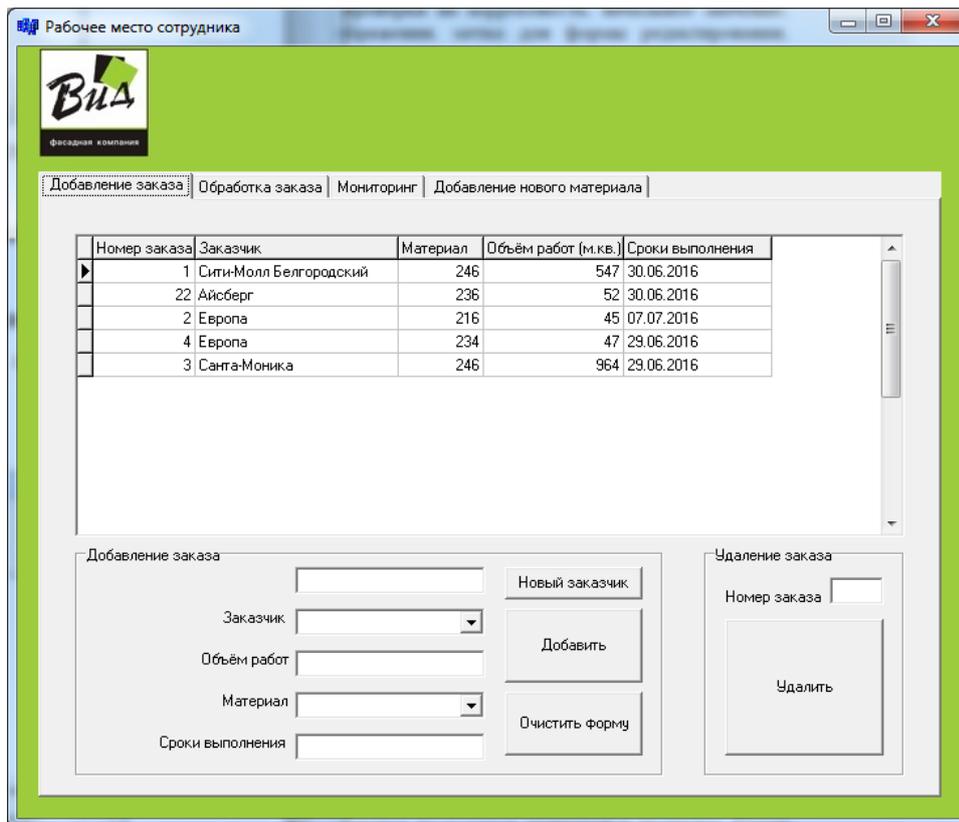


Рисунок 3.13– Экранная форма и список документов «Заказ»

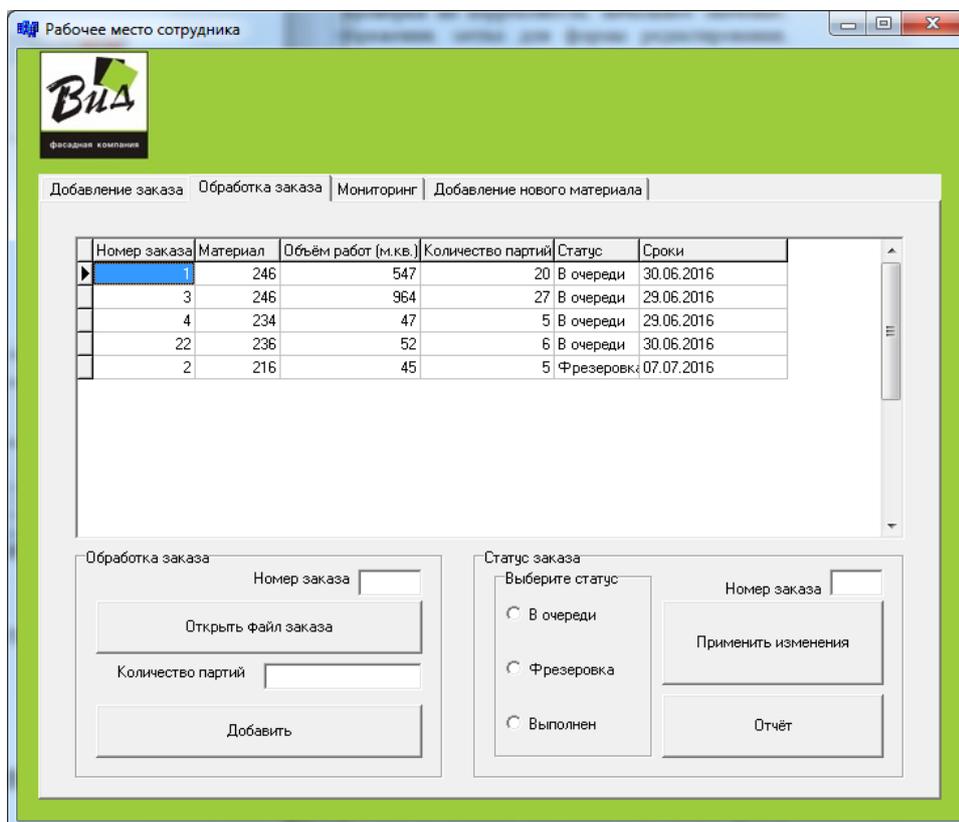


Рисунок 3.15 – Экранная форма и список документов «Обработанный заказ»

Документ «Материал» предназначен для регистрации материалов, доступных для фрезеровки. Форма и список документов представлены на рисунке 3.16:

Номер материала	Вид	Цвет	Толщина
234	Алюмкомпозит	1007 Оранжевый	4,5 мм
246	Алюмкомпозит	1011 Коричневый	4,5 мм
356	Алюмкомпозит	9003 Белый	3 мм
131	Алюмкомпозит	Бесцветный	3 мм
432	Оргстекло	Бесцветный	3,5 мм
532	Оргстекло	Бесцветный	3,5 мм
236	Поликарбонат	Бесцветный	3 мм
754	НРЛ панель	1011 Коричневый	3 мм
456	НРЛ панель	1016 Желтый	4,5 мм
345	МДФ	Бесцветный	4 мм
216	Алюминий	Бесцветный	4 мм

Рисунок 3.16 – Экранная форма и список документов «Материал»

Результатная информация автоматизированной системы учета производственного процесса будет использоваться в результатных документах системы.

3.1.5.2 Характеристика результатных документов

Результатными документами в автоматизированной системе являются отчеты, предоставляющие возможность отслеживать и анализировать информацию о выполненных, находящихся в очереди и выполняемых, на момент формирования отчёта, заказах, зарегистрированных в системе.

На рисунке 3.17 показан сформированный отчет по документу «Обработанный заказ», с помощью которого можно посмотреть информацию о всех обработанных заказах:

Номер заказа	Материал	Сроки	Статус	Объем работ (м.кв.)
1	246	30.06.2016	В очереди	547
3	246	29.06.2016	В очереди	964
4	234	29.06.2016	В очереди	47
22	236	30.06.2016	В очереди	52
2	216	07.07.2016	Фрезеровка	45

Рисунок 3.17 – Сформированный отчет по заказам

На рисунке 3.18 продемонстрирован сформированный отчет по выполненным заказам, с помощью которого можно посмотреть информацию о стадии выполнения заказов:

Номер заказа	Заказчик	Прогресс (%)	Статус
1	Сити-Молл Белгородский	0	В очереди
3	Санта-Моника	18	В очереди
4	Европа	100	В очереди
22	Айсберг	0	В очереди
2	Европа	80	Фрезеровка

Рисунок 3.18 – Сформированный отчет по статусу заказов

Данные отчёты доступны как для просмотра, так и для печати.

3.2 Программное обеспечение задачи

Модель дерева функций относится к функциональному представлению и предназначена для описания иерархической структуры функций (включая

статические связи между ними) бизнес-процессов предприятия.

Функция – это задача, операция или действие, которые выполняются над объектом для достижения одной или нескольких целей.

Дерево функций автоматизированной системы учета производственного процесса показано на рисунке 3.19:

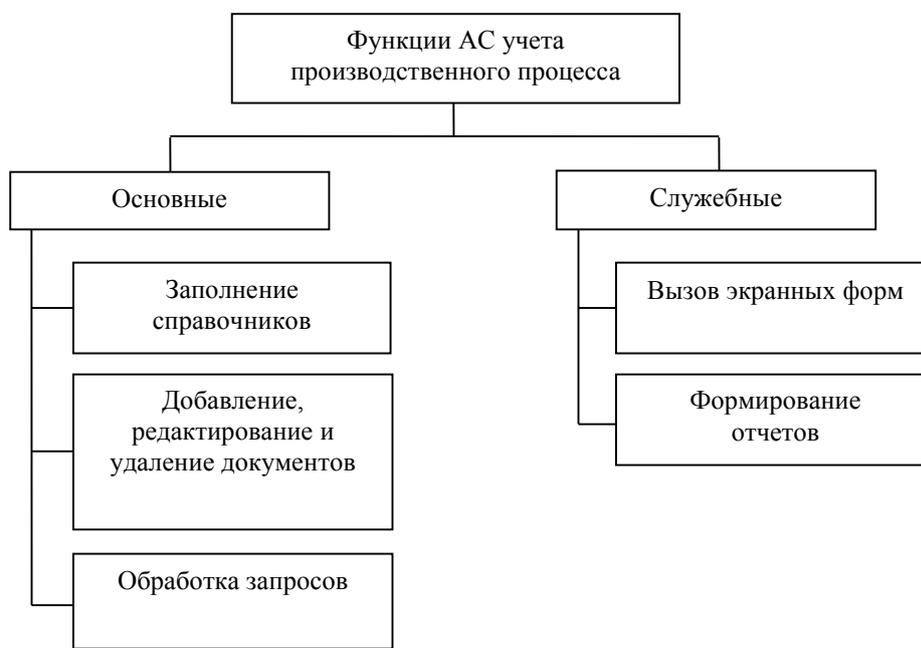


Рисунок 3.19 – Дерево функций

Были выделены следующие подмножества функций:

- основные функции, к которым относятся: заполнение справочников, добавление, редактирование и удаление документов и обработка запросов;
- служебные функции, к которым относятся: вызов экранных форм и формирование отчетов.

3.3 Технологическое обеспечение задачи

3.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

Технологическое обеспечение отражает организацию технологии сбора,

передачи, обработки и выдачи данных и описывает последовательность действий от получения первичной информации и до составления результатных документов [18].

Технологический процесс обработки данных - это упорядоченная последовательность взаимосвязанных действий по обработке информации до получения требуемого пользователю результата [18].

На технологический процесс обработки данных большое влияние оказывают характер выполняемых задач, применяемые технические средства, системы контроля, количество пользователей и иные аспекты. Технологический процесс обработки информации состоит из таких операций, как:

- сбор данных, представляющий собой процесс регистрации, фиксации, записи информации о связях, событиях, объектах, действиях, признаках;
- обработка данных, к которой относятся следующие действия: расчеты, поиск, фильтрация, выборка, объединение, сортировка, слияние и другие;
- генерирование информации, которое заключается в организации, переорганизации и преобразовании данных в форму, необходимую пользователям;
- хранение данных и информации, включающее размещение, накопление, копирование и выработку данных и информации с целью их последующего применения;
- передача информации и данных, которая представляет собой распространение информации и данных между всеми пользователями при помощи средств и систем коммуникаций и путем перемещения данных от источника (отправителя) к приемнику (получателю) [18].

Технологический процесс начинается с загрузки операционной системы. После загрузки операционной системы необходимо запустить программу автоматизированной информационной подсистемы учёта производственного процесса vid.exe (рисунок 3.20).

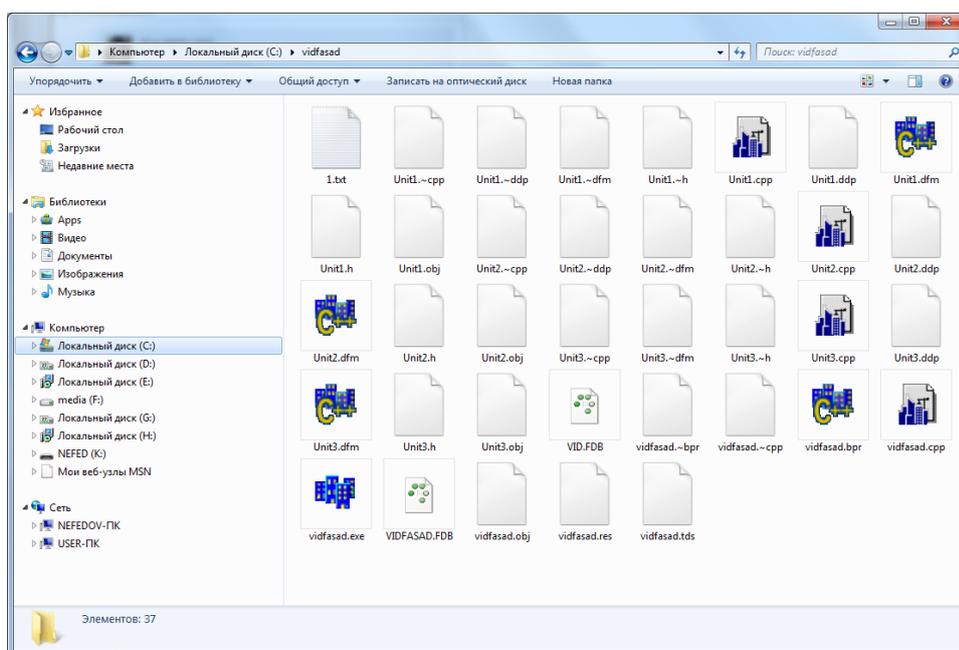


Рисунок 3.20 – Файловая структура программы

После запуска программы выводится информационное окно и активизируется система меню. Работа с программой осуществляется в диалоговом режиме. Главное меню программы содержит такие пункты, как: добавление заказа, обработка заказа, мониторинг, добавление нового материала.

Результатом действия каждого пункта меню является добавление, удаление и редактирование данных, отображение списков документов и отчётов по ним. Всю информацию можно вводить заново, редактировать и удалять. Кроме того, предусмотрена возможность вывода на бумажный носитель (на печать) всех электронных документов.

3.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 3.21 представлен фрагмент схемы технологического процесса сбора, передачи, обработки и выдачи информации на примере работы с документом «Заказ»:

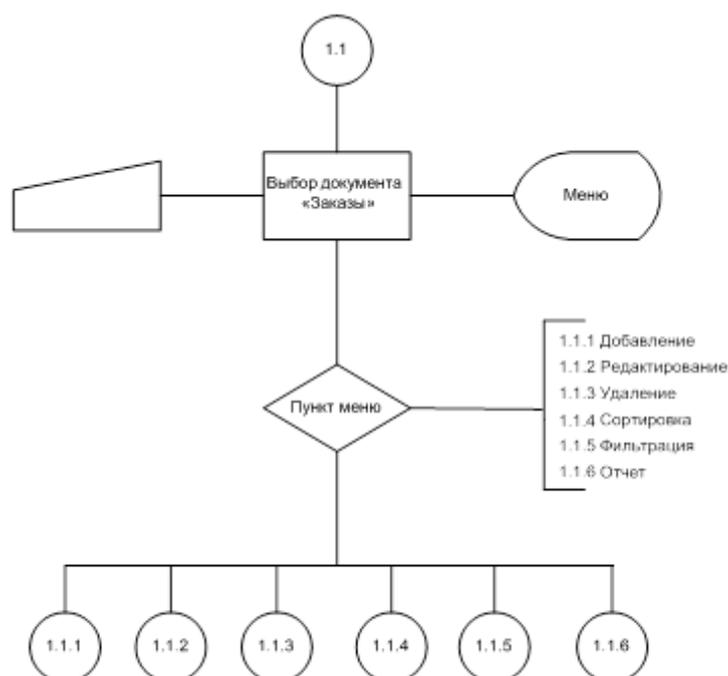


Рисунок 3.21 – Фрагмент схемы технологического процесса

При работе с остальными справочниками и документами, используемыми в автоматизированной системе, технологические процессы сбора, передачи, обработки и выдачи информации осуществляются аналогичным образом.

3.4 Описание контрольного примера реализации проекта

Протестируем разработанную автоматизированную информационную подсистему учета производственного процесса. При открытии программы появляется главное окно, показанное на рисунке 3.22:

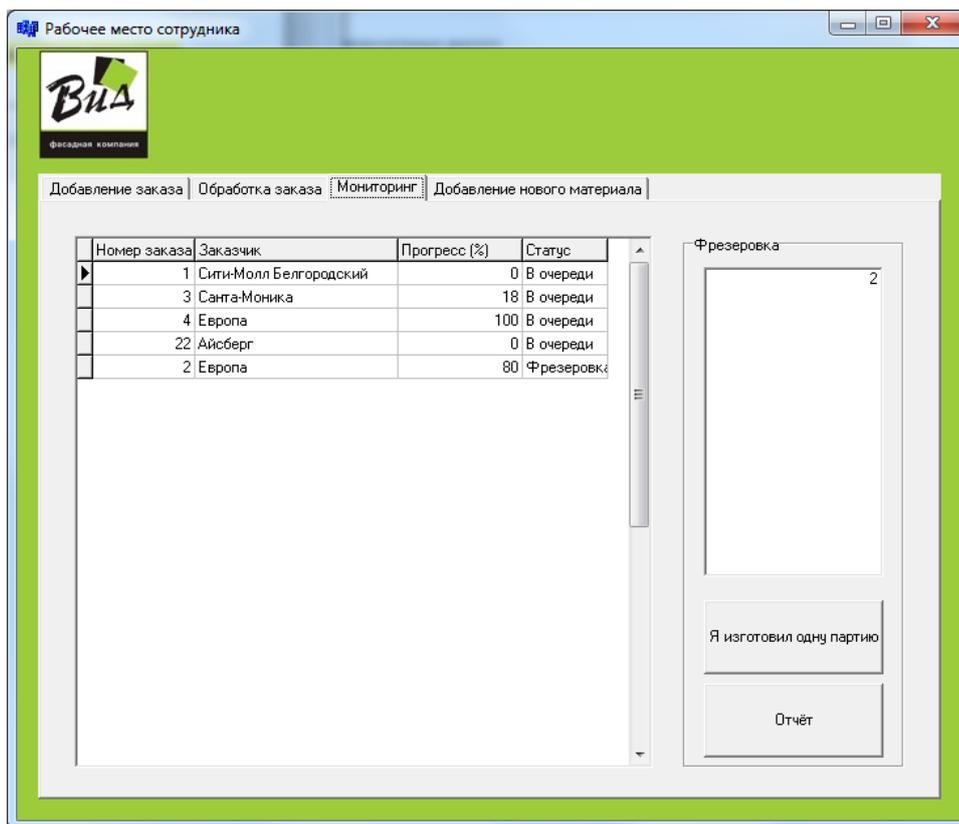


Рисунок 3.22– Главное окно программы

В главном окне программы находится меню, состоящее из следующих разделов:

- добавление заказа;
- обработка заказа;
- мониторинг;
- добавление нового материала;

Для описания контрольного примера работы программного модуля возьмём раздел «Добавление заказа». На рисунках 3.23 и 3.24 представлено добавление информации о новом заказе:

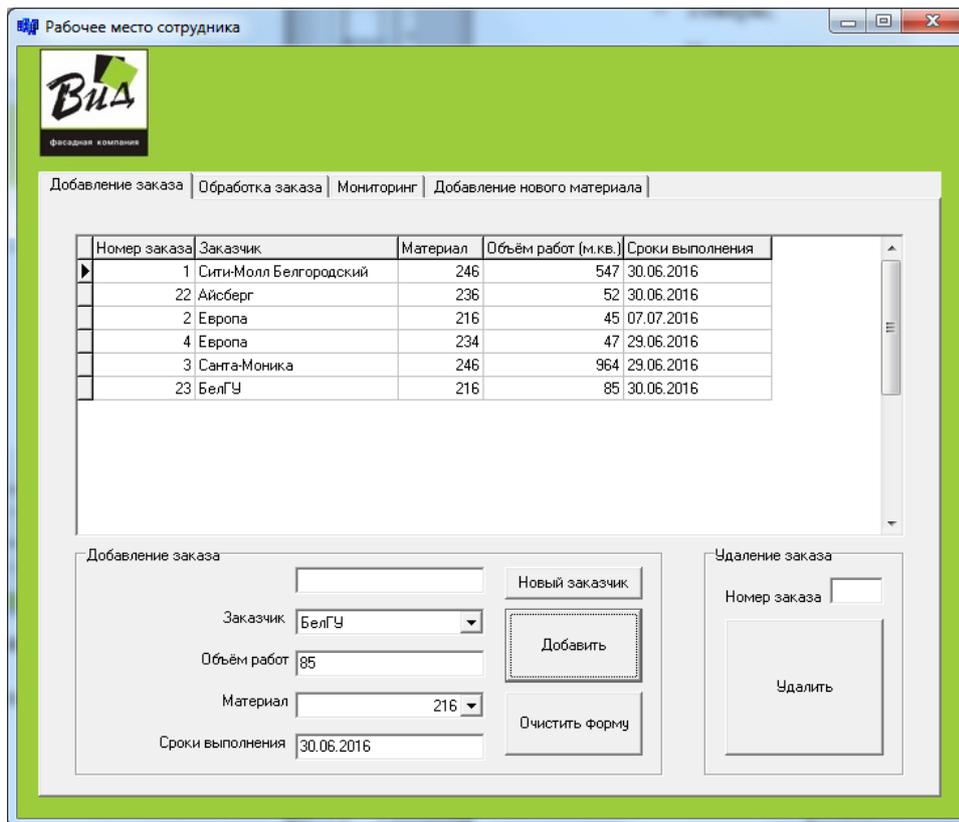


Рисунок 3.23 – Заполненная форма «Добавление заказа»

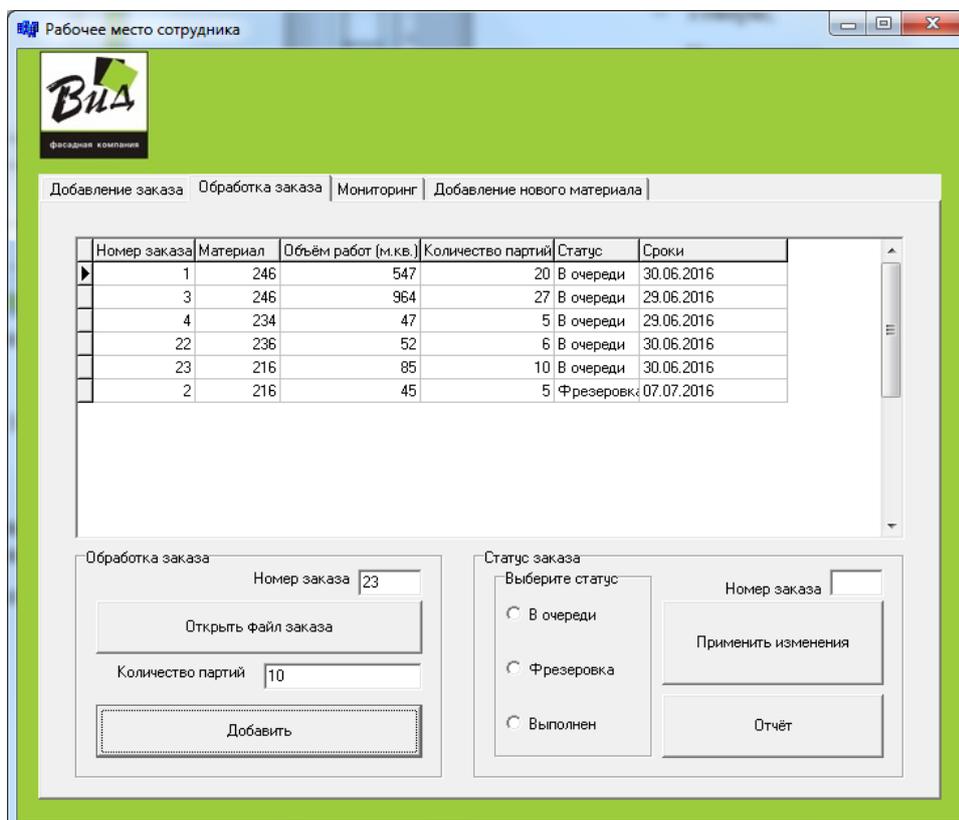


Рисунок 3.24 - Заполненная форма «Обработка заказа»

На рисунке 3.25 представлена форма для удаления заказа из документа:

Рабочее место сотрудника

ВИА
фасадная компания

Добавление заказа | Обработка заказа | Мониторинг | Добавление нового материала

Номер заказа	Заказчик	Материал	Объем работ (м.кв.)	Сроки выполнения
1	Сити-Молл Белгородский	246	547	30.06.2016
22	Айсберг	236	52	30.06.2016
2	Европа	216	45	07.07.2016
4	Европа	234	47	29.06.2016
3	Санта-Моника	246	964	29.06.2016
23	БелГУ	216	85	30.06.2016

Добавление заказа

Заказчик: БелГУ

Объем работ: _____

Материал: 216

Сроки выполнения: _____

Новый заказчик

Добавить

Очистить форму

Удаление заказа

Номер заказа: 23

Удалить

Рисунок 3.25 – Форма «Удаление заказа»

Результат удаления заказа представлен на рисунке 3.26:

Рабочее место сотрудника

ВИА
фасадная компания

Добавление заказа | Обработка заказа | Мониторинг | Добавление нового материала

Номер заказа	Заказчик	Материал	Объем работ (м.кв.)	Сроки выполнения
1	Сити-Молл Белгородский	246	547	30.06.2016
22	Айсберг	236	52	30.06.2016
2	Европа	216	45	07.07.2016
4	Европа	234	47	29.06.2016
3	Санта-Моника	246	964	29.06.2016

Добавление заказа

Заказчик: БелГУ

Объем работ: _____

Материал: 216

Сроки выполнения: _____

Новый заказчик

Добавить

Очистить форму

Удаление заказа

Номер заказа: 23

Удалить

Рисунок 3.26 – Результат удаления заказа

Заполнение формы изменения статуса заказа показано на рисунке 3.27:

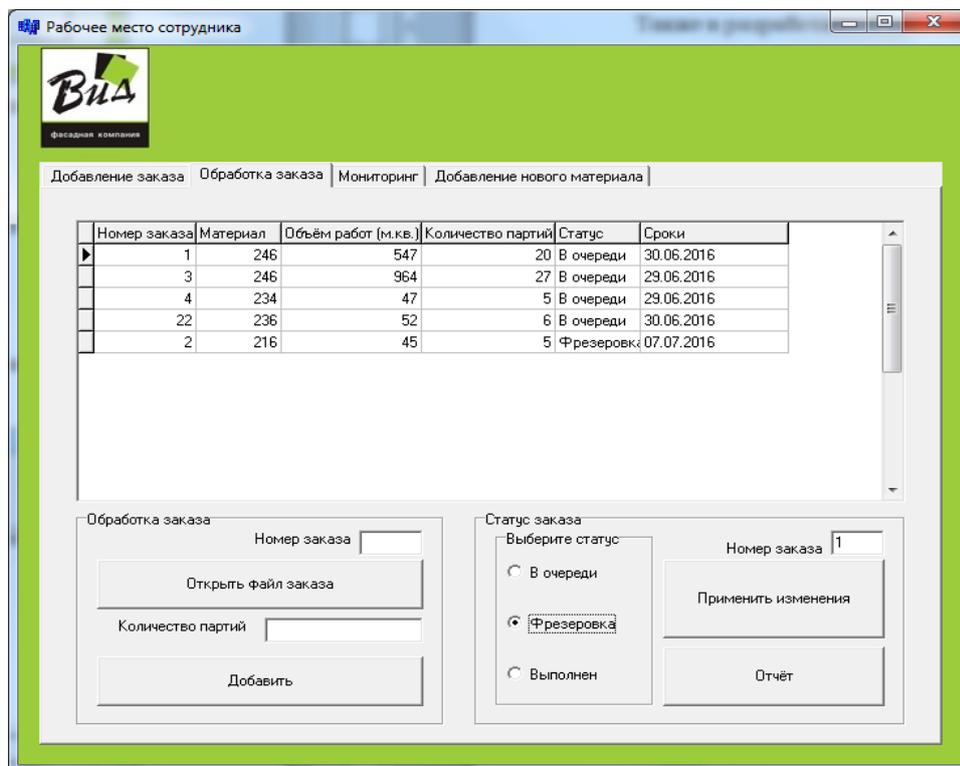


Рисунок 3.27 - Заполнение формы изменения статуса заказа

Также на форме изменения статуса заказа расположена функция формирования отчёта по обработанным заказам.

На рисунке 3.28 отражена работа данного отчёта:

Номер заказа	Материал	Сроки	Статус	Объём работ (м.кв.)
1	246	30.06.2016	В очереди	547
3	246	29.06.2016	В очереди	964
4	234	29.06.2016	В очереди	47
22	236	30.06.2016	В очереди	52
2	216	07.07.2016	Фрезеровка	45

Рисунок 3.28 – Результат формирования отчёта по обработанным заказам

Работа функции документирования выполненной партии деталей отображена на рисунках 3.29 – 3.31:

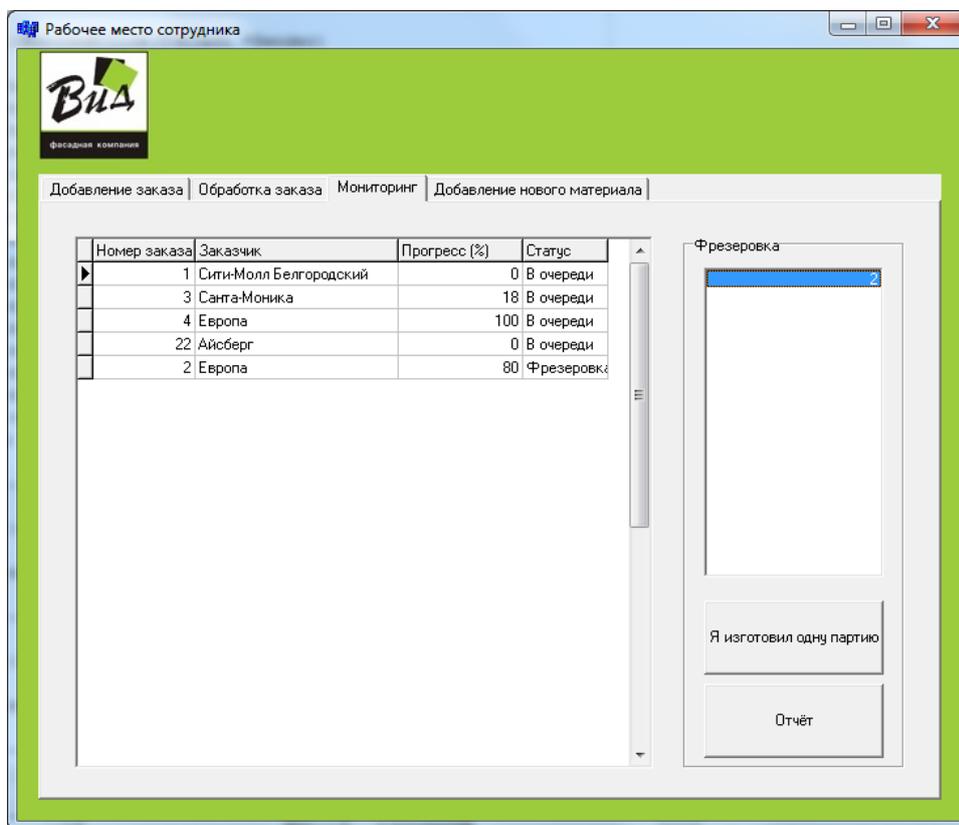


Рисунок 3.29 – Заполнение формы

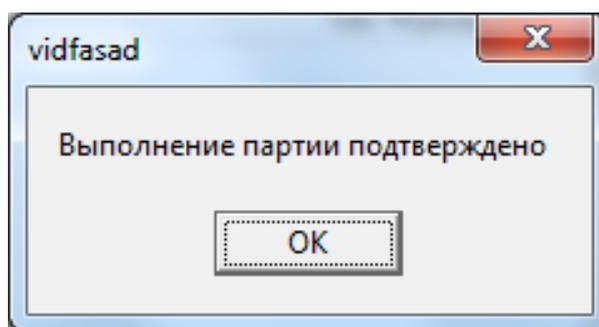


Рисунок 3.30 – Уведомление о выполнении функции

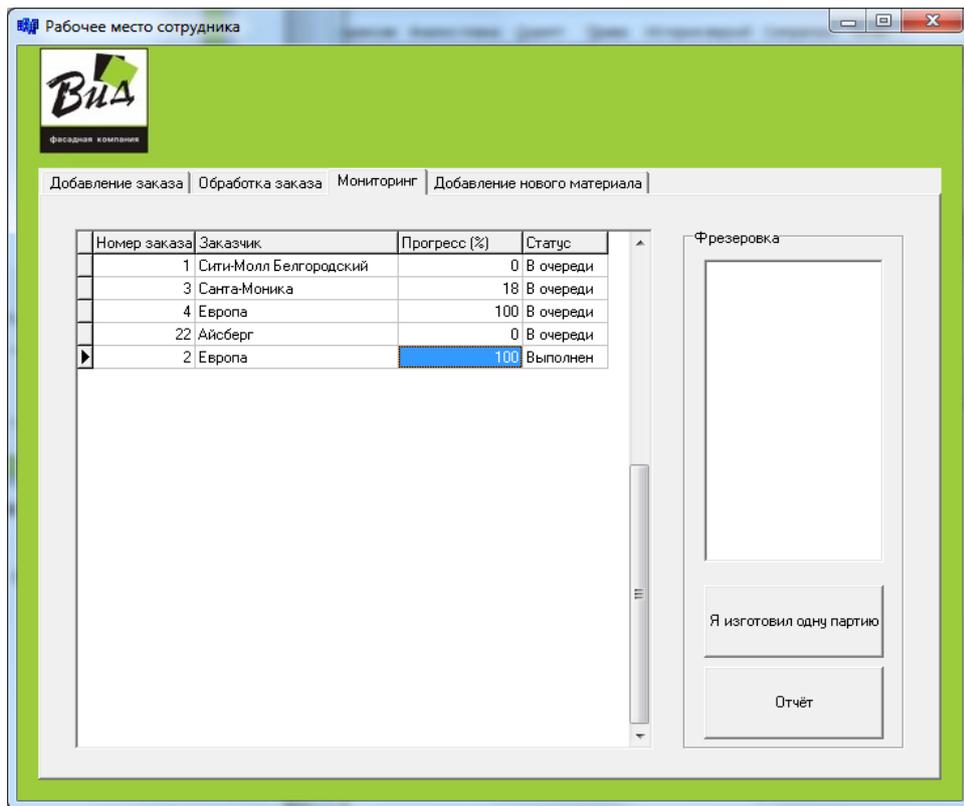


Рисунок 3.31 – Результат выполнения функции

В разработанной автоматизированной системе существует возможность формирования отчета по статусу заказов. На рисунке 3.32 представлен сформированный отчет:



Рисунок 3.32 – Сформированный отчет по статусу заказов

Заполнение формы документа «Материал» и результат добавления представлены на рисунка 3.33 – 3.34:

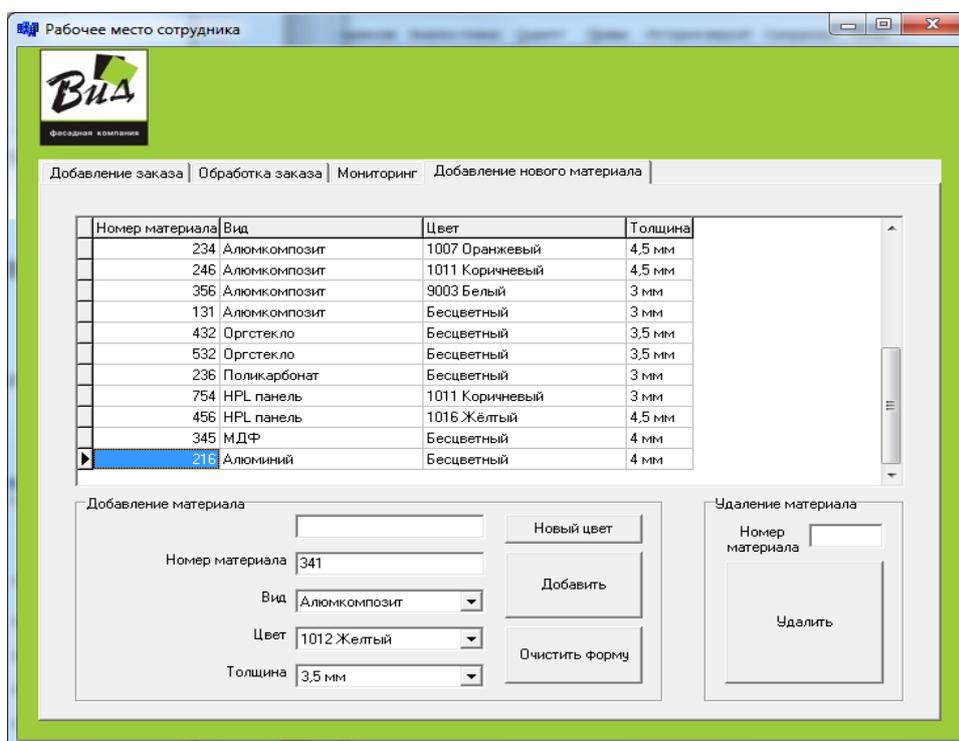


Рисунок 3.33 – Заполненная форма документа «Материал»

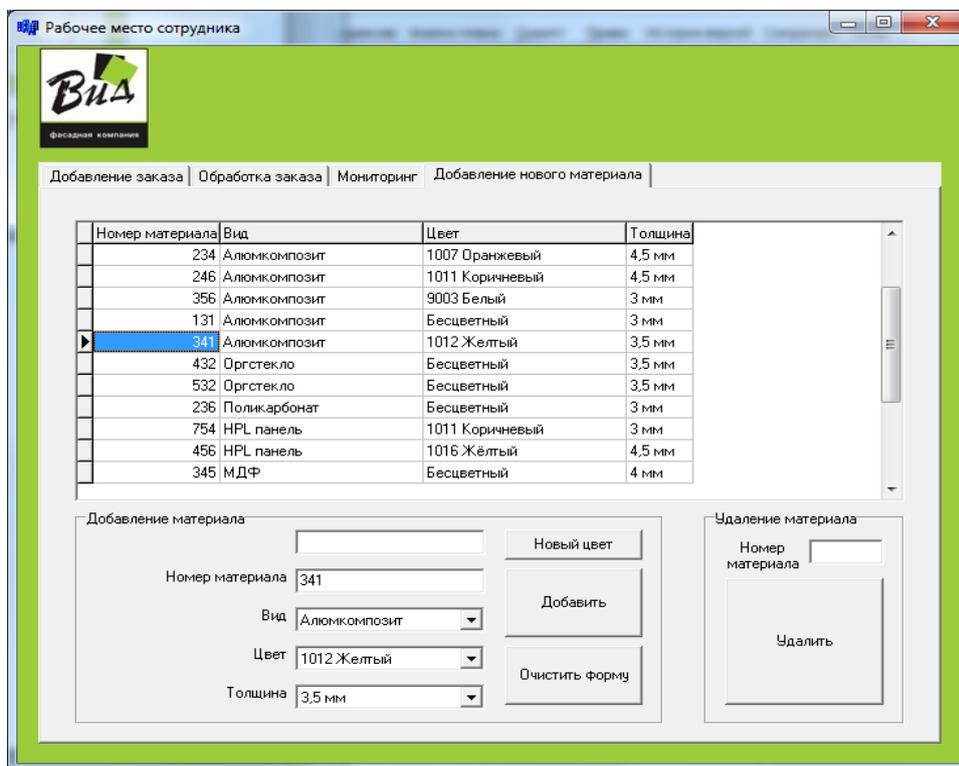


Рисунок 3.34– Результат заполнения формы

На рисунке 3.35 – 3.36 представлены форма для удаления материала из документа и результат данной функции:

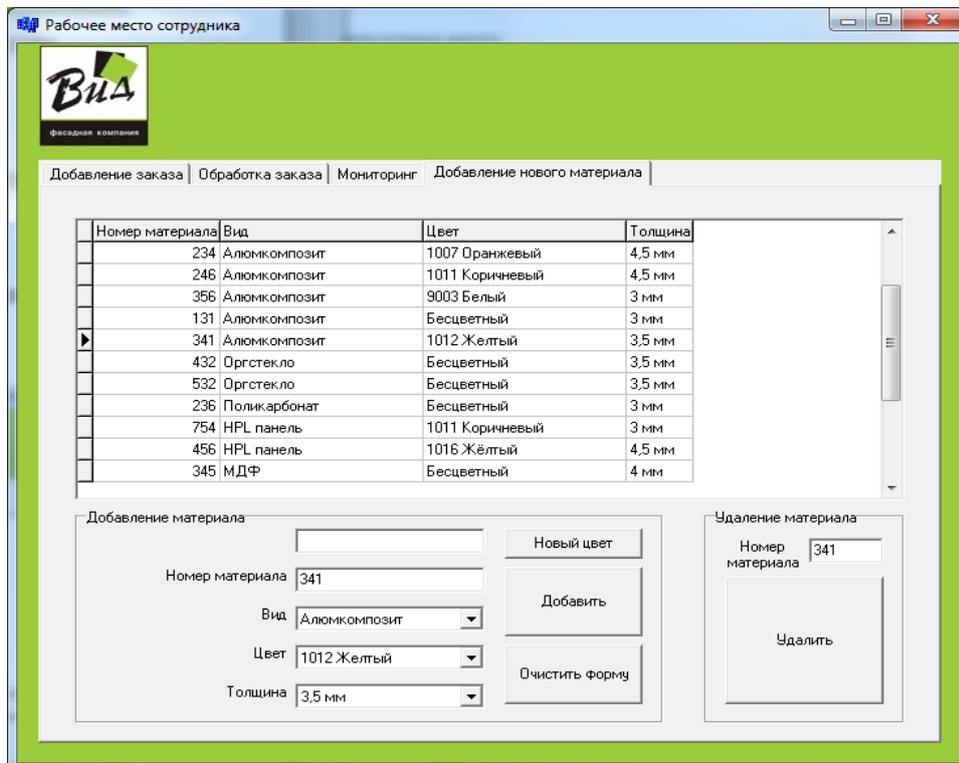


Рисунок 3.35 – Форма «Удаление материала»

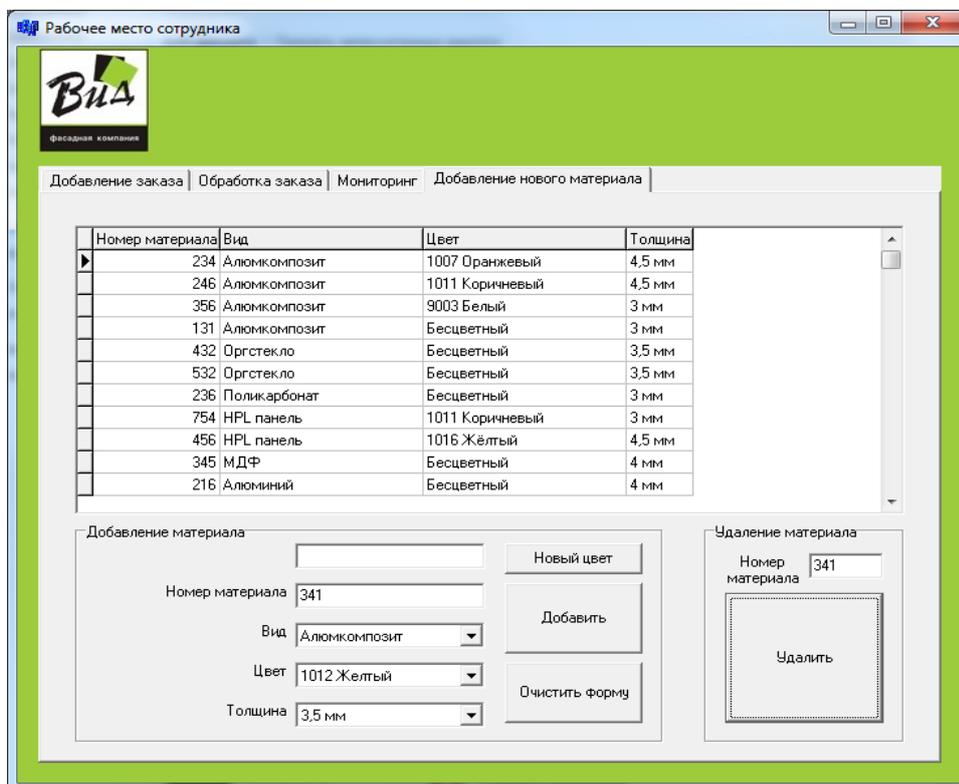


Рисунок 3.36 – Результат удаления материала

В разделах «Добавление заказа» и «Добавление нового материала» представлена функция «Очистить форму», работоспособность которой, на примере раздела «Добавление нового материала», отображена на рисунке 3.37

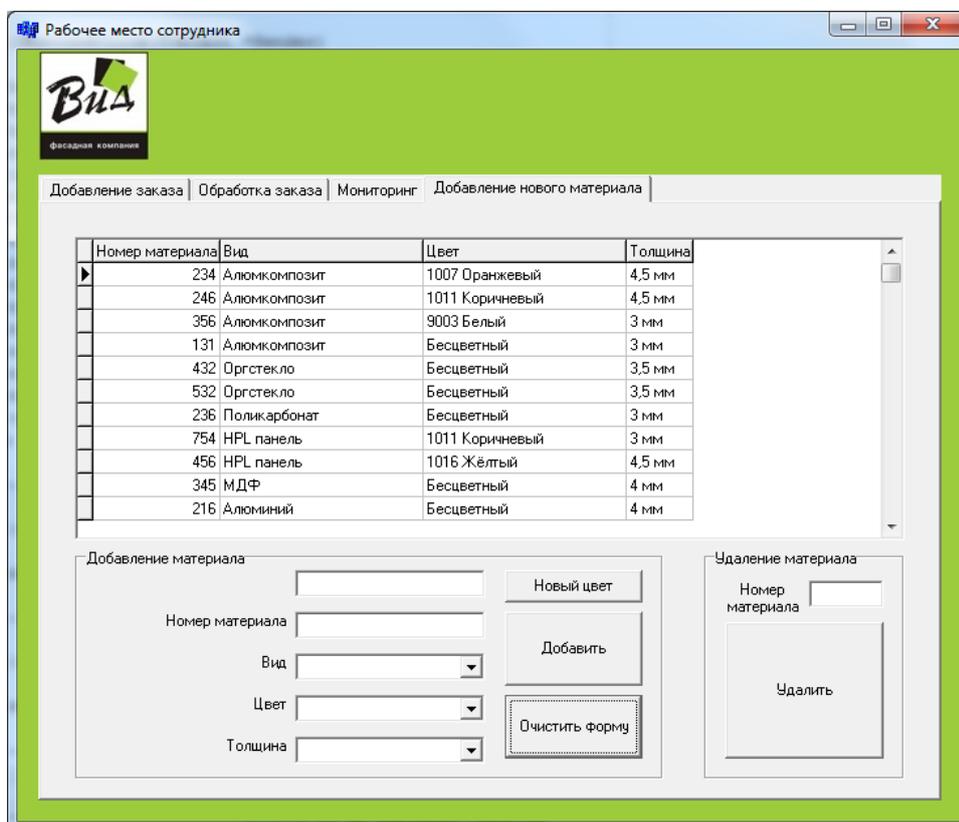


Рисунок 3.36 – Результат очистки формы «Добавление материала»

В процессе ввода тестовых данных было выявлено, что все модули программы работают корректно и без ошибок.

3.5 Расчет экономической эффективности

При проектировании автоматизированной информационной подсистемы учёта производственного процесса фасадной компании «Вид» были определены следующие этапы разработки, а также рассчитано время, затрачиваемое на каждый этап:

- изучение предметной области (10);

- подбор и изучение справочной литературы (15);
- разработка алгоритма и структуры программы (15);
- программирование (25);
- тестирование программного обеспечения (ПО) (10).

Для осуществления работ по каждому этапу определяем состав специалистов: программист и тестировщик.

Затраты на разработку системы подразделяем на капитальные или единовременные и эксплуатационные или текущие. Расчет выполним по отдельным статьям:

- прямые материальные затраты;
- фонд оплаты труда;
- отчисления на социальные нужды;
- амортизационные отчисления;
- накладные расходы;
- прочие расходы.

Расчет материальных затрат приведен в таблице 3.3.

Таблица 3.3 – Расчет прямых материальных затрат

Материалы	Ед. измер.	Кол- во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	Лист	500	0,5	250
Картридж для принтера	шт.	1	1200	1200
Компакт-диск	шт.	5	30	150
Расходы на электроэнергию	кв/ч	8,5	2,47	20,38
Итого				1620,38

К прямым затратам были отнесены:

- бумага;
- картридж для принтера;
- компакт-диск;

– расходы на электроэнергию.

Прямые материальные затраты составили 1620,38 рублей.

Для расчета фонда оплаты труда необходимо рассчитать заработную плату разработчиков подсистемы, а также составить баланс рабочего времени. Данные сведем в таблице 3.4.

Таблица 3.4 – Баланс рабочего времени.

П/п	Наименование показателей	ИТР
1	Число календарных дней в году	365
2	Число выходных и нерабочих дней в году	118
3	Число рабочих дней в году	247
4	Невыходы на работу:	
	А) по болезни	0
	Б) очередной отпуск	0
5	Фактическое число рабочих дней в году	253
6	Продолжительность рабочего дня	8
7	Годовой фонд рабочего времени (час.)	1976

Часовую ставку заработной платы ($Чс$) определяем по формуле:

$$Чс = (З * п * к) / \Phi, \quad (3.1)$$

где $Чс$ - месячная зарплата, руб.;

$П$ - число месяцев в году, исключая отпуск;

$К$ - коэффициент, учитывающий премии из фонда зарплаты;

Φ - фактический годовой фонд рабочего времени, час.

Среднемесячная зарплата программиста и тестировщика составляет 15000 руб.

При $З=15000$ руб., $п = 11$, а $\Phi = 1976$ час., $К=1,15$ получим $Чс = 96,03$ руб./час.

Размер основной заработной платы определяем исходя из времени, затрачиваемого на выполнение работ и стоимости часа работы исполнителя.

Основная заработная плата определяется по графику основных этапов работ. Дополнительная заработная плата может составить до 15% от основной.

Расчет фонда оплаты труда приведен в таблице 3.5.

Таблица 3.5 – Расчет фонда оплаты труда.

Этапы разработки	Время (час)	Часовая ставка (руб.)	Сумма (руб.)
Изучение предметной области	10	96,03	960,3
Подбор, изучение литературы	15	96,03	1440,45
Разработка алгоритма и структуры программы	15	96,03	1440,45
Программирование	25	96,03	2400,75
Тестирование ПО	10	96,03	960,3
Основная заработная плата(итого)			7202,25
Дополнительная заработная плата			1080,34
Итого			8282,59

Размер отчислений на социальные нужды определяется исходя из размера фонда оплаты труда.

Расчет приведен в таблице 3.6.

Таблица 3.6 – Отчисления на социальные нужды.

Отчисления	Доля от фонда оплаты труда (%)	Сумма (руб.)
В пенсионный фонд	22	1822,17
В фонд занятости	1,5	124,24
Медицинское страхование	5,1	422,41
Социальное страхование	2,9	240,2
Итого	31,5	2609,02

Полную смету затрат на разработку системы приведем в таблице 3.7.

Таблица 3.7 – Расчет затрат на разработку ИС.

Статья расхода	Сумма (руб.)
Фонд оплаты труда	8282,59
Отчисления на социальные нужды	2609,02
Материальные затраты	1620,38
Амортизационные отчисления	0
Прочие расходы	0
Накладные расходы	0
Итого	12511,99

Расчет ежемесячных затрат на эксплуатацию системы приведем в таблице 3.7.

Таблица 3.7 – Расчет затрат на эксплуатацию системы

Статья расхода	Сумма (руб.)
Зарплата администратора БД	3000
Отчисления на социальные нужды	945
Затраты на электроэнергию	47,82
Итого	3992,82

Для оценки инвестиционного проекта автоматизированной информационной подсистемы учёта производственного процесса необходимо рассчитать планируемые поступления денежных средств от реализации услуг по месяцам. Расчет приведем в таблице 3.8

Таблица 3.8 – Расчет планируемых поступлений

Месяц	Увеличение объема продаж, %	Сумма, руб.	Затраты, руб.	CF, руб.	$cF_t^{(\Sigma)}$ руб.
1	2	3	4	5	6
1	0	0	3992,82	-3992,82	-3992,82
2	0,1	1500	3992,82	-2492,82	-6485,64
3	0,2	3000	3992,82	-992,82	-7478,46

Продолжение таблицы 3.8

4	0,2	3000	3992,82	-992,82	-8471,28
5	0,3	4500	3992,82	507,18	-7964,1
6	0,4	6000	3992,82	2007,18	-5956,92
7	0,4	6000	3992,82	2007,18	-3949,74
8	0,5	7500	3992,82	3507,18	-442,56
9	0,6	9000	3992,82	5007,18	4564,62
10	0,7	10500	3992,82	6507,18	11071,8
11	0,8	12000	3992,82	8007,18	19078,98
12	0,8	12000	3992,82	8007,18	27086,16

Далее необходимо провести оценку инвестиционного проекта подсистемы учёта.

Первым этапом является расчет NPV (чистой текущей стоимости).

$$NPV = \sum_{t=1}^n \frac{cF_t}{k + q} - I_0, \text{ где} \quad (3.2)$$

NPV – чистая текущая стоимость инвестиций;

CF – поступление денежных средств в конце t- ого периода;

q- банковская ставка;

I – стоимость реализации инвестиционного проекта (инвестиции).

$$NPV = 27086,16 / (1 + 0,11) - 12511,99 = 11889,96$$

Для реализации данного проекта NPV составляет 11889,96 руб. Положительное значение данного показателя указывает на то, что вложение капитала в данный проект является эффективным.

Следующим этапом является расчет рентабельности инвестиций. Расчет рентабельности проводится по следующей формуле:

$$PI = \left[\sum_{t=1}^n \frac{cF_t}{(1+q)^t} \right] / \left[\sum_{t=1}^n \frac{I_t}{(1+q)^t} \right], \text{ где} \quad (3.3)$$

PI – рентабельность инвестиций.

$$PI = (27086,16 / (1 + 0,11)) / 12511,99 = 1,95 = 195\%$$

Данный показатель позволяет определить в какой мере возрастает ценность фирмы в расчете на 1 руб. Для реализации данного проекта по разработке подсистемы учёта рентабельность составляет 195%.

Далее необходимо рассчитать период окупаемости проекта. Расчет окупаемости проводится по следующей формуле:

$$PP = \frac{I_0}{cF_t^{(\Sigma)}}, \quad (3.4)$$

где PP – период окупаемости (лет);

I_0 – первоначальные инвестиции;

$cF_t^{(\Sigma)}$ – годовая сумма денежных поступлений от реализации инвестированного проекта.

$$PP = 12511,99 / (27086,16 / (1 + 0,11)) = 0,51 = 27 \text{ нед.}$$

Проведя расчет периода окупаемости было выявлено, данный показатель составляет 27 недель.

Исходя из полученных результатов можно сделать вывод, что затраты на разработку и внедрение системы окупятся спустя 27 недель со дня введения системы в действие, чистая текущая стоимость инвестиций имеет положительное значение, рентабельность составляет 195 %. Внедрение системы позволяет увеличить количество продаж, уменьшается время заполнения документации.

3.6 SWOT-анализ разработки

SWOT-анализ — это определение сильных и слабых сторон созданной программной разработки, а также возможностей и угроз, исходящих из его ближайшего окружения (внешней среды).

Сильные стороны (Strengths) — преимущества разработки.

Слабости (Weaknesses) — недостатки разработки.

Возможности (Opportunities) — факторы внешней среды, использование которых создаст преимущества программной разработки на рынке.

Угрозы (Threats) — факторы, которые могут потенциально ухудшить положение разработки на рынке [10].

Проанализировать возможности развития автоматизированной информационной подсистемы учёта производственного процесса поможет SWOT-матрица, в которой приводится влияние каждого преимущества или недостатка разработки на существующие возможности развития и возникающие при этом угрозы.

В таблице 3.9 приведена SWOT-матрица разработки.

Таблица 3.9 - SWOT-матрица.

Факторы	Возможности		Угрозы		Итого
	Расширение круга потребителей	Совершенствование разработки	Появление новых конкурентов	Быстрое моральное устаревание	
1	2	3	4	5	6
Сильные стороны					
Низкая стоимость разработки	++	+	++	0	+5
Многофункциональность	+	+	+	+	+4
Обеспечение сопровождения	+	+	+	+	+4
Итого	+4	+3	+4	+2	+13
Слабые стороны					
Недостаточное финансирование	--	--	-	-	-6
Нехватка квалифицированных кадров	--	--	0	-	-5
Итого	-4	-4	-1	-2	-11
Общий итог	0	-1	+3	0	+2

Проанализировав полученную SWOT-матрицу, можно сделать следующие выводы.

Самой сильной стороной разработки является ее низкая стоимость. В дальнейшем необходимо обращать особое внимание на обеспечение и расширение этой стороны разработки.

Все выделенные слабые стороны разработки являются достаточно важными и необходимо принять меры по их устранению.

Из рассмотренных возможностей более реальной представляется возможность расширения круга возможных потребителей, хотя данная возможность при существующих слабостях весьма проблематична. Совершенствование же разработки при недостатке финансирования и нехватке кадров вообще невозможно. Наиболее оптимальным решением этой проблемы является увеличение числа квалифицированных кадров.

Наиболее опасной угрозой представляется быстрое устаревание разработки, но при сложившихся условиях это не столь существенная угроза. Появление конкурентов представляется маловероятным вследствие наличия сильных сторон разработки.

В настоящее время имеются некоторые трудности в дальнейшем развитии разработки, но, тем не менее, разработка является достаточно перспективной, об этом говорит оценка «+4». Первоочередной задачей для совершенствования программного продукта является повышение квалификации кадров.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была достигнута цель и решены поставленные задачи:

- Исследована деятельность производственного цеха организации ООО «Вид»;
- Построена функциональная модель производственного цеха организации ООО «Вид»;
- Проанализированы существующие автоматизированные информационные подсистемы учёта производственного процесса;
- Описаны технологии проектирования автоматизированной информационной подсистемы учёта производственного процесса;
- Даны техническое, технологическое, информационное и программное обоснование проектных решений;
- Дано обоснование экономической эффективности проекта.

Результатом проделанной работы является автоматизированная информационная подсистема системы учёта производственного процесса, которая выполняет функции сбора, хранения и обработки информации.

При выполнении ВКР приобретены навыки практического решения информационных задач в качестве разработчика информационной системы.

В качестве инструментальных средств, используемых при создании и проектировании информационной системы, применялись клиент-серверная СУБД Firebird и утилита IVExpert для разработки базы данных, программный продукт Borland C++ Builder 6.0, предназначенный для разработки пользовательского интерфейса автоматизированной подсистемы.

Разработанная автоматизированная информационная подсистема системы учёта производственного процесса позволяет повысить оперативность и производительность труда сотрудников производственного цеха фасадной компании «Вид». Разработанная система позволяет синхронизировать используемые данные и сократить бумажные архивы, а также предоставляет

полную картину о производстве в цеху.

Разработанное программное средство имеет удобный и интуитивно понятный интерфейс взаимодействия с пользователем, позволяет повысить качество обработки информации, ее достоверность и надежность. В системе предусмотрена возможность формирования отчетов и использования справочников, что позволяет своевременно и оперативно выявлять необходимость в тех или иных видах товаров. Разработанная автоматизированная система соответствует требованиям, предъявляемым к современным программным продуктам.

В итоге разработанный программный продукт позволяет выполнять все задачи, необходимые для эффективного осуществления деятельности по учету производственного процесса в организации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архангельский, А.Я. С++Builder 6 Справочное пособие. Книга 2. Классы и компоненты[Текст] /А.Я Архангельский – М.: Бином-Пресс, 2004. – 528 с.
2. Балдин, К.В. Информационные системы в экономике. Учебное пособие[Текст] / К.В. Балдин, В.Б. Уткин –М.: Дашков и К, 2008. – 395 с.
3. Бобровский, С.И. Технологии С++Builder.Разработка приложений для бизнеса [Текст] / С.И. Бобровский -СПб.: Издательство Питер, 2007. – 1104с.
4. Бондарь, А.Г. Практическое руководство для умных пользователей и начинающих разработчиков[Текст] / А.Г. Бондарь-СПб.: БХВ-Петербург, 2007. – 592 с.
5. Борри, Х.Firebird: руководство разработчика баз данных: Пер. с англ.[Текст] / Х. Борри – СПб.: БХВ-Петербург, 2006. - 1104 с.
6. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем [Текст] / А.М. Вендров - М.: Финансы и статистика, 2000. – 352 с.
7. Вендров, А.М. Современные методы и средства проектирования информационных систем [Текст] / А.М. Вендров - М.: Финансы и статистика, 2008. – 65 с.
8. Голицына, О.Л. Программное обеспечение [Текст] / О. Л. Голицына, И. И. Попов, Т. Л. Партыка. –М.: Форум, 2013.–448с.
9. Гультяев, А.К. Проектирование и дизайн пользовательского интерфейса[Текст] / А.К. Гультяев - СПб.: КОРОНАпринт, 2000. - 349 с.
10. Евдокимова, В.В. Экономическая информатика[Текст] / В.В. Евдокимова -СПб.:Питерпаблишинг, 2008. – 468 с.

11. Иванова, Г.С. Объектно-ориентированное программирование: учебник для вузов[Текст] / Г.С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. - М.: МГТУ им. Н.Э. Баумана, 2003. - 368 с.
12. Илюшечкин, В.М. Основы использования и проектирования баз данных[Текст] / В.М. Илюшечкин - М.: Издательство Юрайт 2010. -213с.
13. Ипатова, Э.Р. Методологии и технологии системного проектирования информационных систем [Текст] / Э.Р. Ипатова, Ю.В. Ипатов.– М.: Флинта, 2008. – 256 с.
14. Карпова, Т.С. Базы данных: модели, разработка, реализация[Текст] / Т.С. Карпова – СПб.: Питер, 2002. – 304с.
15. Конноли, Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3–е издание[Текст] / Т. Конноли, К. Бегг - М.: Издательский дом "Вильямс", 2003. - 1440 с.
16. Культин, Н.Б. Самоучитель С++Builder[Текст] /Н.Б. Культин-СПб.: БХВ-Перербург,2004.-203с.
17. Лавров, С.С. Программирование. Математические основы, средства, теория: учебное пособие. [Текст] / С.С. Лавров - СПб.:БХВ-Петербург, 2001. - 320 с.
18. Левчук, Е.А. Технологии организации, хранения и обработки данных [Текст] /Е.А. Левчук.– Минск: Вышэйшая школа, 2007. – 240 с.
19. Маклаков, С.В. Создание информационных систем с AllFusionModelingSuite[Текст] / С.В. Маклаков– М.: Диалог-МИФИ, 2007. - 432с.
20. Маклаков, С.В. ВРwin, ERwin. CASE-средства разработки информационных систем[Текст] / С.В. Маклаков– М.: ДИАЛОГ-МИФИ, 2007. – 304с.
21. Мандел, Т. Разработка пользовательского интерфейса[Текст] /Т. Мандел– М.:ДМК Пресс, 2008. - 412 с.
22. Михелёв, В.М. Базы данных и СУБД: Учебное пособие[Текст] /В.М. Михелёв– Белгород: Издательство БелГУ, 2007.– 200с.

23. Муромцев, В.В. Проектирование информационных систем: Учебное пособие для студентов вузов заочной формы обучения по спец. 010502 "Прикладная информатика в экономике" [Текст] /В.В. Муромцев, В.А. Ломазов - Белгород: БелГУ, 2007. - 160 с.
24. Пахомов, Б.И. С/C++ и Borland C++ Builder для начинающих [Текст] /Б.И. Пахомов – СПб.: БХВ-Петербург, 2005. – 640 с.
25. Пахомов, Б.И. С\C++ и Borland C++ Builder для студента[Текст] /Б.И. Пахомов СПб.: БХВ-Петербург, 2006. – 448с.
26. Петров, В.Н. Информационные системы [Текст] /В.Н. Петров– СПб.: Питер, 2002. – 688 с.
27. Послед,Б.С. Borland C++ Builder 6. Разработка приложений баз данных[Текст] /Б.С. Послед – СПб.: ООО «ДиаСофтЮП», 2003 –320 с.
28. Смирнова, Г.Н. Проектирование экономических информационных систем. Учебное пособие[Текст] / Г.Н. Смирнова, А.А. Сорокин–М.: Высшая школа, 2002.– 428 с.
29. Титоренко, Г.А. Автоматизированные информационные технологии в экономике: учебное пособие. [Текст] /Г.А. Титоренко, 2003 г. – 245 с.
30. Маклаков, С.В ВРwin и ERwin. CASE – средства разработки информационных систем[Текст] / С.В. Маклаков – Изд.: Диалог–Мифи – 2014.– 295 с.
31. Федоров, Н.В. Проектирование информационных систем на основе современных CASE-технологий: учебное пособие. [Текст] / Н. В. Федоров.- МГИУ, 2008.-128 с.
32. Федорова, Е.Н. Теоретические основы программирования: учебное пособие. [Текст] / Е. Н. Федорова.- МГИУ, 2012.-214 с.
33. Фельдман, Я.А. Создаем информационную систему [Текст] / Я.А. Фельдман. –М.: Солон-Пресс, 2007.–120с.
34. Хомоненко, А.Д. Базы данных: учебник для высших учебных заведений, 4-е издание дополненное и переработанное[Текст] / А.Д. Хомоненко– СПб.: Корона, 2004. – 736 с.

35. Хомоненко, А.Д. Работа с базами данных в C++ Builder[Текст] / А.Д. Хомоненко, С.Е. Ададуров-СПб.:БХВ-Петербург, 2006.-496 с.
36. Черемных, С.В. Структурный анализ систем: IDEF-технологии[Текст] / С.В. Черемных, И.О. Семенов, В.С. Ручкин– М: Финансы и статистика, 2001. – 208с.
37. 1С:Предприятие 8. [Электронный ресурс] / Режим доступа: <http://solutions.1c.ru/catalog/wms/>
38. Основы проектирования реляционных баз данных. [Электронный ресурс] / Режим доступа: http://www.intuit.ru/goods_store/ebooks/8322/
39. Ресурсы информационных систем. [Электронный ресурс] / Режим доступа: <http://www.economica-upravlenie.ru/content/view/204/>
40. Сертифицированные информационные системы[Электронный ресурс] / Режим доступа: <http://certsys.ru/products/index.php>

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Фрагмент программного кода базы данных

```
CREATE GENERATOR GEN_T_ORDER_ID;

CREATE TABLE T_ORDER (
    NOM_O D_NOM NOT NULL /* D_NOM = SMALLINT NOT NULL */,
    NOM_K D_NOM /* D_NOM = SMALLINT NOT NULL */,
    NOM_M D_NOM /* D_NOM = SMALLINT NOT NULL */,
    FOT_O D_PART /* D_PART = SMALLINT */,
    DAT_O D_DATE /* D_DATE = TIMESTAMP */,
    NOW_O D_PART /* D_PART = SMALLINT */,
    ALL_O D_PART /* D_PART = SMALLINT */,
    PRS_O D_PR /* D_PR = NUMERIC(5,0) */,
    NOM_S D_NOM /* D_NOM = SMALLINT NOT NULL */
);

ALTER TABLE T_ORDER ADD CONSTRAINT PK_T_ORDER PRIMARY KEY (NOM_O);

ALTER TABLE T_ORDER ADD CONSTRAINT FK_T_ORDER_1 FOREIGN KEY (NOM_K) REFERENCES
T_KLIENT (NOM_K);
ALTER TABLE T_ORDER ADD CONSTRAINT FK_T_ORDER_2 FOREIGN KEY (NOM_M) REFERENCES
T_MATERIAL (NOM_M);
ALTER TABLE T_ORDER ADD CONSTRAINT FK_T_ORDER_3 FOREIGN KEY (NOM_S) REFERENCES
T_STATUS (NOM_S);

SET TERM ^ ;

/* Trigger: T_ORDER_BI */
CREATE OR ALTER TRIGGER T_ORDER_BI FOR T_ORDER
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
    IF (NEW.NOM_O IS NULL) THEN
        NEW.NOM_O = GEN_ID(GEN_T_ORDER_ID,1);
END
^

SET TERM ; ^

);
SET TERM ^ ;

create or alter procedure ADD_O (
    N_K D_NOM,
    FOT D_PART,
    N_M D_NOM,
    DAT D_DATE)
as
begin
insert into t_order (t_order.nom_k, t_order.fot_o, t_order.nom_m, t_order.dat_o,
t_order.nom_s, t_order.now_o)
values (:n_k,:fot,:n_m,:dat, 1, 0);
suspend;
end^

SET TERM ; ^
```

```

GRANT INSERT ON T_ORDER TO PROCEDURE ADD_O;

GRANT EXECUTE ON PROCEDURE ADD_O TO SYSDBA;

SET TERM ^ ;

create or alter procedure DEL_O (
    N_O D_NOM)
as
begin
delete from t_order
wheret_order.nom_o=:n_o;
suspend;
end^

SET TERM ; ^

GRANT SELECT,DELETE ON T_ORDER TO PROCEDURE DEL_O;

GRANT EXECUTE ON PROCEDURE DEL_O TO SYSDBA;
SET TERM ^ ;

create or alter procedure UPD_ALL (
    N_O D_NOM,
    A_O D_PART)
as
begin
updatet_order set t_order.all_o=:a_o
wheret_order.nom_o=:n_o;
suspend;
end^

SET TERM ; ^

GRANT SELECT,UPDATE ON T_ORDER TO PROCEDURE UPD_ALL;

GRANT EXECUTE ON PROCEDURE UPD_ALL TO SYSDBA;
SET TERM ^ ;

create or alter procedure UPD_NOW (
    N_O D_NOM)
as
begin
updatet_order set t_order.now_o=t_order.now_o+1
wheret_order.nom_o=:n_o;
suspend;
end^

SET TERM ; ^

GRANT SELECT,UPDATE ON T_ORDER TO PROCEDURE UPD_NOW;

GRANT EXECUTE ON PROCEDURE UPD_NOW TO SYSDBA;
SET TERM ^ ;

create or alter procedure UPD_S (
    N_O D_NOM,
    N_S D_NOM)
as
begin
updatet_order set t_order.nom_s=:n_s
wheret_order.nom_o=:n_o;
suspend;
end^

```

ПРИЛОЖЕНИЕ Б

Фрагмент программного код клиентского Win-приложения

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
#include "Unit2.h"  
#include "Unit3.h"  
#include "Unit4.h"  
//-----  
#pragma package (smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1 (TComponent* Owner)  
    : TForm (Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::Button3Click (TObject *Sender)  
{  
ShellExecute (Handle, "open", "c:\\\\vidfasad\\", NULL, NULL, SW_RESTORE);  
}  
//-----  
void __fastcall TForm1::Button8Click (TObject *Sender)  
{  
DataModule2->IBStoredProc1->StoredProcName="ADD_K";  
DataModule2->IBStoredProc1->ParamByName ("nam")->AsString=Edit7->Text;  
DataModule2->IBStoredProc1->Prepare ();  
DataModule2->IBStoredProc1->ExecProc ();  
DataModule2->IBTransaction1->Commit ();  
DataModule2->IBQuery5->Close ();  
DataModule2->IBQuery1->Open ();  
DataModule2->IBQuery2->Open ();  
DataModule2->IBQuery3->Open ();  
DataModule2->IBQuery4->Open ();  
DataModule2->IBQuery5->Open ();  
DataModule2->IBQuery6->Open ();  
DataModule2->IBQuery7->Open ();  
DataModule2->IBQuery8->Open ();  
DataModule2->IBQuery9->Open ();  
DataModule2->IBQuery10->Open ();  
}  
//-----  
  
void __fastcall TForm1::Button1Click (TObject *Sender)  
{  
DataModule2->IBStoredProc2->StoredProcName="ADD_O";  
DataModule2->IBStoredProc2->ParamByName ("n_k")->AsString=DBLookupComboBox1->ListSource->DataSet->FieldByName ("nom_k")->AsString;  
DataModule2->IBStoredProc2->ParamByName ("fot")->AsString=Edit1->Text;  
DataModule2->IBStoredProc2->ParamByName ("n_m")->AsString=DBLookupComboBox5->ListSource->DataSet->FieldByName ("nom_m")->AsString;  
DataModule2->IBStoredProc2->ParamByName ("dat")->AsString=Edit2->Text;  
DataModule2->IBStoredProc2->Prepare ();  
DataModule2->IBStoredProc2->ExecProc ();
```

```

DataModule2->IBTransaction1->Commit ();
DataModule2->IBQuery1->Close ();
DataModule2->IBQuery1->Open ();
DataModule2->IBQuery2->Open ();
DataModule2->IBQuery3->Open ();
DataModule2->IBQuery4->Open ();
DataModule2->IBQuery5->Open ();
DataModule2->IBQuery6->Open ();
DataModule2->IBQuery7->Open ();
DataModule2->IBQuery8->Open ();
DataModule2->IBQuery9->Open ();
DataModule2->IBQuery10->Open ();
}
//-----

void __fastcall TForm1::Button7Click(TObject *Sender)
{
Edit1->Clear ();
Edit2->Clear ();
Edit3->Clear ();
Edit7->Clear ();
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
DataModule2->IBStoredProc3->StoredProcName="DEL_O";
DataModule2->IBStoredProc3->ParamByName ("n_o") ->AsString=Edit3->Text;
DataModule2->IBStoredProc3->Prepare ();
DataModule2->IBStoredProc3->ExecProc ();
DataModule2->IBTransaction1->Commit ();
DataModule2->IBQuery1->Close ();
DataModule2->IBQuery1->Open ();
DataModule2->IBQuery2->Open ();
DataModule2->IBQuery3->Open ();
DataModule2->IBQuery4->Open ();
DataModule2->IBQuery5->Open ();
DataModule2->IBQuery6->Open ();
DataModule2->IBQuery7->Open ();
DataModule2->IBQuery8->Open ();
DataModule2->IBQuery9->Open ();
DataModule2->IBQuery10->Open ();
}
//-----

void __fastcall TForm1::Button4Click(TObject *Sender)
{
DataModule2->IBStoredProc4->StoredProcName="UPD_ALL";
DataModule2->IBStoredProc4->ParamByName ("n_o") ->AsString=Edit4->Text;
DataModule2->IBStoredProc4->ParamByName ("a_o") ->AsString=Edit5->Text;
DataModule2->IBStoredProc4->Prepare ();
DataModule2->IBStoredProc4->ExecProc ();
DataModule2->IBTransaction1->Commit ();
DataModule2->IBQuery2->Close ();
DataModule2->IBQuery1->Open ();
DataModule2->IBQuery2->Open ();
DataModule2->IBQuery3->Open ();
DataModule2->IBQuery4->Open ();
DataModule2->IBQuery5->Open ();
DataModule2->IBQuery6->Open ();
DataModule2->IBQuery7->Open ();
DataModule2->IBQuery8->Open ();
DataModule2->IBQuery9->Open ();
DataModule2->IBQuery10->Open ();
}

```

```

}
//-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{
DataModule2->IBStoredProc5->ParamByName("n_o")->AsString=Edit6->Text;
if (RadioGroup1->ItemIndex==0){DataModule2->IBStoredProc5->ParamByName("n_s")->
>AsString="1";}
else if (RadioGroup1->ItemIndex==1){DataModule2->IBStoredProc5->
>ParamByName("n_s")->AsString="2";}
else if (RadioGroup1->ItemIndex==2){DataModule2->IBStoredProc5->
>ParamByName("n_s")->AsString="3";};
DataModule2->IBStoredProc5->Prepare();
DataModule2->IBStoredProc5->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->IBQuery2->Close();
DataModule2->IBQuery1->Open();
DataModule2->IBQuery2->Open();
DataModule2->IBQuery3->Open();
DataModule2->IBQuery4->Open();
DataModule2->IBQuery5->Open();
DataModule2->IBQuery6->Open();
DataModule2->IBQuery7->Open();
DataModule2->IBQuery8->Open();
DataModule2->IBQuery9->Open();
DataModule2->IBQuery10->Open();
}
//-----

void __fastcall TForm1::Button10Click(TObject *Sender)
{
DataModule2->IBStoredProc6->StoredProcName="ADD_M";
DataModule2->IBStoredProc6->ParamByName("n_m")->AsString=Edit8->Text;
DataModule2->IBStoredProc6->ParamByName("n_v")->AsString=DBLookupComboBox2->
>ListSource->DataSet->FieldByName("NOM_V")->AsString;
DataModule2->IBStoredProc6->ParamByName("n_c")->AsString=DBLookupComboBox3->
>ListSource->DataSet->FieldByName("nom_c")->AsString;
DataModule2->IBStoredProc6->ParamByName("n_t")->AsString=DBLookupComboBox4->
>ListSource->DataSet->FieldByName("nom_t")->AsString;
DataModule2->IBStoredProc6->Prepare();
DataModule2->IBStoredProc6->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->IBQuery7->Close();
DataModule2->IBQuery1->Open();
DataModule2->IBQuery2->Open();
DataModule2->IBQuery3->Open();
DataModule2->IBQuery4->Open();
DataModule2->IBQuery5->Open();
DataModule2->IBQuery6->Open();
DataModule2->IBQuery7->Open();
DataModule2->IBQuery8->Open();
DataModule2->IBQuery9->Open();
DataModule2->IBQuery10->Open();
}
//-----

void __fastcall TForm1::Button11Click(TObject *Sender)
{
Edit8->Clear();
Edit10->Clear();
Edit11->Clear();
}
//-----

```

```

void __fastcall TForm1::Button13Click(TObject *Sender)
{
    DataModule2->IBStoredProc7->StoredProcName="NEW_PROCEDURE";
    DataModule2->IBStoredProc7->ParamByName("n_m")->AsString=Edit11->Text;
    DataModule2->IBStoredProc7->Prepare();
    DataModule2->IBStoredProc7->ExecProc();
    DataModule2->IBTransaction1->Commit();
    DataModule2->IBQuery7->Close();
    DataModule2->IBQuery1->Open();
    DataModule2->IBQuery2->Open();
    DataModule2->IBQuery3->Open();
    DataModule2->IBQuery4->Open();
    DataModule2->IBQuery5->Open();
    DataModule2->IBQuery6->Open();
    DataModule2->IBQuery7->Open();
    DataModule2->IBQuery8->Open();
    DataModule2->IBQuery9->Open();
    DataModule2->IBQuery10->Open();
}
//-----

void __fastcall TForm1::Button12Click(TObject *Sender)
{
    DataModule2->IBStoredProc8->StoredProcName="ADD_Ñ";
    DataModule2->IBStoredProc8->ParamByName("nam")->AsString=Edit10->Text;
    DataModule2->IBStoredProc8->Prepare();
    DataModule2->IBStoredProc8->ExecProc();
    DataModule2->IBTransaction1->Commit();
    DataModule2->IBQuery8->Close();
    DataModule2->IBQuery1->Open();
    DataModule2->IBQuery2->Open();
    DataModule2->IBQuery3->Open();
    DataModule2->IBQuery4->Open();
    DataModule2->IBQuery5->Open();
    DataModule2->IBQuery6->Open();
    DataModule2->IBQuery7->Open();
    DataModule2->IBQuery8->Open();
    DataModule2->IBQuery9->Open();
    DataModule2->IBQuery10->Open();
}
//-----

void __fastcall TForm1::Button6Click(TObject *Sender)
{
    DataModule2->IBStoredProc8->StoredProcName="UPD_NOW";
    DataModule2->IBStoredProc8->ParamByName("n_o")->AsString=DBLookupListBox1-
    >ListSource->DataSet->FieldByName("nom_o")->AsString;
    DataModule2->IBStoredProc8->Prepare();
    DataModule2->IBStoredProc8->ExecProc();
    DataModule2->IBTransaction1->Commit();
    DataModule2->IBQuery8->Close();
    DataModule2->IBQuery1->Open();
    DataModule2->IBQuery2->Open();
    DataModule2->IBQuery3->Open();
    DataModule2->IBQuery4->Open();
    DataModule2->IBQuery5->Open();
    DataModule2->IBQuery6->Open();
    DataModule2->IBQuery7->Open();
    DataModule2->IBQuery8->Open();
    DataModule2->IBQuery9->Open();
    DataModule2->IBQuery10->Open();
    ShowMessage("Äüïëíäíëäíäðèèèíäòâáðæääí");
}
//-----

```

```
void __fastcall TForm1::Button9Click(TObject *Sender)
{
Form3->QuickRep1->Preview();
}
//-----
```

```
void __fastcall TForm1::Button14Click(TObject *Sender)
{
Form4->QuickRep1->Preview();
}
//-----
```