

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Проектирование и разработка десктоп приложения

«Продажи нефтегазового оборудования»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Хашаева Мовлади Алиевича

Научный руководитель

доцент, к.ф.-м.н., доцент

Е.Ю.Крылова

Зав. кафедрой

зав. кафедрой, д.ф.-м.н., доцент

Ю.А.Блинков

Саратов 2023

Введение. Современные производственные предприятия должны оперативно реагировать на изменения рыночной конъюнктуры и постоянно улучшать свои бизнес-процессы. Разработка программного обеспечения является одним из способов улучшения работы на предприятии. В данной работе рассматривается разработка программного продукта для управления производством и поставками нефтегазового оборудования.

Актуальность данной работы состоит в том, что многие компании, занимающиеся продажей нефтегазового оборудования, сталкиваются с проблемами в контроле процесса продаж и управлении продуктами из-за использования разнородных дорогостоящих программных приложений. На сегодняшний день на рынке существует немало программных комплексов, которые предназначены для управления продажами и заказами различных товаров, однако большинство из них не учитывают специфики продаж нефтегазового оборудования и включают в себя ненужные функции, что приводит к ошибкам и задержкам при работе. Разработка специализированного десктоп-приложения может решить указанные проблемы.

Основными инструментами для разработки программного продукта на языке C++ являются библиотека *Qt* и система управления базами данных *SQLite*. Библиотека *Qt* позволяет создавать кроссплатформенные графические приложения с использованием C++ и обладает широкими возможностями для разработки пользовательского интерфейса. Система управления базами данных *SQLite* обеспечивает быстрый и надежный доступ к информации, позволяя хранить данные в удобном для работы формате.

Цель данной работы – разработать программный продукт для управления производством и поставками нефтегазового оборудования на языке C++ с использованием системы управления базами данных *SQLite*, достижение которой позволит повысить эффективность работы компаний рассматриваемой предметной области. Для достижения указанной цели были поставлены следующие задачи:

- глубокое изучение языка программирования C++ и библиотеки *Qt*;
- изучение способов хранения данных в базах данных с использованием *SQLite*;

- реализовать функционал сохранения и загрузки базы данных в формате *.xml* и *.db*.

Бакалаврская работа состоит из следующих разделов:

- содержание;
- введение;
- описание предметной области;
- проектирование приложения;
- проектирование базы данных;
- реализация приложения;
- заключение;
- список литературы;
- приложение А. Фрагменты исходного кода на C++.

Работа представляет собой комплексное исследование методов и инструментов программной инженерии, применяемых для разработки программного обеспечения на языке C++, с использованием библиотеки *Qt* и системы управления базами данных *SQLite*.

Основное содержание работы. В первой главе работы рассматривается нефтегазовая сфера и поставки нефтегазового оборудования. Нефтегазовая сфера является одной из ключевых отраслей экономики, существующей в мире уже несколько десятилетий. Она включает множество различных сегментов, в том числе и поставки нефтегазового оборудования. В данной главе рассматриваются основные принципы функционирования нефтегазовой отрасли, ее значимость, а также рассматриваются практические аспекты поставки нефтегазового оборудования, с которыми сталкиваются компании в данной сфере.

Нефтегазовая промышленность является важным сектором экономики, и поставки нефтегазового оборудования становятся все более востребованными. Нефтяное оборудование включает в себя различные механические устройства и системы, которые необходимы для добычи нефти, ее транспортировки, переработки и хранения. Газовое оборудование используется для переработки и транспортировки газа.

В процессе разработки нефте- и газопроводов необходимо оснащать их нефтегазовым оборудованием, таким как насосы, клапаны, компрессоры,

фильтры, мембранные системы, системы контроля и управления. Не менее важным является оборудование для добычи нефти из скважин, к которому относятся установки для бурения, насосы для закачки флюида в скважину и дополнительное оборудование для очистки нефти и газа.

Важным этапом в поставке нефтегазового оборудования является учет особенностей климата и условий эксплуатации. Оборудование для работы в условиях арктического климата, например, должно быть специальным, выполняться из специальных материалов и иметь усиленную изоляцию.

При поставках нефтегазового оборудования также важно учитывать соблюдение международных стандартов и требований к качеству и безопасности. Кроме того, необходимо учитывать соответствие оборудования техническим требованиям и стандартам региона, где будет применяться это оборудование.

Все вышеупомянутые критерии являются важными аспектами в поставке нефтегазового оборудования, поэтому данный процесс требует высокой квалификации и компетенции со стороны поставщиков оборудования.

Во второй главе работы рассматривается проектирование разрабатываемого приложения.

Объектно-ориентированный подход к проектированию программных продуктов является одним из наиболее распространенных и эффективных подходов в современной программной инженерии. Основная идея объектно-ориентированного подхода заключается в том, чтобы выстраивать программы из объектов, которые являются некими абстрактными сущностями, объединяющими данные и функции, которые над ними оперируют. Каждый объект имеет свойства и методы, которые позволяют с ним взаимодействовать.

Программы, созданные с помощью объектно-ориентированного подхода, легко масштабируются, модифицируются и расширяются. Они также более устойчивы к ошибкам, поскольку объекты, разработанные с заранее определенными правилами взаимодействия, обеспечивают более высокую степень модульности. Библиотеки классов позволяют повторно использовать уже реализованный функционал, ускоряя процесс разработки и повышая качество выпускаемых продуктов.

Одним из главных принципов объектно-ориентированного подхода является инкапсуляция, или скрытие данных и интерфейса на уровне объекта. Это означает, что объект может читать и изменять свои собственные данные, но не может непосредственно взаимодействовать с данными других объектов, за исключением публичного интерфейса объекта.

Другой принцип объектно-ориентированного подхода - это наследование. Наследование позволяет создавать новый класс, на основе уже существующего класса, который наследует свойства и методы от родительского класса, при этом дополнительно может иметь свои собственные свойства и методы.

Третий принцип объектно-ориентированного подхода - это полиморфизм. Полиморфизм позволяет объектам решать одну и ту же задачу разными способами, в зависимости от контекста. Примером полиморфизма может служить метод, который ведет себя по-разному в зависимости от параметров, переданных в него.

Всякое проектирование с помощью объектно-ориентированного подхода начинается с выделения объектов и их разделения на классы, определяются свойства и методы каждого класса. Затем определяются иерархии наследования, устанавливаются интерфейсы, исключения, делегирование. Затем реализуется абсолютно вся логика программы на основе определенных классов и объектов, и далее производится тестирование.

Одним из вариантов проектирования является диаграмма вариантов использования – это структурная диаграмма языка моделирования *UML*, демонстрирующая графическое представление функциональных требований к системе, которое используется для описания того, как различные типы пользователей будут взаимодействовать с системой. Диаграмма вариантов использования для предметной приведена в соответствии с рисунком 1.

Объектно-ориентированный подход к проектированию программных продуктов обеспечивает более гибкий и структурированный способ разработки приложений, позволяет создавать качественные, надежные и легко модифицируемые решения.

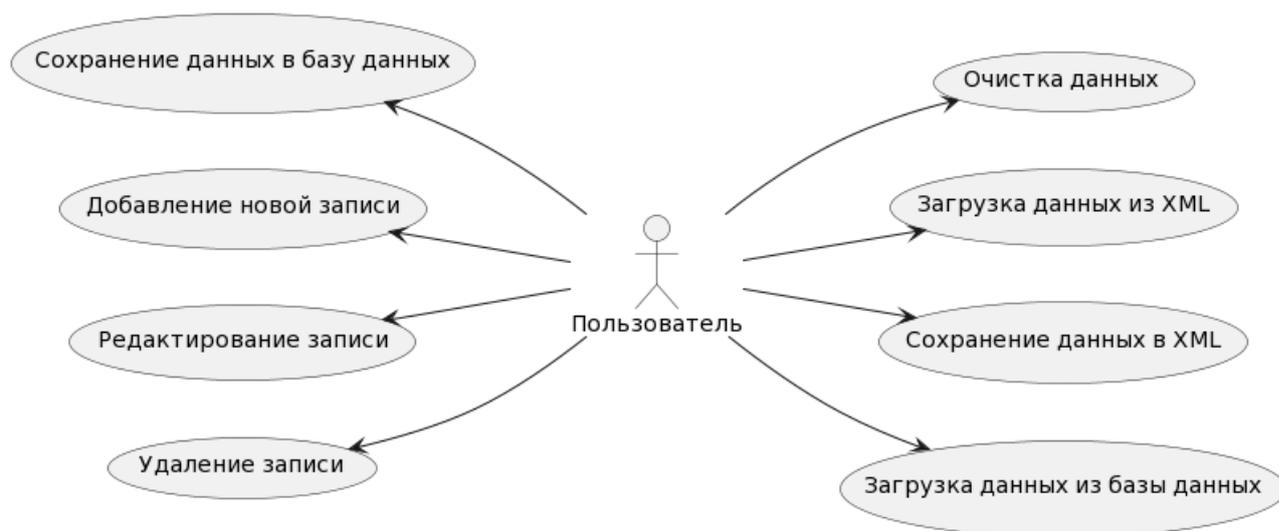


Рисунок 1 – Диаграмма вариантов использования

В третьей главе работы рассматривается проектирование базы данных, используемой в приложении.

Для реализации программного продукта, автоматизирующего учет поставок нефтегазового оборудования, необходимо создать базу данных. В данной главе рассматривается система управления базами данных *SQLite*, которая является легкой, быстрой и компактной СУБД. Описываются основные возможности и преимущества этой СУБД, а также рассматриваются основные аспекты ее использования.

Для проектирования базы данных использовалась ER-диаграмма. ER-диаграммы (Entity-Relationship Diagrams) – это графические элементы, которые используются для описания системы, учитывая ее сущности и их отношения. ER-диаграммы обычно используются в процессах проектирования баз данных и являются полезным инструментом для анализа фундаментальных компонентов данных. Все ER-диаграммы состоят из трех основных компонентов:

1. Сущности - это объекты или идеи, описывающие концепции в системе. Сущности представлены прямоугольниками с названием - это может быть конкретный объект, такой как «пользователь» или «товар», или абстрактный концепт, такой как «заказ» или «компания».
2. Атрибуты - это свойства или характеристики сущностей. Атрибуты представлены в скобках внутри прямоугольника сущности. Например,

для сущности «пользователь» может быть атрибут «имя», «адрес» или «email».

3. Связи - это отношения между сущностями. Связи показывают, как две или более сущностей связаны друг с другом. Указываются типы связи, например как «один-ко-многим» или «многие-ко-многим», а также направление связи, если оно не очевидно. Связи представляются линиями, которые связывают прямоугольники сущностей.

ER-диаграммы могут быть скомпонованы в более сложные структуры, такие как поддиаграммы, которые могут отображаться на одном уровне. Поддиаграммы позволяют скрыть сложность модели, делая ее более понятной для пользователей.

ER-диаграммы могут использоваться для анализа и проектирования баз данных, моделирования процессов бизнеса и создания сущностей для веб-приложений. Они помогают определить необходимые сущности и взаимоотношения между ними, что может снизить количество ошибок и принести больше ясности в разработку проектов.

В соответствии с рисунком 2 приведена ER-диаграмма из 14 сущностей, используемых в приложении.

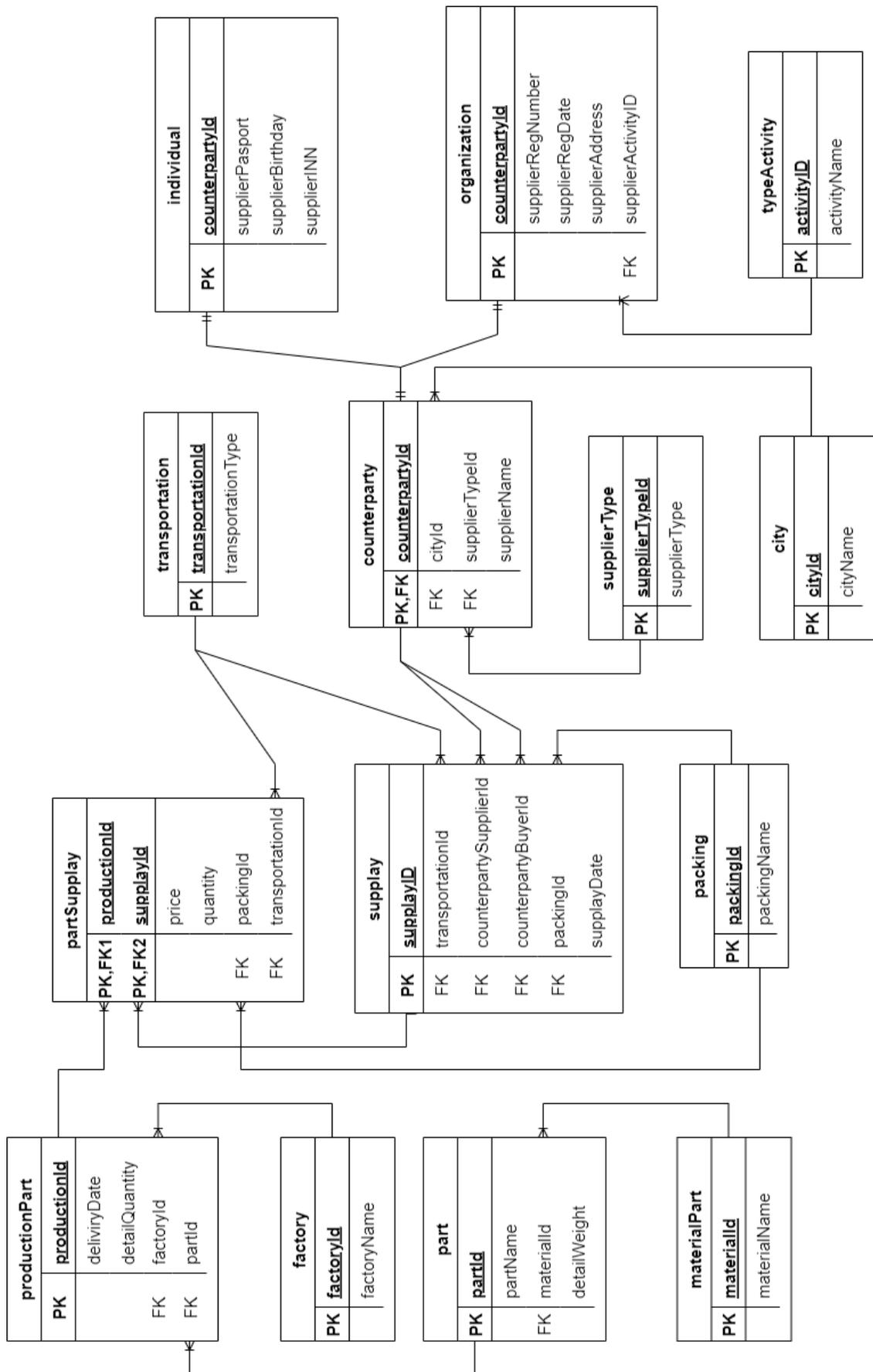


Рисунок 2 – ER-диаграмма

В соответствии с рисунком 3 приведен фрагмент кода создания таблицы *Supplay*.

```
60 CREATE TABLE supplay
61 (
62 supplayid INT PRIMARY KEY,
63 transportationid INT NOT null REFERENCES transportation(transportationid),
64 counterpartysupplierid INT NOT null REFERENCES counterparty(counterpartyid),
65 counterpartybuyerid INT NOT null REFERENCES counterparty(counterpartyid),
66 packingid INT NOT null REFERENCES packing(packingid),
67 supplaydate DATE NOT null DEFAULT current_date,
68 UNIQUE(transportationid, counterpartysupplierid, counterpartybuyerid, packingid),
69 CHECK(counterpartysupplierid!=counterpartybuyerid)
70 );
```

Рисунок 3 – Код таблицы *Supplay*

В четвертой главе подробно рассматривается реализация разрабатываемого приложения.

Для написания программы автоматизации учета поставок нефтегазового оборудования был выбран язык программирования C++ и библиотека *Qt*. C++ является высокопроизводительным и низкоуровневым языком программирования, обеспечивающим быструю работу программы и эффективное использование ресурсов. Библиотека *Qt* является мощным и функциональным инструментом, позволяющим создавать множество различных приложений.

В данной главе детально описывается процесс создания программы автоматизации учета поставок нефтегазового оборудования с использованием языка C++ и библиотеки *Qt*. Рассматриваются основные принципы работы с этими инструментами, а также представляется непосредственная реализация программы.

Пример кода заголовочного файла *supplay.h*:

```
1 #ifndef SUPPLAY_H
2 #define SUPPLAY_H
3 #include <string>
4 using namespace std;
5
6 class Supplay
7 {
```

```

8 private:
9     int id;
10    int transportation_id;
11    int counterparty_supplier_id;
12    int counterparty_buyer_id;
13    int packing_id;
14    string supplay_date;
15
16 public:
17    Supplay(int id = -1, int transportation_id = -1, int
18            counterparty_supplier_id = -1,
19            int counterparty_buyer_id = -1, int packing_id = -1, string
20            supplay_date = "NULL");
21
22    void set_id(int id);
23    void set_transportation_id(int transportation_id);
24    void set_counterparty_supplier_id(int counterparty_supplier_id);
25    void set_counterparty_buyer_id(int counterparty_buyer_id);
26    void set_packing_id(int packing_id);
27    void set_supplay_date(string supplay_date);
28
29    int get_id();
30    int get_transportation_id();
31    int get_counterparty_supplier_id();
32    int get_counterparty_buyer_id();
33    int get_packing_id();
34    string get_supplay_date();
35 };
36 #endif // SUPPLAY_H

```

В соответствии с рисунком 4 представлен внешний вид разработанной программы:

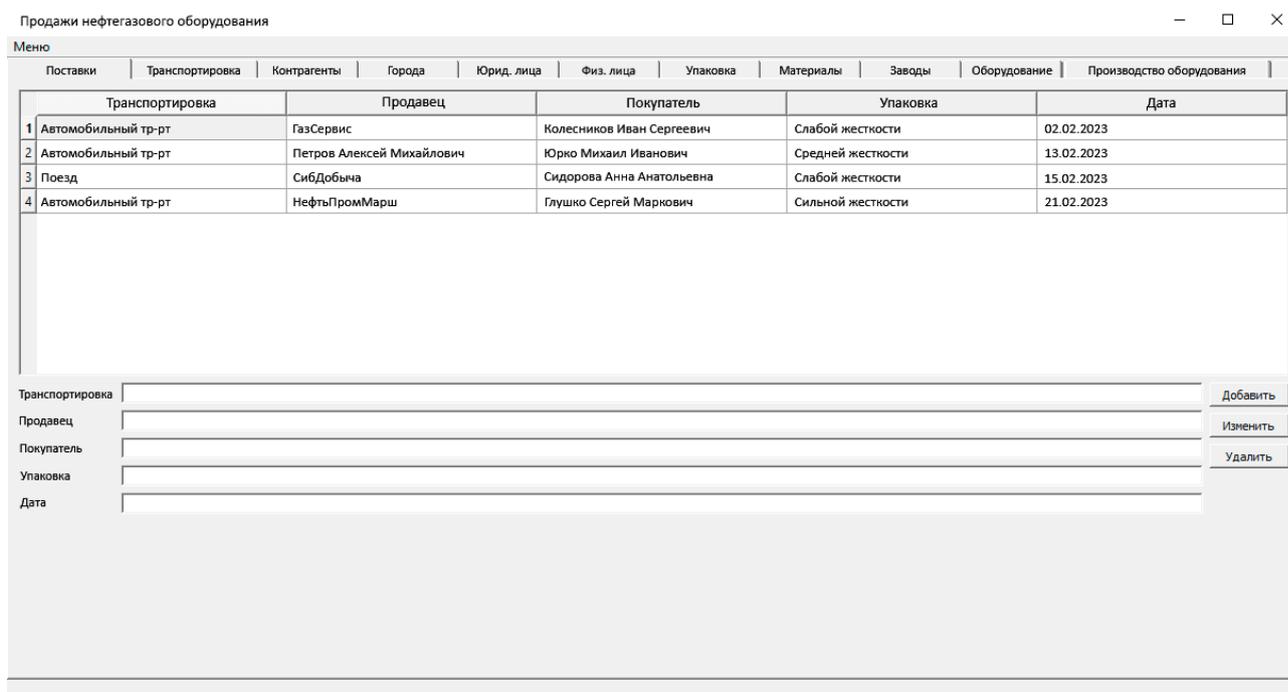


Рисунок 4 – Внешний вид приложения

Заключение. В процессе выполнения работы была создана программа автоматизации учета поставок нефтегазового оборудования с использованием языка программирования C++ и библиотеки *Qt*.

Были проанализированы проблемы, с которыми сталкиваются компании в данной сфере, и предложено решение, позволяющее значительно сократить время на учет и контроль поставок нефтегазового оборудования. Система управления базами данных *SQLite* была использована в качестве основного хранилища информации. Программа была реализована с использованием языка программирования C++ и библиотеки *Qt*.

Важным достижением данной работы является успешное внедрение разработанной программы в компанию «Ойлгаз-Комплект», занимающуюся поставкой нефтегазового оборудования. Полученные отзывы от представителей этих компаний были положительными, и они отметили значительный вклад программы в автоматизацию и оптимизацию бизнес-процессов. Этот результат показывает, что разработанный программный продукт эффективен и может быть использован для повышения эффективности работы компаний, занимающихся поставкой нефтегазового оборудования.

Таким образом, разработка программы автоматизации учета поставок нефтегазового оборудования с использованием языка C++ и библиотеки *Qt* имеет важное практическое значение. Разработанный программный продукт может стать незаменимым инструментом для компаний, занимающихся поставкой нефтегазового оборудования. Развитие данных технологий и новых механизмов обработки информации открывают новые возможности для улучшения данной программы и ее применения в будущем.