

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА МОДУЛЯ УЧЕТА КАЧЕСТВА ПРИНИМАЕМОГО
СЫРЬЯ В ОАО «ЭФКО»**

Выпускная квалификационная работа
обучающегося по направлению подготовки 38.03.05 «Бизнес-информатика»
заочной формы обучения, группы 12001455
Белых Максима Владимировича

Научный руководитель:
к.т.н., доцент
Асадуллаев Р.Г.

БЕЛГОРОД 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 АНАЛИЗ СИСТЕМ УПРАВЛЕНИЯ КАЧЕСТВОМ ПРИНИМАЕМОГО СЫРЬЯ.....	5
1.1 Анализ систем управления качеством	5
1.2 Исследование информационного обеспечения систем управления качеством	10
2 АНАЛИЗ ДЕЯТЕЛЬНОСТИ ОАО «ЭФКО» И СУЩЕСТВУЮЩЕЙ ОЦЕНКИ КАЧЕСТВА СЫРЬЯ	17
2.1 Анализ деятельности компании ОАО «ЭФКО»	17
2.2 Характеристики качества сырья при получении продукции масложирового дивизиона	20
2.3 Описание существующего процесса оценки качества принимаемого сырья ОАО «ЭФКО».....	23
3 РАЗРАБОТКА МОДУЛЯ УЧЕТА КАЧЕСТВА ПРИНИМАЕМОГО СЫРЬЯ ОАО «ЭФКО».....	28
3.1 Проектирование интерфейса модуля учета качества принимаемого сырья ОАО «ЭФКО».....	28
3.2 Проектирование базы данных модуля учета качества принимаемого сырья ОАО «ЭФКО».....	29
3.3 Работа с модулем учета качества принимаемого сырья	35
3.4 Оценка экономического эффекта спроектированного модуля учета качества принимаемого сырья ОАО «ЭФКО».....	48
ЗАКЛЮЧЕНИЕ	54
СПИСОК ЛИТЕРАТУРЫ.....	56
ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ.....	59
ПРИЛОЖЕНИЕ Б. ВЫХОДНАЯ ФОРМА ВХОДНОГО КОНТРОЛЯ.....	68

ВВЕДЕНИЕ

Современное производство отличается сложностью, многообразием связей, форм и методов. Чтобы производство было эффективным, потоки информации — плановые и отчетные документы, производственную документацию, информацию о банковских операциях — необходимо обрабатывать безошибочно и в самые сжатые сроки. Своевременно и правильно обработанная информация становится важным производственным ресурсом. Использование компьютеров и информационных технологий на всех этапах управления способно повысить его эффективность и качество.

Автоматизированная система управления (АСУ) — комплекс технических и программных средств, обеспечивающий в тесном взаимодействии с отдельными специалистами или коллективами управление объектом в производственной, научной или общественной сфере.

В практике разработки автоматизированных систем для областей, где их применение носит массовый характер (в первую очередь это АСУ промышленных предприятий) сложился ряд характерных функциональных подсистем, выделение которых обосновывается объективным делением управляемого процесса производства на отдельные части в соответствии с существующей организационной структурой управления, функциональной спецификой этих частей и ограниченными возможностями каждой из них по обработке необходимых объемов информации.

Все рассмотренные доводы говорят об актуальности выбранной темы исследования, а именно: автоматизации повторяющихся производственных процессов.

Целью исследования является повышение эффективности процесса оценки качества принимаемого сырья в ОАО «ЭФКО».

Гипотеза исследования: если разработать информационную систему оценки качества принимаемого сырья в ОАО «ЭФКО», то это позволит

сократить расходы, связанные с простоем технологического оборудования, ответственного за хранение масла и оценку его качества.

Объект исследования: отдел оценки качества сырья (отдел нормоконтроля) ОАО «ЭФКО».

Предмет исследования: информационное обеспечение отдела оценки качества сырья (отдела нормоконтроля) ОАО «ЭФКО».

Задачи исследования:

- 1) Провести анализ систем управления качеством принимаемого сырья;
- 2) Провести анализ деятельности ОАО «ЭФКО» и учета контроля качества принимаемого сырья масложирового дивизиона;
- 3) Спроектировать модуль оценки качества принимаемого сырья ОАО «ЭФКО».

Методы исследования: методы анализа и синтеза, метод проектирования.

1 Анализ систем управления качеством принимаемого сырья

1.1 Анализ систем управления качеством

Перед тем, как проводить анализ систем управления качеством, необходимо ознакомиться с документами, которые должны быть положены в данные информационные системы. Управление качеством товаров и услуг описано в системе менеджмента качества.

В понятии системы менеджмента качества (СМК) заключена совокупность мероприятий, действий, реализуемых операций, выполняемых в соответствии с определенными нормативными документами. Целью функционирования и поддержания СМК в актуальном состоянии является достижение оптимального уровня качества продукции и услуг.

СМК не является набором случайных действий, это отлаженный механизм, задача которого комплексно работать над улучшением внутренних процессов с целью недопущения возникновения потерь качества.

Система менеджмента качества пищевой промышленности включает российские стандарты:

- ГОСТ Р ИСО 22000-2007. Система менеджмента безопасности пищевой продукции. Требования к организациям, участвующим в цепи создания пищевой продукции;

- ГОСТ Р 51705.1-2001. Управление качеством пищевых продуктов на основе принципов ХАССП. Общие требования.

Настоящий международный стандарт устанавливает требования к системе менеджмента безопасности пищевой продукции для тех случаев, когда организации, находящейся в пищевой цепочке, необходимо продемонстрировать свою способность управлять опасностями в пищевой продукции для обеспечения того, чтобы пищевая продукция была безопасна во время ее потребления человеком.

ГОСТ Р ИСО 22000-2007 содержит требования к системе менеджмента безопасной пищевой продукции, включающей основные признанные элементы:

- интерактивный обмен информацией;
- система менеджмента;
- программы создания предварительных условий (программы-предпосылки);
- принципы анализа опасностей по критическим контрольным точкам (НАССР).

В системе НАССР выделены семь основных принципов [1]:

- идентификация потенциального риска и рисков (опасных факторов) производства;
- выявление критических контрольных точек в производстве для устранения (минимизации) риска;
- в документах системы НАССР следует установить и соблюдать предельные значения параметров в критических контрольных точках;
- разработка системы мониторинга, позволяющая обеспечить контроль критических контрольных точек;
- разработка корректирующих действий и применение их в случае отрицательных результатов мониторинга;
- разработка процедур проверки, которые должны регулярно проводиться;
- документирование всех процедур системы, форм и способов регистрации данных.

Контроль входного сырья включает в себя: органолептические показатели, качество упаковки, маркировка, весовые характеристики, наличие необходимой сопроводительной документации [2].

Среди аналогов информационных систем, соответствующих требованиям вышерассмотренных ГОСТов, можно выделить:

- Altami;
- АСУ ТП «Контроль качества»;
- ИС-ПРО.

Рассмотрим их в таблице 1.1.

Таблица 1.1 – Анализ информационных систем оценки качества [3-5]

Параметр ИС	Область применения	Назначение	Особенности
Altami (разработчик: Altami software)	различные отрасли производства	автоматизация контроля качества изделий и товаров	решают многочисленные задачи по анализу и обработке изображений исследуемой продукции
АСУ ТП «Контроль качества» (разработчик: ООО «Ритм»)	различные отрасли производства	автоматизация входного контроля сырья	- печать отчетов и протоколов испытаний, - хранение всех графиков, технологических режимов, рецептуры, - печатные формы по ГОСТ
ИС-ПРО (разработчик: IS-PRO)	пищевая промышленность	автоматизация всех процессов пищевой промышленности	- ведение нормативно-справочной информации, - автоматический ввод количественных показателей движения сырья, - ведение бухгалтерского и налогового учета

Проанализировав аналоги информационных систем, можно прийти к выводу, что использование ИС-ПРО является более перспективным, т.к. ее область применения – только лишь пищевая промышленность, что говорит о целевой направленности данного продукта.

ИС под названием Altami применять возможно не в любой ситуации, т.к. ее преимущество – только лишь оптическая система по обработке изображения. Нет возможности ведения отчетов, например, по типам сырья. АСУ ТП «Контроль качества» является приложением, которое следует дорабатывать под нужды конкретного предприятия, что повлечет за собой потерю лишнего времени. Доработка также касается и действующих ГОСТов

системы. В качестве более перспективного остановимся на ИС-ПРО от разработчика IS-PRO. Скриншот ее главного окна показан на рисунке 1.1.

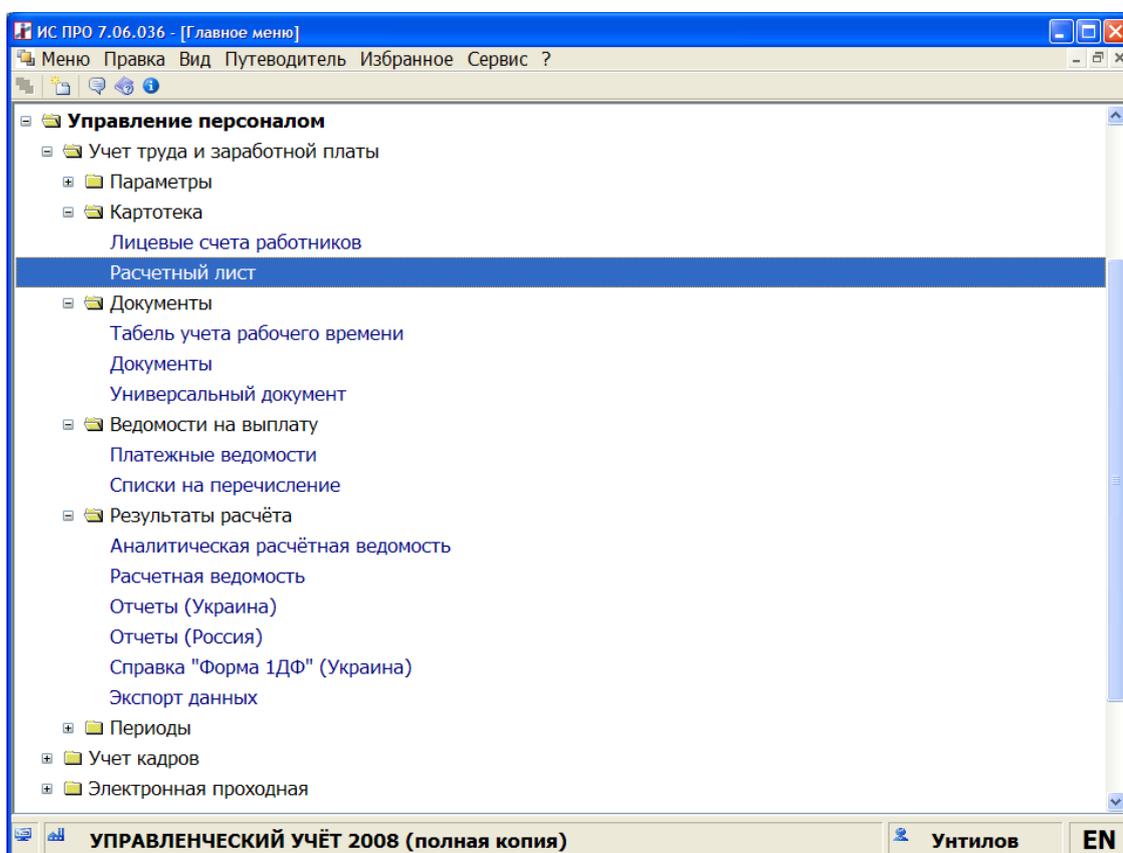


Рисунок 1.1 – Главное окно ИС-ПРО

Программа автоматизации процессов работы предприятия ИС-ПРО представляет собой набор взаимосвязанных подсистем. Каждая подсистема имеет определенную самостоятельность, именно поэтому ИС-ПРО можно собирать в нужной конфигурации, в том числе и для управления качеством (рисунок 1.2). Благодаря такому гибкому подбору функционала программным комплексом осуществляется автоматизация бизнес-процессов для каждого отдельного предприятия. Это позволяет экономить за счет отказа от лишнего функционала [26, с. 38].



Рисунок 1.2 – Ключевые особенности ИС-ПРО

Система ИС-ПРО разработки компании «Интеллект-Сервис» поддерживает стандарт MRP II - один из самых распространенных методов управления, используемый в мировой практике. Стандарт MRP II - это совокупность правил, моделей и процедур управления, обеспечивающая улучшение показателей экономической деятельности предприятий, в которых интегрированы все основные процессы - поставка, планирование, производство, продажи, контроль за материальными и финансовыми ресурсами.

Принципы, заложенные в данный стандарт, позволяют оптимизировать основные процессы, происходящие на предприятии, за счет сокращения расходов и времени, которое тратится на производство продукции. Это совокупность правил, моделей и процедур управления, обеспечивающая улучшение показателей экономической деятельности предприятий.

Визитная карточка системы ИС-ПРО - проработка учетных и управленческих функций для решения задач предприятия, опора на

современные технологии и лучший мировой опыт, оперативная поддержка всех изменений в законодательстве, устойчивость и надежность.

1.2 Исследование информационного обеспечения систем управления качеством

Информационная система (ИС) управления качеством продукции – это определенным образом организованная совокупность взаимосвязанных средств вычислительной техники и программного обеспечения, позволяющая решать те или иные функциональные задачи по управлению качеством.

Информационное обеспечение подобных систем включает в себя непосредственно саму ИС в виде ERP-системы, системы офисного документооборота, с которой она взаимодействует, и базу данных для хранения информации (рисунок 1.3).

Системы офисного документооборота выходят за рамки рассмотрения данного вопроса, поэтому остановимся на том, что ею могут выступать конфигурации ИС: Предприятия, подобранные под нужды организации.

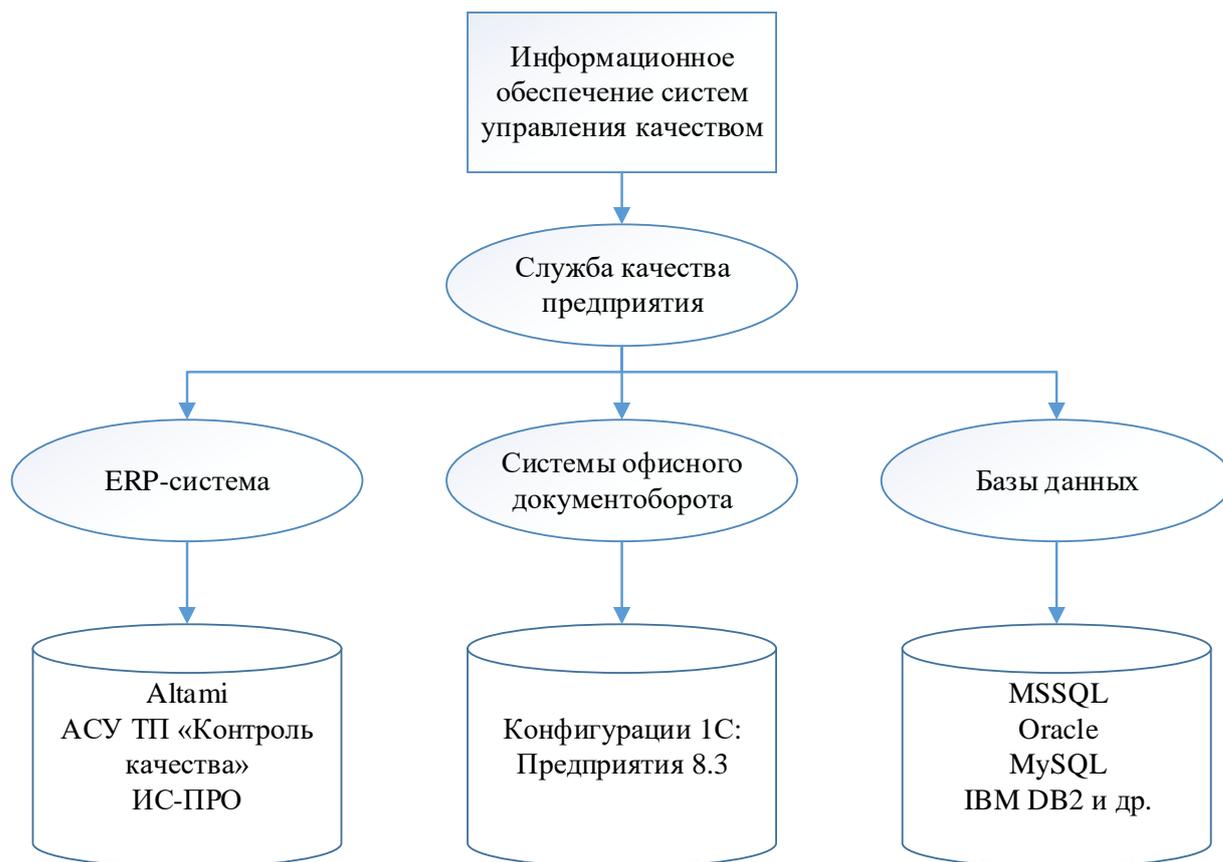


Рисунок 1.3 – Информационное обеспечение систем управления качеством

Проектирование ERP-систем управления качеством продукции состоит из следующих этапов [6, с. 115]:

- формирование требований к ERP-системе;
- разработка концепции ERP-системы;
- техническое задание: разработка и утверждение технического задания на создание ERP-системы;
- эскизный проект: разработка предварительных проектных решений по системе и ее частям;
- технический проект: разработка проектных решений по системе и ее частям; разработка документации на ERP-систему и ее части;
- разработка рабочей документации на ERP-систему и ввод в действие.

Сама же автоматизированная система информационной поддержки управления качеством предприятия ERP-системы представляет собой совокупность трех совместно используемых компонент:

- подсистемы, обеспечивающие информационную поддержку разработки, функционирования и улучшения СМК предприятия;
- средства документирования информации (данных, записей) о качестве;
- технология построения и внедрения ERP-системы.

В число подсистем, обеспечивающих информационную поддержку разработки, функционирования и улучшения СМК предприятия входят специализированные подсистемы (рисунок 1.4).



Рисунок 1.4 – Подсистемы, обеспечивающие информационную поддержку разработки, функционирования и улучшения СМК

В рамках рассмотрения учета качества для нас представляют интерес первые две подсистемы. Их назначение и функции показаны в таблице 1.2.

Таблица 1.2 – Подсистемы учета качества

Параметр	Подсистема	
	Управление документами системы качества (УДСК)	Внутренние проверки качества (ВПК)
Назначение	сохраняются результаты всех действий с документами системы качества (ДСК) начиная с регистрации проекта и до передачи ДСК в архив базы данных	обеспечивает управление подготовкой и проведением внутренних проверок СМК
Основные функции	<ul style="list-style-type: none"> – управление и контроль за ходом документирования СМК; – многоуровневое согласование, утверждение, ввод в действие, корректировку, замену, отмену ДСК; – исключение использования недействующих и наличие на рабочих местах только актуальных ДСК; – исключение ошибок при рассылках ДСК; – исключение «бумажного» документооборота ДСК; – снижение временных затрат на разработку новых и корректировку действующих ДСК; – организацию коллективной работы с ДСК 	<ul style="list-style-type: none"> – ведение реестра внутренних аудиторов, включая управление и контроль за ходом их подготовки; – планирование и контроль проведения годовых проверок СМК; – управление организацией и проведением текущих проверок СМК; – оформление результатов текущих проверок СМК; – контроль за ходом и результатами устранения замечаний, выявленных на текущих проверках СМК

Вторым ключевым параметром информационного обеспечения систем управления качеством в виде ERP-систем являются базы данных (а именно системы управления базами данных), в которых хранится вся информация по управлению качеством.

Система управления базами данных (СУБД) – это набор языковых и программных средств для создания и ведения базы данных: обновления данных базы, обеспечения доступа к данным по запросам и выдачи их пользователям и др.

Рассмотрим основные СУБД.

Oracle – это мощный программный комплекс, позволяющий создавать приложения любой степени сложности [7]. Ядром этого комплекса является база данных, хранящая информацию, количество которой за счет предоставляемых средств масштабирования практически безгранично. С высокой эффективностью работать с этой информацией одновременно может

практически любое количество пользователей (при наличии достаточных аппаратных ресурсов), не проявляя тенденции к снижению производительности системы при резком увеличении их числа.

Встраивание в СУБД Oracle JavaVM, полномасштабная поддержка серверных технологий (Java Server Pages, Java-сервлеты, модули Enterprise JavaBeans, интерфейсы прикладного программирования CORBA), привело к тому, что Oracle на сегодняшний день де-факто является стандартом СУБД для Internet.

Поскольку СУБД MS Access [8] может работать в режиме коллективного доступа к базам данных, она является идеальным средством разработки приложений для рабочих групп, которые хранят данные на серверах локальных сетей своих подразделений и в то же время периодически пользуются приложениями других подразделений и сохраняют свои данные на серверах учреждения или предприятия. Если речь идет о небольших рабочих группах, то хранение и коллективный доступ к данным могут осуществляться только при помощи MS Access. Большие приложения для хранения данных используют специальный сервер (например, SQL Server), а MS Access на рабочей станции выступает в этом случае в роли клиента. Кроме того, в крупных учреждениях MS Access может использоваться как пользовательская среда для обработки данных.

Лидирующие позиции на рынке среди СУБД занимает Microsoft SQL Server. Доля рынка Microsoft SQL Server, по данным Gartner, составляет 46,8%, остальная часть рынка принадлежит Oracle и IBM DB2, относительно небольшую часть рынка занимают СУБД Open Source, такие как PostgreSQL и Firebird [9].

Microsoft SQL Server 2012 [10] является новейшей и мощнейшей системой управления базами данных. Помимо стандартных для СУБД функций, SQL Server 2012 содержит большой набор интегрированных служб по анализу данных. Доступ к данным, расположенным на SQL Server могут получить любые приложения, разработанные на .Net и VisualStudio, а также

приложения пакета Microsoft Office 2007. SQL Server 2012 обеспечивает высочайшую в своём классе масштабируемость, производительность и безопасность.

IBM DB2 использует общепринятую на текущий момент клиент-серверную архитектуру реляционной СУБД, с обеспечением хранения информации на сервере и подключением приложений-клиентов к базам данных локально либо через сеть [11].

В DB2 реализована поддержка всех распространенных промышленных стандартов доступа приложений к данным, включая стандартный язык запросов SQL, интерфейсы ODBC и JDBC, работу с типовыми текстовыми табличными форматами и т.п. Кроме того, DB2 включает в себя развитые возможности по хранению и работе с полуструктурированными данными в форматах XML, JSON/BSON.

Для разработки хранимых процедур в DB2 реализована поддержка множества процедурных языков, включая:

- стандартный для DB2 язык SQL PL,
- используемый в СУБД корпорации Oracle язык SQL/PL.

Выводы по главе: модули систем оценки качества являются одними из важных в производственных процессах. Среди аналогов информационных систем, соответствующих требованиям вышерассмотренных ГОСТов, можно выделить:

- Altami;
- АСУ ТП «Контроль качества»;
- ИС-ПРО.

Информационное обеспечение подобных систем включает в себя непосредственно саму ИС в виде ERP-системы, системы офисного документооборота, с которой она взаимодействует, и базу данных для хранения информации.

Сама же автоматизированная система информационной поддержки управления качеством предприятия ERP-системы представляет собой совокупность трех совместно используемых компонент:

- подсистемы, обеспечивающие информационную поддержку разработки, функционирования и улучшения СМК предприятия;
- средства документирования информации (данных, записей) о качестве;
- технология построения и внедрения ERP-системы.

Среди баз данных рассмотрены такие СУБД, как Oracle, MS Access, Microsoft SQL Server и IBM DB2.

2 Анализ деятельности оао «эфко» и существующей оценки качества сырья

2.1 Анализ деятельности компании ОАО «ЭФКО»

Группа компаний «ЭФКО» была основана в 1994 году в г. Алексеевке. Она уверенно занимает позицию ведущего российского производителя жиров специального назначения и маргаринов, которые используются в качестве ингредиентов для производства продуктов питания в кондитерской, хлебопекарной, молочной и других отраслях пищевой промышленности.

Производство маргаринов и жиров специального назначения осуществляется на трех заводах компании, расположенных в городе Алексеевке Белгородской области, Краснодарском крае на территории морского порта Тамань и в г. Алматы Республики Казахстан, с 2016 года контрактное производство размещается также на площадке ООО «Евдаково».

Мощная техническая база обеспечивает серьезные преимущества перед конкурентами. В настоящее время Группа компаний «ЭФКО» является единственным в России производителем, способным осуществлять процесс фракционирования при производстве альтернатив масла какао, используемых в кондитерской промышленности [12].

Группа Компаний «ЭФКО» включает в себя множество отделов, каждый из которых производит функционирование своей положенной ему отрасли. В представленной работе интерес представляет только отдел, занимающийся оценкой качества сырья. Упрощенная организационная структура отдела оценки качества сырья ОАО «ЭФКО» представлена на рисунке 2.1.



Рисунок 2.1 – Организационная структура отдела оценки качества сырья
ОАО «ЭФКО»

Менеджер по поставкам проводит сотрудничество с поставщиками сырья из России и других стран для масложирового дивизиона (масел, майонезов). В особенности, он работает только с поставщиками.

Научно-исследовательская лаборатория разрабатывает технологии оценки качества сырья для пищевых ингредиентов, а также проводит оценку качества.

Лабораторная оценка качества сырья производится в научно-исследовательской лаборатории при каждом сотрудничестве с фирмой-поставщиком. Лаборатория анализирует характерные параметры для того или иного вида сырья и выносит решение по каждой отдельной партии. Научно-исследовательская лаборатория производит основную работу по безопасности продуктов, используя стандарты качества ГОСТ Р ИСО 22000-2007 и ГОСТ Р 51705.1-2001. Также лаборатория руководствуется и собственными техническими условиями (ТУ) и стандартами предприятия (СТП), работа над которыми ведется в лаборатории.

Менеджер по работе с договорами ведет всю бухгалтерскую и

юридическую работу по связям с поставщиками.

Приемщики сырья непосредственно производят приемку сырья для оценки его качества.

В рамках бизнес-сегмента «Продажа продуктов маслопереработки и сельскохозяйственной продукции» Группа компаний «ЭФКО» осуществляет переработку масличных культур, таких как семена подсолнечника и рапса, соевых бобов – и реализует следующие основные продукты:

– Масло подсолнечное рафинированное дезодорированное, высший сорт, ГОСТ 1129-2013 (наливом);

– Шрот подсолнечный тостированный, высокопротеиновый, ГОСТ 11246-96 (гранулированный, протеин – 44 % на а. с. в.);

– Шрот подсолнечный тостированный, ГОСТ 11246-96 (гранулированный, протеин – 39 % на а. с. в.);

– Масло соевое гидратированное, ГОСТ 31760-2012 (наливом);

– Шрот соевый кормовой тостированный, высокопротеиновый, ГОСТ Р 53799-2010 (негранулированный, обогащенный липидами, протеин – 51-52 % на а. с. в.);

– Шрот соевый кормовой тостированный, базовый, ГОСТ Р 53799-2010 (негранулированный, обогащенный липидами, протеин – 47 % на а. с. в.);

– Оболочка сои гранулированная, ТУ 9147-048-00333693-2014;

– Масло рапсовое нерафинированное, ГОСТ 31759-2012 (наливом);

– Шрот рапсовый тостированный, ГОСТ 30257-95 (негранулированный);

– Жир растительный сухой ULTRA FEED F, СТО 04402528-001-2019.

Компания использует новейшие разработки в области системы менеджмента качества, благодаря чему производимая продукция не только соответствует всем российским и мировым стандартам качества и ГОСТ, но и превосходит их. Продукция производится из сырья, не подверженного генной модификации (без ГМО).

Проверка качества сырья для пищевых ингредиентов ОАО «ЭФКО» осуществляется по рецептурам и технологиям, разработанным собственной научно-исследовательской лабораторией, и сертифицировано в соответствии с межгосударственными и Мировыми стандартами:

- ГОСТ Р ИСО 9001-2015 (ISO 9001:2015);
- Международным стандартом на пищевую продукцию (International Featured Standards – IFS Food);
- Международным стандартом пищевой безопасности Британского розничного консорциума (BRC Global Standard for Food Safety);
- Стандарту RSPO;
- Kosher;
- Halal;
- Sedex.

2.2 Характеристики качества сырья при получении продукции масложирового дивизиона

Система менеджмента качества при получении продукции масложирового дивизиона включает российские стандарты:

- ГОСТ Р ИСО 22000-2007. Система менеджмента безопасности пищевой продукции. Требования к организациям, участвующим в цепи создания пищевой продукции;
- ГОСТ Р 51705.1-2001. Управление качеством пищевых продуктов на основе принципов ХАССП. Общие требования.

Настоящий международный стандарт устанавливает требования к системе менеджмента безопасности пищевой продукции для тех случаев, когда организации, находящейся в пищевой цепочке, необходимо продемонстрировать свою способность управлять опасностями в пищевой

продукции для обеспечения того, чтобы пищевая продукция была безопасна во время ее потребления человеком.

ГОСТ Р ИСО 22000-2007 [13] содержит требования к системе менеджмента безопасной пищевой продукции, включающей основные признанные элементы:

- интерактивный обмен информацией;
- система менеджмента;
- программы создания предварительных условий (программы-предпосылки);
- принципы анализа опасностей по критическим контрольным точкам (НАССР).

В системе НАССР выделены семь основных принципов [14]:

- идентификация потенциального риска и рисков (опасных факторов) производства;
- выявление критических контрольных точек в производстве для устранения (минимизации) риска;
- в документах системы НАССР следует установить и соблюдать предельные значения параметров в критических контрольных точках;
- разработка системы мониторинга, позволяющая обеспечить контроль критических контрольных точек;
- разработка корректирующих действий и применение их в случае отрицательных результатов мониторинга;
- разработка процедур проверки, которые должны регулярно проводиться;
- документирование всех процедур системы, форм и способов регистрации данных.

Для рассмотрения сырья, выберем выпускаемую продукцию масложирового дивизиона ОАО «ЭФКО»:

– линейку растительных масел «Слобода»: включает в себя рафинированное дезодорированное подсолнечное масло «Слобода» для салатов и жарки, рафинированное дезодорированное подсолнечное масло «Слобода» Дачное, рафинированное дезодорированное подсолнечное масло «Слобода» для жарки и фритюра, нерафинированное подсолнечное масло «Слобода» Ароматное, рафинированное дезодорированное кукурузное масло «Слобода», рафинированное дезодорированное масло подсолнечное с добавлением оливкового «Слобода»;

– линейку растительных масел Altero: Altero Golden, Altero Vitality, Altero Guarana, Altero Bouquet отборное подсолнечное масло смешивается с особыми ингредиентами – с оливковым маслом, маслом зародышей пшеницы, с экзотическим экстрактом гуараны, лепестками роз;

– продукцию марки EFKO FOOD professional товары, ориентированные на сегмент HoReCa – профессиональных поваров, у которых есть свои, особенные требования к продуктам. Для их удобства под брендом EFKO FOOD professional выпускаются майонезы различной жирности, подсолнечное рафинированное масло, масло для фритюра в емкостях большого объема, удобных для постоянного и активного приготовления пищи.

Таким образом, в рамках продукции масложирового дивизиона рассмотрим характеристики качества сырья, используемого в наибольшем количестве. К нему относится такое сырье, как [15]:

- смешанное масло;
- пресловое масло.

Качественные характеристики смешанного, соевого и преслового подсолнечного масла на этапе приемки показаны в табл. 2.1.

Все рассматриваемые характеристики будут использоваться при составлении информационной системы оценки качества сырья.

Таблица 2.1 – Количественные характеристики смешанного, соевого и прессового подсолнечного масла на этапе приемки

Характеристика	Номинальное значение	Основание	Периодичность проверки
Кислотное число, мгКОН/г: -смешанное масло -прессовое масло	Не более 6,0 Не более 4,0	ГОСТ 31933 [16]	Каждое транспортное средство
Перекисное число, ммоль активного кислорода/кг	Не более 10,0	ГОСТ 26593 [17]	Каждое транспортное средство
Массовая доля влаги и летучих веществ, %: -смешанное масло -прессовое масло	Не более 0,3 Не более 0,2	ГОСТ 11812 [18]	Каждое транспортное средство
Цветное число, мг йода: -смешанное масло -прессовое масло	Не более 35 Не более 25	ГОСТ 5477 [19]	Каждое транспортное средство
Массовая доля фосфорсодержащих веществ (в пересчете на стеароолеолецитин), %: -смешанное масло -прессовое масло	Не более 0,8 Не более 0,6	ГОСТ 31753 [20]	Каждое транспортное средство

2.3 Описание существующего процесса оценки качества принимаемого сырья ОАО «ЭФКО»

Обязательным условием приемки продукции является наличие в комплекте сопроводительной документации, в общем случае: сертификат соответствия, паспорт на партию; удостоверение качества; гигиенический сертификат; счет-фактура, накладная.

Для приемки сырья масложирового дивизиона предназначена маслосливная ЦПСП.

Маслосливная ЦПСП предназначена для приема, хранения и отгрузки в производство или в авто и ж/д цистерны смешанного нерафинированного, прессового подсолнечного и соевого масел.

Масло прессовое, выработанное из смеси семян молочной спелости и полнозрелых подсолнечника, поступает на хранение в емкость для прессового масла.

Емкости маслосливной ЦПСИ оснащены ручной запорной арматурой, мешалками, насосами, люками, приборами контроля температуры и количества продукции (уровнемеры). Температуру масла и уровень наполнения (%) каждой из емкостей можно наблюдать в пультовой на мониторе компьютера.

Ассортимент масла, закачиваемый в емкости:

- Экстракционное масло - масло, извлеченное из масличного материала экстракционным методом;
- Прессовое масло - растительное масло, получаемое прессованием масличной мезги и мезги, приготовленной из форпрессового жмыха, на прессах;
- Смешанное масло - смесь прессового и экстракционного масла.

Рассмотрим непосредственно процесс приемки.

После поступления ж/д или автоцистерны под налив соевым гидратированным маслом, Оператор линии получает у оператора ЭВМ необходимое количество накопительных сосудов, в которые оператор линии будет осуществлять отбор проб.

Отбор проб осуществляется в соответствии с ГОСТ 32190 [21]. При наполнении ж/д и авто цистерн отбор проб проводят при перекачке масла по трубопроводу. Объединенную пробу отбирают штуцерным пробоотборником в накопительный сосуд. Пробу в накопительном сосуде тщательно перемешивают.

При отборе проб должны соблюдаться следующие условия:

- средства для отбора проб и емкости для хранения проб должны быть чистыми, сухими и не иметь постороннего запаха;
- средства отбора проб и отобранные пробы должны быть защищены от загрязнения и атмосферных осадков;

– отбор проб требуется осуществлять только в местах, защищенных от атмосферных осадков.

Объем одной пробы, отобранной при наполнении одной ж/д или автоцистерны должен составлять не менее 1 л (или равняться объему одного полностью наполненного выданного лабораторией сосуда).

Отбор проб из автоцистерн производится в соответствии со следующим алгоритмом:

- открыть штуцерный пробоотборник и отбирать объединенную пробу (по 1/3 в начале, в середине и в конце загрузки);
- объем пробы не менее 1 л;
- закрыть штуцерный пробоотборник.

Отобранные пробы с соевым гидратированным маслом оператор линии передает на весовую оператору ЭВМ.

После того, как получено положительное заключение, подключается шланг насоса и масло откачивается из цистерн в резервуары. На производстве имеется семь резервуаров: отдельно для смешанного, соевого или прессового масла. Если же продукция не соответствует требованиям закупки, то производится возврат продукции.

Весь процесс приемки в среднем составляет около 40 минут.

В ОАО «ЭФКО» ведется учет поступивших заявок в накопительной папке отдела снабжения, а также в системе электронного документооборота предоставляются отчетные формы (приказами устанавливается периодичность обновления).

В связи с этим, имеется необходимость разработки автоматизированной информационной системы проверки параметров для каждой единицы сырья, которая будет анализировать выходные параметры поступающего сырья и выносить окончательное решение о необходимости приемки или отказа от приемки партии. На данный момент формы входного контроля формируются вручную.

Также в компании внедрена система контроля качества «Органик-контроль», которая предполагает использование ингредиентов натурального нехимического происхождения.

Проведение анализа данных производится в специализированных производственных лабораториях, которые производят оценку качества сырья, которое ему поставляется. Каждая из лабораторий подчиняется правилам и методам оценки по ГОСТу и внутренним СМК того сырья, который ему вверен. Также лаборатории могут подчиняться собственным техническим условиям (ТУ), которые разработаны на основе ГОСТов и СМК. Данный этап может занимать продолжительное время в зависимости от размеров партии.

Анализ данных представляет собой получение параметров исследуемых характеристик после проведения серии лабораторных испытаний. Выходные параметры представляют собой значения характеристик пищевых продуктов.

Автоматизация оценок и анализа выходных характеристик качества сырья масложирового дивизиона в ОАО «ЭФКО» обеспечит:

- уменьшение временных издержек, связанных с долгим принятием решения о качестве партии сырья;
- составление отчетов об анализе характеристик качества в автоматизированном режиме;
- увеличить количество принимаемых автоцистерн в сутки, что является, несомненно, большим плюсом для производства;
- рост производительности труда;
- повышение эффективности работы предприятия, путем сокращения штата сотрудников, в связи с автоматизацией работы отдела;
- контроль над графиком проведения оценок качества сырья.

Автоматизация оценок и анализа выходных характеристик подразумевает также обучение определенного числа сотрудников работе с автоматизированной системой.

Выводы по главе: в представленной работе интерес представляет только отдел, занимающийся оценкой качества сырья. В ходе анализа организационной структуры отдела, выявлено, что непосредственно основную работу по оценке качества продуктов, используя стандарты качества ГОСТ Р ИСО 22000-2007 и ГОСТ Р 51705.1-2001 производит научно-исследовательская лаборатория отдела.

Для рассмотрения характеристик сырья выбраны три основные брендовые марки масложировой продукции ОАО «ЭФКО»: линейка растительных масел «Слобода», линейка растительных масел Altero и продукция марки EFKO FOOD professional. В рамках продукции масложирового дивизиона рассмотрим такое сырье, как смешанное масло и пресовое масло.

Контроль входного сырья включает в себя оценку таких параметров, как: кислотное число, перекисное число, массовая доля влаги и летучих веществ, цветное число, массовая доля фосфорсодержащих веществ (в пересчете на стеароолеолецитин).

В ОАО «ЭФКО» ведется учет поступивших заявок в накопительной папке отдела снабжения, а также в системе электронного документооборота предоставляются отчетные формы (приказами устанавливается периодичность обновления).

В связи с этим, имеется необходимость разработки автоматизированной информационной системы проверки параметров для каждой единицы сырья, которая будет анализировать выходные параметры поступающего сырья и выносить окончательное решение о необходимости приемки или отказа от приемки партии.

3 Разработка модуля учета качества принимаемого сырья оао «эфко»

3.1 Проектирование интерфейса модуля учета качества принимаемого сырья ОАО «ЭФКО»

В качестве средств реализации выбрана среда MS Visual Studio и язык программирования С#, являющийся в последнее время особенно актуальным. При проектировании программы были использованы следующие компоненты среды программирования [24, 25] (таблица 3.1).

Таблица 3.1 – Компоненты среды

Наименование	Описание
1	2
Label	Используется для размещения на формах и других контейнерах текста, который не изменяется пользователем. Компонент визуальный
Button	Используется для создания кнопок, на которых располагается текстовая надпись. Компонент визуальный
GroupBox	Является контейнером для группирования органов управления. Графическая панель можно использовать также для построения полос состояния. Компонент визуальный
ComboBox	Используется для вывода выпадающего списка. Компонент визуальный
DataGridView	Используется для вывода табличных данных из базы данных. Компонент визуальный
menuStrip	Используется для отображения пунктов меню программы. Компонент визуальный
pictureBox	Используется для отображения изображения на главной форме приложения. Компонент визуальный
textBox	Используется для добавления и редактирования записей базы данных программы. Компонент визуальный
dateTimePicker	Используется для отображения дат из базы данных программы. Компонент визуальный

В работе программы помимо модулей основных форм для работы с программой, имеются основные классы, в которые включены основные алгоритмы работы программы:

– database.cs – класс, позволяющий работать с базой данных: добавлять и изменять в ней информацию, а также заполнять этими данными списки comboBox;

– ExportExcel.cs – класс, вызывающий основной алгоритм вынесения результатов проверки о разрешении/запрещении запуска в производство в формат MS Excel.

Разработанный модуль учета качества поступающего сырья позволяет работать с параметрами оценки для каждого вида сырья и производить занесение параметров оценки для каждой партии сырья, поступающего автоцистернами. Для работы программы требуется установить на компьютер поставщик базы данных Microsoft.ACE.OLEDB.12.0, который позволяет работать с базой данных (формата .accdb), и обладающей последними разработками компании Microsoft [28].

Для работы же самой программной оболочки требуется установить NET. Framework последней версии на сайте Microsoft, но не меньше версии 4.5. База данных должна обязательно располагаться в одной папке с программной оболочкой.

3.2 Проектирование базы данных модуля учета качества принимаемого сырья ОАО «ЭФКО»

При проектировании физической структуры [27] MS Access создадим семь таблиц:

- 1) Сырье – данные о сырье и его назначении;
- 2) ПараметрыОтчета – данные по параметрам оценки качества того или иного сырья;
- 3) ПараметрыСырья – информация о проведенных проверках в ИС;
- 4) Отчет – информация о значениях параметров сырья;

5) Поставщики – информация о поставщиках сырья и датах сотрудничества с ними;

6) Сотрудники – информация о сотрудниках, которые трудятся в лабораториях, проверяющих сырье в соответствии со стандартами качества и приемщиками сырья;

7) Лаборатории – информация о лабораториях, которые производят оценку качества сырья.

В таблице «Сырье» поле «id_сырья» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.1.

The image shows two screenshots of the Microsoft Access interface for a table named 'Сырье' (Raw Material). The top screenshot is in Design View, showing the field names and their data types. The bottom screenshot is in Datasheet View, showing the data rows.

Имя поля	Тип данных
idСырья	Счетчик
idПоставщика	Числовой
наименование	Короткий текст
назначение	Длинный текст

idСырья	idПоставщи	наименова	назначение
3	2	Смешенное м	Для производ
4	1	Прессовое ма	Для производ
(№)			

Рисунок 3.1 – Таблица «Сырье»

В таблице «ПараметрыОтчета» ни одно поле не является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.2.

ПараметрыОтчета	
Имя поля	Тип данных
idПараметра	Числовой
idОтчета	Числовой
ПолученноеЗначение	Короткий текст

ПараметрыОтчета		
idПараметр	idОтчета	Полученное
1	3	14
2	4	3,9
3	3	15
4	4	8
5	3	45
6	4	0,15
7	3	45
8	4	25
9	3	4
10	4	0,4
*	0	0

Рисунок 3.2 – Таблица «ПараметрыОтчета»

В таблице «Отчет» поле «idПроверки» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.3.

Отчет	
Имя поля	Тип данных
idПроверки	Счетчик
idЛаборатории	Числовой
ДатаОтчета	Дата и время
idСырья	Числовой
idСотрудника	Числовой
Партия	Короткий текст
КолвоСырья	Числовой
ТранспортСредство	Короткий текст
ДатаИзготовления	Дата и время

Отчет								
idПроверки	idЛаборато	ДатаОтчета	idСырья	idСотрудни	Партия	КолвоСырь	ТранспортС	ДатаИзгото
1	1	29.04.2019 14:36:34	3	9	2	3	1	29.04.2019
4	1	29.04.2019 18:43:38	4	9	147	25160	P 3228AC	28.04.2019
*	(№)	0				0		

Рисунок 3.3 – Таблица «Отчет»

В таблице «ПараметрыСырья» поле «idПараметра» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.4.

The image shows two screenshots of the Microsoft Access interface for the 'ParametersRaw' table. The top screenshot is in Design View, showing the field names and their data types. The bottom screenshot is in Datasheet View, showing the table's data with columns for 'наименование', 'idПараметр', 'ЭталонноеЗначение', and 'idСырья'.

ПараметрыСырья	
Имя поля	Тип данных
наименование	Короткий текст
idПараметра	Счетчик
ЭталонноеЗначение	Короткий текст
idСырья	Числовой

ПараметрыСырья			
наименование	idПараметр	ЭталонноеЗначение	idСырья
Кислотное число, мг	1	6,0	3
Кислотное число, мг	2	4,0	4
Перекисное число,	3	10,0	3
Перекисное число,	4	10,0	4
Массовая доля влаг	5	0,3	3
Массовая доля влаг	6	0,2	4
Цветное число, мг й	7	35	3
Цветное число, мг й	8	25	4
Массовая доля фосф	9	0,8	3
Массовая доля фосф	10	0,6	4
*	(№)		0

Рисунок 3.4 – Таблица «ПараметрыСырья»

В таблице «Поставщики» поле «idПоставщика» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.5.

The image shows a screenshot of the Microsoft Access interface for the 'Suppliers' table in Design View, showing the field names and their data types.

Поставщики	
Имя поля	Тип данных
idПоставщика	Счетчик
наименование	Короткий текст
ДатаНачала	Дата и время
ДатаОкончания	Дата и время
город	Короткий текст

Поставщики				
idПоставщи	наименова	ДатаНачала	ДатаОконч	город
1	ООО Сай	30.10.2018	12.07.2019	Краснодар
2	ООО Самарин	21.02.2018	22.11.2019	Алушта

Рисунок 3.5 – Таблица «Поставщики»

В таблице «Сотрудники» поле «idСотрудника» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.6.

Сотрудники	
Имя поля	Тип данных
idСотрудника	Счетчик
ФИОСотрудника	Короткий текст
Должность	Короткий текст

Сотрудники		
idСотрудни	ФИОСотруд	Должность
3	Захарченко Т.	Инженер СКК
9	Семенов К.А.	Приемщик

Рисунок 3.6 – Таблица «Сотрудники»

В таблице «Лаборатории» поле «idЛаборатории» является ключевым и индексируемым.

Окончательно таблица в режиме конструктора и в режиме выполнения показана на рисунке 3.7.

Лаборатории	
Имя поля	Тип данных
idЛаборатории	Счетчик
Лаборатория	Короткий текст
Телефон	Короткий текст
Заведующий	Числовой

Лаборатории			
idЛаборато	Лаборатория	Телефон	Заведующий
1	Лаборатория 1	788-55-45	8
2	Лаборатория 2	458-99-15	8

Рисунок 3.7 – Таблица «Лаборатории»

Схема данных полученной физической структуры будет выглядеть как на рисунке 3.8.

Данная схема данных соответствует целостности структуры данных [29] и обеспечивает уникальность данных. Каждый поставщик может поставлять несколько видов сырья, поэтому связываем таблицу Поставщики и Сырье как один ко многим.

Каждое сырье должно иметь свои параметры, свойственные только ему и данные параметры должны иметь эталонные значения. Поэтому таблицу Сырье и ПараметрыСырья связываем как один ко многим.

Для проведения отчета, заносится в таблицу номер партии проверяемого сырья, идентификатор лаборатории, которая проводила проверку, и номер сотрудника, который ее проводил. Заносятся параметры автоцистерны и количество сырья. Поэтому таблицу Сырье и Отчеты связываем как один ко многим. Таким образом, занеся один раз в таблицу сырье, мы можем проводить проверки его разных партий, не занеся в таблицу Сырье избыточные данные.

Также таблицы Лаборатории и Сотрудники связываем как один ко многим с таблицей Отчет, т.к. одна и та же лаборатория и сотрудник может проводить несколько отчетов.

Так как один отчет может иметь несколько параметров, причем параметр может повторяться с другим выходным значением, то свяжем таблицы Отчет и ПараметрыОтчета как один ко многим.

Чтобы к полученным значениям в результате проверки в таблице не приписывать соответствующие эталонные значения, которые у нас есть в таблице ПараметрыСырья, свяжем таблицы ПараметрыСырья и ПараметрыОтчета как один ко многим. Так как один и тот же параметр сырья может появиться в отчете несколько раз. Такая структура данных избавит от

избыточности данные, и соответственно, от ошибок в процессе разработки информационной системы, связанные с неверным представлением данных.

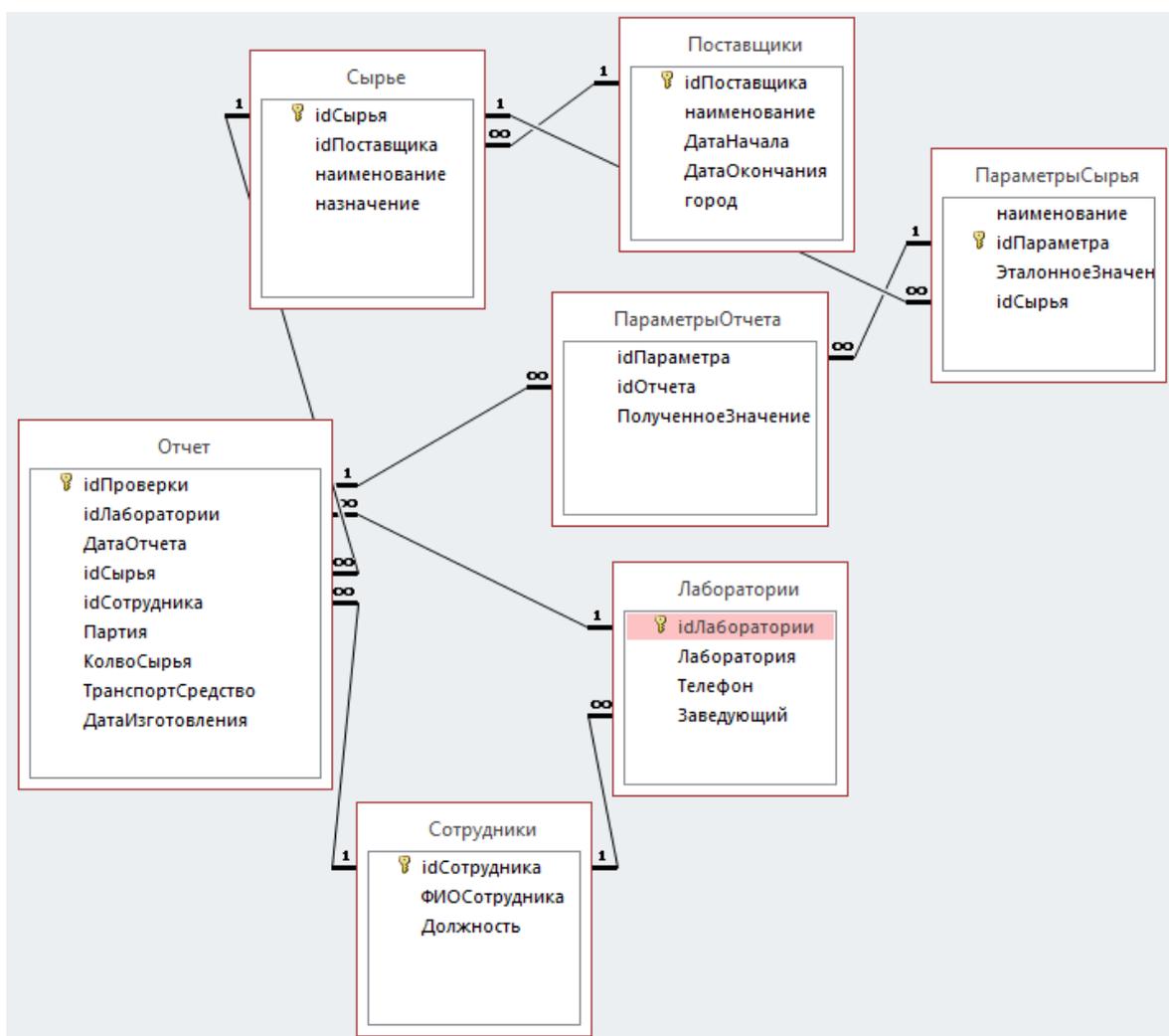


Рисунок 3.8 – Схема физической структуры данных в MS Access

3.3 Работа с модулем учета качества принимаемого сырья

В результате анализа учета качества принимаемого сырья масложирового дивизиона ОАО «ЭФКО» были составлены алгоритмы работы информационной системы учета качества принимаемого сырья [22, 23]. Алгоритм работы модуля учета качества представлен на рисунок 3.9.

При запуске программы с целью запуска алгоритма проведения проверок должны быть заполнены те справочные таблицы и данные параметров, без которых не возможен дальнейший ход работы программы:

- справочник «Сотрудники»;
- справочник «Лаборатории»;
- справочник «Поставщики»;
- справочник «Сырье»;
- проведение проверки и составление отчета;
- выгрузка результатов проверки в выходной форме формата Excel.



Рисунок 3.9 – Алгоритм работы модуля учета качества

Разберем алгоритм этапа формирования выходной формы подробнее (рисунок 3.10 – 3.11), т.к. алгоритм работы остальных этапов является не основным и производит работу лишь с базой данных:

- в самом начале работы алгоритма производится получение информации о проверках из таблицы «Отчеты» и подсчитывается их количество. Если же число проверок равно нулю (проверки не проводились), то тут алгоритм заканчивается, иначе производится запрос к базе данных по выбранному номеру проверки (или партии);

- далее выносится решение о разрешении запуска продукции в производство, т.е. если все полученные параметры отчета менее или равны эталонным, то выносится положительное заключение, иначе – отрицательное;

- производится запуск библиотеки Interop.Excel и в фоновом режиме создается рабочий лист книги;

- производится подготовка формы листа: объединяются нужные ячейки, заполняются шаблонными данными и производится форматирование ячеек;

- далее необходимые ячейки формы заполняются результатами запроса и сформированным решением о возможности приемки сырья;

- после этого происходит закрытие книги MS Excel, ее сохранение на диске и уведомление о том, что процесс прошел успешно.

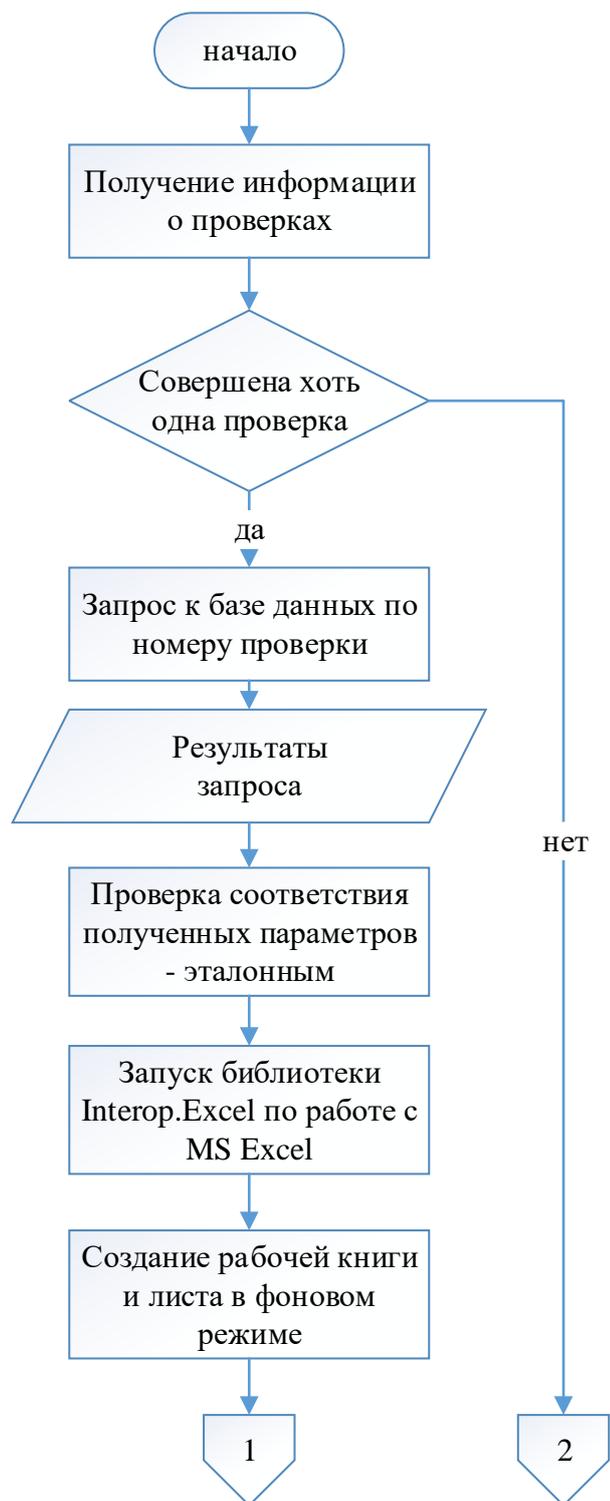


Рисунок 3.10 – Алгоритм работы формирования выходной формы проверки

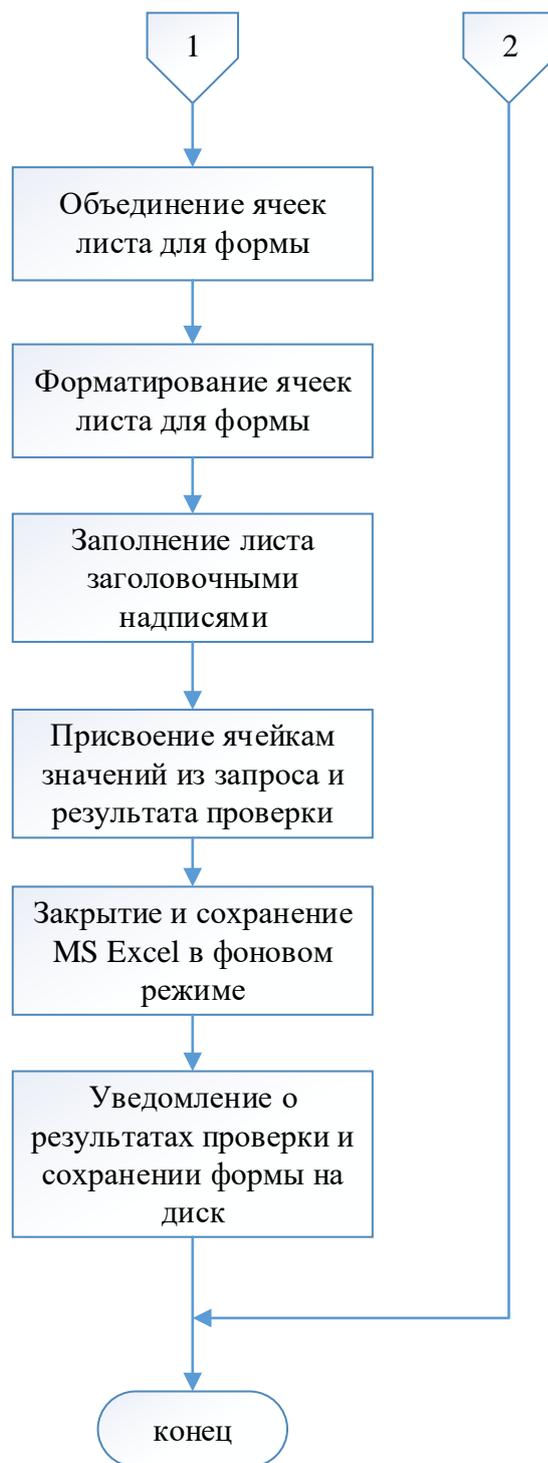


Рисунок 3.11 – Алгоритм работы формирования выходной формы проверки (окончание)

Сформированные алгоритмы (рисунок 3.9 – 3.11) были использованы при разработке модуля учета качества сырья на этапе получения.

Далее рассмотрим процесс работы непосредственно с программой.

Запуск программы осуществляется запуском файла Bin/Release/QualityModule.exe, в результате чего появляется окно как на рисунке 3.12.

Т.к. в модуль занесено два отчета, то система будет на главном окне показывать номер последней проверенной партии и дату и время этой проверки. Результаты этой проверки можно оперативно выгрузить, нажав на главной форме «Экспорт последних результатов в Excel», после чего отчет появится в папке reports.

Интерфейс программы также представлен выпадающим меню, включающим в себя:

- пункт меню «Справочники», где ведется информация о лабораториях, поставщиках, сотрудниках и сырье;
- пункт меню «Оценка качества», где ведется информация о параметрах качества сырья и можно выполнить проверку качества;
- пункт меню «Экспорт», где можно производить выгрузку результатов.

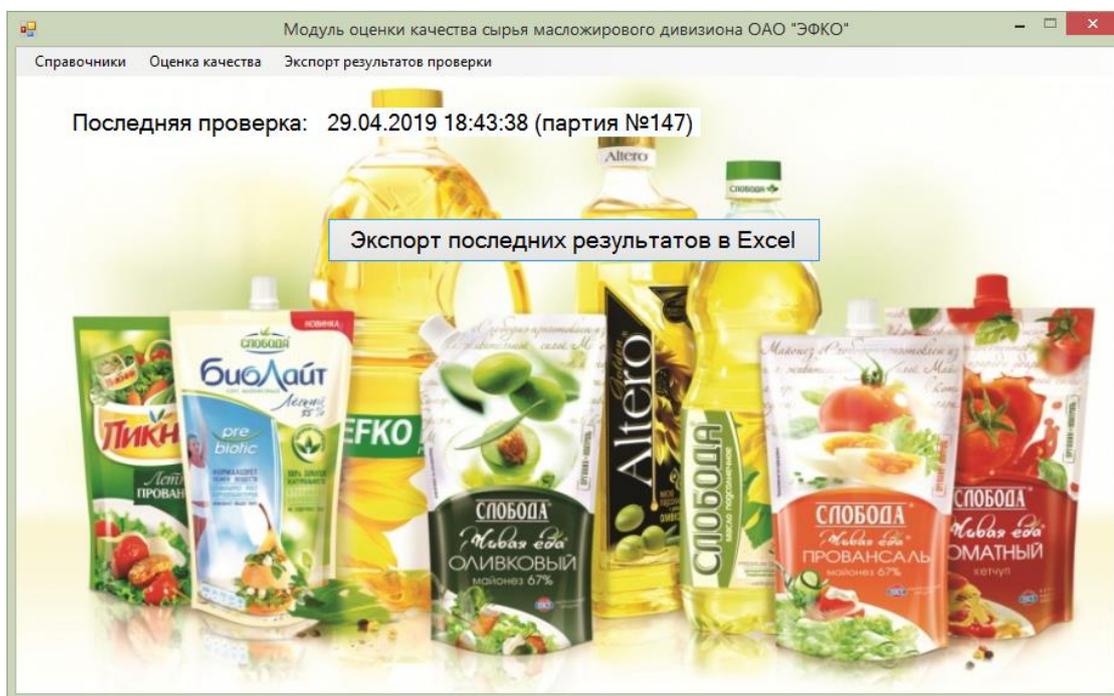


Рисунок 3.12 – Главное окно программы

Редактирование сотрудников и добавление их производится вызовом меню «Справочники – Сотрудники» (рисунок 3.13). Для добавления сотрудника требуется:

- заполнить поле «ФИО сотрудника»,
- заполнить поле «Должность», выбором из выпадающего списка, после чего нажать на «Добавить». Редактирование производится после выделения строки с записью в таблице, редактирования появившихся данных и нажатия «Изменить». Удаление производится аналогично, после чего данные сразу же отобразятся.

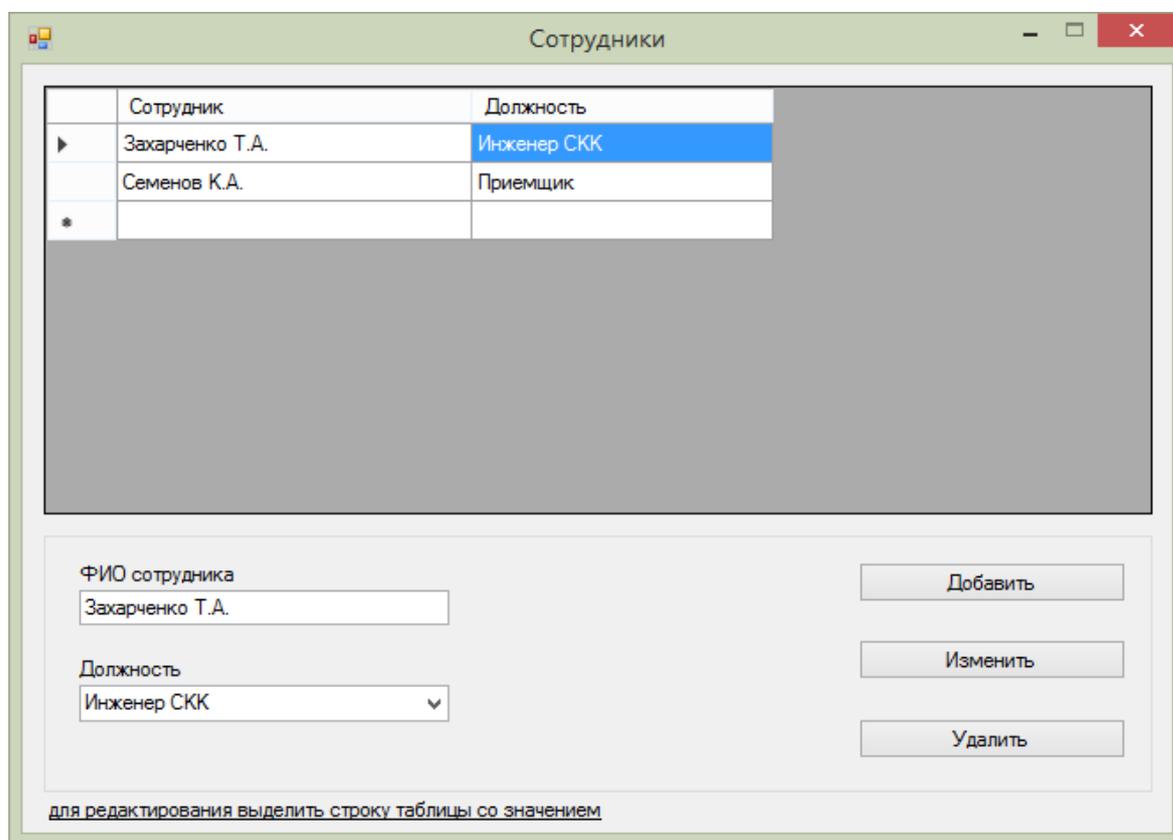


Рисунок 3.13 – Добавление сотрудников

Редактирование лабораторий и добавление их производится вызовом меню «Справочники – Лаборатории» (рисунок 3.14). Для добавления лаборатории требуется:

- заполнить поле «Лаборатория»,

- заполнить поле «Заведующий», выбором из выпадающего списка,
- заполнить поле «Телефон»,

после чего нажать на «Добавить». Редактирование производится после выделения строки с записью в таблице, редактирования появившихся данных и нажатия «Изменить». Удаление производится аналогично, после чего данные сразу же отобразятся.

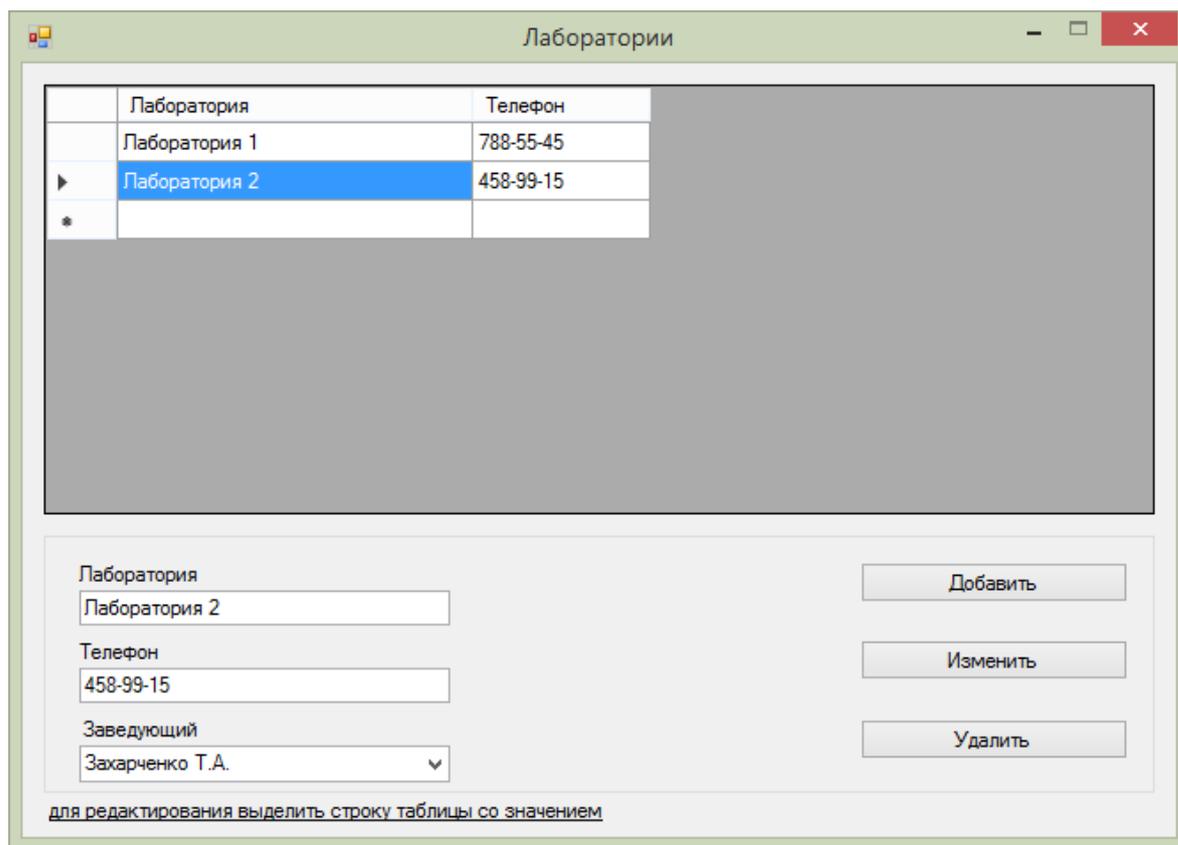


Рисунок 3.14 – Добавление лабораторий

Редактирование поставщиков и добавление их производится вызовом меню «Справочники – Поставщики» (рисунок 3.15). Для добавления поставщика требуется:

- заполнить поле «Наименование»,
- заполнить поле «Город»,
- заполнить поле «Дата начала», выбором даты из списка,
- заполнить поле «Дата окончания», выбором даты из списка,

после чего нажать на «Добавить». Редактирование производится после выделения строки с записью в таблице, редактирования появившихся данных и нажатия «Изменить». Удаление производится аналогично, после чего данные сразу же отобразятся.

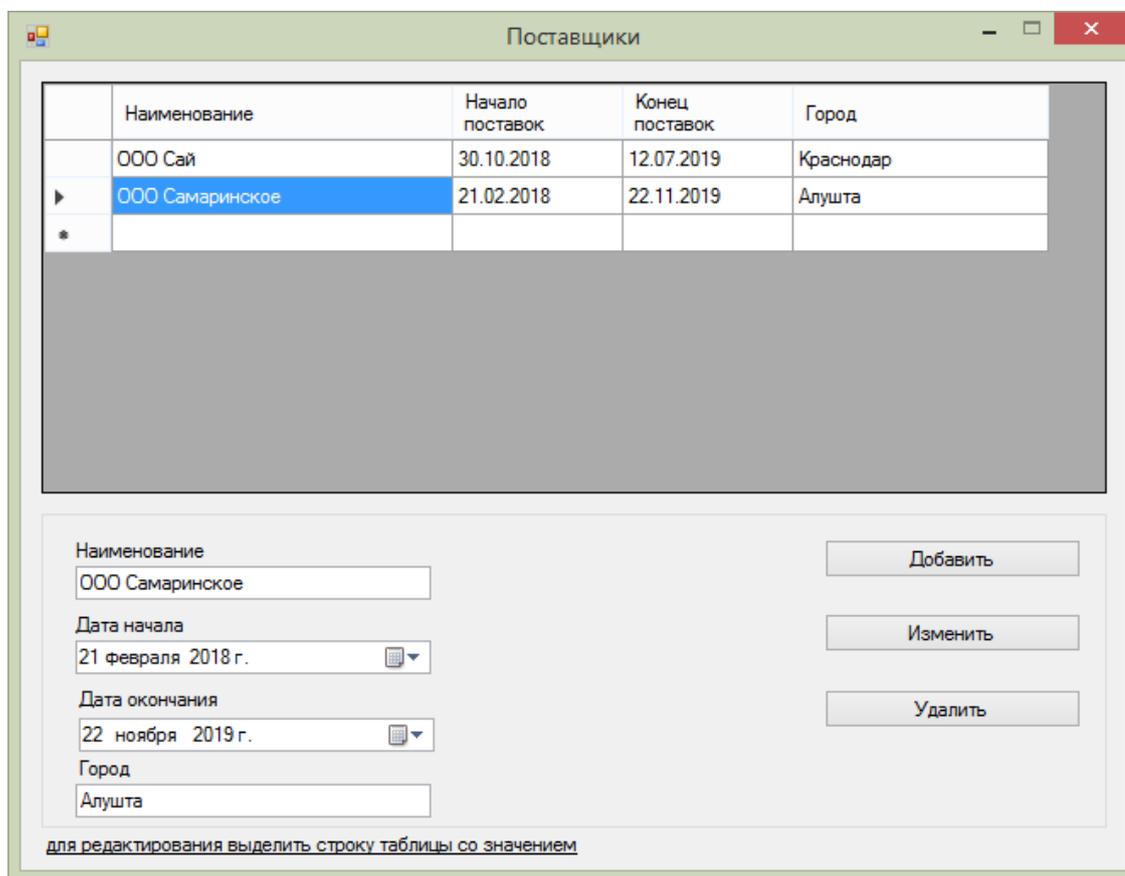


Рисунок 3.15 – Добавление поставщиков

Редактирование сырья и добавление их производится вызовом меню «Справочники – Сырье» (рисунок 3.16). Для добавления сырья требуется:

- заполнить поле «Поставщик», выбором из списка,
- заполнить поле «Наименование»,
- заполнить поле «Назначение»,

после чего нажать на «Добавить». Редактирование производится после выделения строки с записью в таблице, редактирования появившихся данных и нажатия «Изменить». Удаление производится аналогично, после чего данные сразу же отобразятся.

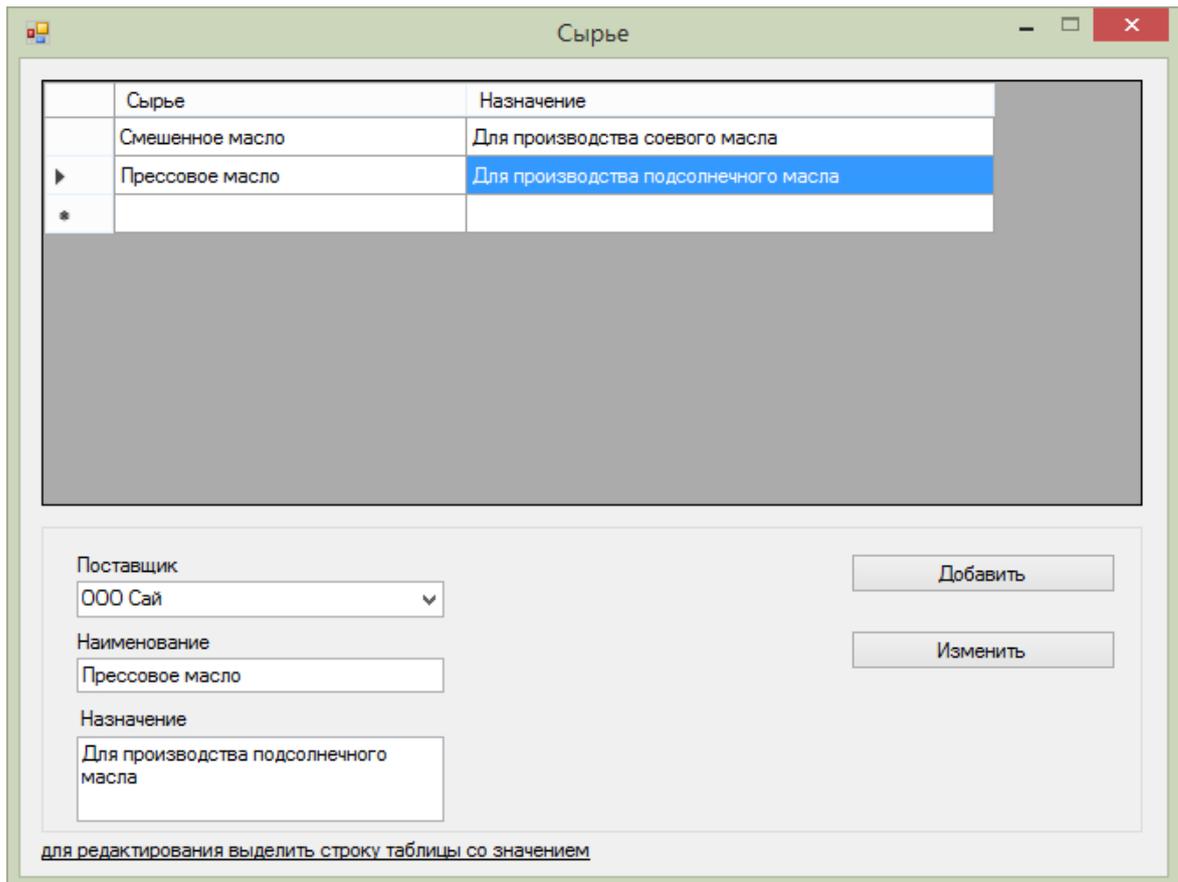


Рисунок 3.16 – Добавление сырья

Редактирование параметров сырья и добавление их производится вызовом меню «Оценка качества – Параметры сырья» (рисунок 3.17). Для добавления параметров сырья требуется:

- заполнить поле «Наименование»,
- заполнить поле «Эталонное значение»,
- заполнить поле «Сырье», выбором из списка,

после чего нажать на «Добавить». Редактирование производится после выделения строки с записью в таблице, редактирования появившихся данных и нажатия «Изменить». После чего данные сразу же отобразятся.

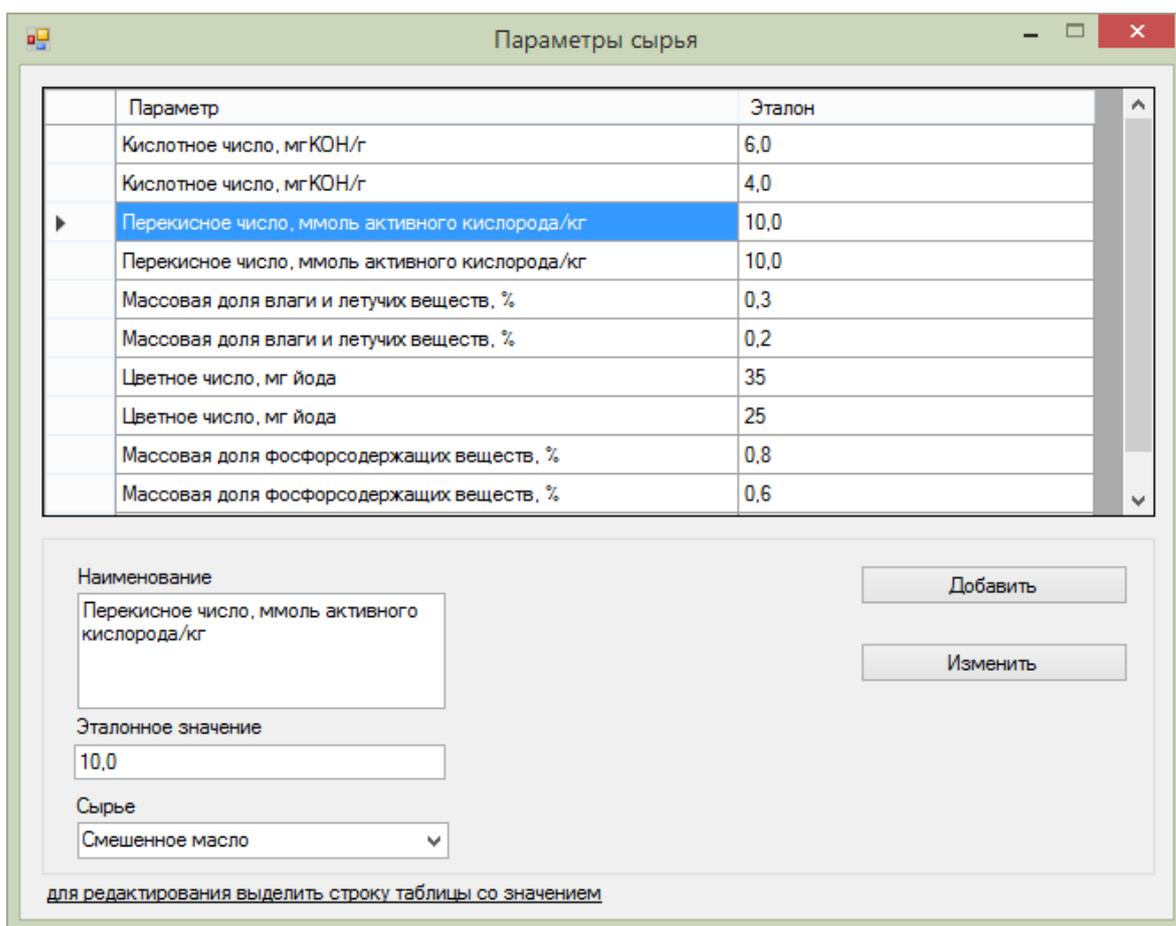


Рисунок 3.17 – Добавление параметров проверки качества сырья

Проведение проверки производится вызовом меню «Оценка качества – Проведение проверки» (рисунок 3.18). Для осуществления проведения проверки требуется:

- заполнить поле «Сотрудник», выбором из списка сотрудника, производящего снятие проб,
- заполнить поле «Лаборатория», выбором из списка лаборатории, которая будет производить анализ качества проб,
- заполнить поле «Транспортное средство», указанием госномера автомобиля,
- заполнить поле «Номер партии» по накладной, либо указать номер самой накладной,
- заполнить поле «Количество», где указать количество принимаемого сырья,

- заполнить поле «Дата изготовления», выбором даты из списка, либо оставить его без изменений,
- заполнить поле «Сырье», выбором из списка сырья, для которого будет производиться анализ качества проб и нажать «выбрать».

После выбора сырья появится таблица с параметрами качества этого сырья, куда следует занести значения, которые выявились в ходе проверки, после чего нажать на «Добавить». Добавление возможно лишь после указания всех значений параметров. После чего отобразится уведомление что отчет создан.

Параметр	Значение
Кислотное число, мгКОН/г	1
▶ Перекисное число, ммоль активного кислорода/кг	0,8
Массовая доля влаги и летучих веществ, %	6
Цветное число, мг йода	14,8
Массовая доля фосфорсодержащих веществ, %	0,7

Рисунок 3.18 – Проведение проверки качества сырья

Проведение проверки производится вызовом меню «Экспорт результатов проверки» (рисунок 3.19). Для осуществления экспорта результатов проверки требуется выбрать номер партии для экспорта из списка и нажать «Экспортировать». После этого запустится алгоритм экспорта выходной формы и отобразятся результаты проверки в виде уведомления (рисунок 3.20).

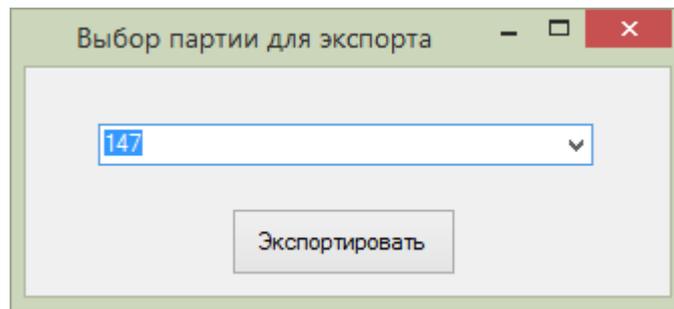


Рисунок 3.19 – Выбор партии для экспорта

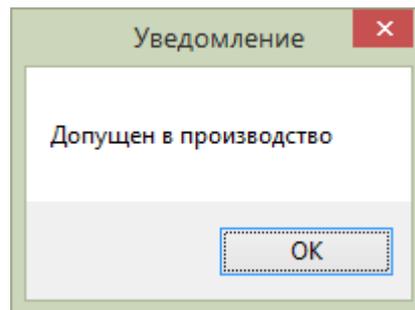


Рисунок 3.20 – Уведомление о решении по партии

После нажатия на «ОК», производится формирование выходной формы с результатом проверки в формат MS Excel и его сохранение под именем в формате «control_номер партии». В случае, если данный файл уже существует, появится уведомление о необходимости перезаписать его (рисунок 3.21), а после перезаписи – уведомление, что запись уже окончена с путем, где находится отчет (рисунок 3.22).

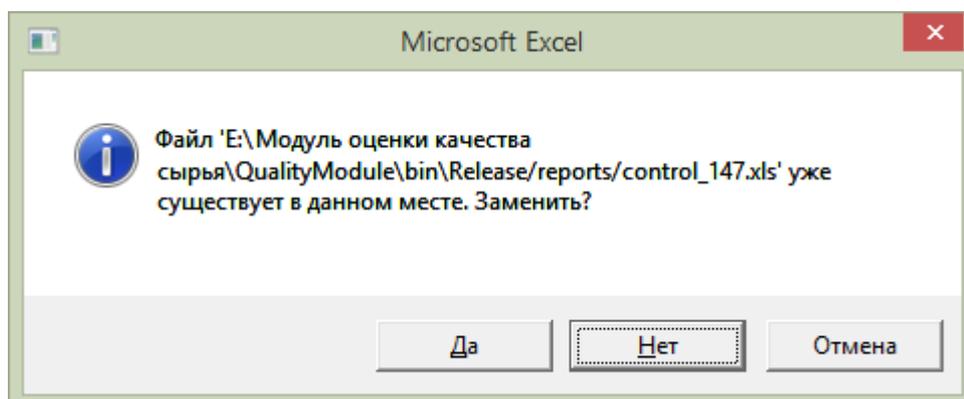


Рисунок 3.21 – Уведомление о сохранении формы

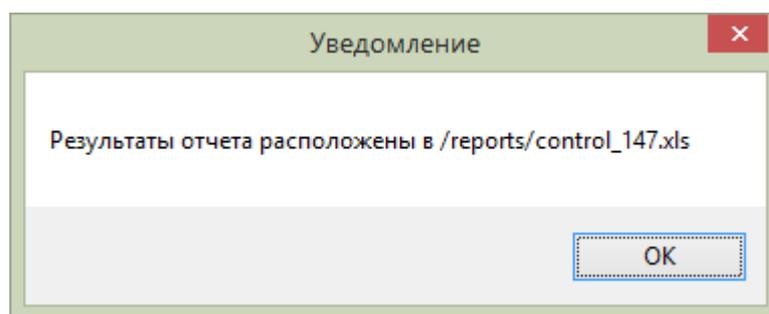


Рисунок 3.22 – Уведомление о местоположении результатов отчета

Представленный отчет можно распечатать, зайдя в папку reports рядом с запускаемым файлом программы (рисунок 3.23) и выбрав необходимый отчет.

Имя	Дата изменения	Тип	Размер
 control_147	30.04.2019 13:21	Лист Microsoft Ex...	30 КБ

Рисунок 3.23 – Вид папки со сформированной выходной формой

Внешний вид полученной формы в режиме MS Excel и в режиме печати показан в Приложении Б. Таким образом, разработанная программа позволяет автоматизировать процесс учета качества принимаемого сырья ОАО «ЭФКО», исключив необходимость занесения результатов проверки в выходную форму входного контроля вручную. Тем самым исчезает вероятность внесения в форму некорректных данных, связанных с человеческим фактором.

3.4 Оценка экономического эффекта спроектированного модуля учета качества принимаемого сырья ОАО «ЭФКО»

Исходными данными для расчета экономической эффективности проекта являются затраты на подготовку и реализацию проекта, объем входной и выходной информации, различные коэффициенты, фонд рабочего

времени, используя руководство [30]. Значение этих показателей приведены в таблице 3.2.

Таблица 3.2 – Основные показатели, используемые для расчета экономической эффективности программного средства

Наименование показателя	Величина показателя	
	в базовом варианте	в новом варианте
Предпроизводственные затраты, т. руб. (Кп)	-	27,00
Капитальные вложения, т. руб. (Кк)	-	16,00
Годовой объем информации, тыс. зн.:		
Входной (Qвх)	210,00	235,00
Выходной (Qвых)	200,00	210,00
Норма выработки при ручной обработке информации, зн./час (Нв)	700	-
Коэффициент, учитывающий дополнительные затраты времени на логические операции при ручной обработке информации, (Гд)	1,15	-
Среднемесячная зарплата пользователя, руб (Зм)	14000,00	16500,00
Норма выработки при подготовке информации тыс.зн./час (Нва)	-	5,0
Коэффициент начисления на заработную плату, (W)	1,4	1,4
Затраты времени на обработку 1000 зн. Информации с использованием ПС ВТ час/тыс.зн. (Тз)	-	0,5
Стоимость одного машинного часа работы ПЭВМ, руб./час (Цм)	-	22 0,00
Коэффициент, учитывающий накладные расходы (Кн)	2	2
Среднемесячный фонд рабочего времени, час (Т)	167, 8	16 7,8
Граничная величина коэффициента эффективности капитальных вложений (Е)	-	0,3 3

Рассчитываем стоимость одного часа ручной обработки информации по формуле:

$$Ц_p = З_m \times W \times К_n / T$$

(1)

$$Ц_p = 14000 \times 1,4 \times 2/167,8 = 233,611 \text{ руб./час.}$$

Затраты на подготовку и обработку информации в базовом варианте:

$$C_p = (Q_{\text{вх.б.}} + Q_{\text{вых.б.}}) \times Ц_p \times \Gamma_{\text{д/Нв}} \quad (2)$$

$$C_p = (210 + 200) \times 233,611 \times 1,15/700 = 157,354 \text{ т. руб.}$$

Затраты на подготовку информации для реализации функций, автоматизированных в ПС ВТ:

$$C_{\text{п}} = (Ц_p + Ц_{\text{м}}) \times Q_{\text{вх.н/Нвыр.а}} \quad (3)$$

$$C_{\text{п}} = (233,611 + 350) \times (235/5,0) = 21,319 \text{ тыс. руб.}$$

Затраты машинного времени в новом варианте на обработку информации:

$$T_{\text{м}} = (Q_{\text{вх.н.}} + Q_{\text{вых.н.}}) \times T_{\text{з}} \quad (4)$$

$$T_{\text{м}} = (235 + 210) \times 0,5 = 222,5 \text{ часов.}$$

Затраты на машинное время для реализации функций, автоматизированных при использовании внедренного ПС ВТ:

$$C_{\text{об}} = T_{\text{м}} \times Ц_{\text{м}} \quad (5)$$

$$C_{об} = 222,5 \times 220 = 48,950 \text{ тыс. руб.}$$

Затраты на обработку информации при использовании ПС ВТ:

$$C_M = C_{П} + C_{об} \tag{6}$$

$$C_M = 21,319 + 48,950 = 70,269 \text{ тыс. руб.}$$

Экономия затрат на обработку информации, связанная с внедрением ПС ВТ:

$$\mathcal{E} = C_{р} - C_M \tag{7}$$

$$\mathcal{E} = 157,354 - 70,269 = 87,084 \text{ тыс. руб.}$$

Расчет единовременных затрат на создание и внедрение ПС ВТ:

$$K = K_{П} + K_{к} \tag{8}$$

$$K = 27 + 16 = 43 \text{ тыс. руб.}$$

Расчетный коэффициент эффективности капитальных вложений:

$$E = \mathcal{E}/K \tag{9}$$

$$E = 87,084/47 = 2,025$$

Срок окупаемости капитальных вложений на создание и разработку ПС
ВТ:

$$T_0 = K/\dot{Э}$$

10)

$$T_0 = 47/87,084 = 0,494 \text{ года}$$

Проведенные расчеты показали, что годовой экономический эффект от внедрения задачи составляет 47 тыс. руб., расчетный коэффициент эффективности капитальных вложений составляет 2,025, что превышает значение нормативного коэффициента, принимаемого здесь равным 0,33, срок окупаемости капитальных вложений равен 0,494 года.

Таким образом, проведенные расчеты говорят о том, что разработка и внедрение модуля оценки качества эффективна и целесообразна.

Выводы по главе: в качестве средств реализации выбрана среда MS Visual Studio и язык программирования C#, являющийся в последнее время особенно актуальным.

Разработанный модуль оценки качества поступающего сырья позволяет работать с параметрами оценки для каждого вида сырья и производить занесение параметров оценки для каждой партии сырья, поступающего автоцистернами.

В результате анализа оценки качества принимаемого сырья масложирового дивизиона ОАО «ЭФКО» были составлены алгоритмы работы информационной системы оценки качества принимаемого сырья, на основании которых должны быть заполнены те справочные таблицы и данные параметров, без которых не возможен дальнейший ход работы программы:

- справочник «Сотрудники»;

- справочник «Лаборатории»;
- справочник «Поставщики»;
- справочник «Сырье»;
- проведение проверки и составление отчета;
- выгрузка результатов проверки в выходной форме формата Excel.

Разработанная программа позволяет автоматизировать процесс оценки качества принимаемого сырья ОАО «ЭФКО», исключив необходимость занесения результатов проверки в выходную форму входного контроля вручную. Тем самым исчезает вероятность внесения в форму некорректных данных, связанных с человеческим фактором.

Годовой экономический эффект разработанного модуля говорит о том, что срок окупаемости капитальных вложений равен 0,494 года, что говорит об эффективности разработки.

ЗАКЛЮЧЕНИЕ

В представленной выпускной работе интерес автоматизирован отдел, занимающийся оценкой качества сырья. В ходе анализа организационной структуры отдела, выявлено, что непосредственно основную работу по оценке качества продуктов, используя стандарты качества ГОСТ Р ИСО 22000-2007 и ГОСТ Р 51705.1-2001 производит отдел нормоконтроля.

Для рассмотрения характеристик сырья выбраны три основные брендовые марки масложировой продукции ОАО «ЭФКО»: линейка растительных масел «Слобода», линейка растительных масел Altero и продукция марки EFKO FOOD professional.

В рамках продукции масложирового дивизиона рассмотрим характеристики качества сырья, используемого в наибольшем количестве. К нему относится такое сырье:

- смешанное масло;
- прессовое масло.

Контроль входного сырья включает в себя оценку таких параметров:

- Кислотное число;
- Перекисное число;
- Массовая доля влаги и летучих веществ;
- Цветное число;
- Массовая доля фосфорсодержащих веществ (в пересчете на

стеароолеолецитин).

В ОАО «ЭФКО» ведется учет поступивших заявок в накопительной папке отдела снабжения, а также в системе электронного документооборота предоставляются отчетные формы. В связи с этим, имелась необходимость разработки автоматизированной информационной системы проверки параметров для каждой единицы сырья, которая будет анализировать выходные параметры поступающего сырья и выносить окончательное

решение о необходимости приемки или отказа от приемки партии. На данный момент формы входного контроля формируются вручную.

В качестве средств реализации выбрана среда MS Visual Studio и язык программирования C#, являющийся в последнее время особенно актуальным.

Разработанный модуль оценки качества поступающего сырья позволяет работать с параметрами оценки для каждого вида сырья и производить занесение параметров оценки для каждой партии сырья, поступающего автоцистернами.

В результате анализа оценки качества принимаемого сырья масложирового дивизиона ОАО «ЭФКО» были составлены алгоритмы работы информационной системы оценки качества принимаемого сырья, на основании которых должны быть заполнены те справочные таблицы и данные параметров, без которых не возможен дальнейший ход работы программы:

- справочник «Сотрудники»;
- справочник «Лаборатории»;
- справочник «Поставщики»;
- справочник «Сырье»;
- проведение проверки и составление отчета;
- выгрузка результатов проверки в выходной форме формата Excel.

Разработанная программа позволяет автоматизировать процесс оценки качества принимаемого сырья ОАО «ЭФКО», исключив необходимость занесения результатов проверки в выходную форму входного контроля вручную. Тем самым исчезает вероятность внесения в форму некорректных данных, связанных с человеческим фактором.

Срок окупаемости капитальных вложений равен 0,494 года, что говорит об эффективности разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р ИСО 22000-2007. Система менеджмента безопасности пищевой продукции. Требования к организациям, участвующим в цепи создания пищевой продукции
2. ГОСТ Р 51705.1-2001. Управление качеством пищевых продуктов на основе принципов ХАССП. Общие требования
3. Автоматизация контроля качества [Электронный ресурс]. – Режим доступа: http://altamisoft.ru/article/quality_control_automation/ (дата обращения: 24.03.2019)
4. Автоматизированные системы контроля качества [Электронный ресурс]. – Режим доступа: <https://ritm.pro/resheniya/avtomatizirovannye-ispysatelnue-stendy> (дата обращения: 24.03.2019)
5. Система управления предприятием ИС-Про [Электронный ресурс]. – Режим доступа: <http://is-pro.ru> (дата обращения: 24.03.2019)
6. Агарков, А.П. Управление качеством: Учебное пособие/ А.П. Агарков. – М.: Дашков и К, 2015. – 228 с.
7. Oracle Россия и СНГ [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/ru/INDEX.HTML> (дата обращения: 24.03.2019)
8. MS Access [Электронный ресурс]. – Режим доступа: <https://products.office.com/en-us/Access?rtc=1> (дата обращения: 24.03.2019)
9. Система управления базами данных Microsoft SQL Server 2012 [Электронный ресурс]. – Режим доступа: <https://www.lankey.ru/kis/sql-server/> (дата обращения: 24.03.2019)
10. SQL Server 2016 [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016> (дата обращения: 24.03.2019)
11. Инструменты IBM Db2 [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/ru-ru/analytics/db2/tools> (дата обращения: 24.03.2019)

12. Продукты питания [Электронный ресурс]. – Режим доступа: <http://www.efko.ru/produkciya/brendy/234/> (дата обращения: 24.03.2019)

13.ГОСТ Р ИСО 22000-2007 «Системы менеджмента безопасности пищевой продукции. Требования к организациям, участвующим в цепи создания пищевой продукции»

14.ГОСТ Р 51705.1-2001 «Управление качеством пищевых продуктов на основе принципов ХАССП. Общие требования»

15.Осипова, Г. А. Методы контроля качества сырья, полуфабрикатов и макаронных изделий [Текст]: учебное пособие. Ч. 1. / Г. А. Осипова, С. Я. Корячкина, Н. А. Березина. – Орел: ФГБОУ ВПО «Госунiversитет – УНПК», 2013. - 130 с.

16. ГОСТ 31933-2012 «Масла растительные. Методы определения кислотного числа»

17. ГОСТ 26593-85 «Масла растительные. Метод измерения перекисного числа»

18. ГОСТ 11812-66 «Масла растительные. Методы определения влаги и летучих веществ»

19. ГОСТ 5477-2015 «Масла растительные. Методы определения цветности»

20. ГОСТ 31753-2012 «Масла растительные. Методы определения фосфорсодержащих веществ»

21. ГОСТ 32190-2013 «Масла растительные. Правила приемки и методы отбора проб»

22. ГОСТ 19.002-80. Схема алгоритмов и программ. Правила выполнения

23. ГОСТ 19.701-90 (ИСО 5807-85) Единая система программной документации (ЕСПД). Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения

24. Попова Л.К. Проектирование информационных систем в экономике [Текст]: учебное пособие для студентов специальности 0719.00

«Информационные системы в экономике». – Ростов-на-Дону.: РГЭА, 2015 – 152 с.

25. Жизненный цикл и процессы разработки ПО [Электронный ресурс] // НОУ «ИНТУИТ». URL: <https://www.intuit.ru/studies/courses/64/64/lecture/1868> (дата обращения: 26.05.2019).

26. Дембовский, В. В. Автоматизация управления производством. - СПб.: СЗТУ, 2011. – 82 с.

27. Ботыгин И.А. Исследование схем распределенного информационного взаимодействия компонентов программных комплексов с базами данных / Каликин К.А. Ботыгин И.А Известия Томского политехнического университета. 2015. Т. 312. № 5

28. Безручко В.Т. Информатика [Текст]: учебное пособие / В.Т. Безручко. – М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. – 432 с.

29. Информатика. Базовый курс[Текст]: учебное пособие для студентов техн. вузов / под ред. С.В. Симоновича. – 3-е изд. – СПб.: Питер, 2014. – 640 с.

30. Абрамов, А.А. Методические указания по расчету показателей экономической эффективности дипломного проекта для специальности 080801 «Прикладная информатика (в экономике)» /Гришина Н.И., Абрамов А.А. Н.Новгород. Изд-во НГУ, 2013. – 23 с.

ПРИЛОЖЕНИЕ А

Листинг программы

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;
namespace QualityModule
{
    class ExportExcel
    {
        public void StartExcel(DataGridView grid,
            DataGridView grid2, string save, string title)
        {
            Excel.Application xlApp;
            Excel.Workbook xlWorkBook;
            Excel.Worksheet xlWorkSheet;
            object misValue
            = System.Reflection.Missing.Value;

            Int16 i;
            xlApp = new Excel.Application();
            xlWorkBook
            = xlApp.Workbooks.Add(misValue);

            xlWorkSheet
            = (Excel.Worksheet)
            xlWorkBook.Worksheets.get_Item(1);

            //ТЕКТОВЫЕ ДАННЫЕ
            Excel.Range _excelCells1
            = (Excel.Range)xlWorkSheet.get_Range("A1",
            "I1").Cells;
            _excelCells1.Merge(Type.Missing);
            xlWorkSheet.Cells[1, 1] = "Входной
            контроль"; xlWorkSheet.get_Range("A1",
            "A1").Font.Size = 16;

            xlWorkSheet.get_Range("A1").HorizontalAlignment
            = -4108; //-4131 лево, -4152 право

            Excel.Range _excelCells2
            = (Excel.Range)xlWorkSheet.get_Range("A2",
            "I2").Cells;
            _excelCells2.Merge(Type.Missing);
            xlWorkSheet.Cells[2, 1] = title + " на запуск
            продукции в производство № " +
            grid.Rows[0].Cells[0].Value;
            xlWorkSheet.get_Range("A2", "A2").Font.Size = 12;

            xlWorkSheet.get_Range("A2").HorizontalAlignment
            = -4108;

            Excel.Range _excelCells3
            = (Excel.Range)xlWorkSheet.get_Range("A3",
            "I3").Cells;
            _excelCells3.Merge(Type.Missing);
            xlWorkSheet.Cells[3, 1] = "От " +
            grid.Rows[0].Cells[1].Value;

            xlWorkSheet.get_Range("A3").HorizontalAlignment
            = -4108;

            Excel.Range _excelCells4
            = (Excel.Range)xlWorkSheet.get_Range("A5",
            "C5").Cells; _excelCells4.Merge(Type.Missing);
            Excel.Range _excelCells5
            = (Excel.Range)xlWorkSheet.get_Range("A6",
            "C6").Cells; _excelCells5.Merge(Type.Missing);
            Excel.Range _excelCells6
            = (Excel.Range)xlWorkSheet.get_Range("A7",
            "C7").Cells; _excelCells6.Merge(Type.Missing);
            Excel.Range _excelCells7
            = (Excel.Range)xlWorkSheet.get_Range("A8",
            "C8").Cells; _excelCells7.Merge(Type.Missing);
            Excel.Range _excelCells8
            = (Excel.Range)xlWorkSheet.get_Range("A9",
            "C9").Cells; _excelCells8.Merge(Type.Missing);
            Excel.Range _excelCells9
            = (Excel.Range)xlWorkSheet.get_Range("D5",
            "I5").Cells; _excelCells9.Merge(Type.Missing);
            Excel.Range _excelCells10
            = (Excel.Range)xlWorkSheet.get_Range("D6",
            "I6").Cells; _excelCells10.Merge(Type.Missing);
            Excel.Range _excelCells11
            = (Excel.Range)xlWorkSheet.get_Range("D7",
            "I7").Cells; _excelCells11.Merge(Type.Missing);
            Excel.Range _excelCells12
            = (Excel.Range)xlWorkSheet.get_Range("D8",
            "I8").Cells; _excelCells12.Merge(Type.Missing);
            Excel.Range _excelCells13
            = (Excel.Range)xlWorkSheet.get_Range("D9",
            "I9").Cells; _excelCells13.Merge(Type.Missing);

            xlWorkSheet.Cells[5, 1] = "Наименование
            продукции";
            xlWorkSheet.get_Range("D5", "I5").Borders[Excel.XlB
            ordersIndex.xlEdgeBottom].LineStyle = true;
            xlWorkSheet.Cells[5, 4]
            = grid.Rows[0].Cells[2].Value;
            xlWorkSheet.Cells[6, 1] = "Изготовитель";
            xlWorkSheet.get_Range("D6",
            "I6").Borders[Excel.XlBordersIndex.xlEdgeBottom].Li
            neStyle = true; xlWorkSheet.get_Range("D7",
```

```

"I7").Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = true;
xlWorkSheet.Cells[6, 4] = grid.Rows[0].Cells[3].Value;
xlWorkSheet.Cells[7, 1] = "Дата изготовления";
xlWorkSheet.get_Range("G8", "I8").Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = true;
xlWorkSheet.Cells[7, 4] = grid.Rows[0].Cells[4].Value;
xlWorkSheet.Cells[8, 1] = "№ партии";
xlWorkSheet.Cells[8, 4] = grid.Rows[0].Cells[5].Value;
xlWorkSheet.Cells[8, 6] = "а/м";
xlWorkSheet.Cells[8, 7] = grid.Rows[0].Cells[6].Value;
xlWorkSheet.get_Range("D8", "E8").Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = true;
xlWorkSheet.Cells[9, 1] = "Количество";
xlWorkSheet.get_Range("D9", "I9").Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = true;
xlWorkSheet.Cells[9, 4] = grid.Rows[0].Cells[7].Value;

//табличные данные

for (i = 0; i <= grid.RowCount - 2; i++)
{
    xlWorkSheet.Cells[i + 10, 1] = grid.Rows[i].Cells[8].Value.ToString() + ": " + grid.Rows[i].Cells[9].Value.ToString() + " (норма - не более " + grid.Rows[i].Cells[10].Value.ToString() + ")";
}
xlWorkSheet.Cells[10 + grid.RowCount, 1] = "Инженер СКК";
xlWorkSheet.Cells[10 + grid.RowCount, 5] = "_____";
xlWorkSheet.Cells[10 + grid.RowCount, 7] = grid2.Rows[0].Cells[1].Value.ToString();

xlWorkbook.SaveAs(@save, Excel.XlFileFormat.xlWorkbookNormal, misValue, misValue, misValue, misValue, Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue, misValue, misValue);
xlWorkbook.Close(true, misValue, misValue);
xlApp.Quit();

releaseObject(xlWorkSheet);
releaseObject(xlWorkbook);
releaseObject(xlApp);
}

private void releaseObject(object obj)
{
    try
    {
        obj = null;
    }
    catch (Exception ex)
    {
        obj = null;
        MessageBox.Show("Exception Occured while releasing object " + ex.ToString());
    }
    finally
    {
        GC.Collect();
    }
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Linq;
using System.Xml;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public string strSQL;
        database db = new database();
        private void Form1_Load(object sender, EventArgs e)
        {
            strSQL = "SELECT Max(Отчет.ДатаОтчета) AS [Max-ДатаОтчета], Отчет.Партия, Отчет.idПроверки FROM Отчет GROUP BY Отчет.Партия, Отчет.idПроверки";
            db.selectSQL(strSQL, dataGridView1);
            label2.Text = "не проводилась";
            if (dataGridView1.Rows[0].Cells[0].Value != null) {
                label2.Text = dataGridView1.Rows[0].Cells[0].Value.ToString() + " (партия №"+dataGridView1.Rows[0].Cells[1].Value.ToString()+")";
                label3.Text = dataGridView1.Rows[0].Cells[2].Value.ToString();
                экспортToolStripMenuItem.Enabled = true;
            }
        }
    }
}

```

```

private void СотрудникиToolStripMenuItem_Click(object sender, EventArgs e)
{
    sotr s = new sotr();
    s.Show();
}

private void ЛабораторииToolStripMenuItem_Click(object sender, EventArgs e)
{
    labs s = new labs();
    s.Show();
}

private void ПоставщикиToolStripMenuItem_Click(object sender, EventArgs e)
{
    postav s = new postav();
    s.Show();
}

private void СырьеToolStripMenuItem_Click(object sender, EventArgs e)
{
    sir s = new sir();
    s.Show();
}

private void ПараметрыСырьяToolStripMenuItem_Click(object sender, EventArgs e)
{
    @params s = new @params();
    s.Show();
}

private void ПроведениеПроверкиToolStripMenuItem_Click(object sender, EventArgs e)
{
    report s = new report();
    s.Show();
}

private void ЭкспортToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows[0].Cells[0].Value != null)
    {
        string str = "Разрешение";
        strSQL = "SELECT Отчет.idПроверки, Отчет.ДатаОтчета, Сырье.наименование, Поставщики.наименование, Отчет.ДатаИзготовления, Отчет.Партия, Отчет.ТранспортСредство, Отчет.КолвоСырья, ПараметрыСырья.наименование, ПараметрыОтчета.ПолученноеЗначение, ПараметрыСырья.ЭталонноеЗначение FROM (Поставщики INNER JOIN Сырье ON Поставщики.idПоставщика = Сырье.idПоставщика) INNER JOIN (ПараметрыСырья INNER JOIN (Отчет INNER JOIN ПараметрыОтчета ON Отчет.idПроверки = ПараметрыОтчета.idОтчета) ON ПараметрыСырья.idПараметра = ПараметрыОтчета.idПараметра) ON (Сырье.idСырья = ПараметрыСырья.idСырья) AND (Сырье.idСырья = Отчет.idСырья) WHERE (((Отчет.idПроверки)=" + Convert.ToInt32(label3.Text) + ")";
        db.selectSQL(strSQL, dataGridView1);
        strSQL = "SELECT Отчет.idПроверки, Сотрудники.ФИОСотрудника FROM Сотрудники INNER JOIN (Лаборатории INNER JOIN Отчет ON Лаборатории.idЛаборатории = Отчет.idЛаборатории) ON Сотрудники.idСотрудника = Лаборатории.Заведующий WHERE (((Отчет.idПроверки)=" + Convert.ToInt32(label3.Text) + ")";
        db.selectSQL(strSQL, dataGridView2);
        for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
        {
            if (Convert.ToDouble(dataGridView1.Rows[i].Cells[9].Value) > Convert.ToDouble(dataGridView1.Rows[i].Cells[10].Value)) { str = "Запрещение"; }
            if (str == "Запрещение") { MessageBox.Show("Запуск запрещен", "Уведомление"); } else MessageBox.Show("Запуск разрешен", "Уведомление");
            ExportExcel ee = new ExportExcel();
            ee.StartExcel(dataGridView1, dataGridView2, Environment.CurrentDirectory + "/reports/control_" + dataGridView1.Rows[0].Cells[5].Value.ToString() + ".xls", str);
            MessageBox.Show("Результаты отчета расположены в /reports/control_" + dataGridView1.Rows[0].Cells[5].Value.ToString() + ".xls", "Уведомление");
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    proverki s = new proverki();
    s.t = "Excel";
    s.Show();
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class labs : Form
    {
        public labs()
        {
            InitializeComponent();
        }
        public string strSQL;
        int num;
        database db = new database();
        private void labs_Load(object sender, EventArgs e)
        {
            strSQL = "SELECT * FROM [Лаборатории]";
            db.selectSQL(strSQL, dataGrid);
            dataGrid.Columns[0].Visible = false;
            dataGrid.Columns[1].HeaderText = "Лаборатория";
            dataGrid.Columns[1].Width = 200;
            dataGrid.Columns[2].HeaderText = "Телефон";
            dataGrid.Columns[2].Width = 100;
            dataGrid.Columns[3].Visible = false;
            strSQL = "SELECT ФИОСотрудника,
            idСотрудника FROM [Сотрудники]";
            db.comboSQL(strSQL, comboBox1,
            "ФИОСотрудника", "idСотрудника");
        }

        private void addbutton_Click(object sender,
        EventArgs e)
        {
            strSQL = "INSERT INTO [Лаборатории]
            (Лаборатория, Телефон, Заведующий) VALUES ('" +
            textBox1.Text + ",'" + textBox2.Text + "','" +
            comboBox1.SelectedValue + "')";
            db.insertSQL(strSQL);
            strSQL = "SELECT * FROM [Лаборатории]";
            db.selectSQL(strSQL, dataGrid);
        }

        private void changebutton_Click(object sender,
        EventArgs e)
        {
            strSQL = "UPDATE [Лаборатории] SET
            Лаборатория='" + textBox1.Text + "', Телефон='" +
            textBox2.Text + "', Заведующий='" +
            comboBox1.SelectedValue + "' WHERE
            (idЛаборатории = " + num + ")";
            db.insertSQL(strSQL);
            strSQL = "SELECT * FROM [Лаборатории]";
            db.selectSQL(strSQL, dataGrid);
        }

        private void deletebutton_Click(object sender,
        EventArgs e)
        {
            strSQL = "DELETE FROM [Лаборатории]
            WHERE (idЛаборатории = " + num + ")";
            db.insertSQL(strSQL);
            strSQL = "SELECT * FROM [Лаборатории]";
            db.selectSQL(strSQL, dataGrid);
        }

        private void dataGrid_CellClick(object sender,
        DataGridViewCellEventArgs e)
        {
            num =
            (int)dataGrid.Rows[e.RowIndex].Cells[0].Value;
            textBox1.Text =
            dataGrid.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text =
            dataGrid.Rows[e.RowIndex].Cells[2].Value.ToString();
            comboBox1.SelectedValue =
            dataGrid.Rows[e.RowIndex].Cells[3].Value.ToString();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class @params : Form
    {
        public @params()
        {
            InitializeComponent();
        }
        public string strSQL;
        int num;
        database db = new database();
        private void params_Load(object sender,
        EventArgs e)
        {
            strSQL = "SELECT * FROM
            [ПараметрыСырья]";
            db.selectSQL(strSQL, dataGrid);
            dataGrid.Columns[1].Visible = false;
            dataGrid.Columns[3].Visible = false;
            dataGrid.Columns[0].HeaderText =
            "Параметр";
            dataGrid.Columns[0].Width = 350;
            dataGrid.Columns[2].HeaderText = "Эталон";
            dataGrid.Columns[2].Width = 200;
            strSQL = "SELECT наименование, idСырья
            FROM [Сырье]";
            db.comboSQL(strSQL, comboBox1,
            "наименование", "idСырья");
        }
    }
}

```

```

private void dataGrid_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    num
    (int)dataGrid.Rows[e.RowIndex].Cells[1].Value;
    comboBox1.Selected.Value
    dataGrid.Rows[e.RowIndex].Cells[3].Value.ToString();
    textBox1.Text
    dataGrid.Rows[e.RowIndex].Cells[0].Value.ToString();
    textBox2.Text
    dataGrid.Rows[e.RowIndex].Cells[2].Value.ToString();
}

private void addbutton_Click(object sender,
EventArgs e)
{
    strSQL = "INSERT INTO [ПараметрыСырья]
(наименование, ЭталонноеЗначение, idСырья)
VALUES ('" + textBox1.Text + "','" + textBox2.Text +
"', " + comboBox1.Selected.Value + ")";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM
[ПараметрыСырья]";
    db.selectSQL(strSQL, dataGrid);
}

private void changebutton_Click(object sender,
EventArgs e)
{
    strSQL = "UPDATE [ПараметрыСырья] SET
наименование='" + textBox1.Text + "',
ЭталонноеЗначение='" + textBox2.Text + "',
idСырья=" + comboBox1.Selected.Value + " WHERE
(idПараметра = " + num + ")";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM
[ПараметрыСырья]";
    db.selectSQL(strSQL, dataGrid);
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class postav : Form
    {
        public postav()
        {
            InitializeComponent();
        }
        public string strSQL;
        int num;
        database db = new database();

private void postav_Load(object sender, EventArgs
e)
{
    strSQL = "SELECT * FROM [Поставщики]";
    db.selectSQL(strSQL, dataGrid);
    dataGrid.Columns[0].Visible = false;
    dataGrid.Columns[1].HeaderText =
    "Наименование";
    dataGrid.Columns[1].Width = 200;
    dataGrid.Columns[2].HeaderText = "Начало
поставок";
    dataGrid.Columns[2].Width = 100;
    dataGrid.Columns[3].HeaderText = "Конец
поставок";
    dataGrid.Columns[3].Width = 100;
    dataGrid.Columns[4].HeaderText = "Город";
    dataGrid.Columns[4].Width = 150;
}

private void addbutton_Click(object sender,
EventArgs e)
{
    strSQL = "INSERT INTO [Поставщики]
(наименование, ДатаНачала, ДатаОкончания, город)
VALUES ('" + textBox1.Text + "','" +
date1.Value.ToShortDateString() + "','" +
date2.Value.ToShortDateString() + "','" +
textBox2.Text + "')";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM [Поставщики]";
    db.selectSQL(strSQL, dataGrid);
}

private void changebutton_Click(object sender,
EventArgs e)
{
    strSQL = "UPDATE [Поставщики] SET
наименование='" + textBox1.Text + "',
ДатаНачала=" +
date1.Value.ToShortDateString() + "',
ДатаОкончания=" + date2.Value.ToShortDateString()
+ "', город=" + textBox2.Text + "' WHERE
(idПоставщика = " + num + ")";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM [Поставщики]";
    db.selectSQL(strSQL, dataGrid);
}

private void deletebutton_Click(object sender,
EventArgs e)
{
    strSQL = "DELETE FROM [Поставщики]
WHERE (idПоставщика = " + num + ")";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM [Поставщики]";
    db.selectSQL(strSQL, dataGrid);
}

private void dataGrid_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    num
    (int)dataGrid.Rows[e.RowIndex].Cells[0].Value;
}
}
}

```

```

        textBox1.Text = INNER JOIN ПараметрыОтчета ON
dataGrid.Rows[e.RowIndex].Cells[1].Value.ToString(); Отчет.idПроверки = ПараметрыОтчета.idОтчета) ON
        date1.Value = ПараметрыСырья.idПараметра =
Convert.ToDateTime(dataGrid.Rows[e.RowIndex].Cells[2].Value); ПараметрыОтчета.idПараметра) ON (Сырье.idСырья
        date2.Value = ПараметрыСырья.idСырья) AND (Сырье.idСырья =
Convert.ToDateTime(dataGrid.Rows[e.RowIndex].Cells[3].Value); = Отчет.idСырья) WHERE (((Отчет.idПроверки)=" +
        textBox2.Text = comboBox1.SelectedValue + ")");
dataGrid.Rows[e.RowIndex].Cells[4].Value.ToString(); db.selectSQL(strSQL, dataGridView1);
        strSQL = "SELECT Отчет.idПроверки,
    } Сотрудники.ФИОСотрудника FROM Сотрудники
    } INNER JOIN (Лаборатории INNER JOIN Отчет ON
using System; Лаборатории.idЛаборатории =
using System.Collections.Generic; Отчет.idЛаборатории) ON
using System.ComponentModel; Сотрудники.idСотрудника =
using System.Data; Лаборатории.Заведующий WHERE
using System.Drawing; (((Отчет.idПроверки)=" + comboBox1.SelectedValue
using System.Linq; + ")");
using System.Text; db.selectSQL(strSQL, dataGridView2);
using System.Threading.Tasks; for (int i = 0; i < dataGridView1.Rows.Count
using System.Windows.Forms; - 1; i++)
    {
        if
        (Convert.ToDouble(dataGridView1.Rows[i].Cells[9].Value) >
        Convert.ToDouble(dataGridView1.Rows[i].Cells[10].Value)) { str = "Запрещение"; }
        if (str == "Запрещение") {
        MessageBox.Show("Запуск запрещен",
        "Уведомление"); } else MessageBox.Show("Запуск
        разрешен", "Уведомление");
        ExportExcel ee = new ExportExcel();
        ee.StartExcel(dataGridView1,
        dataGridView2, Environment.CurrentDirectory +
        "/reports/control_"
        + dataGridView1.Rows[0].Cells[5].Value.ToString()
        + ".xls", str);
        MessageBox.Show("Результаты отчета
        расположены в /reports/control_"
        + dataGridView1.Rows[0].Cells[5].Value.ToString()
        + ".xls", "Уведомление");
        Close();
    }
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class proverki : Form
    {
        public proverki()
        {
            InitializeComponent();
        }
        public string t = "";
        public string strSQL;
        database db = new database();
        private void proverki_Load(object sender,
        EventArgs e)
        {
            strSQL = "SELECT Партия, idПроверки
            FROM [Отчет]";
            db.comboSQL(strSQL, comboBox1, "Партия",
            "idПроверки");
        }

        private void button1_Click(object sender,
        EventArgs e)
        {
            if (t == "Excel")
            {
                string str = "Разрешение";
                strSQL = "SELECT Отчет.idПроверки,
                Отчет.ДатаОтчета, Сырье.наименование,
                Поставщики.наименование,
                Отчет.ДатаИзготовления, Отчет.Партия,
                Отчет.ТранспортСредство, Отчет.КолвоСырья,
                ПараметрыСырья.наименование,
                ПараметрыОтчета.ПолученноеЗначение,
                ПараметрыСырья.ЭталонноеЗначение FROM
                (Поставщики INNER JOIN Сырье ON
                Поставщики.idПоставщика = Сырье.idПоставщика)
                INNER JOIN (ПараметрыСырья INNER JOIN (Отчет

```

```

        InitializeComponent();
    }
    public string strSQL;
    database db = new database();
    private void report_Load(object sender,
EventArgs e)
    {
        var column0 = new
DataGridViewTextBoxColumn();
        var column = new
DataGridViewTextBoxColumn();
        var column2 = new
DataGridViewTextBoxColumn();
        column0.Name = "n0";
        column.Name = "n1";
        column2.Name = "n2";
        column.HeaderText = "Параметр";
        column2.HeaderText = "Значение";
        dataparams.Columns.AddRange(new
DataGridViewColumn[] { column0, column, column2
});
        dataparams.Columns[0].Visible = false;
        dataparams.Columns[1].Width = 400;
        dataparams.Columns[2].Width = 100;
        strSQL = "SELECT ФИОСотрудника,
idСотрудника FROM [Сотрудники]";
        db.comboSQL(strSQL, comboBox1,
"ФИОСотрудника", "idСотрудника");
        strSQL = "SELECT Лаборатория,
idЛаборатории FROM [Лаборатории]";
        db.comboSQL(strSQL, comboBox2,
"Лаборатория", "idЛаборатории");
        strSQL = "SELECT наименование,
idСырья FROM [Сырье]";
        db.comboSQL(strSQL, comboBox3,
"наименование", "idСырья");
    }

    private void button1_Click(object sender,
EventArgs e)
    {
        comboBox4.DataSource = null;
        comboBox4.DisplayMember = null;
        comboBox4.ValueMember = null;
        comboBox4.Items.Clear();
        dataparams.Rows.Clear();
        if (dataparams.RowCount>2) {
            for(int i = 0; i <
dataparams.Rows.Count; i++)
                dataparams.Rows.Remove(dataparams.Rows[i]);
            strSQL = "SELECT наименование,
idПараметра FROM [ПараметрыСырья] WHERE
(idСырья = " + comboBox3.SelectedValue + ")";
            db.comboSQL(strSQL, comboBox4,
"наименование", "idПараметра");
            for (int i = 0; i <
comboBox4.Items.Count; i++)
                {
                    object o = (comboBox4.Items[i] as
DataRowView)[0];
                    object o0 = (comboBox4.Items[i] as
DataRowView)[1];
                    string[] row = new string[] {
Convert.ToString(o0), Convert.ToString(o), "" };
                    dataparams.Rows.Add(row);
                }
            private void addbutton_Click(object sender,
EventArgs e)
            {
                bool e0 = true;
                for (int i = 0; i <
dataparams.Rows.Count-1;i++)
                    {
                        if
(dataparams.Rows[i].Cells[2].Value.ToString() == "") {
e0 = false; }
                    }
                if (e0==true)
                {
                    strSQL = "INSERT INTO [Отчет]
(idлаборатории, ДатаОтчета, idСырья,
idСотрудника, Партия, КолвоСырья,
ТранспортСредство, ДатаИзготовления) VALUES ("
+ comboBox2.SelectedValue + ",
"+DateTime.Now.ToString()+ ", "
+ comboBox3.SelectedValue + ", "
+ comboBox1.SelectedValue + ", "+textBox2.Text+ ", "
+ textBox3.Text + ", " + textBox1.Text + ",
"+date.Value.ToShortDateString()+")";
                    db.insertSQL(strSQL);
                    strSQL = "SELECT
Max(Отчет.ДатаОтчета) AS [Max-ДатаОтчета],
Отчет.Партия, Отчет.idПроверки FROM Отчет
GROUP BY Отчет.Партия, Отчет.idПроверки";
                    db.selectSQL(strSQL,
dataGridView1);
                    for (int i = 0; i <
dataparams.Rows.Count - 1; i++)
                        {
                            strSQL = "INSERT INTO
[ПараметрыОтчета] (idОтчета, idПараметра,
ПолученноеЗначение) VALUES ("
+ dataGridView1.Rows[0].Cells[2].Value + ", "
+ dataparams.Rows[i].Cells[0].Value + ", "
+ dataparams.Rows[i].Cells[2].Value.ToString() + ")";
                            db.insertSQL(strSQL);
                        }
                    MessageBox.Show("Отчет создан.
Вы можете сформировать результаты
проверки","Уведомление");
                }
            }
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class sir : Form
    {
        public sir()
        {
            InitializeComponent();
        }
        public string strSQL;
        int num;
        database db = new database();
        private void dataGrid_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            num = (int)dataGrid.Rows[e.RowIndex].Cells[0].Value;
            comboBox1.SelectedValue = dataGrid.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox1.Text = dataGrid.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox2.Text = dataGrid.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void addbutton_Click(object sender, EventArgs e)
        {
            strSQL = "INSERT INTO [Сырье] (idПоставщика, наименование, назначение) VALUES (" + comboBox1.SelectedValue + ", " + textBox1.Text + ", " + textBox2.Text + ")";
            db.insertSQL(strSQL);
            strSQL = "SELECT * FROM [Сырье]";
            db.selectSQL(strSQL, dataGrid);
        }

        private void sir_Load(object sender, EventArgs e)
        {
            strSQL = "SELECT * FROM [Сырье]";
            db.selectSQL(strSQL, dataGrid);
            dataGrid.Columns[0].Visible = false;
            dataGrid.Columns[1].Visible = false;
            dataGrid.Columns[2].HeaderText = "Сырье";
            dataGrid.Columns[2].Width = 200;
            dataGrid.Columns[3].HeaderText = "Назначение";
            dataGrid.Columns[3].Width = 300;
            strSQL = "SELECT наименование, idПоставщика FROM [Поставщики]";
            db.comboSQL(strSQL, comboBox1, "наименование", "idПоставщика");
        }
    }
}

```

```

private void changebutton_Click(object sender, EventArgs e)
{
    strSQL = "UPDATE [Сырье] SET idПоставщика=" + comboBox1.SelectedValue + ", наименование=" + textBox1.Text + ", назначение=" + textBox2.Text + " WHERE (idСырья = " + num + ")";
    db.insertSQL(strSQL);
    strSQL = "SELECT * FROM [Сырье]";
    db.selectSQL(strSQL, dataGrid);
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace QualityModule
{
    public partial class sotr : Form
    {
        public sotr()
        {
            InitializeComponent();
        }
        public string strSQL;
        int num;
        database db = new database();
        private void sotr_Load(object sender, EventArgs e)
        {
            strSQL = "SELECT * FROM [Сотрудники]";
            db.selectSQL(strSQL, dataGrid);
            dataGrid.Columns[0].Visible = false;
            dataGrid.Columns[1].HeaderText = "Сотрудник";
            dataGrid.Columns[1].Width = 200;
            dataGrid.Columns[2].HeaderText = "Должность";
            dataGrid.Columns[2].Width = 170;
        }

        private void addbutton_Click(object sender, EventArgs e)
        {
            strSQL = "INSERT INTO [Сотрудники] (ФИОСотрудника, Должность) VALUES (" + textBox1.Text + ", " + comboBox1.SelectedItem + ")";
            db.insertSQL(strSQL);
            strSQL = "SELECT * FROM [Сотрудники]";
            db.selectSQL(strSQL, dataGrid);
        }
    }
}

```

```

    }

    private void changebutton_Click(object sender, EventArgs e)
    {
        strSQL = "UPDATE [Сотрудники] SET
        ФИОСотрудника=" + textBox1.Text + ",
        Должность=" + comboBox1.SelectedItem + "
        WHERE (idСотрудника = " + num + ")";
        db.insertSQL(strSQL);
        strSQL = "SELECT * FROM
        [Сотрудники]";
        db.selectSQL(strSQL, dataGrid);
    }

    private void deletebutton_Click(object sender, EventArgs e)
    {
        strSQL = "DELETE FROM
        [Сотрудники] WHERE (idСотрудника = " + num +
        ")";

        db.insertSQL(strSQL);
        strSQL = "SELECT * FROM
        [Сотрудники]";
        db.selectSQL(strSQL, dataGrid);
    }

    private void dataGrid_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        num =
        (int)dataGrid.Rows[e.RowIndex].Cells[0].Value;
        textBox1.Text =
        dataGrid.Rows[e.RowIndex].Cells[1].Value.ToString();
        comboBox1.SelectedItem =
        dataGrid.Rows[e.RowIndex].Cells[2].Value.ToString();
    }
}
}

```

ПРИЛОЖЕНИЕ Б.

ВЫХОДНАЯ ФОРМА ВХОДНОГО КОНТРОЛЯ

	A	B	C	D	E	F	G	H	I	
1	Входной контроль									
2	Разрешение на запуск продукции в производство № 4									
3	От 29.04.2019 18:43:38									
4										
5	Наименование продукции	Прессовое масло								
6	Изготовитель	ООО Сай								
7	Дата изготовления								28.04.2019	
8	№ партии	147 а/м				P 3228AC				
9	Количество								25160	
10	Кислотное число, мгКОН/г: 3,9 (норма - не более 4,0)									
11	Перекисное число, ммоль активного кислорода/кг: 8 (норма - не более 10,0)									
12	Массовая доля влаги и летучих веществ, %: 0,15 (норма - не более 0,2)									
13	Цветное число, мг йода: 25 (норма - не более 25)									
14	Массовая доля фосфорсодержащих веществ, %: 0,4 (норма - не более 0,6)									
15										
16	Инженер СКК							Захарченко Т.А.		

Рисунок Б.1 – Выходная форма в режиме листа в MS Excel

Входной контроль	
Разрешение на запуск продукции в производство № 4	
От 29.04.2019 18:43:38	
Наименование продукции	Прессовое масло
Изготовитель	ООО Сай
Дата изготовления	28.04.2019
№ партии	147 а/м P 3228AC
Количество	25160
Кислотное число, мгКОН/г: 3,9 (норма - не более 4,0)	
Перекисное число, ммоль активного кислорода/кг: 8 (норма - не более 10,0)	
Массовая доля влаги и летучих веществ, %: 0,15 (норма - не более 0,2)	
Цветное число, мг йода: 25 (норма - не более 25)	
Массовая доля фосфорсодержащих веществ, %: 0,4 (норма - не более 0,6)	
Инженер СКК	Захарченко Т.А.

Рисунок Б.2 – Выходная форма в режиме печати в MS Excel