

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

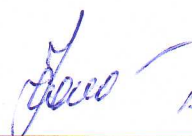
Направление подготовки 09.03.01 Информатика и вычислительная техника

Интеллектуальная система оценки бизнес-процессов аэропорта

Пояснительная записка

ОГУ 09.03.01.1219.046 ПЗ

Заведующий кафедрой
д-р техн. наук, профессор

 17.06.19

Н.А. Соловьев

инициалы, фамилия


Руководитель
канд. техн. наук, доцент

 7.06

А.М. Семенов

инициалы, фамилия

Студент

 7.06.2019

А.К. Яхина

инициалы, фамилия

Оренбург 2019

Утверждаю
заведующий кафедрой _____

_____ ПОВТАС
(наименование кафедры)
_____ А.Ч. Соловьев
подпись инициалы фамилия
« 10 » 10 20 19 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту _____ Джинай Анна Камилловна
(фамилия имя отчество)

по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника
код, наименование

1 Тема ВКР _____ Интеллектуальная система оценки бизнес-процессов аэропорта

2 Срок сдачи студентом ВКР « 08 » _____ июля 20 19 г.

3 Цель и задачи ВКР _____ автоматизации информационных процессов
аэропорта в рамках эффективного бизнес-управления

4 Исходные данные к ВКР _____ результаты предпринятой практики

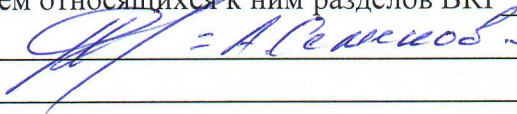
5 Перечень вопросов, подлежащих разработке _____

- 1 Аналитический раздел
 - 1.1 Анализ кредитной области
 - 1.2 Анализ аналогов программных средств
 - 1.3 Выбор математического аппарата
 - 1.4 Техническое задание на разработку программной системы
- 2 Разработка информационного и программного обеспечения системы
 - 2.1 Выбор инструментальных средств разработки программного средства
 - 2.2 Разработка организационного назначения
 - 2.3 Разработка информационной структура данных
 - 2.4 Разработка алгоритма программного средства
 - 2.5 Тестирование программного средства
- 3 Разработка технологической документации
 - 3.1 Руководство системного администратора
 - 3.2 Руководство пользователя
 - 3.3 Техническое обоснование
Замечание
Список использованных источников
Приложение

6 Перечень графического (иллюстративного) материала

- 1 Функциональная модель
- 2 Иерархия модулей
- 3 Функциональная схема
- 4 ЕК - диаграмма
- 5 Амортизи математического аппарата
- 6 Упрощенной амортизи программного средства
- 7 Тестирование программного средства
- 8 Патологическая модель

7 Консультанты, с указанием относящихся к ним разделов ВКР

1-3  = А. С. Семенов

Дата выдачи и получения задания

Руководитель ВКР

« 1 » августа 20 18 г.



подпись

А. С. Семенов
инициалы фамилия

Студент

« 8 » августа 20 18 г.


подпись

А. К. Жуков
инициалы фамилия

Дата выполнения студентом задания консультанта

Консультант

« ___ » _____ 20 ___ г.

подпись

инициалы фамилия

Аннотация

Данная выпускная квалификационная работа посвящена разработк «Интеллектуальной системы оценки бизнес-процессов аэропорта», целью которо является автоматизация информационных и бизнес-процессов на территории ГУИ Оренбургской области «Аэропорт Оренбург».

В данной пояснительной записке описывается анализ информационны потоков предметной области, обзор существующих аналогов, выбо инструментальных средств разработки, постановка задачи, архитектура АИС проектирование и создание БД, разработка алгоритмов АИС, тестировани компонентов, руководство системного администратора, руководство пользовател

Работа содержит 79 листов, 35 рисунков, 30 таблиц, 2 приложения. Графическая часть выполнена на 9 листах А4.

					ОГУ 09.03.01. 1219. 046 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Интеллектуальная система оценки бизнес-процессов аэропорта	Лит.	Лист	Листов
Разраб.		Яхина А.К.		07.08		В К Р	3	79
Пров.		Семенов А.М.		07.08		15ИВТ(ба)ОП-1		
Н. контр.		Чернопрудова Е.Н.		18.06				
Зав. каф.		Соловьев Н.А.		12.06.19				

The annotation

This final qualifying work is devoted to the development of an automated information system «Intelligent evaluation system of business processes of the airport», the purpose of which is automation of information and business processes in the territory State Unitary Enterprise of Orenburg region "Orenburg Airport".

This explanatory note describes the analysis of information flows of the subject area, a review of existing analogues, the choice of development tools, the formulation of the problem, the intelligent evaluation system architecture, the design and creation of databases, the development of automated information system algorithms, component testing, an application programmer's guide, a user guide.

The work contains 79 sheets, 35 pictures, 30 tables, 2 application. The graphic part is made on 9 sheets of A4.

Содержание

Введение.....	5
1 Аналитический раздел	6
1.1 Анализ предметной области.....	6
1.2 Анализ аналогов программных средств.....	10
1.3 Выбор математического аппарата	12
1.4 Техническое задание на разработку программной системы	14
2 Разработка информационного и программного обеспечения системы.....	16
2.1 Выбор инструментальных средств разработки ПС	16
2.2 Разработка функционального назначения.....	20
2.3 Разработка информационной структуры данных	24
2.4 Разработка алгоритма программного средства	38
2.5 Тестирование программного средства	39
3 Разработка технологической документации	46
3.1 Руководство системного администратора	46
3.2 Руководство пользователя.....	47
3.3 Экономическое обоснование	54
Заключение	60
Список использованных источников	61
Приложение А (обязательное) ER-диаграмма	63
Приложение Б (обязательное) Даталогическая модель	64
Приложение В (справочное) SQL-скрипты.....	65
Приложение Г (обязательное) Листинг программы.....	71

Введение

Работа любого предприятия становится гораздо эффективнее, если производить анализ системы и периодически внедрять проекты, оптимизирующие и повышающую производительность всей системы. Аэропорты — это сложные предприятия. Объемы, скорость и разнородность информации, используемой их руководителями, растут с каждым днем. Потому приходится принимать сложные решения в процессе изучения того, какой проект будет выводить аэропорт на новый уровень трудоспособности.

Вопрос повышения эффективности предприятия всегда актуален и требует ответственного подхода. Поскольку процесс выявления оптимального бизнес-процесса достаточно трудная задача даже для экспертов в этой области, необходимо разработать информационную систему, способную произвести анализ всех альтернатив на основе выделенных критерий. Таким образом, задача сводится к выбору наилучшего варианта посредством многокритериального анализа, создание приложения поможет эффективно руководить процессами на предприятии ГУП Оренбургской области «Аэропорт Оренбург».

Целью выпускной квалификационной работы является автоматизация информационных и бизнес-процессов на территории аэропорта.

Для достижения поставленной цели необходимо решить ряд задач:

1) задачи анализа:

- анализ предметной области;
- анализ информационных потоков предприятия, выделение потоков подлежащих автоматизации;
- анализ аналогов программного средства;
- выбор, обоснование и разработка метода оценки бизнес-процессов;
- выбор инструментальных средств разработки приложения и базы данных;

2) задачи синтеза:

- разработка архитектуры программного средства;
- разработка информационной структуры данных (концептуальная информационно-логическая, даталогическая и физическая модели);
- разработка алгоритма приложения;
- разработка технологической документации интеллектуальной системы;

3) тестирование компонентов интеллектуальной системы.

Разработанная интеллектуальная система позволит выполнить оценку бизнес-процессов, выявляя наилучший из них, и повысить эффективность работы аэропорта.

1 Аналитический раздел

1.1 Анализ предметной области

Государственное унитарное предприятие Оренбургской области «Аэропорт Оренбург» им. Ю. А. Гагарина предлагает обслуживание и авиаперевозки пассажиров и их багажа [1].

Организационную структуру аэропорта составляют различные подразделения и отделы, в том числе отдел стратегического развития аэропорта, состоящий из руководителя и сотрудника. Данный отдел занимается вопросами увеличения пассажиропотока и прибыли аэропорта. Эффективное развитие предприятия зависит от грамотного управления всеми процессами.

В области управления аэропортом и их бизнес – результаты входят: защита и безопасность, энергоснабжение, физические активы и кадровые ресурсы, недвижимое имущество и пространство [2].

Для того, чтоб увеличить пассажиропоток международного аэропорта города Оренбург, обратим внимание на сферу недвижимого имущества и пространства. Бизнес – процессами данной сферой являются:

- 1 реконструкция терминала международных воздушных линий;
- 2 реконструкция грузового комплекса;
- 3 строительство современной контейнерной площадки для обработки грузов;
- 4 строительство дополнительного топливо – заправочного комплекса;
- 5 развитие летной зоны с реконструкцией рулежных дорожек к ней
- 6 расширение автомобильной стоянки

Разрабатываемая программная система должна поддерживать оценку каждого бизнес – процесса по заданным критериям. Анализ должен производиться по следующим критериям: стоимость мероприятия (в рублях), срок реализации (в месяцах), срок окупаемости (в месяцах).

Результат оценки поможет определить, какой из них станет определяющим в обеспечении комфортного функционирования аэропорта.

Информация о бизнес – процессах и их описание хранится в базе данных, которую ведет отдел стратегического развития. Все изменения в описаниях бизнес – процессов должны фиксироваться в базе данных. В базе учитываются следующие данные: название процесса, описание критериев и их значения, а также дата сбора фактов, правила измерения фактов, степень ответственности, факты измерения.

Система управления включает людей, выполняющих следующие роли:

– сотрудник отдела стратегического развития, который выполняет роль оператора АС, его функции – добавление, изменение и обновление данных о бизнес-процессах в БД;

– руководитель отдела стратегического развития (выполняет роль руководителя и лица, принимающего решения) – анализирует результаты проведенной оценки бизнес – процессов, принимает окончательное решение в выборе внедряемого проекта.

Конечные пользователи АС – это руководитель и сотрудник отдела стратегического развития.

Обработка данных охватывает следующие информационные потоки (ИП) [3]:

– информационные потоки (ИП), выявленные в ходе анализа предметной области:

– ИП1 (внешний, входной) – типы, виды, характеристики бизнес – процессов;

– ИП2 (внешний, выходной) – пустой;

– ИП3 (внутренний, входной) – дата, время сбора фактов, правила измерения фактов, степень ответственности, факты измерения;

– ИП4 (внутренний, выходной) – статистические отчеты, результат обработки на основе математического метода (метода анализа иерархий).

Охватываемые обработкой данных функции управления:

– оценка бизнес – процессов аэропорта;

– описание бизнес – процессов;

– формирование отчетов.

В таблице 1 отображены документы / фрагменты документов, подлежащие автоматизированной обработке, а также описан состав реквизитов входных и выходных документов.

Таблица -1 Входные и выходные документы АС

Номер документа	Вид документа (вх)/(вых)	Качественные реквизиты	Количественные показатели	Принадлежность к информационному потоку (ИП)
1	2	3	4	5
1 Сведения об аэропорте и сотрудниках	вх	Полное, краткое название предприятия, фамилия, имя, отчество сотрудников	Код предприятия, табельный номер	ИП-1 (документ о состоянии бизнес - процессов)
2 Сведения о бизнес – процессах	вх	Полное, краткое название бизнес – процессов	Номер бизнес – процесса; Дата начала; Дата окончания.	ИП-1 (документ о состоянии бизнес - процессов)

Продолжение таблицы 1

1	2	3	4	5	
3 Сведения о характеристиках бизнес - процессов	ВХ	Полное, название	краткое	Номер	ИП-1 (позиция документа о состоянии бизнес - процессов)
4 Оценка бизнес - процессов	ВЫХ	Полное, название	краткое	Дата начала; Дата окончания.	ИПЗ (параметры отображения)
5 Отчет	ВЫХ	Полное, название, описание	краткое	Номер; Дата.	ИП4 – (обработка результатов)

На рисунке 1 отображена модель потоков данных предприятия в нотации DFD. На диаграмме присутствуют следующие элементы [4]:

- внешние сущности: сотрудник отдела автоматизации и руководитель отдела стратегического развития;
- потоки данных, пронумерованные в соответствии с таблицей 1;
- АС, реализующая процессы обработки данных в автоматизированном режиме (с участием человека).

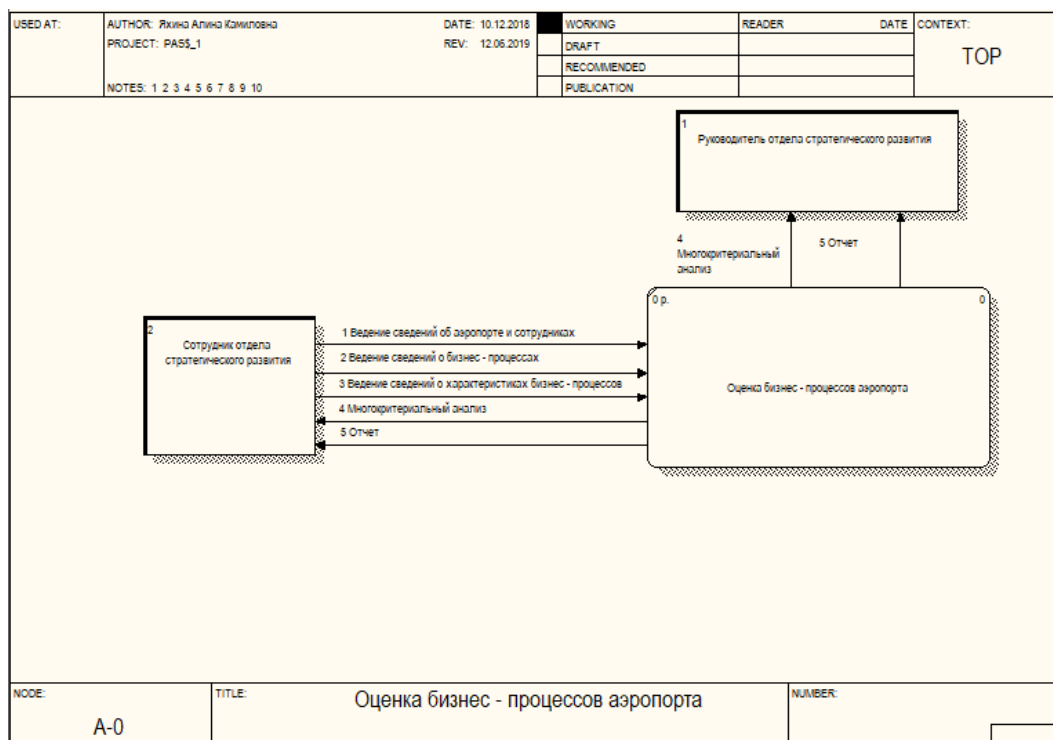


Рисунок 1 – Диаграмма потоков данных в нотации DFD

На рисунке 2 представлена контекстная (0-го уровня) функциональная модель деятельности предприятия в рамках решения задач автоматизации средствами интеллектуальной системы [5].

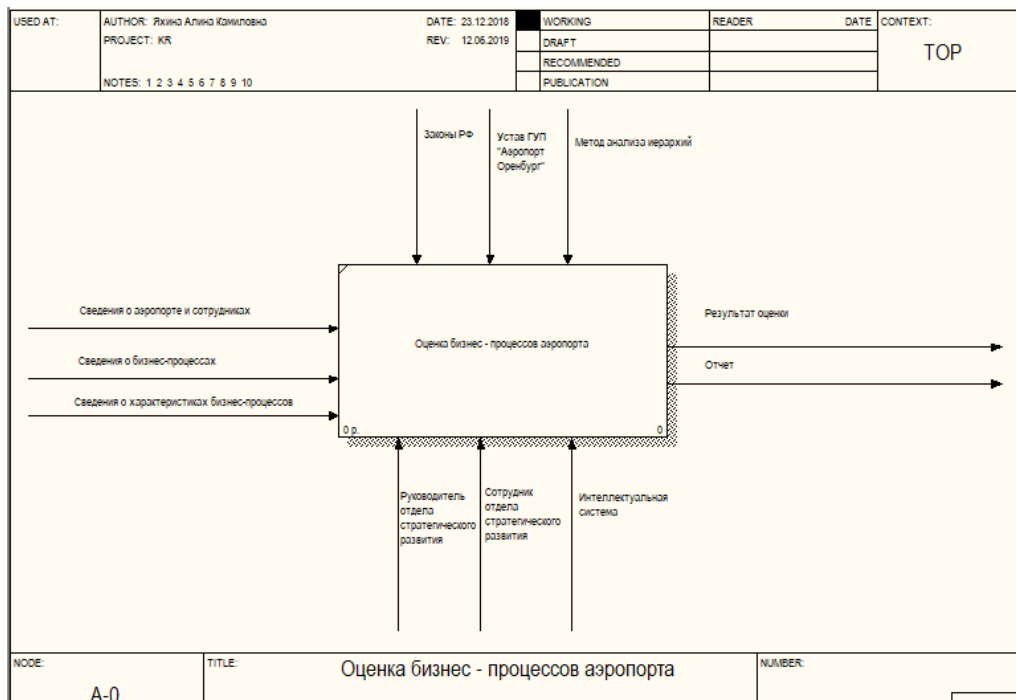


Рисунок 2 – Контекстная диаграмма функциональной модели

На рисунке 3 представлена диаграмма декомпозиции первого уровня, являющаяся дальнейшей детализацией функциональной модели [4].

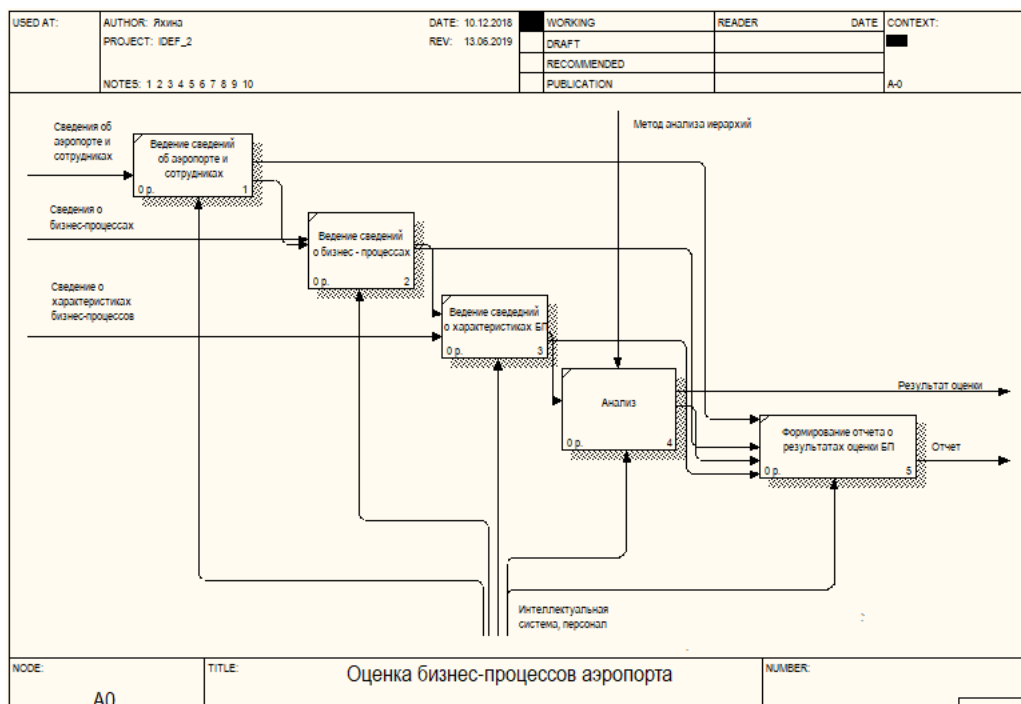


Рисунок 3 - Декомпозиция поставленной задачи

Функциональная модель первого уровня включает следующие задачи:

- 1 Ведение сведений об аэропорте и сотрудниках;
- 2 Ведение сведений о бизнес - процессах;
- 3 Ведение сведений о характеристиках бизнес – процессов;
- 4 Анализ;
- 5 Формирование отчета о результатах оценки бизнес – процессов;

Разрабатываемая программная система позволит поддерживать оценку по заданным критериям каждого бизнес – процесса для достижения наиболее точных результатов. А результат оценки, в свою очередь, поможет определить, какой из них станет определяющим в обеспечении комфортного и прибыльного функционирования аэропорта.

1.2 Анализ аналогов программных средств

Один из аналогов средств оценки – бизнес-процессов – это "Мастер Проектов. Предварительная оценка" - программа для оценки рентабельности инвестиционных проектов и расчета основных финансовых показателей эффективности инвестиций при минимуме исходных данных (рисунок 4).

Методика расчетов, используемая в модели "Мастер Проектов. Предварительная оценка", полностью соответствует общепринятым мировым стандартам инвестиционного анализа.

Заключение об инвестиционной привлекательности проекта делается на основании полного набора "классических" финансовых показателей, в число которых входят NPV, IRR и срок окупаемости инвестиций.

Работа с моделью и оформление отчета может вестись как на русском, так и на английском языке.

Компьютерная модель "Мастер Проектов. Предварительная оценка" относится к классу открытых систем, что означает возможность настройки модели пользователем для решения специфических задач [6].

II. MODEL SETUP INFORMATION			
28	Currency unit	тыс. руб. Change	
29	Planning interval	кварт. Change	
30	Duration of the planning interval	90 дней	
31	Number of planning intervals	12 Change	
32	Project lifetime	3,0 лет	
III. INITIAL DATA			
	Item description	Стоимость, тыс. руб.	Норма амортизации
37	Land	0	0%
38	Buildings	0	2%
39	Equipment	6000	10%
40	Vehicles	0	20%
41	Rights & licenses	0	10%
42	Other fixed assets	0	10%
43	Other non-period fixed payments	2000	33%
44	Fixed investment costs	8000	Hide List
	Product name	Сумма, тыс. руб. в кварт.	Комментарий
46	Product	3600	
47	Product	4000	Quantity
48	Sales revenue	7600	Hide List
55	Variable production costs	4365	Show List
64	Fixed production costs	989	Show List
72	Net working capital requirement	4618	Show List

Рисунок 4 – "Мастер Проектов. Предварительная оценка"

Другой аналог АИС оценки бизнес – процессов – «Project Expert» [7].

Аналитическая система Project Expert — программа, позволяющая «прожить» планируемые инвестиционные решения без потери финансовых средств, предоставить необходимую финансовую отчётность потенциальным инвесторам и кредиторам, обосновать для них эффективность участия в проекте (рисунок 5).

Незаменим для создания и выбора оптимального плана развития бизнеса, проработки финансовой части бизнес-плана, оценки инвестиционных проектов.

Позволяет моделировать деятельность различных отраслей и масштабов — от небольших венчурных компаний до холдинговых структур. Программа широко используется для финансового моделирования и разработки бизнес-планов производства и оказания услуг в банковском бизнесе, телекоме, строительстве, нефтедобыче и нефтепереработке, транспорте, химии, перерабатывающей и лёгкой промышленности, машиностроении, аэрокосмической отрасли, энергетике.

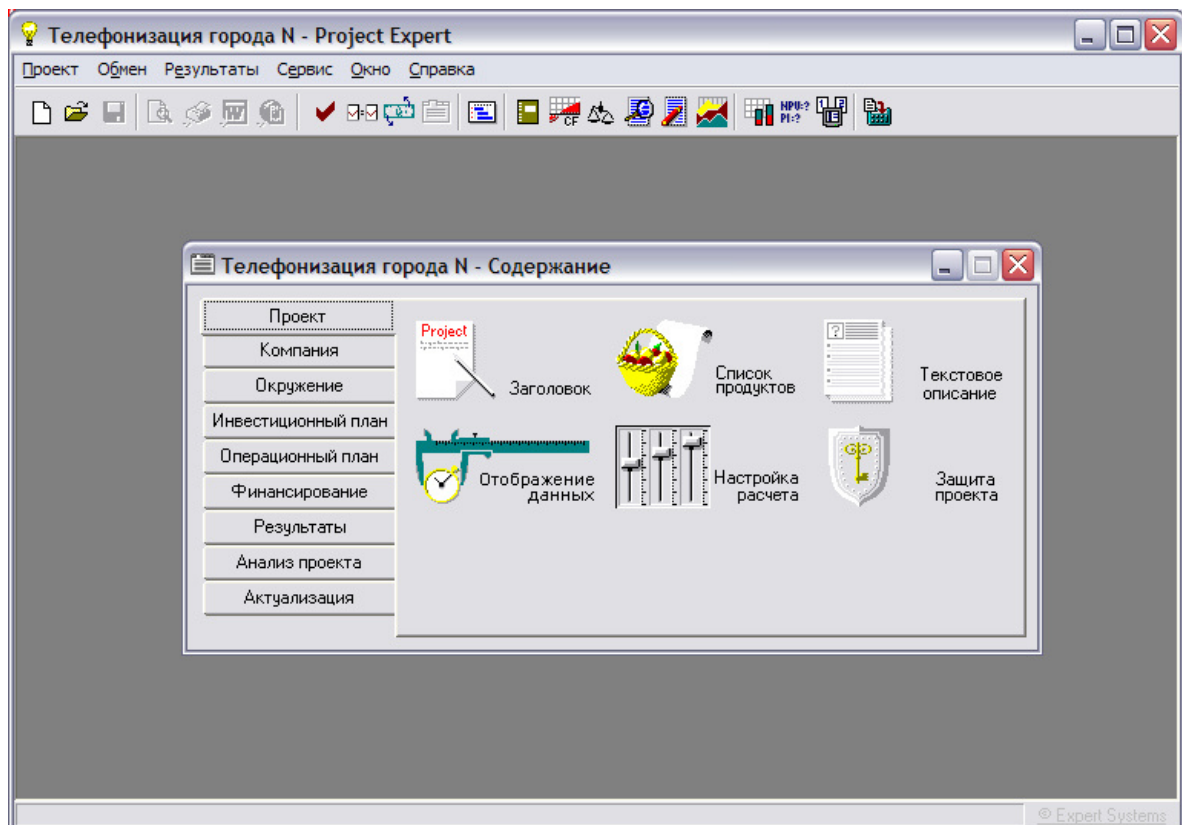


Рисунок 5 – «Project Expert»

Таким образом, рассмотрев информационные системы, автоматизирующие оценку проектов и бизнес-процессов на предприятиях, можно сделать вывод, что, несмотря на широкий диапазон функциональных возможностей, рассмотренные системы обладают некоторыми недостатками:

- в первом рассмотренном аналоге «Мастер Проектов. Предварительная оценка» нет функции сравнительного анализа бизнес – процессов, каждый проект рассматривается отдельно, а значит перед лицом, принимающим решение, остается задача сравнения и выбора наиболее эффективного проекта.

- «Project Expert» является дорогостоящим продуктом (от 80 тысяч рублей), который нуждается в сопровождении программистом, обученном работе в этой программе, что влечет за собой дополнительные расходы;

Следовательно, доказана необходимость разработки интеллектуальной системы оценки бизнес-процессов аэропорта.

1.3 Выбор математического аппарата

В работе для выбора наилучшего бизнес-процесса используется метод многокритериального выбора в иерархиях с различным числом и составом альтернатив под критериями [8].

Задача подбора организаций относится к типу задач, когда ранжируемые по множеству критериев альтернативы оцениваются не по всем критериям. То есть имеется разное количество альтернатив под каждым критерием или под их частью.

Ниже приведена методика определения вектора приоритета альтернатив, когда иерархия имеет один уровень критериев, объединенных целью, и разное количество альтернатив у каждого критерия [8].

Шаг 1. Строится иерархия проблемы.

Шаг 2. Экспертная оценка альтернатив по критериям на основе метода парного сравнения. Построение матрицы A (1):

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \dots & \dots & \dots & \dots \\ a_{r1} & a_{r2} & \dots & a_{rp} \end{bmatrix}, \quad (1)$$

где экспертные оценки a_{ij} представляют векторы приоритетов альтернатив относительно критериев E_j . Если альтернатива A_i не сравнивается по критерию E_j , то $a_{ij} = 0$.

Шаг 3. В результате обработки матрицы попарных сравнений критериев относительно фокуса определяется вектор приоритетов критериев относительно цели \bar{X} .

Шаг 4. Формируются диагональные матрицы S (2) и L (3).

$$S = \begin{bmatrix} \left(\sum_{i=1}^r a_{i1}\right)^{-1} & 0 & \dots & 0 \\ 0 & \left(\sum_{i=1}^r a_{i2}\right)^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \left(\sum_{i=1}^r a_{ip}\right)^{-1} \end{bmatrix}, \quad (2)$$

$$L = \begin{bmatrix} R_1/N & 0 & \dots & 0 \\ 0 & R_2/N & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & R_p/N \end{bmatrix}, \quad (3)$$

где R_j - число альтернатив, находящихся под критерием E_j , $N = \sum_{j=1}^p R_j$ - суммарное число альтернатив, находящихся под всеми критериями.

С помощью матрицы S нормируются векторы приоритетов альтернатив, образующих матрицу A , путем ее умножения на S справа. Критерий L позволяет эксперту изменять вес альтернатив, связанных с критериями пропорционально отношению R_j/N . За счет этого повышается приоритет альтернатив в группах с их относительно небольшим числом.

Шаг 5. Определяется вектор приоритетов альтернатив W относительно критериев по формуле (4) в случае ненормированных оценок в матрице A , по формуле (5) - в случае нормированных оценок.

$$W = [A] \cdot [S] \cdot [L] \cdot \bar{X} \cdot [B]. \quad (4)$$

$$W = [A] \cdot [L] \cdot \bar{X} \cdot [B]. \quad (5)$$

Матрица B рассчитывается по формуле (6).

$$B = \begin{bmatrix} \left(\sum_{i=1}^r x_i\right)^{-1} & 0 & \dots & 0 \\ 0 & \left(\sum_{i=1}^r x_i\right)^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \left(\sum_{i=1}^r x_{ii}\right)^{-1} \end{bmatrix}, \quad (6)$$

где x_i - значения ненормированного вектора приоритетов альтернатив, полученные после последовательного перемножения матриц $[A] \cdot [S] \cdot [L] \cdot \bar{X}$, r - число альтернатив [8].

На основе полученного вектора принимается решение о выборе наилучшей альтернативы относительно совокупности критериев.

1.4 Техническое задание на разработку программной системы

Задача сводится к выбору наилучшего бизнес-процесса в аэропорту. Создание приложения поможет собрать произвести оценку всех бизнес-процессов по заданным критериям и принять решение.

Для достижения поставленной задачи был выбран метод анализа иерархий, так как он позволяет решать задачи выбора

Система разрабатывается на основе приказа № 1301-С от 08.08.2018 «Об утверждении руководителей и тем выпускных квалификационных работ обучающимся направления подготовки 09.03.01 «Информатика и вычислительная техника».

Первая версия системы предназначена для автоматизации оценки бизнес – процессов. В последующих версиях предполагается увеличение количества решаемых задач.

Система должна представлять совокупность методических и программных средств решения следующих задач:

- описание с помощью характеристик бизнес - процессов;
- формирование отчетности;
- оценка бизнес – процессов аэропорта.

Методическое обеспечение должно быть реализовано в пользовательском интерфейсе системы, который должен предполагать выбор необходимой задачи; ввод и сохранение данных исходных данных, промежуточных и окончательных результатов в базе данных для последующего анализа.

Требования к надежности.

– Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой.

– Обеспечить корректное завершение вычислений с соответствующей диагностикой при превышении имеющихся вычислительных ресурсов. Возможность автоматизации создания отчетной документации по результатам обработки информации.

- Обеспечить целостность информации, хранящейся в базе данных.
- Разграничение пользовательских прав.
- Возможность резервного копирования базы данных.
- Требования к составу и параметрам технических средств.

Система должна работать на IBM-совместимых персональных компьютерах.

Минимальная конфигурация:

- процессор с тактовой частотой не ниже 1,3 ГГц;
- размер ОЗУ не менее 2 ГБ;
- не менее 5 ГБ свободного места на жестком диске;
- видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей);
- разрешение экрана монитора 1280 x 720 и выше.

Необходимо наличие монитора, манипулятора «мышь», клавиатуры и принтера.

Необходимы следующие программные требования:

- операционная система Windows 7/8/Server 2017/Server 2012;
- офисные приложения Microsoft Office 2016 (Word);
- инструментальная среда разработки Visual Studio 2017;
- система управления базами данных (СУБД) SQL Server 2008;
- драйвера для принтера.

Разрабатываемая система должна включать справочную информацию о работе системы и подсказки пользователю.

В состав сопровождающей документации должны входить:

- пояснительная записка;
- руководство администратора,
- руководство пользователя (сотрудника отдела стратегического развития).

2 Разработка информационного и программного обеспечения системы

2.1 Выбор инструментальных средств разработки ПС

Был проведен анализ 4-х СУБД: Oracle Personal edition (PE), Microsoft SQL Server 2017, Visual FoxPro и Access [9, 10, 11]. Основные характеристики представлены в таблице 4.

Таблица 4 – Сравнительные характеристики СУБД

Параметр	СУБД			
	Oracle Personal edition (PE)	SQL Server 2017	Visual FoxPro	Access
Название	2	3	4	5
1 Название, версия, фирма производитель	Oracle	MS SQL Server 2017 Management Studio,	Microsoft	Microsoft Access 2016
2 Поддерживаемые операционные системы.	Windows 2000, Windows NT, Windows XP, Windows Vista и Windows Server 2003/32-bit и 64-bit версии	Windows	Windows NT/2000, Linux, FreeBSD, NetBSD, UNIX	Windows
3 Требования к аппаратному обеспечению	Процессор: Intel Xeon, Intel EM64T, Intel Core 2duo, Intel Pentium IV Pentium 200, максимум RAM ОС, объем дисковой памяти должен быть не меньше 100 Мбайт	Процессор: AMD Ryzen, AMD Athlon 64, Intel Xeon, Intel EM64T, Intel Pentium IV; Память: 4 ГБ; Жесткий диск: 100 ГБ или более	Процессор: Intel Pentium 2020M, 32 Мбайт RAM, объем дисковой памяти должен быть не меньше 100 Мбайт	Процессор: AMD A8 8700 64Pentium 75, 32 Мбайт RAM, 100 Мбайт свободного места на диске

Продолжение таблицы 4

1	2	3	4	5
4 Поддерживаемые объекты БД.	База данных, схема данных, таблица, индекс, хранимая процедура, функция, представление, триггер, роль	База данных, схема данных, таблица, представление, синоним, индекс, процедура, функция, пакет, синоним, триггер, кластер	Таблица, хранимая процедура, функция, представление, триггер, домен	Таблица, индекс, представление (запрос), форма, отчёт, макрос, схема БД.
5 Формат файла БД.	.ora	.mdf	.lod	.dbf .dbc .bdf
6 Технология создания БД и объектов БД.	Визуальная и командная технология. Management Packs не включены.	Визуальная (Management Studio) и командная технология.	Командная технология. Визуальные средства сильно ограничены.	Визуальная технология
7 Оптимальный размер БД.	1 Гбайт	10 Гбайт	1 Гбайт	100 Мбайт
8 Удобство разработки и администрирования.	Предоставляет одному пользователю средства разработки и развертывания приложений, которые требуют полной совместимости с бд Oracle Database Standard Edition One.	Оптимальное соотношение мощности и удобства администрирования	Очень слабые средства администрирования.	СУБД для ПК, использующая реляционные РБД, имеющая объектно-ориентированный алгоритмический язык для работы с информацией, методы визуального программирования.

Продолжение таблицы 4

1	2	3	4	5
9 Сложность или простота работы с СУБД.	Лицензируется только на одного пользователя. Не предназначена для использования на мобильных устройствах.	Просто реализуется администрирование, интерфейс частично русифицирован.	Сложность в освоении и использовании Интерфейс полностью русифицирован.	СУБД проста в освоении за счёт русифицированного интерфейса.
10 Количество часов работы разработчика в среде	Около 2 часов	Полгода	Около 2 часов	Около 24 часов

В ходе сравнительного анализа четырех СУБД, сделан вывод, что Microsoft SQL Server 2017 достаточно удобна в администрировании, легко интегрируема с различными средами разработки, а главное наиболее изучена разработчиком.

Затем был проведен анализ 3 сред программирования Project Rider, Eclipse и Visual Studio 2015 [12, 13]. Подробные сравнительные характеристики представлены в таблице 5.

Таблица 5 – Сравнительные характеристики инструментальных средств

Параметр	Инструментальная среда		
	Project Rider	Eclipse	Visual Studio
Название			
1	2	3	4
1 Название, версия, фирма производитель	JetBrains	Eclipse Foundation	Visual Studio 2015, Microsoft
2 Поддерживаемые операционные системы	Windows, MacOS, Linux	Windows, MacOS, Linux	OC Windows (XP – 10)

Продолжение таблицы 5

1	2	3	4
3 Требования к аппаратному обеспечению	AMD A4, 512 Мбайт RAM 100 Мбайт свободного пространства на жёстком диске	AMD A8 - 9600, 200 Мбайт RAM, 500 Мбайт свободного пространства на жёстком диске	AMD A4 или Pentium 1ГГц или совместимый, 512 Мбайт RAM, 750 Мбайт свободного пространства на жёстком диске
4 Встроенный язык программирования	C#	C#, Java	Visual C++, Visual C#, Visual J#, Visual Basic .NET
6 Поддержка технологии ООП	+	+	+
7 Механизмы доступа к БД	dbExpress, BDE	JDBC	ADO .NET
8 Средства построения отчётов	QReport	Ireport, Birt	SQL Reporting Services
9 Утилиты для работы с БД	Интегрированная поддержка MySQL	Интегрированная поддержка Microsoft Access 2016	Интегрированная поддержка Microsoft SQL Server 2017
10 Уровень подготовки к работе	Интуитивно понятная	Большая функциональность требует дополнительного обучения для работы в этой среде.	Интуитивно понятная, дополняется бесплатными руководством и справочником основных функций.

Продолжение таблицы 5

1	2	3	4
11 Доступность	Самая дешевая версия Project Rider обойдется в 139 долларов за первый год использования.	Распространяется бесплатно.	Бесплатная версия Community
12 Количество часов работы разработчика в среде	Около 2 часов	Около 2 часов	3 года

В ходе анализа выбрана среда разработки Visual Studio, так как она является простой в использовании, интуитивно понятной, а также совместима с Microsoft SQL Server 2017, распространяется бесплатно, а также достаточно изучена разработчиком.

2.2 Разработка функционального назначения

2.2.1 Обоснование архитектуры ПО

Архитектура программного средства – это его строение, представление программного продукта как системы, состоящей из некоторой совокупности взаимодействующих подсистем [14]. В качестве подсистем будут выступать программные модули, так как разрабатываемая интеллектуальная система имеет модульную структуру.

Система включает в себя программные модули, которые организуют работу программного средства. Модули системы являются независимыми, однако, функционирование модулей отдельно не имеет никакого смысла. Модули программного средства содержат процедуры и функции для создания и редактирования базы данных, формирования отчетов.

Декомпозиция задачи на отдельные подзадачи привела к построению модульной архитектуры программной системы. Архитектура программы позволяет модифицировать отдельные модули без необходимости внесения изменений в остальные модули, подключать новые модули, а также способствует более быстрому пониманию функционирования системы в целом. Структура модулей программного комплекса представляет собой иерархическую структуру, которая демонстрирует порядок взаимодействия основных модулей программы. Иерархическая структура модулей программного средства на рисунке 6. Описание разработанных модулей представлено в таблице 6.

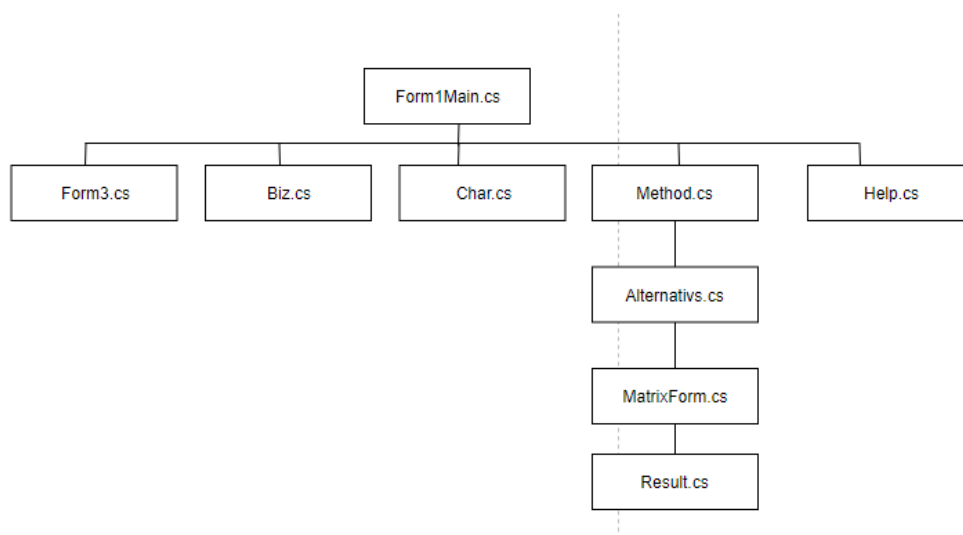


Рисунок 6 – Архитектура интеллектуальной системы

Таблица 6 – Спецификация модулей

Название модуля	Назначение модуля	Вызывающий модуль
Form1Main.cs	Главный модуль	
Form3.cs	Модуль информации об аэропорте и сотрудниках	Form1Main.cs
Biz.cs	Модуль информации о бизнес - процессах	Form1Main.cs
Char.cs	Модуль информации о характеристиках бизнес-процессах	Form1Main.cs
Method.cs	Модуль оценки бизнес-процессов	Form1Main.cs
Alternativs.cs	Модуль выбора альтернатив и критериев	Method.cs
MatrixForm.cs	Модуль редактирования матриц предпочтительности	Alternativs.cs
Result.cs	Модуль представления результатов	Method.cs
Help.cs	Модуль информации о программе.	Method.cs

2.2.2 Обоснование функционального назначения

Разрабатываемая программная система должна выполнять определенный ряд функций, которые можно объединить в иерархию, представленную на рисунке 7. Среди них обязательными функциями для систем, работающих с помощью баз данных являются функции «Авторизация», «Ведение справочных данных», «Ведение учетных данных», «Формирование отчетов». Кроме того, в программе реализована система поддержки принятия решения, на основе метода анализа иерархий.

АС	Авторизация	Логин/Пароль	Добавление/Обновление	Ф1
			Просмотр	Ф2
	Ведение справочных данных	Единицы измерения	Просмотр	Ф3
		Тип аэропорта	Просмотр	Ф4
		Тип характеристики бизнес-процессов	Просмотр	Ф5
		Тип телефона	Просмотр	Ф6
		Перечень бизнес-процессов(БП)	Просмотр	Ф7
	Ведение данных об аэропорте и сотрудниках	Данные об аэропорте	Добавление/Обновление	Ф8
			Просмотр	Ф9
		Данные о сотрудниках	Добавление/Обновление	Ф10
			Просмотр	Ф11
	Ведение данных о бизнес - процессах	Выбор бизнес-процесса	Формирование	Ф12
			Просмотр	Ф13
		Период действия	Добавление/Обновление	Ф14
			Просмотр	Ф15
	Ведение данных о характеристиках БП	Данные о характеристиках	Добавление/Обновление	Ф16
			Просмотр	Ф17
	Многокритериальный анализ	Выбор бизнес-процессов	Формирование	Ф18
			Просмотр	Ф19
		Выбор критериев	Формирование	Ф20
			Просмотр	Ф21
	Формирования отчета о результатах оценки	Сохранение в .docx	Формирование	Ф22
			Просмотр	Ф23

Рисунок 7 – Иерархия функций

На рисунке 8 представлена функциональная схема интеллектуальной системы.

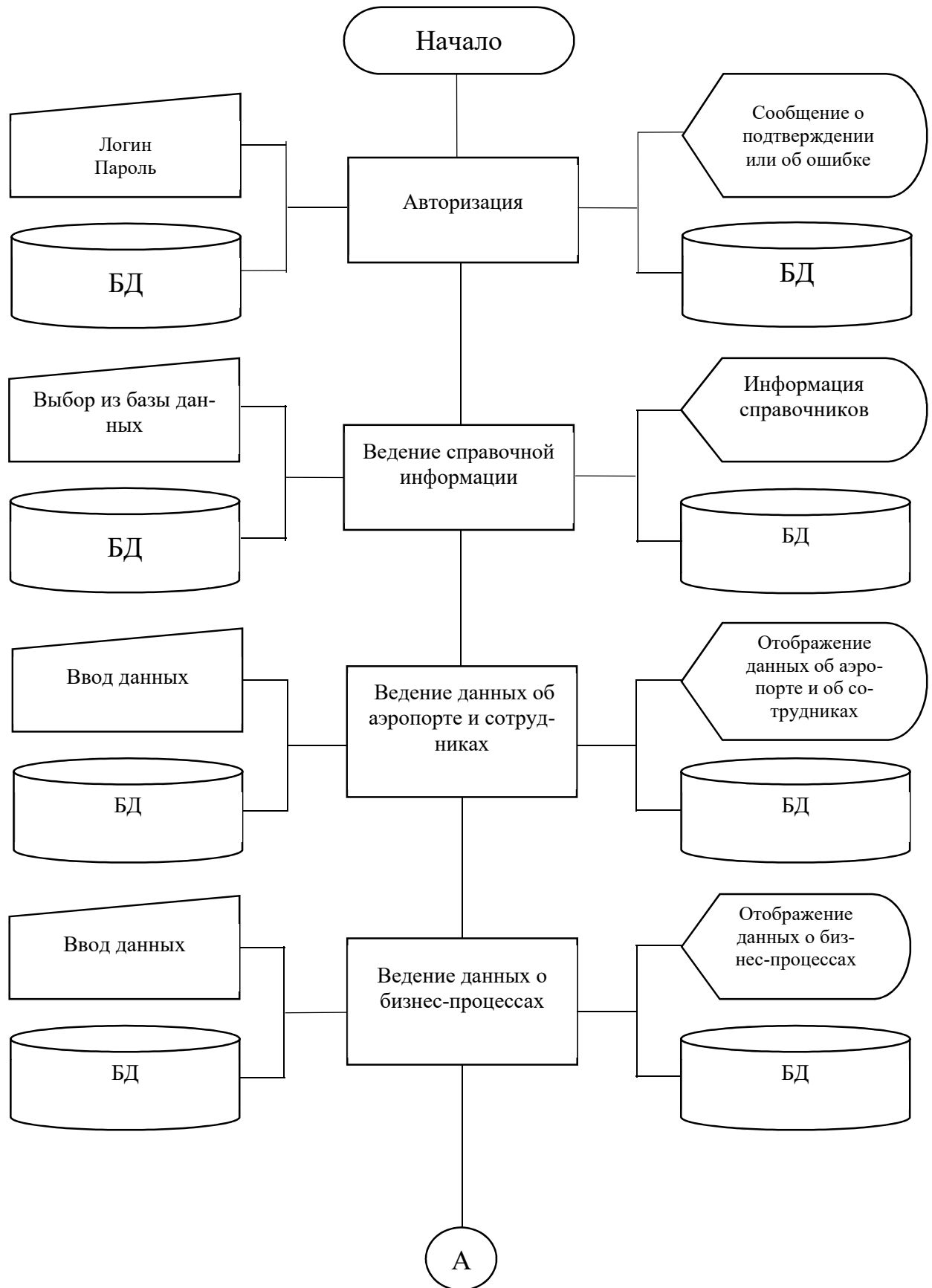


Рисунок 8 – Функциональная схема

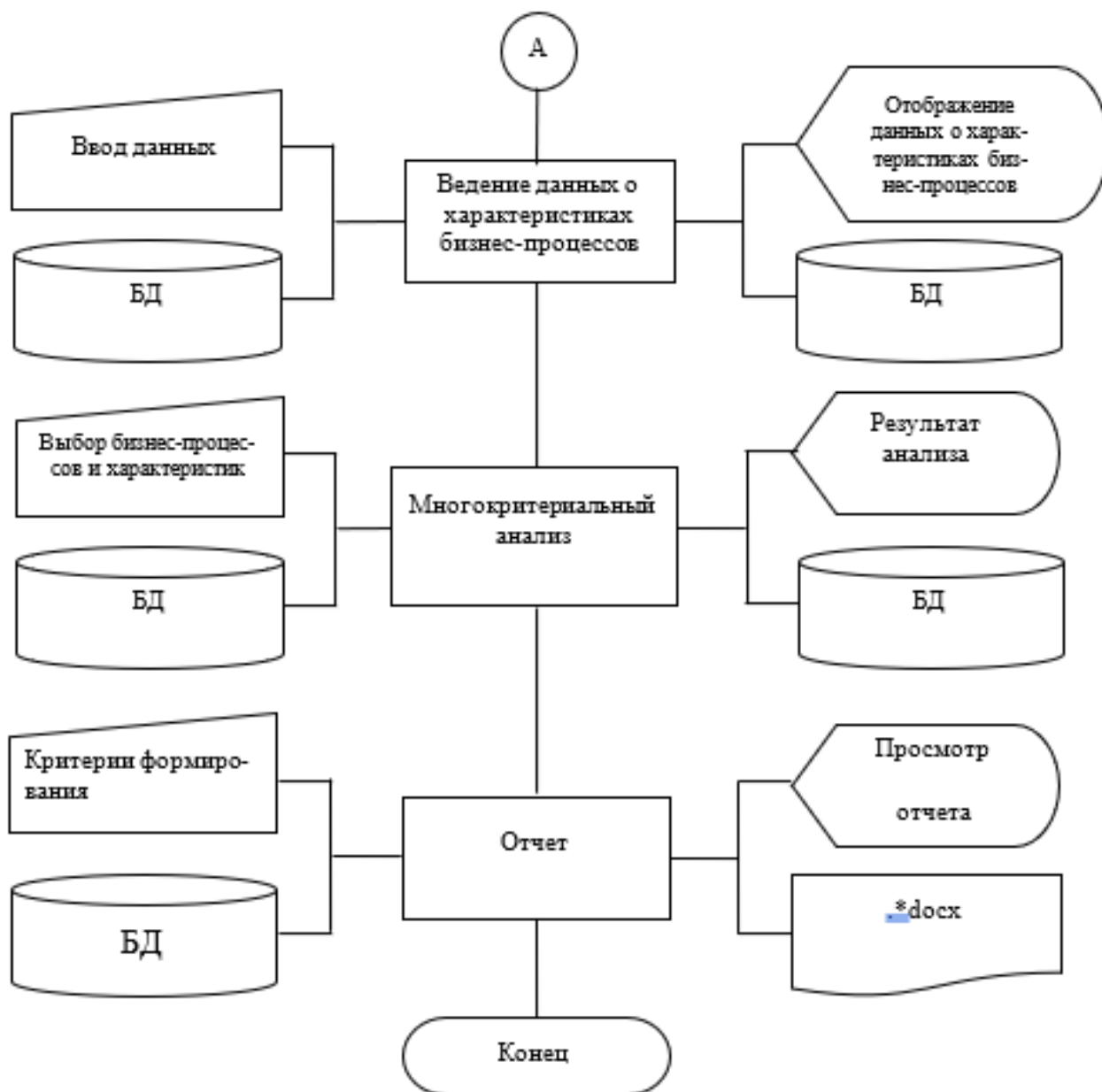


Рисунок 8, лист 23

2.3 Разработка информационной структуры данных

2.3.1 Внешний уровень архитектуры БД

В результате анализа предметной области было получено формализованное описание предметной области. В таблице 7 представлены классы объектов предметной области и их свойства.

Таблица 7 – Формализованное описание классов объектов

Объект (тип сущности) / свойство (атрибут)	Уникальный идентификатор	Физические характеристики (тип, длина)	Обязательность значения (м.б., д.б.)	Логические ограничения (диапазон значений, прописные, строчные буквы для символьных свойств и т.п.)	Процессы (генерация, ввод значений, возможность обновления, просмотра)
1	2	3	4	5	6
Физическое лицо					
Код физического лица	УИ1, ПК	число, 10	Должно быть	>0	Ген, Пр
Табельный номер		число, 10	Должно быть	>0	Вв, Пр
Фамилия		символы, 40	Должно быть	Первая буква заглавная	Вв, Пр, Об
Имя		символы, 40	Должно быть	Первая буква заглавная	Вв, Пр, Об
Отчество		символы, 40	Должно быть	Первая буква заглавная	Вв, Пр, Об
Документ о состоянии бизнес - процессов					
Номер документа	УИ2, ПК	число, 10	Должно быть	>0	Вв, Пр
Название		символы, 50	Должно быть	Первая буква заглавная	Вв, Пр
Дата		дата	Должно быть	дд.мм.гггг	Вв, Пр
Аэропорт					
Код аэропорта	УИ3, ПК	число, 10	Должно быть	>0	Ген, Пр
Название аэропорта		символы, 60	Должно быть	Первая буква заглавная	Вв, Пр, Об
Краткое название		символы, 5	Должно быть	Первая буква заглавная	Вв, Пр Об
Тип аэропорта					
Код типа аэропорта	УИ4, ПК	число, 10	Должно быть	>0	Ген, Пр
Название типа аэропорта		символы, 20	Должно быть	Первая буква заглавная	Вв, Пр Об
Краткое название		символы, 10	Должно быть	Первая буква заглавная	Вв, Пр Об
Бизнес – процессы (БП)					
Номер бизнес - процесса	УИ5, ПК	число, 10	Должно быть	>0	Ген, Пр Об
Название БП		символы, 50	Должно быть	Первая буква заглавная	Вв, Пр Об
Краткое название БП		символы, 10	Должно быть	Первая буква заглавная	Вв, Пр Об
Дата начала		дата	Должно быть	дд.мм.гггг	Вв, Пр Об

Продолжение таблицы 7

1	2	3	4	5	6
Дата окончания		дата	Должно быть	дд.мм.гггг	Вв, Пр Об
Номер работы	УИ6, ПК	число, 10	Должно быть	>0	Ген, Пр
Дата начала		дата	Должно быть	дд.мм.гггг	Вв, Пр Об
Дата окончания		дата	Должно быть	дд.мм.гггг	Вв, Пр Об
Позиция документа о состоянии бизнес – процесса					
Номер	УИ7, ПК	число, 10	Должно быть	>0	Ген, Пр
Значение		число, 10	Должно быть	>0	Вв, Пр
Единицы измерения					
Номер	УИ8, ПК	число, 10	Должно быть	>0	Ген, Пр
Название		символы, 50	Должно быть	Первая буква за- главная	Вв, Пр
Краткое назва- ние		символы, 10	Должно быть	Первая буква за- главная	Вв, Пр
Характеристика бизнес – процессов					
Номер	УИ9, ПК	число, 10	Должно быть	>0	Ген, Пр
Название		символы, 50	Должно быть	Первая буква за- главная	Вв, Пр Об
Краткое назва- ние		символы, 10	Должно быть	Первая буква за- главная	Вв, Пр Об
Должность					
Код	УИ10, ПК	число, 10	Должно быть	>0	Ген, Пр
Название		символы, 70	Должно быть	Первая буква за- главная	Вв, Пр
Краткое назва- ние		символы, 50	Должно быть	Первая буква за- главная	Вв, Пр
Перечень бизнес-процессов					
Номер	УИ11, ПК	число, 10	Должно быть	>0	Ген, Пр
Название		символы, 100	Должно быть	Первая буква за- главная	Вв, Пр
Краткое назва- ние		символы, 50	Должно быть	Первая буква за- главная	Вв, Пр
Тип характеристики					
Номер	УИ12, ПК	число, 10	Должно быть	>0	Ген, Пр

Продолжение таблицы 7

1	2	3	4	5	6
Название		символы, 70	Должно быть	Первая буква заглавная	Вв, Пр
Краткое название		символы, 50	Должно быть	Первая буква заглавная	Вв, Пр
Тип телефона					
Номер	УИ13, ПК	число, 10	Должно быть	>0	Ген, Пр
Название		символы, 50	Должно быть	Первая буква заглавная	Вв, Пр
Краткое название		символы, 10	Должно быть	Первая буква заглавная	Вв, Пр
Телефон					
Код	УИ14, ПК	число, 10	Должно быть	>0	Ген, Пр
Номер телефона		число, 10	Должно быть	>0	Вв, Пр

В таблице 7 используются следующие сокращения [15]:

- УИ – уникальный идентификатор;
- ПК – первичный ключ;
- д.б./м.б. – значение должно/может быть;
- Вв – ввод значения;
- Пр – просмотр значения;
- Об – обновление значения;
- Ген. – генерация значения.

Связи между классами объектов представлены в таблице 7. В таблице 7 используются следующие сокращения:

- 1 – мощность связи «один»;
- М – мощность связи «много»;
- д.б./м.б. – связь должна быть/связь может быть.

Таблица 8 – Формализованное описание отношений между классами объектов

Связь классов объектов		Название связи со стороны к/о		Тип связи со стороны к/о		Опц. связи к/о	
глав	подч	глав	подч	глав	подч	глав	подч
1	2	3	4	5	6	7	8
Единица измерения	Позиция документа о состоянии бизнес - процесса	относится	содержит	1	М	Д.б.	М.б.

Продолжение таблицы 8

1	2	3	4	5	6	7	8
Характеристика бизнес - процесса	Позиция документа о состоянии бизнес - процесса	соответствует	относится	1	М	Д.б.	Д.б.
Документ о состоянии бизнес – процессов	Позиция документа о состоянии бизнес - процесса	содержит	относится	1	М	Д.б.	Д.б.
Трудовой договор	Документ о состоянии бизнес - процессов	относится	содержит	1	М	Д.б.	Д.б.
Бизнес - процесс	Документ о состоянии бизнес - процессов	соответствует	относится	1	М	Д.б.	Д.б.
Физическое лицо	Трудовой договор	относится	содержит	1	М	М.б.	Д.б.
Аэропорт	Бизнес - процесс	соответствует	относится	1	М	Д.б.	Д.б.
Аэропорт	Трудовой договор	соответствует	относится	1	М	Д.б.	Д.б.
Тип аэропорта	Аэропорт	содержит	относится	1	М	Д.б.	Д.б.
Перечень бизнес-процессов	Бизнес-процесс	содержит	относится	1	М	Д.б.	Д.б.
Тип характеристики	Характеристика бизнес-процесса	содержит	относится	1	М	Д.б.	Д.б.
Тип телефона	Телефон	содержит	относится	1	М	Д.б.	Д.б.
Телефон	Физическое лицо	соответствует	относится	1	М	Д.б.	Д.б.
Должность	Физическое лицо	соответствует	относится	1	М	Д.б.	Д.б.

Каждая ПОЗИЦИЯ ДОКУМЕНТА О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССА содержит одну ЕДИНИЦУ ИЗМЕРЕНИЯ, каждая ЕДИНИЦА ИЗМЕРЕНИЯ может относиться нескольким ПОЗИЦИЯМ ДОКУМЕНТА О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССА.

Каждая ХАРАКТЕРИСТИКА БИЗНЕС-ПРОЦЕССА относится к ПОЗИЦИИ ДОКУМЕНТА О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССА, к каждой ПОЗИЦИИ ДОКУМЕНТА О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССА может относиться только одна ХАРАКТЕРИСТИКА БИЗНЕС-ПРОЦЕССА.

Каждый ДОКУМЕНТ О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССА содержит много ПОЗИЦИЙ этого документа, каждая ПОЗИЦИЯ ДОКУМЕНТА О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССОВ относится только к одному ДОКУМЕНТУ.

Каждый ДОКУМЕНТ О СОСТОЯНИИ БИЗНЕС-ПРОЦЕССОВ содержит много информации о ТРУДОВОМ ДОГОВОРЕ, каждая информация о ТРУДОВОМ ДОГОВОРЕ относится к одному ДОКУМЕНТУ.

Каждый БИЗНЕС-ПРОЦЕСС соответствует одному ДОКУМЕНТУ, каждый ДОКУМЕНТ содержит много БИЗНЕС-ПРОЦЕССОВ.

Одно ФИЗИЧЕСКОЕ ЛИЦО относится к одному ТРУДОВОМУ ДОГОВОРУ, ТРУДОВОЙ ДОГОВОР содержит много ФИЗИЧЕСКИХ ЛИЦ.

Одному АЭРОПОРТУ соответствует много БИЗНЕС-ПРОЦЕССОВ, БИЗНЕС-ПРОЦЕСС относится к одному АЭРОПОРТУ.

АЭРОПОРТ относится к одному ТИПУ АЭРОПОРТА, ТИП АЭРОПОРТА содержит много АЭРОПОРТОВ.

Таблица 9– Состав и роли персонала АС .

Доступные операции	Персонал АС		
	Технический персонал	Конечные пользователи	
	Роль «Прикладной программист (ПП)»	Роль «Оператор» - сотрудник отдела стратегического развития	Роль «Руководитель/ ЛПР» - руководитель отдела стратегического развития
Добавление данных Insert (I)	-	+	-
Обновление данных Update (U)	-	+	-
Удаление данных Delete (D)	+	-	-
Выборка/ чтение данных Select/ Read (R)	+	+	+

2.3.2 Концептуальный уровень архитектуры БД

2.3.2.1 Информационно – логическая модель предметной области

Исходными данными для построения информационно-логической модели предметной области (ИЛМ) являются результаты анализа предметной области, представленные в виде описания классов объектов и связей между ними. Чаще всего ИЛМ предметной области представляют в терминах семантической модели данных, в виде ER-диаграммы предметной области [15].

В настоящее время существуют разнообразные нотации построения ER-модели. Подробно рассмотрим самую распространённую из них – методологию Ричарда Баркера[16].

Методология Ричарда Баркера построена на базе следующих элементов: класс объектов, свойство класса объектов, уникальные идентификаторы, опциональность свойств, мощность (тип), опциональность и переносимость связей, уникальность объектов из связей, супертипы, подтипы, арки.

В методологии используются следующие соглашения [15]:

- класс объектов отображается в виде четырехугольника с закругленными углами, а имя класса объектов указывается внутри четырехугольника, это имя существительное в единственном числе, отображенное заглавными буквами;
- свойства записываются внутри четырехугольника, отображающего класс объектов строчными буквами, это имя существительное в единственном числе;
- четырехугольник, отображающий класс объектов, можно увеличивать до любых размеров, четырехугольники могут быть разных размеров;
- опциональность свойств помечается: обязательное свойство – звездочкой (*), необязательное – кружочком (o);
- уникальный идентификатор помечается #, если уникальных идентификаторов несколько, тогда каждый помечается номером, указанным в скобках, например, # (1), #(2);
- обязательная связь помечается сплошной линией, необязательная связь пунктирной линией;
- тип (мощность) связи «один» помечается линией, «много» — «вороньей лапой».

Каждый объект обладает определённым набором свойств. Для объектов одного класса набор этих свойств одинаков, а эти значения могут различаться[15].

При описании предметной области необходимо отразить связь между объектами разных классов. Различают связи типа «один к одному» (1:1), «один ко многим» (1:M), «многие ко многим» (M:M)[15].

Концептуальная инфологическая модель предметной области, построенная по методологии Ричарда Баркера, представлена на рисунке 9.

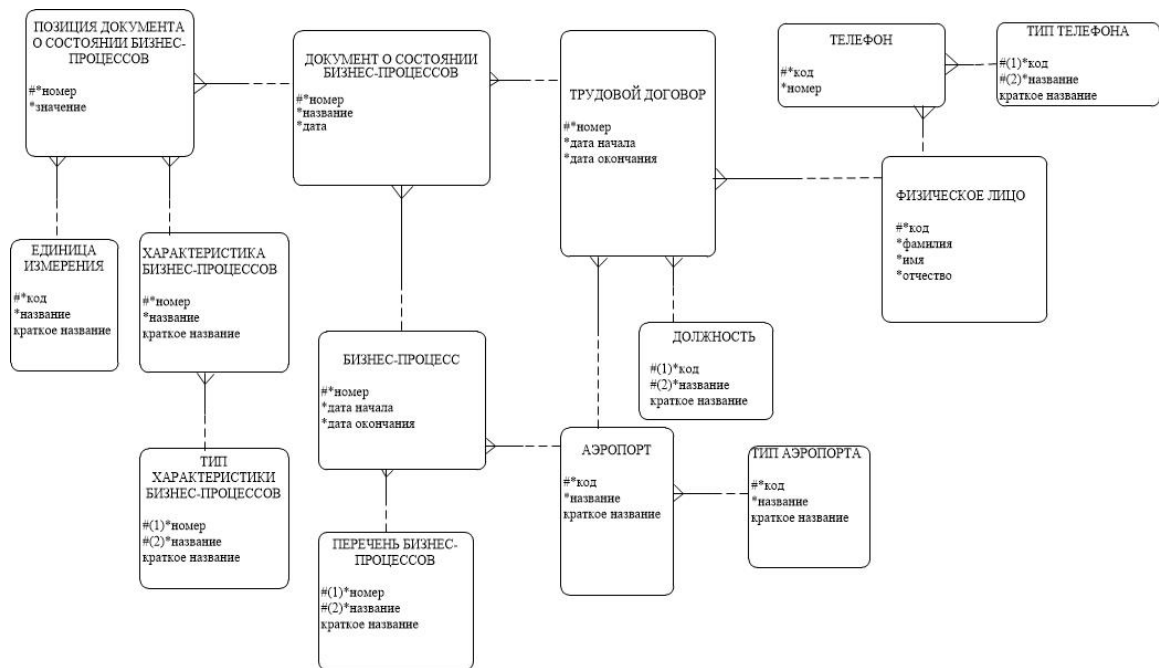


Рисунок 9 – ER – диаграмма

2.3.2.2 Даталогическая модель БД

Построение даталогической модели базы данных – это следующий этап проектирования базы данных методом «нисходящего» проектирования. Исходными данными для даталогического проектирования является информационно-логическая модель предметной области [15]. В результате даталогического моделирования должна быть получена логическая структура базы данных, описанная в терминах реляционной модели данных на основе физических записей.

При переходе к даталогической модели следует помнить, что инфологическая модель включает в себя всю информацию о предметной области, необходимую и достаточную для проектирования баз данных [15].

Рассмотрим некоторые основные правила формирования логической структуры реляционной базы данных на основе ER-диаграммы предметной области.

1) Преобразование простых классов объектов. Это классы объектов, информация о которых первой появляется в предметной области.

2) Преобразование связи 1:M. Связь реализуется копированием первичного ключа из реляционного отношения на стороне «один» в реляционное отношение на стороне «много», из главного отношения в подчиненное. Новому появившемуся атрибуту присваивается уникальное в пределах отношения имя. Этот вновь появившийся атрибут помечается как внешний ключ.

3) Преобразование связи 1:1. В ER-диаграмме связь 1:1 может иметь разную опциональность. От этого зависит ее отображение в схеме БД. Внешний ключ создается копированием первичного ключа из главного отношения в подчиненное.

4) Преобразование рекурсивной связи. Поскольку рекурсивная связь – это связь между объектами одного класса объектов, то внешний ключ создается путем копирования первичного ключа в эту же схему отношения.

5) Реализация связей М:М. Наличие в ER-диаграмме связей М:М – это недообследование предметной области и в этом случае модель предметной области не совсем адекватна.

Концептуальная даталогическая модель базы данных представлена на рисунке 10.

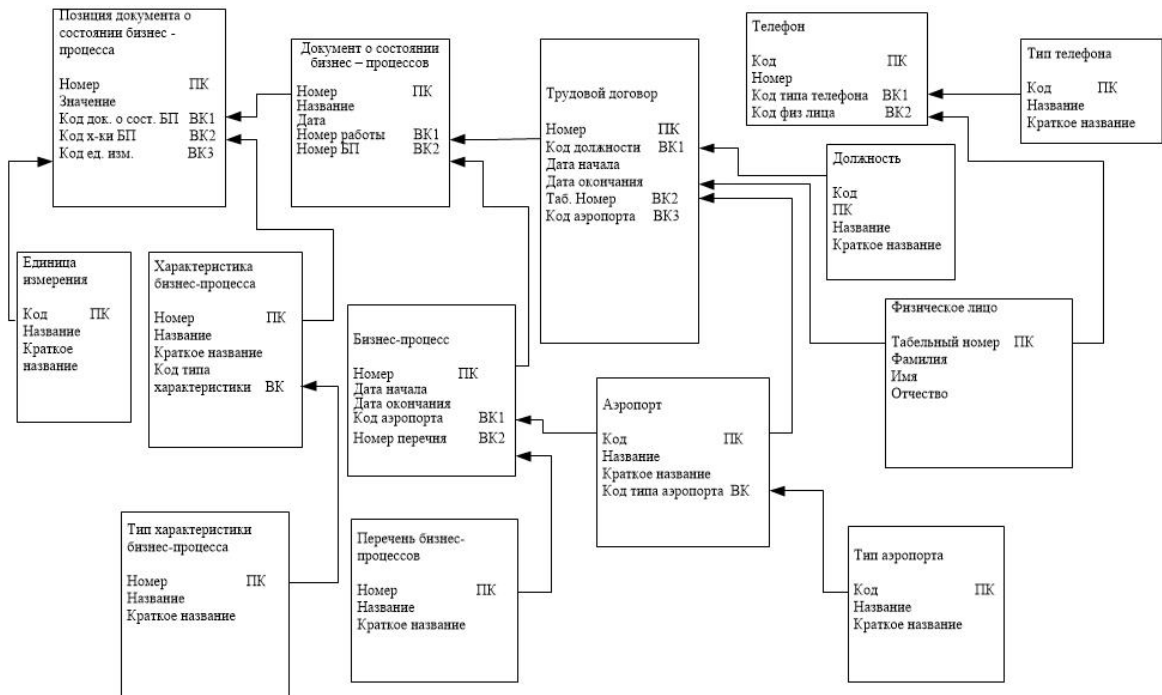


Рисунок 10 – Даталогическая модель

Далее проверим соответствие полученной модели предметной области выполнению состава функций АИС. Это делается с помощью перекрестной проверки полученной модели. Проверка приводится в формализованном виде, в таблице 10.

Таблица 10 – Перекрестная проверка модели предметной области и иерархии функций

Функции АС	Позиция документа о состоянии Б	Документ о состоянии БП	Физическое лицо	Единица измерения	Характеристика БП	Бизнес – процесс	Тип аэропорта	Аэропорт	Перечень БП	Тип характеристики БП	Должность	Трудовой договор	Телефон	Тип телефона
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Продолжение таблицы 10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ф(1)			IU											
Ф(2)			R											
Ф(3)				R										
Ф(4)							R							
Ф(5)										R				
Ф(6)														R
Ф(7)									R					
Ф(8)								IU						
Ф(9)								R						
Ф(10)			IU								IU	IU	IU	
Ф(11)			R								R	R	R	
Ф(12)						R			R					
Ф(13)						R			R					
Ф(14)					IU									
Ф(15)					R									
Ф(16)	IU	IU			IU									
Ф(17)	R	R			R									
Ф(18)						IU			R					
Ф(19)						R								
Ф(20)					IU					R				
Ф(21)					R					R				
Ф(22)	IU	IU			IU	IU								
Ф(23)	R					R								

В таблице использованы сокращения: I – операция добавления; U – операция обновления, R – операция чтения (выборки).

Перекрестная проверка иерархии функций и модели данных позволяет выявить избыточность или недостаточность построенной модели данных. Если в таблице присутствует пустой столбец, это означает, что модель избыточна, если же пустая строка, то недостаточна. Т.к. в таблице нет пустых строк и столбцов, то из этого следует, что построенная нами модель полностью отвечает функциональным возможностям разрабатываемой АИС.

2.3.3 Внутренний уровень архитектуры БД

Внутренний уровень архитектуры базы данных отображается составом соответствующих объектов. В проекте, реализуемом средствами современной СУБД, могут быть использованы домены (типы), таблицы, последовательности, триггеры, хранимые процедуры, представления (просмотры), пользователи, роли, курсоры и др. В рамках курсовой работы создается техническое описание таблиц проектируемой базы данных на языке определения данных (ЯОД) выбранной СУБД. В таблице 9 приведен пример технического описания реляционной таблицы «Сотрудник» на ЯОД СУБД SQL Server 2017. Описание представлено конструкциями команд языка SQL. Стандарт языка SQL поддерживается всеми современными СУБД [15].

Таблица 11 – Техническое описание таблицы «Аэропорт»

Имя отношения в схеме РБД – «Аэропорт»				
Имя поля	Kod	Kod_TA	Nazv	Kr_Nazv
Ключ	Primary Key	Foreign Key		
Тип, длина	Integer	Integer	Varchar(50)	Varchar(10)
Обязательность значения	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)	Check (Kod_TA >0)		
Примеры данных	412	23	Оренбург	Орб

Таблица 12 – Техническое описание таблицы «Тип аэропорта»

Имя отношения в схеме РБД – «Тип аэропорта.»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(50)	Varchar(10)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	27	Имя	И

Таблица 13 – Техническое описание таблицы «Характеристика бизнес-процесса»

Имя отношения в схеме РБД – «Характеристика бизнес - процесса»			
Имя поля	Nomer	Nazv	Kr nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(50)	Varchar(10)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Nomer >0)		
Примеры данных	46	Имя	И

Таблица 14 – Техническое описание таблицы «Бизнес - процесс»

Имя отношения в схеме РБД – «Бизнес - процесс»						
Имя поля	Nomer	Date_n	Date_o	Kod air	Nazv	Kr nazv
1	2	3	4	5	6	7
Ключ	Primary Key			Foreign Key		
Тип, длина	Integer	Date	Date	Integer	Varchar(50)	Varchar(10)

Продолжение таблицы 14

1	2	3	4	5	6	7
Обязательность Значения	Not Null	Not Null	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Nomer >0)			Check (Kod_air >0)		
Примеры данных	412	12.03.2017	12.12.2018	35	Имя	И

Таблица 15 – Техническое описание таблицы «Позиция документа о состоянии бизнес - процесса»

Имя отношения в схеме РБД – «Позиция документа о состоянии бизнес - процесса»					
Имя поля	Nomer	Znach	Kod_doc_sost	Kod_xar_bp	Kod_ei
Ключ	Primary Key		Foreign Key	Foreign Key	Foreign Key
Тип, длина	Integer	Integer	Integer	Integer	Integer
Обязательность значения	Not Null	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Nomer >0)	Check (Znach >0)	Check (Kod_doc_sost >0)	Check (Kod_xar_bp >0)	Check (Kod_ei >0)
Примеры данных	7	87	4	456	234

Таблица 16 – Техническое описание таблицы «Единица измерения»

Имя отношения в схеме РБД – «Единица измерения»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(50)	Varchar(10)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	23	Название	Назв

Таблица 17 – Техническое описание таблицы «Физическое лицо»

Имя отношения в схеме РБД – «Физическое лицо»				
Имя поля	Tab_Nom	Familia	Imya	Otch
Ключ	Primary Key			
Тип, длина	Integer	Varchar(50)	Varchar(50)	Varchar(50)
Обязательность значения	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Tab_Nom >0)			
Примеры данных	347	Петров	Петр	Петрович

Таблица 18 – Техническое описание таблицы «Трудовой договор»

Имя отношения в схеме РБД – «Трудовой договор»						
Имя поля	Nomer	Tab_No m	Date_n	Date_o	Kod_dol	Kod_air
Ключ	Primary Key	Foreign Key				Foreign Key
Тип, длина	Integer	Varchar	Date	Date	Integer	Integer
Обязательность значения	Not Null	Not Null	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Nomer >0)				Check (Kod_air >0)	Check (Kod_air >0)
Примеры данных	6421	Имя	1.03.2018	2.03.2018	346	562

Таблица 19 – Техническое описание таблицы «Документ о состоянии бизнес – процессов»

Имя отношения в схеме РБД – «Документ о состоянии бизнес – процессов»					
Имя поля	Nomer	Nazv	Date	Nom_rab	Nom_bp
Ключ	Primary Key			Foreign Key	Foreign Key
Тип, длина	Integer	Varchar	Date	Integer	Integer
Обязательность значения	Not Null	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Nomer >0)			Check (Nom_rab >0)	Check (Nom_bp>0)
Примеры данных	7	Название	12.03.2017	456	234

Таблица 20 – Техническое описание таблицы «Должность»

Имя отношения в схеме РБД – «Должность»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(70)	Varchar(50)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	23	Название	Назв

Таблица 21 – Техническое описание таблицы «Перечень бизнес-процессов»

Имя отношения в схеме РБД – «Перечень бизнес-процессов»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(100)	Varchar(50)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	23	Название	Назв

Таблица 22 – Техническое описание таблицы «Тип характеристики»

Имя отношения в схеме РБД – «Тип характеристики»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(50)	Varchar(20)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	23	Название	Назв

Таблица 23 – Техническое описание таблицы «Тип телефона»

Имя отношения в схеме РБД – «Тип телефона»			
Имя поля	Kod	Nazv	Kr_nazv
Ключ	Primary Key		
Тип, длина	Integer	Varchar(50)	Varchar(10)
Обязательность значения	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		
Примеры данных	23	Название	Назв

Таблица 24 – Техническое описание таблицы «Телефон»

Имя отношения в схеме РБД – «Телефон»				
Имя поля	Kod	Nomer_tlf	Kod_TT	Kod_FL
Ключ	Primary Key		Foreign Key	Foreign Key
Тип, длина	Integer	Varchar(50)	Integer	Integer
Обязательность значения	Not Null	Not Null	Not Null	Not Null
Логическое ограничение	Check (Kod >0)		Check (Kod_TT >0)	
Примеры данных	412	23	89198432067	413

2.4 Разработка алгоритма программного средства

Укрупненная схема алгоритма работы интеллектуальной системы представлена на рисунке 11.

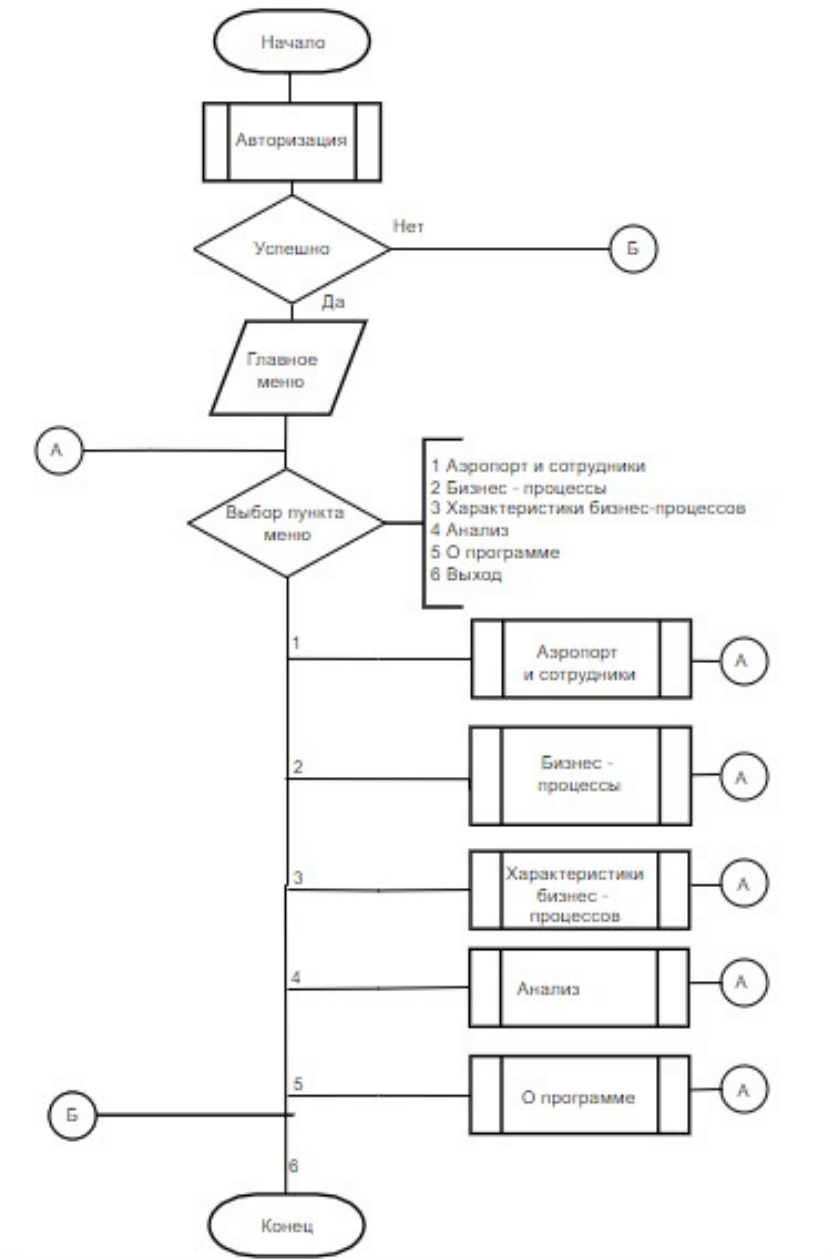


Рисунок 11 – Укрупненная схема алгоритма работы интеллектуальной системы

Алгоритм метода многокритериального выбора в иерархиях с различным числом и составом альтернатив под критериями представлен диаграммой деятельности (рисунок 12).



Рисунок 12 – Алгоритм многокритериального выбора в иерархиях с различным числом и составом альтернатив под критериями

2.5 Тестирование программного средства

Тестирование интеллектуальной системы позволяет выявить нестандартные ситуации при ее эксплуатации, приводящие к потере данных, исключениям в процессе вычислений.

Для корректной работы системы подобные ситуации были обработаны и сопровождаются уведомлениями пользователю.

Поскольку работа интеллектуальной системы основана на взаимодействии с базой данных, существует вероятность возникновения ситуаций, при которых обработка данных невозможна вследствие отсутствия соединения с базой. В подобных ситуациях пользователь уведомляется сообщением об ошибке (рисунок 13).

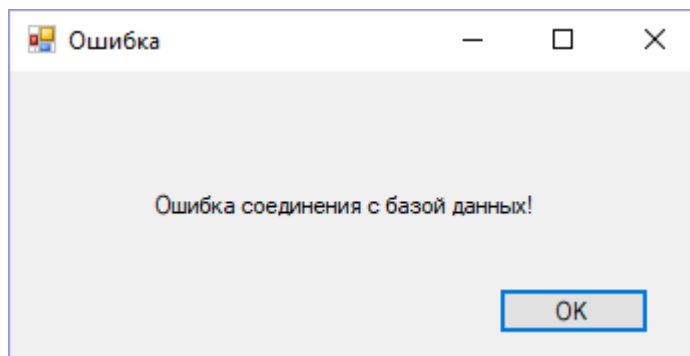


Рисунок 13 – Сообщение об ошибке

В соответствии с разработанной моделью данных существуют обязательные и необязательные для заполнения поля ввода. С целью предупреждения потери данных при попытке сохранения и наличии незаполненных обязательных полей пользователь уведомляется сообщением (рисунок 14).

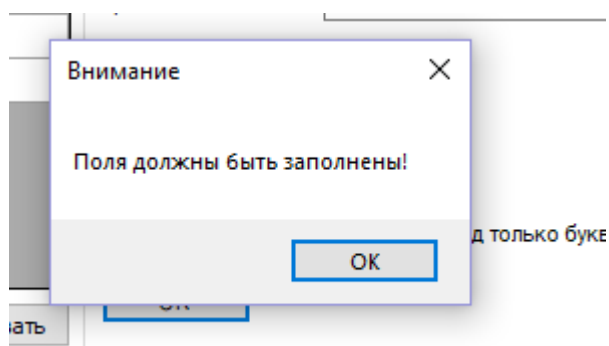


Рисунок 14 – Сообщение об ошибке

В случае ошибочности введения данных или попытке ввести пустую запись, программа подскажет об этом (рисунок 15).

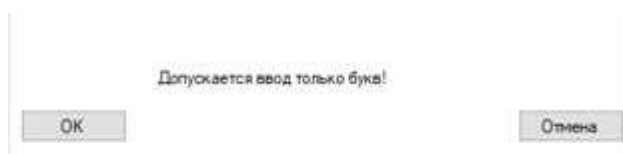


Рисунок 15 – Сообщение пользователю

Тестирование математической интеллектуальной системы выполнено в программном пакете MathCAD.

Исходные данные для тестирования представлены на рисунках 16 - 18.

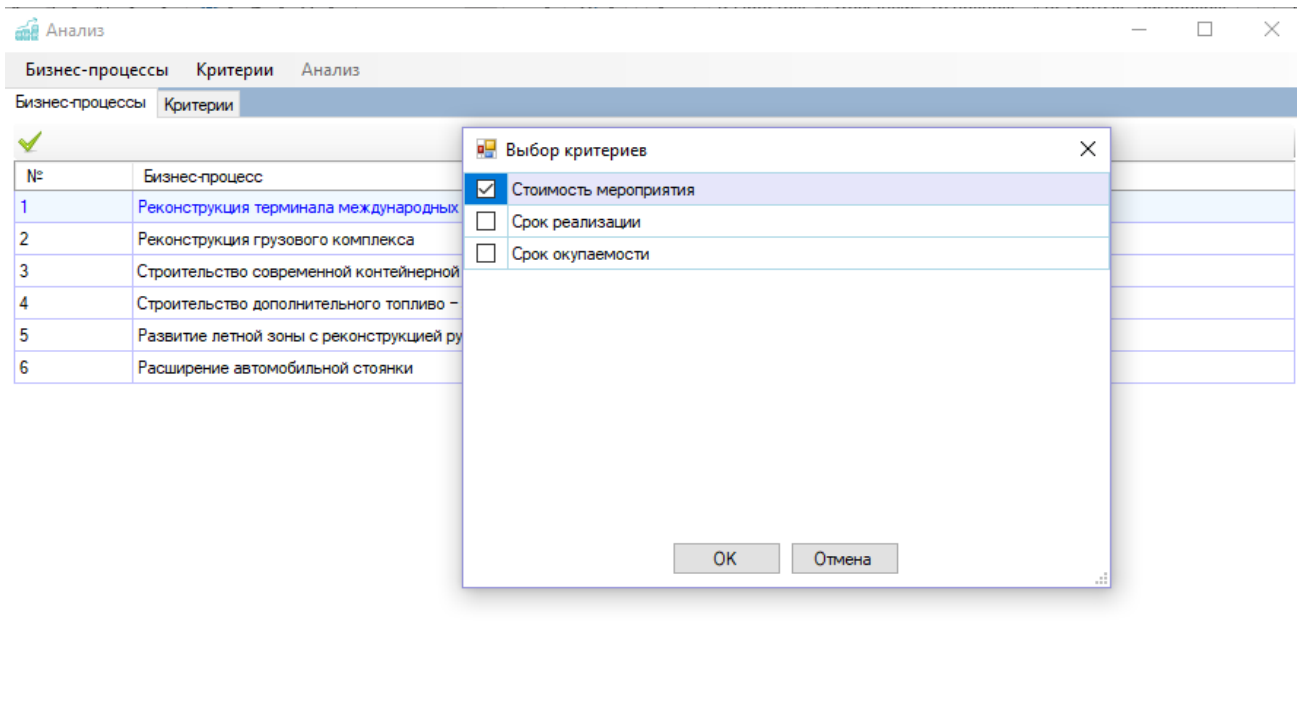


Рисунок 16 – Исходные данные

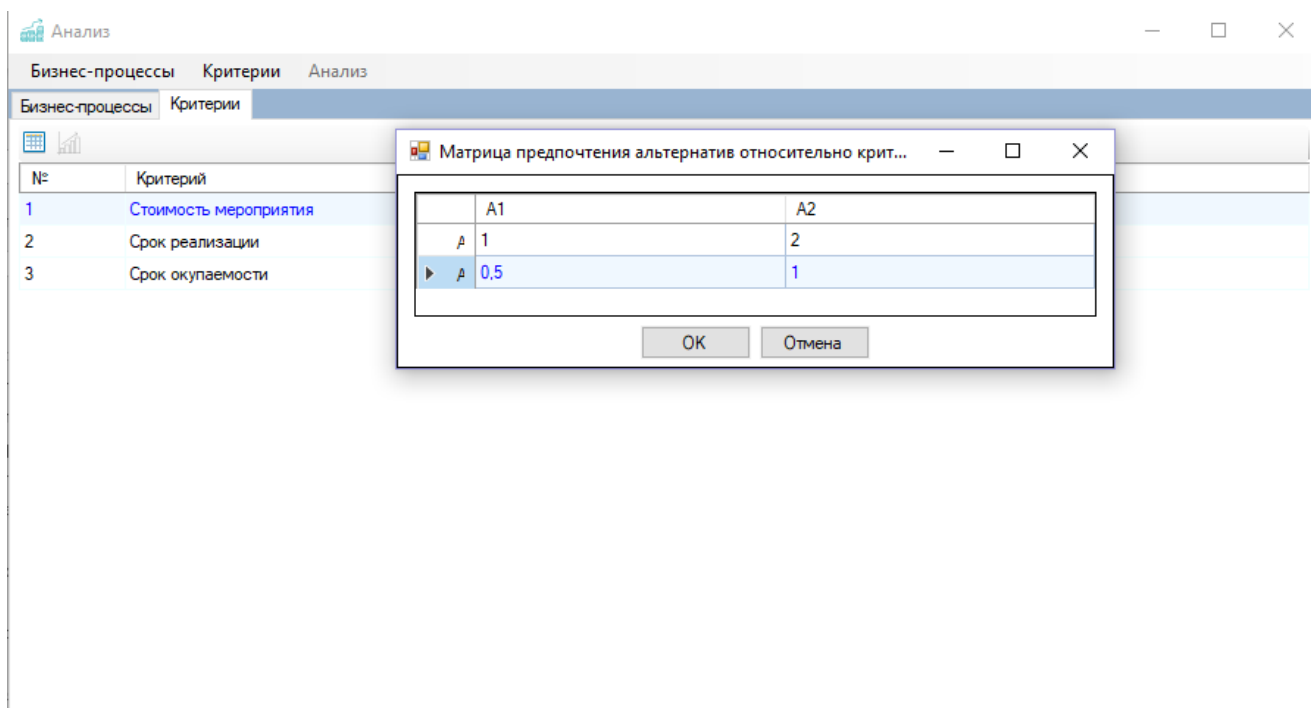


Рисунок 17 – Матрица предпочтения альтернатив относительно критериев

	K1	K2	K3
K1	1	1	1
K2	1	1	1
K3	1	1	1

Рисунок 18 – Матрица предпочтений альтернатив относительно Фокуса

В соответствии с заданными критериями сравнения бизнес -процессов и матрицами предпочтений альтернатив относительно выбранных критериев был выполнен анализ, в результате которого выбрана наилучшая альтернатива (рисунок 19).

Приоритет альтернатив по критерию K2
0,255
0,325
0,42

Матрица предпочтения альтернатив относительно критерия K3
1 0,333
3 1

$c=1,77766666666667$
Приоритет альтернатив по критерию K3
0,25
0,75

Приоритет альтернатив относительно фокуса F
0,333
0,333
0,333

Матрица A
0,667 0 0
0,333 0 0
0 0,255 0
0 0,325 0
0 0,42 0,25
0 0 0,75

Матрица L
0,286 0 0
0 0,429 0
0 0 0,286

$k=0,3333333333333333$
Матрица B
3 0 0 0 0 0
0 3 0 0 0 0
0 0 3 0 0 0
0 0 0 3 0 0
0 0 0 0 3 0
0 0 0 0 0 3

Вектор приоритетов альтернатив
0,19 0,095 0,109 0,139 0,252 0,214

Альтернатива "Развитие летной зоны с реконструкцией рулежных дорожек" является наилучшей.
Приоритет: 0,252

Рисунок 19 – Результат анализа

Результатом явилась альтернатива «Развитие летной зоны с реконструкцией рулежных дорожек», ее приоритет равен 0,252.

Реализация вычислений в программном пакете MathCAD представлена на рисунках 21-23.

Матрица предпочтения альтернатив относительно критерия K1

$$K1 := \begin{pmatrix} 1 & 2 \\ 0.5 & 1 \end{pmatrix}$$

Матрица предпочтения альтернатив относительно критерия K2

$$K2 := \begin{pmatrix} 1 & 0.333 & 2 \\ 3 & 1 & 0.25 \\ 0.5 & 4 & 1 \end{pmatrix}$$

Матрица предпочтения альтернатив относительно критерия K3

$$K3 := \begin{pmatrix} 1 & 0.333 \\ 3 & 1 \end{pmatrix}$$

Матрица предпочтения альтернатив относительно фокуса

$$F := \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$N := 3$ Число критериев

$$Y := \begin{pmatrix} \frac{1}{N} & \frac{1}{N} & \frac{1}{N} \end{pmatrix} = (0.333 \ 0.333 \ 0.333)$$

Рисунок 20 – Задание исходных данных

Приоритет альтернатив по критерию K1

$$K1 := K1 \cdot \begin{pmatrix} \frac{1}{N} \\ \frac{1}{N} \end{pmatrix} = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \quad c := \sum K1 = 1.5 \quad K1 := \frac{K1}{c} = \begin{pmatrix} 0.667 \\ 0.333 \end{pmatrix}$$

Приоритет альтернатив по критерию K2

$$K2 := K2 \cdot \begin{pmatrix} \frac{1}{N} \\ \frac{1}{N} \\ \frac{1}{N} \end{pmatrix} = \begin{pmatrix} 1.111 \\ 1.417 \\ 1.833 \end{pmatrix} \quad c := \sum K2 = 4.361 \quad K2 := \frac{K2}{c} = \begin{pmatrix} 0.255 \\ 0.325 \\ 0.42 \end{pmatrix}$$

Приоритет альтернатив по критерию K3

$$K3 := K3 \cdot \begin{pmatrix} \frac{1}{N} \\ \frac{1}{N} \end{pmatrix} = \begin{pmatrix} 0.444 \\ 1.333 \end{pmatrix} \quad c := \sum K3 = 1.778 \quad K3 := \frac{K3}{c} = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix}$$

Рисунок 21 – Определение приоритета альтернатив по критериям

Приоритет критериев относительно фокуса F

$$F := F \cdot \begin{pmatrix} \frac{1}{N} \\ \frac{1}{N} \\ \frac{1}{N} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad c := \sum F = 3 \quad F := \frac{F}{c} = \begin{pmatrix} 0.333 \\ 0.333 \\ 0.333 \end{pmatrix}$$

$$A := \begin{pmatrix} K1_{0,0} & 0 & 0 \\ K1_{1,0} & 0 & 0 \\ 0 & K2_{0,0} & 0 \\ 0 & K2_{1,0} & 0 \\ 0 & K2_{2,0} & K3_{0,0} \\ 0 & 0 & K3_{1,0} \end{pmatrix} = \begin{pmatrix} 0.667 & 0 & 0 \\ 0.333 & 0 & 0 \\ 0 & 0.255 & 0 \\ 0 & 0.325 & 0 \\ 0 & 0.42 & 0.25 \\ 0 & 0 & 0.75 \end{pmatrix}$$

Рисунок 22 – Определение приоритета альтернатив по критериям относительно фокуса

$$R_K1 := 2 \quad R_K2 := 3 \quad R_K3 := 2 \quad N := R_K1 + R_K2 + R_K3 = 7$$

$$I := \begin{pmatrix} \frac{R_K1}{N} & 0 & 0 \\ 0 & \frac{R_K2}{N} & 0 \\ 0 & 0 & \frac{R_K3}{N} \end{pmatrix} = \begin{pmatrix} 0.286 & 0 & 0 \\ 0 & 0.429 & 0 \\ 0 & 0 & 0.286 \end{pmatrix}$$

$$B := \begin{pmatrix} \frac{1}{k} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{k} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{k} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{k} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{k} \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

$$(A \cdot L \cdot F)^T \cdot B = (0.19 \quad 0.095 \quad 0.109 \quad 0.139 \quad 0.252 \quad 0.214)$$

Рисунок 23 – Результат анализа

В результате вычислений в программном пакете MathCAD были получены результаты, совпадающие с расчётными значениями разработанной интеллектуальной системы. Следовательно, программа работает корректно.

3 Разработка технологической документации

3.1 Руководство системного администратора

Требования к аппаратным ресурсам

Минимальная конфигурация:

- процессор AMD A6 и выше;
- тактовая частота процессора 1,7 ГГц и выше;
- RAM 1024 Мб;
- свободное пространство на жёстком диске не менее 30 Мб плюс размер файла базы данных;
- SVGA 1280×1024;
- клавиатура;
- принтер;
- мышь.

Разработанное программное средство предназначено для поддержки принятия решений. ПС позволяет осуществлять поиск нужной информации. Система поддержки принятия решения позволяет проводить выбор оптимального самолета с помощью нечеткого многокритериального анализа, повышая тем самым эффективность и обоснованность принимаемых решений.

Программная средство должно работать под управлением операционной системы MS Windows 7 и выше на персональных компьютерах. Так же, требуется наличие следующего программного обеспечения: SQL Server 2017.

В проводнике Windows необходимо, при помощи указателя мыши перейти к той папке, в которой вы ранее сохраняли проект. Дважды щёлкните на папку, которую следует открыть.

Затем необходимо отыскать файл Project1.exe. Используйте горизонтальную и вертикальную полосы прокрутки, в том случае, если в текущей папке много файлов и каталогов. Дважды щёлкните на данном файле левой кнопкой мыши.

Если в папке «Мой компьютер» или ее подпапках нет файла или папки, которые требуется открыть, найдите этот файл или папку, нажав кнопку Поиск. Чтобы запустить средство поиска, нажмите кнопку Пуск и выберите команду Найти.

Входными данными для программного средства являются данные о сотрудниках, аэропорте, бизнес-процессах, характеристиках бизнес-процессов, трудовых договорах, физических лицах и договорах.

К выходным данным программного продукта можно отнести результаты и процесс принятия решения с помощью нечеткого многокритериального решения, выбор лучшего бизнес-процесса.

В базе данных интеллектуальной системы находятся 14 взаимосвязанных таблиц:

- Air – таблица аэропортов;
- Tip Air – таблица типа аэропорта;
- BP – таблица бизнес-процессов;

- Doc_Sost – таблица документа о состоянии бизнес-процессов;
- Dolgnost – таблица должностей;
- EI – таблица единиц измерения;
- Fiz_lit – таблица физических лиц;
- Per_BP – таблица перечня бизнес-процессов;
- Poz_doc– таблица позиции документа о состоянии бизнес-процессов;
- T_Xar– таблица типов характеристик;
- Telefon– таблица телефонов;
- Tip_Tel – таблица типов телефонов;
- Trud_dgvr – таблица трудовых договоров;
- Xar_BP– таблица характеристик бизнес-процессов.

Для возможности расширения или изменения некоторых функций интеллектуальной системы предусмотрены сообщения программисту, который будет заниматься сопровождением данной СППР. В программном коде содержатся комментарии, поясняющие, какие действия выполняются при обращении к той или иной процедуре или функции.

3.2 Руководство пользователя

Разработанное программное средство предназначено для поддержки принятия решений. Система поддержки принятия решений позволяет проводить выбор оптимального бизнес-процесса, повышая тем самым эффективность и обоснованность принимаемых решений.

Программное средство должно работать под управлением операционной системы MS Windows. Необходимо наличие процессора AMD A6 с тактовой частотой 1,7 ГГц и выше, объём оперативной памяти 1024 Мб и выше, свободное пространство на жёстком диске не менее 30 Мб плюс размер файла базы данных, принтер, клавиатура, мышь.

Так же, требуется наличие следующего программного обеспечения: MS SQL Server 17.

В проводнике Windows необходимо, при помощи указателя мыши перейти к той папке, в которой вы ранее сохраняли проект. Дважды щёлкните на папку, которую следует открыть.

Если в папке «Мой компьютер» или ее подпапках нет файла или папки, которые требуется открыть, найдите этот файл или папку, нажав кнопку Поиск. Чтобы запустить средство поиска, нажмите кнопку Пуск и выберите команду Найти.

Для начала работы с программой «Интеллектуальная система оценки бизнес-процессов аэропорта файл «Project1.exe».

Работа пользователя начинается с ввода логина и пароля, как показано на рисунке 24.

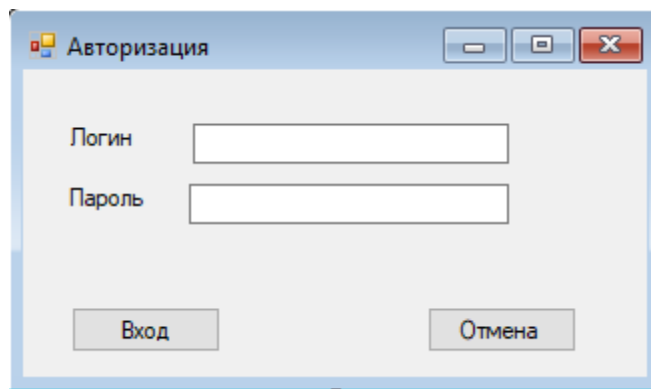


Рисунок 24– Окно авторизации

Нажимаем кнопку «Вход». Если введенные данные верны, открывается главное окно программы – рисунок 25.

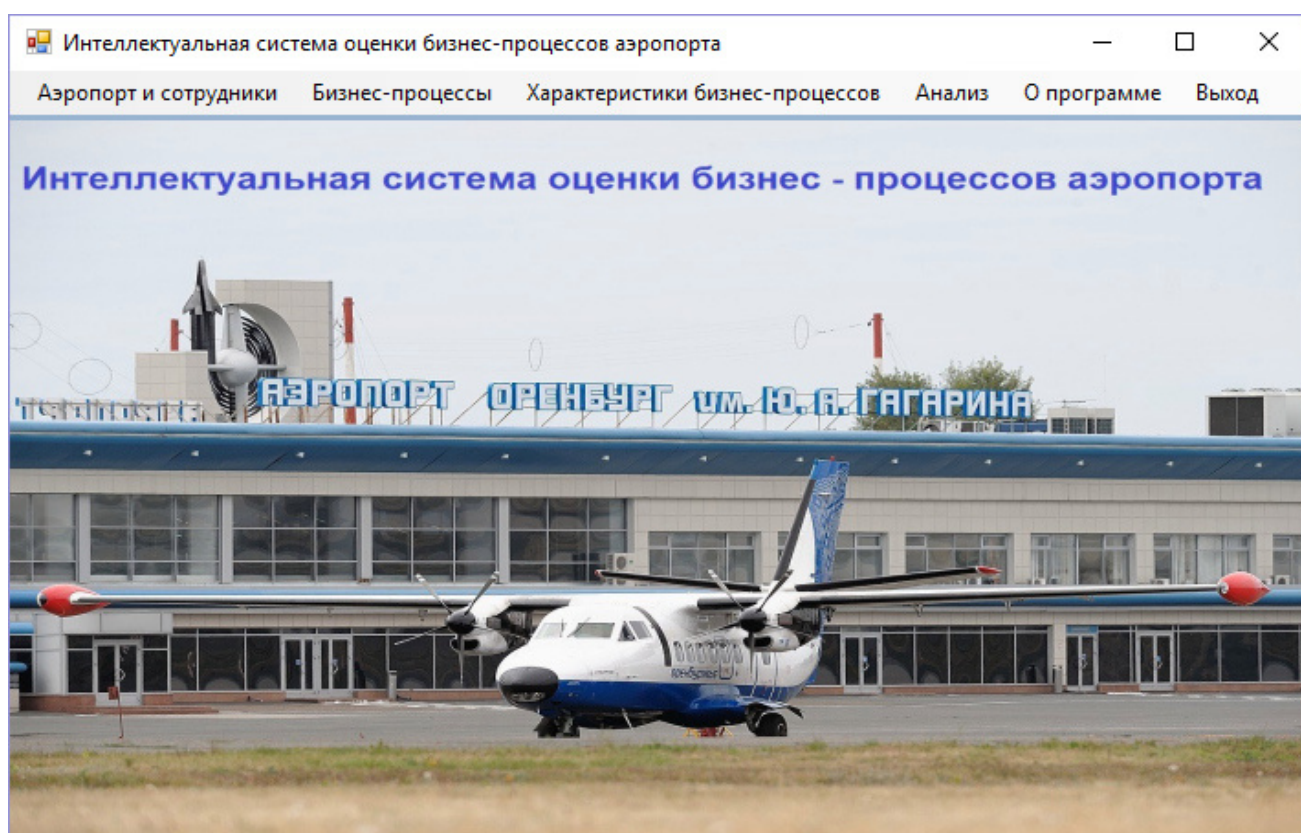


Рисунок 25 – Главное окно программы

Главное меню программы состоит из следующих пунктов:

- 1) Аэропорт и сотрудники
- 2) Бизнес-процессы
- 3) Характеристики бизнес-процессов
- 4) Анализ
- 5) О программе
- 6) Выход.

При выборе пункта меню «Аэропорт и сотрудники» предоставляется возможность просмотреть информацию об аэропорте и сотрудниках на рисунке 26.

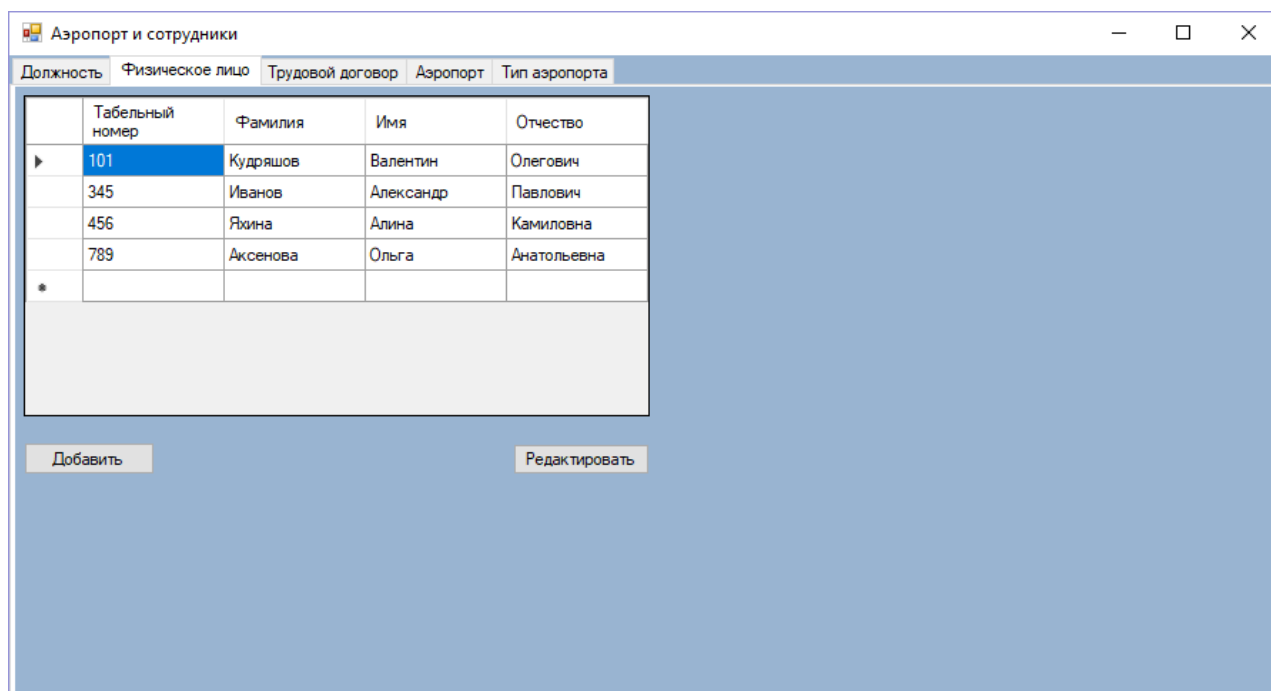


Рисунок 26 – Вкладка «Аэропорт и сотрудники»

Также имеется возможность добавления и редактирования данных в таблицах (рисунки 27-28).

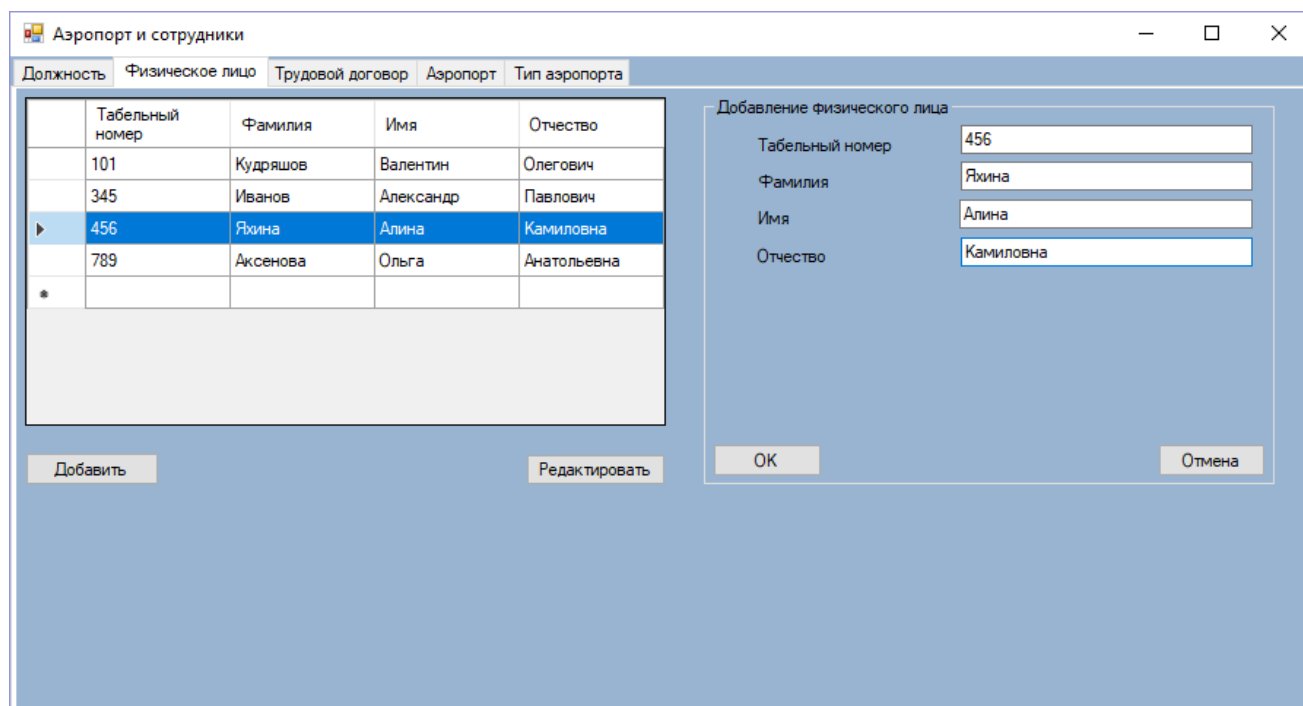


Рисунок 27 – Добавление записи

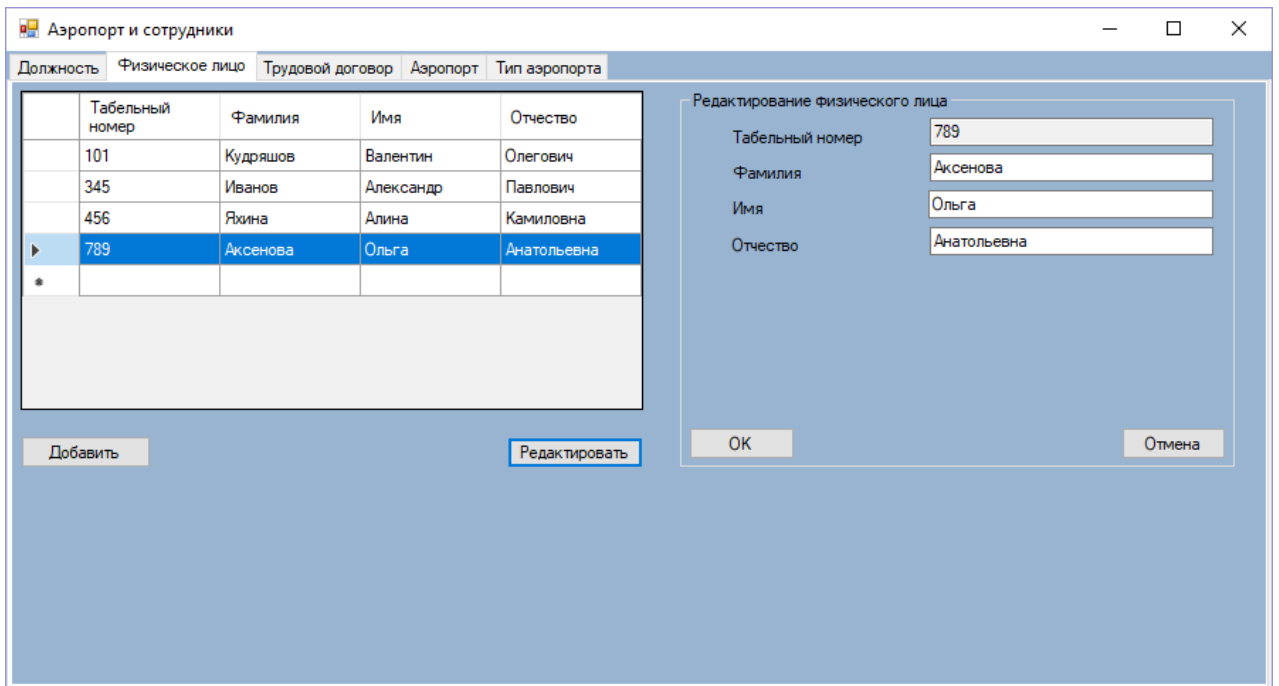


Рисунок 28 – Редактирование записи

На рисунке 29 представлена вкладка с информацией о бизнес – процессах, с возможностью их добавления и редактирования.

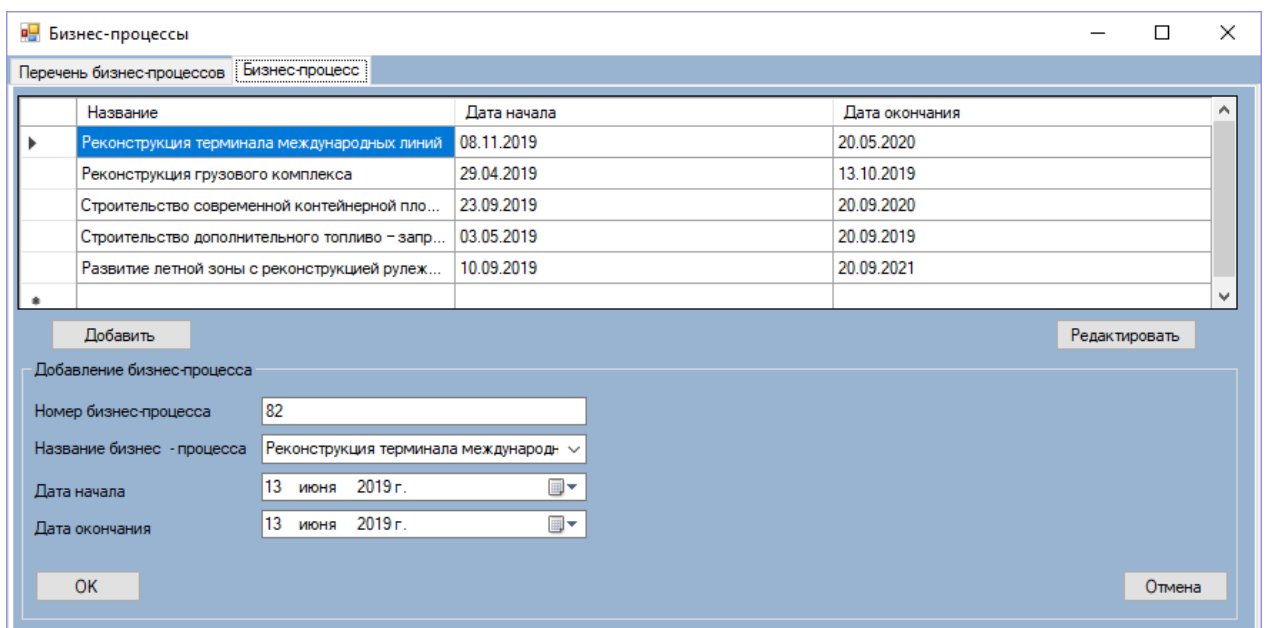


Рисунок 29– Вкладка «Бизнес - процессы»

На рисунке 30 представлена вкладка с информацией о характеристиках бизнес – процессов.

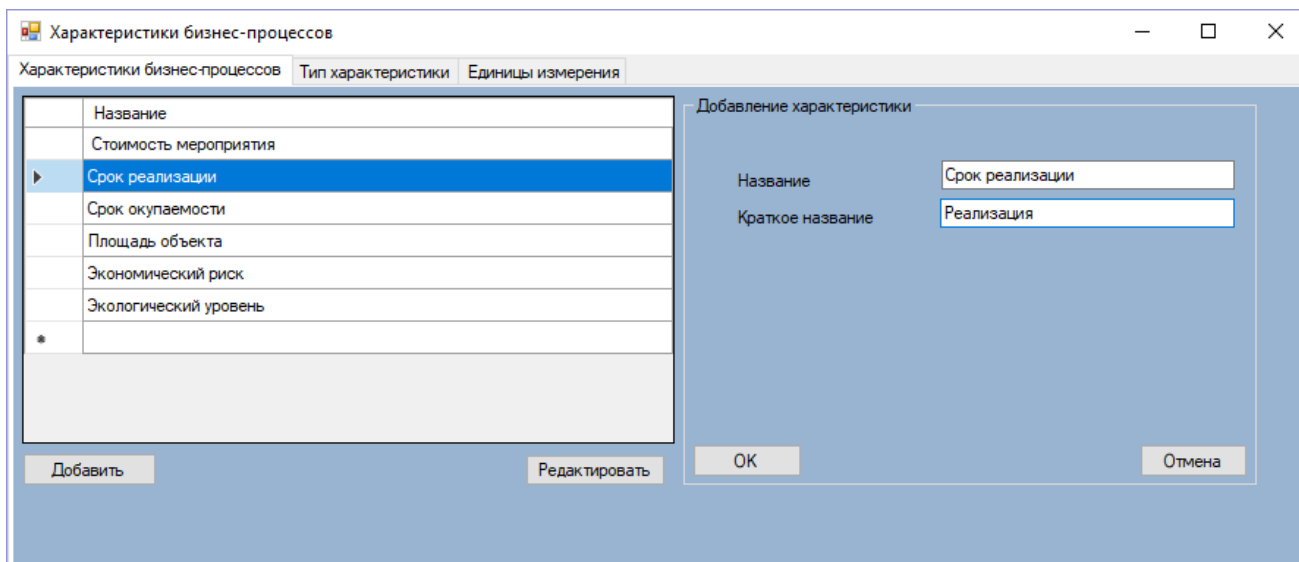


Рисунок 30 – Вкладка «Характеристики бизнес-процесса»

Вкладка «Анализ» представлена на рисунке 31. Предлагается выбор бизнес-процессов и критериев, по которым производится анализ. Форма содержит вкладки: «Бизнес - процессы», «Критерии», «Анализ» и «Отчет».

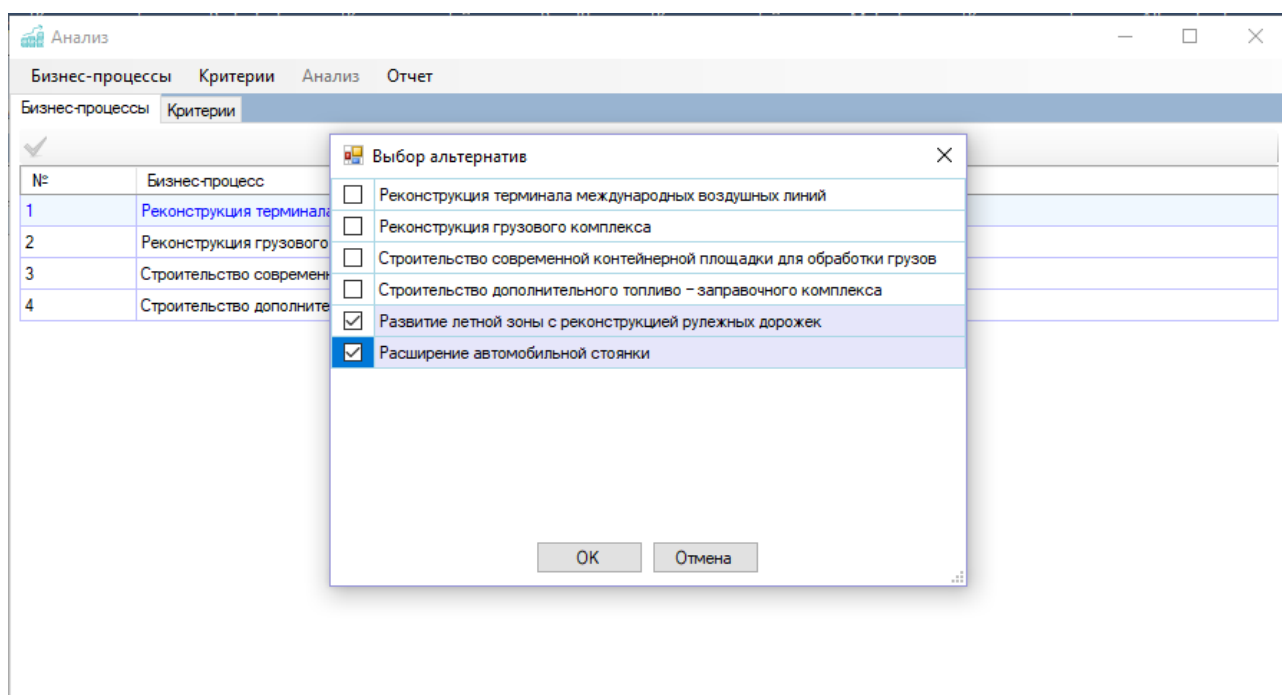


Рисунок 31 – Вкладка «Анализ»

На рисунке 32 представлена возможность выбора альтернатив и критериев, на основе которого будет производиться многокритериальный выбор.

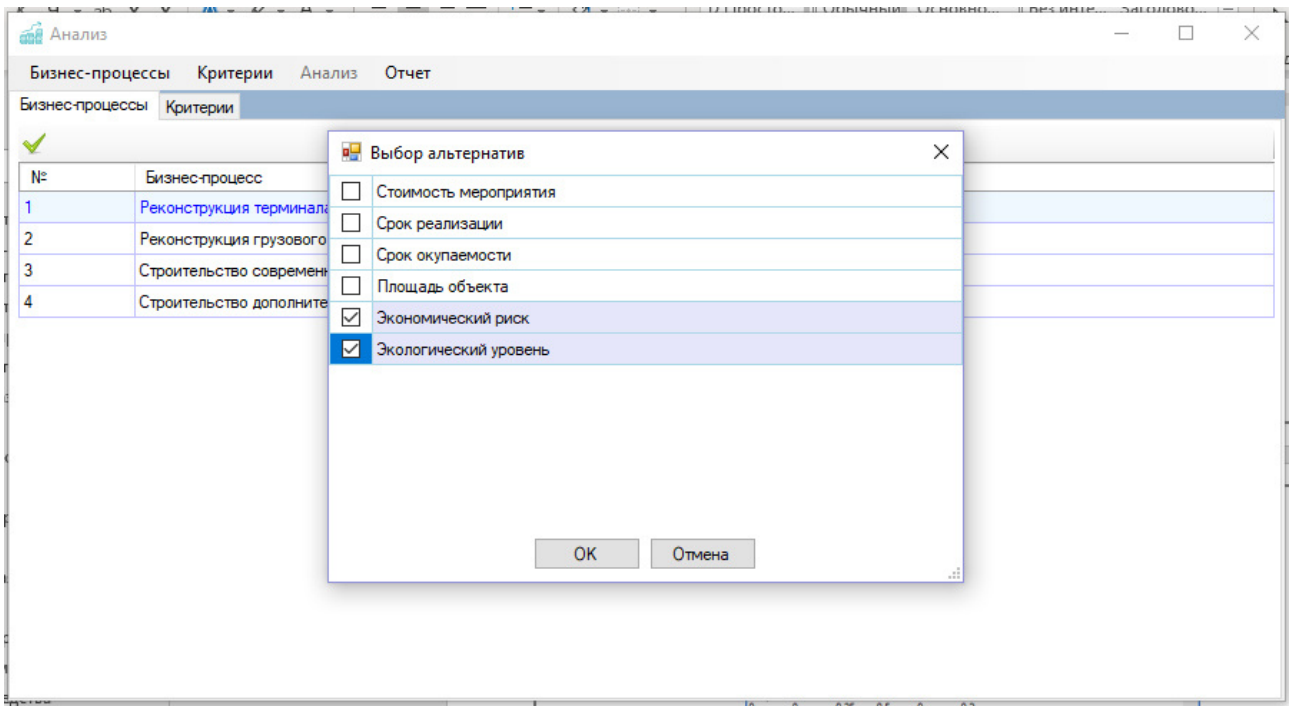


Рисунок 32 – Выбор альтернатив и критериев.

Результаты анализа представлены на рисунке 33.

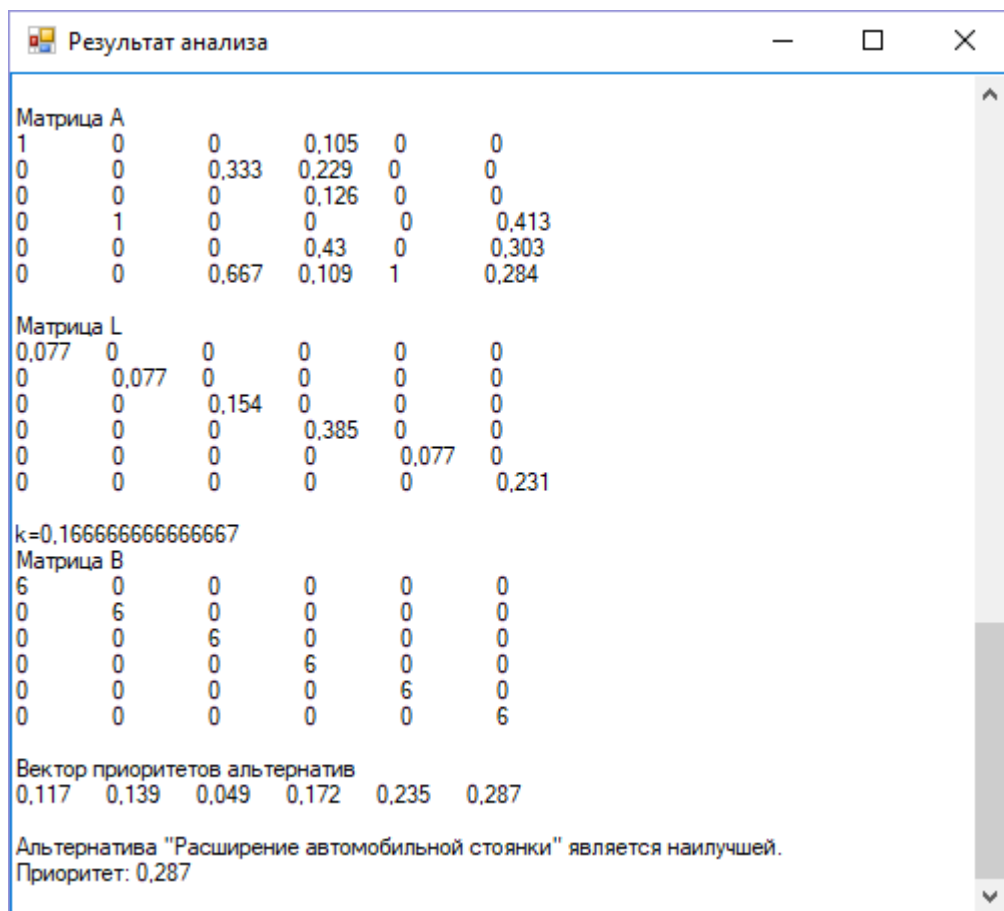


Рисунок 33 – Результаты анализа

Также имеется возможность генерации отчета с результатами оценки бизнес-процессов в формате .docx на рисунке 34.

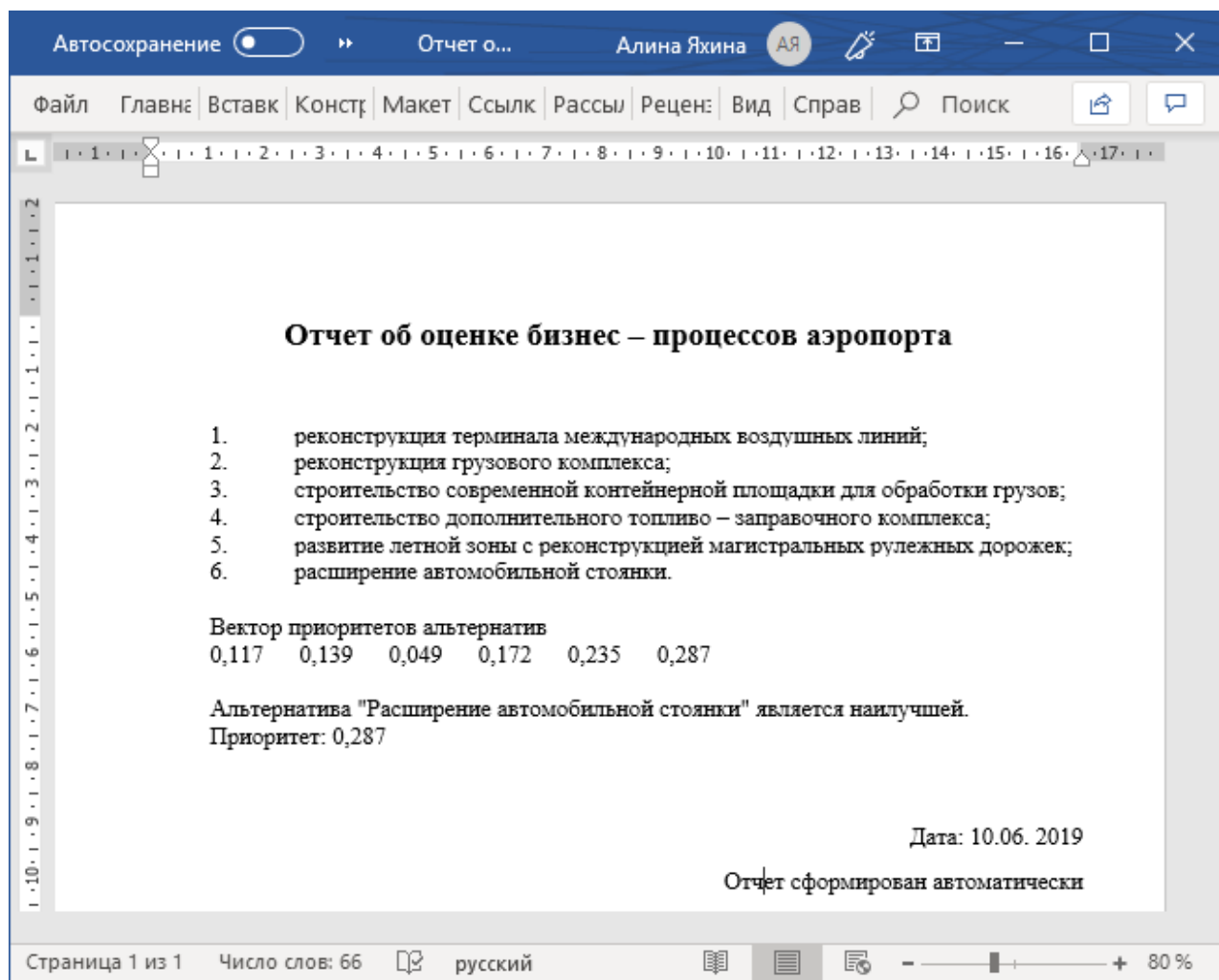


Рисунок 34 – Отчет о результатах оценки бизнес -процессов аэропорта

В случае выхода программа запросит подтверждение (рисунок 35).

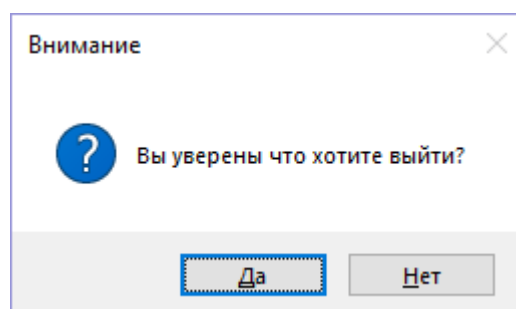


Рисунок 35 – Окно выхода

3.3 Экономическое обоснование

Для оценки экономической эффективности воспользуемся методом инвестиционного анализа. Этот метод позволяет оценить экономические параметры внедрения информационных систем по аналогии с оценкой любого другого инвестиционного проекта [16].

Таблица 25 – График выполнения работ по разработке Интеллектуальной системы оценки бизнес-процессов аэропорта

№ п/п	Наименование работ	Исполнитель	Длительность работы в апреле-мае 2019 года	
			в днях	в часах
1	Разработка технического задания	Яхина А.К.	2	16
2	Планирование АИС	Яхина А.К.	3	16
3	Рабочее проектирование АИС	Яхина А.К.	10	80
4	Отладка и тестирование АИС	Яхина А.К.	4	32
5	Обобщение и оценка результатов	Яхина А.К.	3	24
	Итого		32	168

На разработку «Интеллектуальной системы оценки бизнес-процессов аэропорта» в апреле-мае было затрачено 32 дня либо 168 человеко-часов.

При расчете стоимости (составлении сметы затрат) разработки интеллектуальной системы учитываются следующие виды расходов:

- стоимость материалов и покупных изделий;
- основная заработная плата;
- дополнительная заработная плата;
- страховые взносы;
- накладные расходы;
- затраты на машинное время (затраты на электроэнергию).

Перечень затрат на материалы и покупные изделия приведен в таблице 26.

Таблица 26 – Затраты на материалы и покупные изделия

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу, руб.	Стоимость, руб.
1	2	3	4	5	6
1	Специальная литература	шт	1	200	200
2	Бумага формата А4	шт	2	300	600

Продолжение таблицы 26

1	2	3	4	5	6
3	Ручки	шт	3	10	30
4	Папка	шт	1	20	20
5	USB-флеш-накопитель	шт	1	599	599
6	Итого				1449
7	Расход на транспорт (10% от стоимости)				144,9
8	Итого				1593,9

Транспортные расходы учитываются в объеме 10% от суммы затрат на материалы и покупные изделия, что составляет 145 руб. Таким образом, затраты на материалы и покупные изделия равны:

$$Z_{\text{м}} = \frac{1449 + 1449 \cdot 10}{100} = 1593,9 \text{ руб} \quad (7)$$

Где:

$Z_{\text{м}}$ – затрата на материал;

Результаты расчета фонда заработной платы представлены в таблице 27.

Таблица 27 – Расчет фонда заработной платы

№ п/п	Должность: программист	Количество рабочих дней в апрель-май 2019 г.	Количество проработанных дней в апрель-май 2019 г.	Размер дневной оплаты, руб.	Зарплата в месяц, руб. (апрель-май 2019 г.)
1	Основная зарплата	32	32	1500	48000
2	Доп. зарплата (10%)				4800
3	Итого				52800

В статью «Дополнительная заработная плата» входят выплаты, предусмотренные трудовым договором с руководством на разработку АИС. Размер дополнительной заработной платы разработчика определяется в размере 10 процентов от основной заработной платы.

$$Z_{\text{доп}} = Z_{\text{осн}} \cdot \frac{10}{100} = 48000 \cdot \frac{10}{100} = 4800 \text{ руб} \quad (8)$$

Где:

$Z_{\text{доп}}$ – зарплата дополнительная;

$Z_{\text{осн}}$ – зарплата основная.

Следовательно, разработчику в мае 2019 года всего начислено:

$$Z_{\text{нач}} = Z_{\text{осн}} + Z_{\text{доп}} = 48000 + 4800 = 52800 \text{ руб} \quad (9)$$

Где: $Z_{\text{нач}}$ – зарплата начисленная.

Таким образом, фонд заработной платы разработчика в мае 2019 года составляет 52800 руб.

К отчислениям на социальные нужды относят страховые взносы в ПФР, ФСС, ФФОМС и взносы на страхование от несчастных случаев на производстве и профзаболеваний.

Страховые взносы рассчитываются в размере 32,2 процентов от фонда заработной платы, что составит:

$$СВ = Z_{\text{нач}} * \frac{32,3}{100} = 52800 * \frac{32,3}{100} = 17001,4 \text{ руб} \quad (10)$$

Где: СВ -Страховые взносы.

Отчисления в пенсионный фонд составляют 24 процента от фонда заработной платы и равны:

$$Z_{\text{пф}} = (Z_{\text{осн}} + Z_{\text{доп}}) * \frac{24}{100} = 52800 * \frac{24}{100} = 12672 \text{ руб} \quad (11)$$

Где: $Z_{\text{пф}}$ – затрата пенсионного фонда.

Отчисления в фонд обязательного медицинского страхования равны:

$$Z_{\text{мс}} = (Z_{\text{осн}} + Z_{\text{доп}}) * \frac{5,1}{100} = 52800 * \frac{5,1}{100} = 2692 \text{ руб} \quad (12)$$

Где: $Z_{\text{мс}}$ – затрата медицинского страхования.

Отчисления на социальное страхование равны:

$$Z_{\text{сс}} = (Z_{\text{осн}} + Z_{\text{доп}}) * \frac{2,9}{100} = 52800 * \frac{2,9}{100} = 1531,2 \text{ руб} \quad (13)$$

Где: $Z_{\text{сс}}$ – затрата медицинского страхования.

Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний равны:

$$Z_{\text{нс}} = (Z_{\text{осн}} + Z_{\text{доп}}) * \frac{0,2}{100} = 52800 * \frac{0,2}{100} = 105,8 \text{ руб} \quad (14)$$

Где: $Z_{\text{нс}}$ – затрата несчастные случаи.

Численные значения отчислений на социальные нужды (страховые взносы), представлены в таблице 31.

Таблица 28 – Расчет отчислений на социальные нужды (страховые взносы)

№ п/п	Отчисления на социальные нужды (страховые взносы)	Тарифы страховых взносов, в %	Суммы страховых взносов (руб.)
1	Отчисления в ПФР	24	12672
2	Отчисления в ФОМС	5,1	2692,8
3	Отчисления в ФСС	2,9	1531,2
4	Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	0,2	105,6
5	Всего		17001,6

Размеры тарифов страховых взносов устанавливаются Федеральными законами. На момент разработки проекта необходимо руководствоваться действующим законодательством.

Накладные расходы, косвенные затраты – расходы, затраты, сопровождающие, сопутствующие основному производству, но не связанные с ним напрямую, не входящие в стоимость труда и материалов - дополнительные к основным затратам расходы для обеспечения процессов производства и обращения.

Накладные расходы фирмы составляют 20 процентов (условно) от суммы основной и дополнительной заработной платы

$$Z_{\text{н}} = (Z_{\text{осн}} + Z_{\text{доп}}) * \frac{20}{100} = 52800 * \frac{20}{100} = 10560 \text{ руб} \quad (15)$$

Где: $Z_{\text{н}}$ – затрата накладная.

Как следует из данных таблицы 29, на разработку и последующую отладку АИС потребовалось 32 рабочих дня (Дн).

В среднем с учетом перерывов программист работает за компьютером 6 часов в день. Себестоимость одного кВт/ч электроэнергии (С1кВт/ч) для организаций составляет 2 рубля 99 копеек. При проведении расчетов в проекте необходимо в расчеты брать существующие на дату расчета тарифы. Суммарная мощность энергопотребителей для АРМ программиста складывается из мощности, потребляемой системным блоком персонального компьютера, монитором, принтером и другим периферийным оборудованием, которая составляет 1,2 кВт. Следовательно, за 6 часов работы программиста суммарное энергопотребление за день составит:

$$P = 1,2 * 6 = 7,2 \text{ кВт/ч} \quad (16)$$

Где: P – энергопотребление.

Таким образом, стоимость машинного времени $Z_{\text{маш}}$, необходимого для разработки АИС, составит:

$$Z_{\text{маш}} = P * \text{Дн} * \frac{\text{кВт}}{\text{ч}} = 7,2 \frac{\text{кВт}}{\text{ч}} * 32 * 2,99 \text{ руб.} \frac{\text{кВт}}{\text{ч}} = 688 \text{ руб} \quad (17)$$

Затраты на машинное время учитываются как затраты на электроэнергию.

В результате выше произведенных расчетов мы получили итоговые затраты на разработку АИС (Таблица 29).

Таблица 29 – Итоговая смета затрат

№ п/п	Наименование статей расходов	Сумма, руб.
1	Стоимость материалов и покупных изделий	1593,9
2	Основная заработная плата	48000
3	Дополнительная заработная плата	4800
4	Отчисления на социальные нужды (32,2% от п.2 и п.3)	17001,6
5	Накладные расходы (20% от п.2 и п.3)	10560
6	Затраты на машинное время (затраты на электроэнергию)	688,896
7	Итого	82644,396

Цена программного продукта (Ц) определяется итоговыми затратами и прибылью, которая, в свою очередь, составляет 30 процентов (условно) от фонда заработной платы:

$$Ц = 82644,396 + 52800 * \frac{30}{100} = 98484 \text{ руб} \quad (18)$$

Расчет экономической эффективности по этому методу проведен на основе данных о количестве принятых и обработанных заказов на программное обеспечение до и после размещения «Интеллектуальной системы оценки бизнес-процессов аэропорта». Усредненные данные о реализации программного обеспечения приведены в таблице 30.

Из приведенных данных можно сделать выводы о том, что дополнительные доходы ($D_{\text{доп}}$) от реализации программного средства составили:

$$D_{\text{доп}} = (4 - 1) * 98484 = 295453 \text{ руб} \quad (19)$$

Расходы на разработку АИС (цена программного продукта) составили 98484 рублей, поэтому экономическая эффективность Э (П) от размещения «Интеллектуальной системы оценки бизнес-процессов аэропорта» составляет:

$$Э(П) = D_{\text{доп}} - Ц = 294453 - 98484 = 196968 \text{ руб} \quad (20)$$

Где: $D_{\text{доп}}$ – дополнительные доходы;

Ц - цена программного продукта.

Таким образом, экономический эффект Э в 3 раза превышает затраты на разработку АИС.

Рентабельность проекта (Р) составила:

$$P = \frac{\Pi}{Ц} * 100 = \frac{196968}{98484} = 300\%$$

Срок окупаемости (Т_{ок}) проекта равен:

$$T_{ок} = \frac{Ц}{\Pi} = \frac{98484}{196968} = 182 \text{ день} \quad (21)$$

Следовательно, разработка «Интеллектуальной системы оценки бизнес-процессов аэропорта» была экономически оправдана. Основные технико-экономические показатели приведены в таблице 30.

Таблица 30– Основные технико-экономические показатели проекта

Наименование работ	Основные характеристики	Единицы измерения	Проект
Разработка технического задания Планирование Рабочее проектирование Отладка и тестирование Обобщение и оценка результатов	Трудоемкость выполняемых работ	чел./час.	208
	Себестоимость проекта	руб.	
	Цена программного продукта	руб.	98484,396
	Дополнительные доходы	руб.	295453,188
	Экономическая эффективность (прибыль)	руб.	196698,792
	Экономический эффект	раз	3,0
	Рентабельность	(%)	300
	Срок окупаемости проекта	дни	182

Приведенные данные говорят о заметном экономическом эффекте от размещения «Интеллектуальной системы оценки бизнес-процессов аэропорта» на предприятии ГУП Оренбургской области «Аэропорт Оренбург», превышающем затраты на разработку и внедрение системы.

Заключение

В данной работе был проведён анализ предметной области и обозначены информационные потоки, подлежащие автоматизации, кроме того, были рассмотрены существующие аналоги, определены функции программного средства, обоснован выбор и изложены теоретические основы математического аппарата. В результате комплекса проведённых работ была формализована постановка задачи в форме технического задания.

Также была разработана модульная архитектура программного средства, осуществлена реализация функционального назначения приложения и разработан алгоритм, дан обзор и произведено обоснование выбора инструментальных сред программирования и создания базы данных. Были описаны уровни доступа пользователей, а так же, построена ER-диаграмма, даталогическая модель, произведено физическое проектирование таблиц, было проведено тестирование программного средства и сделан вывод о том, что система работает надёжно. Разработаны руководства по установке ПС и руководство пользователю.

Проведен расчет экономической эффективности данной интеллектуальной системы. Результаты оказались удовлетворительными, следовательно, разработка интеллектуальной системы экономически оправдана.

Таким образом, цель дипломной работы достигнута. Результатом дипломной работы является интеллектуальная система оценки бизнес-процессов аэропорта.

Список использованных источников

- 1 Официальный сайт ОАО «Международный аэропорт Оренбург» [Электронный ресурс] –Режим доступа: <http://www.orenairport.ru> - 03.06.2019.
- 2 Современный аэропорт: преодоление сложностей в эпоху растущих ожиданий [Электронный ресурс] –Режим доступа: <http://raedium.aero/articles/26-sovremennyj-aeroport-preodolenie-slozhnostej-v-epokhu-rastushchikh> - 03.06.2019
- 3 Волкова, Т.В. Проектирование компонентов автоматизированных систем в примерах [Электронный ресурс] : учебное пособие для обучающихся по образовательным программам высшего образования по направлениям подготовки 09.03.01 Информатика и вычислительная техника и 09.03.04 Программная инженерия / Т. В. Волкова, Е. Н. Чернопрудова; М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования "Оренбург. гос. ун-т", Каф. прогр. обеспечения вычисл. техники и автоматизир. систем. - Оренбург: ОГУ. - 2017. - 177 с.
- 4 Похилько А. Ф. CASE-технология моделирования процессов с использованием средств BPWin и ERWin. уч. пособие / А. Ф. Похилько, И. В. Горбачев. – Ульяновск: УлГТУ, 2008. – 120 с. – ISBN 978-5-9795-0000-0.
- 5 Ричард Баркер, CASE Method – Моделирование взаимосвязей между сущностями. [Электронный ресурс] – Режим доступа: <https://ftp.opennet.ru/pub/docs/case/casemeth> - 21.05.2019.
- 6 Сайт Консультационной группы "Воронов и Максимов" [Электронный ресурс] – Режим доступа: <http://www.vmgroupp.ru/soft/2-mp-tr> - 11.10.2018.
- 7 Инвестиционный консалтинг и решения для развития бизнеса [Электронный ресурс] – Режим доступа: <https://www.expert-systems.com/financial/pe/> - 11.10.2018.
- 8 Андрейчиков, А.В. Анализ, синтез, планирование решений в экономике ОАО «Финансы и статистика» [Электронный ресурс] / А. В. Андрейчиков, О. Н. Андрейчикова; – Научное сообщество студентов XXI столетия. ЭКОНОМИЧЕСКИЕ НАУКИ: сб. ст. по мат. LXI междунар. студ. науч.-практ. конф. № 1. – Режим доступа: <https://sibac.info/archive/economy/1> - 25.05.2019.
- 9 Сайт СУБД MS SQL Microsoft.com [Электронный ресурс] – Режим доступа: <https://www.microsoft.com> - 05.06.2019.
- 10 Сайт СУБД Oracle [Электронный ресурс] – Режим доступа: <https://www.oracle.com/ru/index.html> - 05.06.2019.
- 11 Сайт СУБД Visual FoxPro [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Visual_FoxPro - 05.06.2019.
- 12 Сайт среды разработки JetBrains [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/rider/> - 05.06.2019.
- 13 Сайт среды разработки Eclipse [Электронный ресурс] – Режим доступа: <https://www.eclipse.org/> (дата визита - 05.06.2019).

15 ГОСТ 34.003–90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. [Электронный ресурс] – Режим доступа: <http://gostrf.com/normadata/1/4294824/4294824585.pdf> - 01.06.2019

16 Волкова, Т.В. Проектирование и создание БД / Т. В. Волкова; – М-во образования и науки РФ, Гос. образоват. учреждение высш. проф. образования «ОГУ». – Оренбург : ГОУ ОГУ. – 2006. – 140 с.

17 Щелоков, С.А. Экономико-правовые основы рынка программного обеспечения [Электронный ресурс]: учебно-методическое пособие для обучающихся по образовательным программам высшего образования по направлениям подготовки: 09.03.01 Информатика и вычислительная техника; 09.03.04 Программная инженерия / С. А. Щелоков, И. М. Соколова; М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования «Оренбург. гос. ун-т», Каф. прогр. обеспечения вычисл. техники и автоматизир. систем – Оренбург: ОГУ, 2017. – 316 с

18 Соловьев, И. В. Проектирование информационных систем. Фундаментальный курс : учеб. пособие для вузов / И. В. Соловьев, А. А. Майоров ; Моск. гос. ун-т геодезии и картографии. – Москва : Акад. проект, 2015. – 399 с.

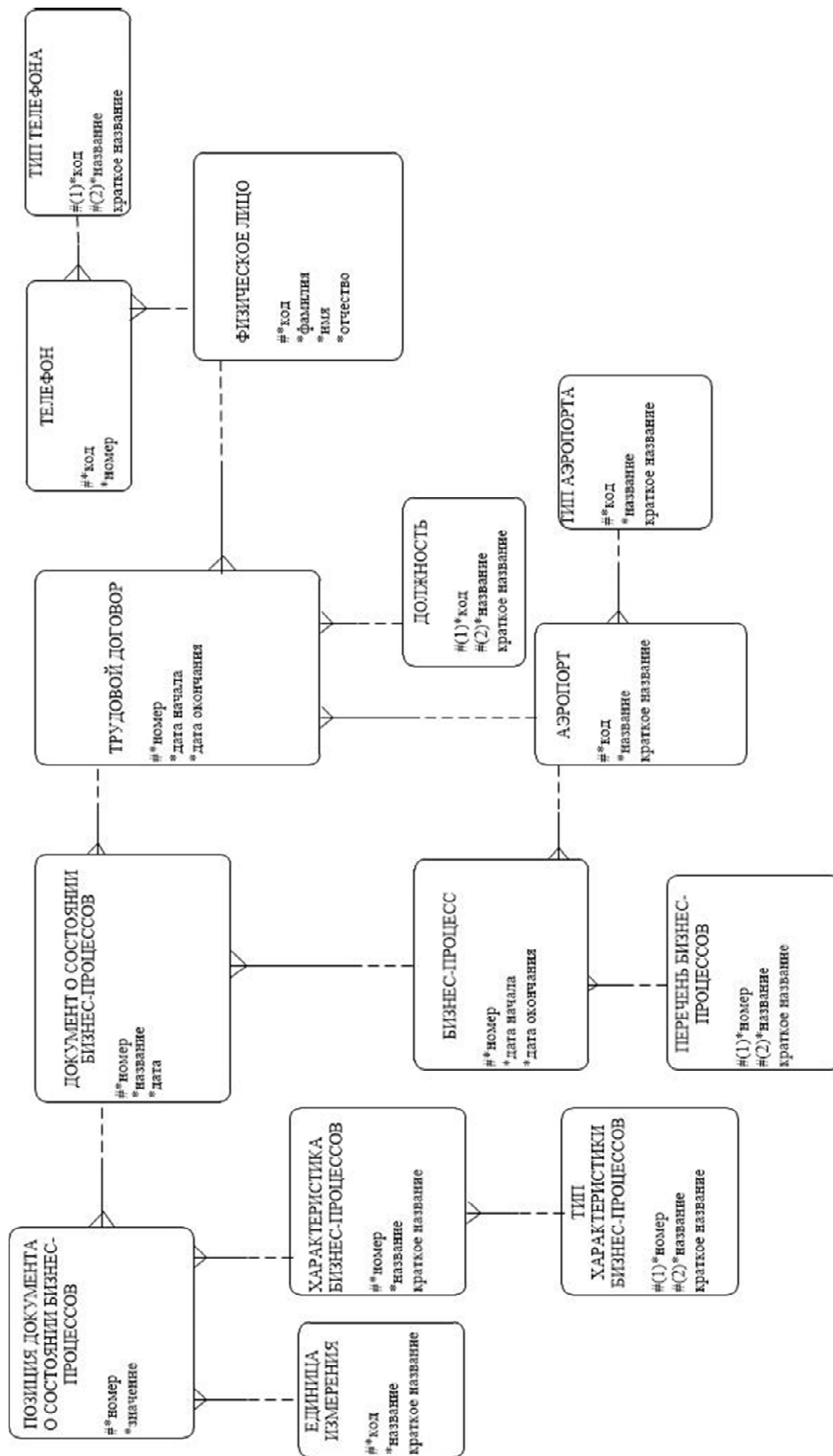
19 Соловьев, Н. А. Основы теории принятия решений для программистов: Уч. пособие: /Н. А. Соловьев, Е.Н. Чернопрудова, Д.А. Лесовой; под ред. Н.А. Соловьев. – Оренбург: ООО ИПК «Университет», 2012. С. 153. ISBN 978-5-4417-0092-4.

20 ГОСТ 34.003–90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. [Электронный ресурс] – Режим доступа: <http://gostrf.com/normadata/1/4294824/4294824585.pdf> - 01.06.2019

21 Волкова, Т. В. Выпускная квалификационная работа: Методические рекомендации /Т. В. Волкова, Н.А. Соловьев, Л.А. Юркевская – Оренбург: ОГУ, 2015 г. - 105 с.

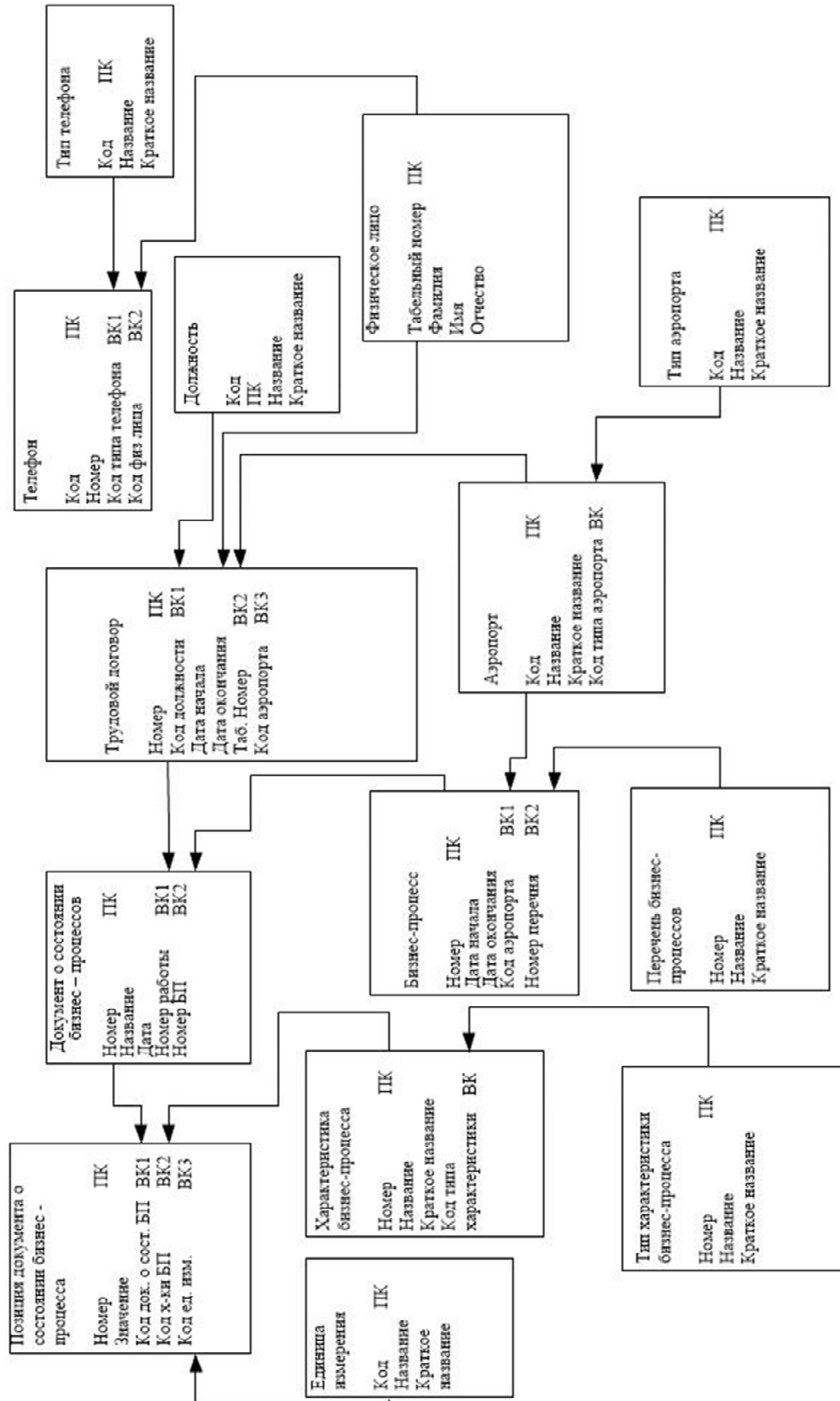
Приложение А (Обязательное)

ER – диаграмма



Приложение Б (Обязательное)

Даталогическая модель



Приложение В (Справочное)

SQL - скрипты

SQL-скрипт создания таблицы «Аэропорт»:

```
CREATE TABLE [dbo].[Air](
    [Kod] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
    [Kod_TA] [int] NOT NULL CHECK (Kod_TA>0),
    [Nazv] [nvarchar](50) NOT NULL UNIQUE,
    [Kr_nazv] [nchar](10) NOT NULL,
    CONSTRAINT [PK_Air] PRIMARY KEY CLUSTERED
(
    [Kod] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Air] WITH CHECK ADD CONSTRAINT
[FK_Air_Tip Air] FOREIGN KEY([Kod_TA])
REFERENCES [dbo].[Tip Air] ([Kod])
GO
ALTER TABLE [dbo].[Air] CHECK CONSTRAINT [FK_Air_Tip Air]
GO
```

SQL-скрипт создания таблицы «Тип аэропорта»:

```
CREATE TABLE [dbo].[Tip Air](
    [Kod] [int] NOT NULL CHECK (Kod>0) UNIQUE,
    [Nazv] [nvarchar](50) NOT NULL UNIQUE,
    [Kr_nazv] [nchar](10) NOT NULL,
    CONSTRAINT [PK_Tip Air] PRIMARY KEY CLUSTERED
([Kod] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

SQL-скрипт создания таблицы «Характеристика бизнес-процессов»:

```
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Xar_BP](
```

```

        [Nomer] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
        [Nazv] [nvarchar](50) NOT NULL UNIQUE,
        [Kr_Nazv] [nchar](10) NULL,
    CONSTRAINT [PK_Xar_BP] PRIMARY KEY CLUSTERED
    (
        [Nomer] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Бизнес - процесс»:

```

CREATE TABLE [dbo].[BP](
    [Nomer] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
    [Date_n] [date] NOT NULL UNIQUE,
    [Date_o] [date] NOT NULL UNIQUE,
    [Kod_air] [int] NOT NULL CHECK (Kod_air>0),
    [Nazv] [nvarchar](50) NOT NULL UNIQUE,
    [Kr_nazv] [nchar](10) NOT NULL,
    CONSTRAINT [PK_BP] PRIMARY KEY CLUSTERED
    (
        [Nomer] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IG-
    NORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
    = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[BP] WITH CHECK ADD CONSTRAINT
[FK_BP_Air] FOREIGN KEY([Kod_air])
REFERENCES [dbo].[Air] ([Kod])
GO
ALTER TABLE [dbo].[BP] CHECK CONSTRAINT [FK_BP_Air]
GO

```

SQL-скрипт создания таблицы «Позиция документа о состоянии бизнес-процессов»:

```

CREATE TABLE [dbo].[Poz_doc](
    [Nomer] [int] NOT NULL CHECK (Nomer>0)UNIQUE,
    [Znach] [int] NOT NULL CHECK (Znach>0) UNIQUE,
    [Kod_doc_sost] [int] NOT NULL CHECK (Kod_doc_sost>0),
    [Kod_xar_bp] [int] NOT NULL CHECK (Kod_xar_bp>0),
    [Kod_ei] [int] NOT NULL CHECK (Kod_ei>0),
    CONSTRAINT [PK_Poz_doc_1] PRIMARY KEY CLUSTERED

```

```

(
    [Nomer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Poz_doc] WITH CHECK ADD CONSTRAINT
[FK_Poz_doc_Doc_sost] FOREIGN KEY([Kod_doc_sost])
REFERENCES [dbo].[Doc_sost] ([Nomer])
GO
ALTER TABLE [dbo].[Poz_doc] CHECK CONSTRAINT
[FK_Poz_doc_Doc_sost]
GO
ALTER TABLE [dbo].[Poz_doc] WITH CHECK ADD CONSTRAINT
[FK_Poz_doc_EI] FOREIGN KEY([Kod_ei])
REFERENCES [dbo].[EI] ([Kod])
GO
ALTER TABLE [dbo].[Poz_doc] CHECK CONSTRAINT [FK_Poz_doc_EI]
GO
ALTER TABLE [dbo].[Poz_doc] WITH CHECK ADD CONSTRAINT
[FK_Poz_doc_Xar_BP] FOREIGN KEY([Kod_xar_bp])
REFERENCES [dbo].[Xar_BP] ([Nomer])
GO
ALTER TABLE [dbo].[Poz_doc] CHECK CONSTRAINT
[FK_Poz_doc_Xar_BP]
GO

```

SQL-скрипт создания таблицы «Единица измерения»:

```

CREATE TABLE [dbo].[EI](
    [Kod] [int] NOT NULL CHECK (Kod>0) UNIQUE,
    [Nazv] [nvarchar](50) NOT NULL UNIQUE,
    [Kr_nazv] [nchar](10) NULL,
    CONSTRAINT [PK_EI] PRIMARY KEY CLUSTERED
(
    [Kod] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IG-
NORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Физическое лицо»:

```

CREATE TABLE [dbo].[Fiz_lit](
    [Tab_nom] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
    [Familia] [nvarchar](50) NOT NULL UNIQUE,
    [Imya] [nvarchar](50) NOT NULL UNIQUE,
    [Otch] [nvarchar](50) NOT NULL UNIQUE,
    CONSTRAINT [PK_Fiz_lit] PRIMARY KEY CLUSTERED
(
    [Tab_nom] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Трудовой договор»:

```

CREATE TABLE [dbo].[Rabota](
    [Nomer] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
    [Tab_nom] [int] NOT NULL,
    [Date_n] [date] NOT NULL UNIQUE,
    [Date_o] [date] NOT NULL UNIQUE,
    [Kod_air] [int] NOT NULL,
    CONSTRAINT [PK_Rabota] PRIMARY KEY CLUSTERED
(
    [Nomer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Rabota] WITH CHECK ADD CONSTRAINT
[FK_Rabota_Fiz_lit] FOREIGN KEY([Tab_nom])
REFERENCES [dbo].[Fiz_lit] ([Tab_nom])
GO
ALTER TABLE [dbo].[Rabota] CHECK CONSTRAINT [FK_Rabota_Fiz_lit]
GO

```

SQL-скрипт создания таблицы «Документ о состоянии бизнес-процессов»:

```

CREATE TABLE [dbo].[Doc_sost](
    [Nomer] [int] NOT NULL CHECK (Nomer>0) UNIQUE,
    [Nazv] [nvarchar](50) NOT NULL UNIQUE,
    [Date] [date] NOT NULL UNIQUE,
    [Nom_rab] [int] NOT NULL,
    [Nom_bp] [int] NOT NULL,

```

```

CONSTRAINT [PK_Doc_sost] PRIMARY KEY CLUSTERED
(
    [Nomer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IG-
NORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Doc_sost] WITH CHECK ADD CONSTRAINT
[FK_Doc_sost_BP] FOREIGN KEY([Nom_bp])
REFERENCES [dbo].[BP] ([Nomer])
GO
ALTER TABLE [dbo].[Doc_sost] CHECK CONSTRAINT
[FK_Doc_sost_BP]
GO
ALTER TABLE [dbo].[Doc_sost] WITH CHECK ADD CONSTRAINT
[FK_Doc_sost_Rabota] FOREIGN KEY([Nom_rab])
REFERENCES [dbo].[Rabota] ([Nomer])
GO
ALTER TABLE [dbo].[Doc_sost] CHECK CONSTRAINT
[FK_Doc_sost_Rabota]
GO

```

SQL-скрипт создания таблицы «Должность»:

```

CREATE TABLE [dbo].[Dolgnost](
    [Kod] [int] IDENTITY(1,1) NOT NULL,
    [Nazv] [varchar](70) NOT NULL,
    [Kr_nazv] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Dolgnost] PRIMARY KEY CLUSTERED
(
    [Kod] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Перечень бизнес-процессов»:

```

CREATE TABLE [dbo].[Per_BP](
    [Nomer] [int] IDENTITY(68,1) NOT NULL,
    [Nazv] [varchar](100) NOT NULL,
    [Kr_nazv] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Per_BP] PRIMARY KEY CLUSTERED
(

```

```

        [Nomer] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Тип характеристики бизнес-процессов»:

```

CREATE TABLE [dbo].[T_Xar](
    [Nomer] [int] NOT NULL,
    [Nazv] [varchar](50) NOT NULL,
    [Kr_nazv] [varchar](20) NOT NULL,
    CONSTRAINT [PK_T_Xar] PRIMARY KEY CLUSTERED
(
    [Nomer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

SQL-скрипт создания таблицы «Телефон»:

```

CREATE TABLE [dbo].[Telefon](
    [Kod] [int] IDENTITY(9081,1) NOT NULL,
    [Nomer_tlf] [varchar](15) NOT NULL,
    [Kod_TT] [int] NOT NULL,
    [Kod_FL] [int] NOT NULL,
    CONSTRAINT [PK_Telefon] PRIMARY KEY CLUSTERED
(
    [Kod] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```


Приложение Г (Обязательное)

Листинг программы

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

namespace Method
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            _matrix = new List<double[,]>();
        }

        private int _m;
        private int _n;
        private List<double[,]> _matrix;
        private double[,] _a;
        private double[,] _f;

        private void выбратьToolStripMenuItem_Click(object sender, EventArgs e)
        {
            var form = new AlternativeForm(@"\Struct\alternative.xml", true, null);
            if (form.ShowDialog() == DialogResult.OK)
            {
                int j = 0;
                for (int i = 0; i < form.grid.RowCount; i++)
                {
                    DataGridViewCheckBoxCell cell = (DataGridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
                    if ((bool)cell.EditedFormattedValue == true)
                    {
                        aGrid.Rows.Add();
                        aGrid[0, j].Value = j + 1;
                        aGrid[1, j++].Value = form.grid[1, i].Value;
                    }
                }
                _m = j;
                if (_n > 0 && _m > 0)
                {
                    _a = new double[_m, _n];
                    toolStripButton1.Enabled = true;
                }
            }
        }

        private void редактироватьToolStripMenuItem_Click(object sender, EventArgs e)
        {
            var form = new RedactorForm(@"\Struct\alternative.xml");
            form.ShowDialog();
        }

        private void редактироватьToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            var form = new RedactorForm(@"\Struct\kritery.xml") { Text = "Редактор критериев" };
            form.ShowDialog();
        }

        private void выбратьToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            var form = new AlternativeForm(@"\Struct\kritery.xml", true, null);
            if (form.ShowDialog() == DialogResult.OK)
            {
                int j = 0;
                for (int i = 0; i < form.grid.RowCount; i++)
                {
                    DataGridViewCheckBoxCell cell = (DataGridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
                    if ((bool)cell.EditedFormattedValue == true)
                    {
                        kGrid.Rows.Add();
                        kGrid[0, j].Value = j + 1;
                        kGrid[1, j++].Value = form.grid[1, i].Value;
                    }
                }
                _n = j;
                if (_m > 0 && _n > 0)
                {
                    _a = new double[_m, _n];
                    toolStripButton1.Enabled = true;
                }
            }
        }

        private void aGrid_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
        {
            var form = new AlternativeForm(@"\Struct\kritery.xml", false, kGrid) { Text = "Выбор критериев" };
            for (int i = 0; i < form.grid.Rows.Count; i++)
```

```

        {
            DataGridViewCheckBoxCell cell = (Data-
GridCheckBoxCell)form.grid.Rows[i].Cells[0];
            if (_a[e.RowIndex, i] > 0)
                cell.Value = true;
        }

        if (form.ShowDialog() == DialogResult.OK)
        {
            for (int i = 0; i < form.grid.Rows.Count; i++)
            {
                DataGridViewCheckBoxCell cell = (Data-
GridCheckBoxCell)form.grid.Rows[i].Cells[0];
                if ((bool)cell.EditedFormattedValue == true)
                    _a[e.RowIndex, i] = 1;
                else _a[e.RowIndex, i] = 0;
            }

            aGrid.Rows[e.RowIndex].DefaultCellStyle.Back-
Color = Color.Honeydew;
        }

        private void kGrid_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
        {
            var form = new MatrixForm();
            double m = Math.Round(Math.Sqrt(_ma-
trix[e.RowIndex].Length));

            for (int i = 0; i < m; i++)
                form.grid.Columns.Add("col" + i, "");
            for (int i = 0; i < m; i++)
                form.grid.Rows.Add();

            int ind = 0;
            for (int i = 0; i < aGrid.Rows.Count; i++)
            {
                if (_a[i, e.RowIndex] == 1)
                {
                    form.grid.Rows[ind].HeaderCell.Value = "A"
+ (i + 1);
                    form.grid.Columns[ind++].HeaderText = "A"
+ (i + 1);
                }
            }

            for (int i = 0; i < form.grid.Rows.Count; i++)
                for (int j = 0; j < form.grid.Columns.Count; j++)
                    form.grid[j, i].Value = _matrix[e.RowIndex][i,
j];

            if (form.ShowDialog() == DialogResult.OK)
            {
                for (int i = 0; i < form.grid.Rows.Count; i++)
                    for (int j = 0; j < form.grid.Columns.Count;
j++)
                        _matrix[e.RowIndex][i, j] = Convert.ToDou-
ble(form.grid[j, i].Value);

                kGrid.Rows[e.RowIndex].DefaultCellStyle.Back-
Color = Color.Honeydew;
            }
        }

        }

        private void toolStripButton2_Click(object sender,
EventArgs e)
        {
            var form = new MatrixForm();
            int m = kGrid.Rows.Count;

            for (int i = 0; i < m; i++)
                form.grid.Columns.Add("col" + i, "K" + (i + 1));
            for (int i = 0; i < m; i++)
            {
                form.grid.Rows.Add();
                form.grid.Rows[i].HeaderCell.Value = "K" + (i +
1);
                for (int j = 0; j < form.grid.Columns.Count; j++)
                    form.grid[j, i].Value = 1;
            }

            if (form.ShowDialog() == DialogResult.OK)
            {
                _f = new double[form.grid.Rows.Count,
form.grid.Columns.Count];
                for (int i = 0; i < form.grid.Rows.Count; i++)
                    for (int j = 0; j < form.grid.Columns.Count;
j++)
                        _f[i, j] = Convert.ToDouble(form.grid[j,
i].Value);
                toolStripButton3.Enabled = true;
                анализToolStripMenuItem.Enabled = true;
            }
        }

        private void toolStripButton3_Click(object sender,
EventArgs e)
        {
            try
            {
                StringBuilder sb = new StringBuilder();
                sb.AppendLine("Решение");
                int n = _matrix.Count;
                int N = 0;
                double[,] l = new double[n, n];

                for (int i = 0; i < n; i++)
                {
                    sb.AppendLine(string.Format("Матрица пред-
почтения альтернатив относительно критерия K{0}", i +
1));

                    var m = Math.Sqrt(_matrix[i].Length);
                    sb.AppendLine(write(_matrix[i], Con-
vert.ToInt32(m), Convert.ToInt32(m)));

                    var a = new double[Convert.ToInt32(m), 1];

                    for (int j = 0; j < m; j++)
                        a[j, 0] = 1.0 / n;

                    var mm = ProductMatrix(_matrix[i], a, Con-
vert.ToInt32(m), Convert.ToInt32(m), 1);
                }
            }
        }
    }
}

```

```

double c = 0.0;
for (int j = 0; j < m; j++)
    c += mm[j, 0];

sb.AppendLine(string.Format("c={0}", c));

for (int j = 0; j < m; j++)
    mm[j, 0] /= c;

sb.AppendLine(string.Format("Приоритет
альтернатив по критерию K{0}", i + 1));
sb.AppendLine(write(mm, Convert.ToInt32(m), 1));

int ind = 0;
for (int j = 0; j < aGrid.Rows.Count; j++)
    if (_a[j, i] > 0)
        _a[j, i] = mm[ind++, 0];

N += Convert.ToInt32(m);
}

var mF = Math.Sqrt(_f.Length);
var aF = new double[Convert.ToInt32(mF), 1];

for (int j = 0; j < mF; j++)
    aF[j, 0] = 1.0 / n;

_f = ProductMatrix(_f, aF, Convert.ToInt32(mF),
Convert.ToInt32(mF), 1);
double cF = 0.0;
for (int j = 0; j < mF; j++)
    cF += _f[j, 0];
for (int j = 0; j < mF; j++)
    _f[j, 0] /= cF;

sb.AppendLine("Приоритет альтернатив относительно
фокуса F");
sb.AppendLine(write(_f, Convert.ToInt32(mF),
1));

for (int i = 0; i < n; i++)
{
    var m = Math.Sqrt(_matrix[i].Length);
    l[i, i] = m / N;
}

sb.AppendLine("Матрица A");
sb.AppendLine(write(_a, aGrid.Rows.Count,
kGrid.Rows.Count));

sb.AppendLine("Матрица L");
sb.AppendLine(write(l, n, n));

var mK = ProductMatrix(_a, l,
aGrid.Rows.Count, kGrid.Rows.Count, n);
mK = ProductMatrix(mK, _f, aGrid.Rows.Count,
n, 1);

double k = 0.0;
for (int i = 0; i < aGrid.Rows.Count; i++)
    k += mK[i, 0];

sb.AppendLine(string.Format("k={0}", k));

double[,] b = new double[aGrid.Rows.Count,
aGrid.Rows.Count];
for (int i = 0; i < aGrid.Rows.Count; i++)
    b[i, i] = 1 / k;

sb.AppendLine("Матрица B");
sb.AppendLine(write(b, aGrid.Rows.Count,
aGrid.Rows.Count));
_n = j;
if (_m > 0 && _n > 0)
{
    _a = new double[_m, _n];
    toolStripButton1.Enabled = true;
}
}

private void aGrid_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
{
    var form = new AlternativeForm(@"Struct\kriteriy.xml", false, kGrid) { Text =
"Выбор критериев" };
    for (int i = 0; i < form.grid.Rows.Count; i++)
    {
        DataGridViewCheckBoxCell cell = (DataGridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
        if (_a[e.RowIndex, i] > 0)
            cell.Value = true;
    }

    if (form.ShowDialog() == DialogResult.OK)
    {
        for (int i = 0; i < form.grid.Rows.Count; i++)
        {
            DataGridViewCheckBoxCell cell = (DataGridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
            if ((bool)cell.EditedFormattedValue == true)
                _a[e.RowIndex, i] = 1;
            else _a[e.RowIndex, i] = 0;
        }

        var alfT = FloopMatrix(mK, aGrid.Rows.Count,
1);
        var priority = ProductMatrix(alfT, b, l,
aGrid.Rows.Count, aGrid.Rows.Count);

        sb.AppendLine("Вектор приоритетов альтернатив");
        sb.AppendLine(write(priority, 1,
aGrid.Rows.Count));

        n = 0;
        for (int i = 1; i < aGrid.Rows.Count; i++)
            if (priority[0, i] > priority[0, i - 1]) n = i;
    }
}

```

```

        sb.AppendLine(string.Format("Альтернатива
\"{0}\" является наилучшей.\r\nПриоритет: {1}",
aGrid.Rows[n].Cells[1].Value, Math.Round(priority[0, n],
3)));

        var form = new ResultForm();
        form.textBox1.Text = sb.ToString();
        form.Show();
    } catch (Exception ex)
    {
    }
    //toolStripButton2.Enabled = toolStripButton3.Enabled = анализToolStripMenuItem.Enabled = false;
}

//Произведение двух матриц
public double[,] ProductMatrix(double[,] a, double[,]
b, int amountRowA, int amountColumnA, int
amountColumnB)
{
    double[,] c = new double[amountRowA,
amountColumnB];

    double sum = 0;

    for (int k = 0; k < amountColumnB; k++)
        for (int i = 0; i < amountRowA; i++)
            {
                sum = 0;

                for (int j = 0; j < amountColumnA; j++)
                    sum += a[i, j] * b[j, k];

                c[i, k] = sum;
            }

    return c;
}

//Транспонирование матрицы
public double[,] FloopMatrix(double[,] matrix, int m,
int n)
{
    double[,] matrixT = new double[n, m];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            matrixT[i, j] = matrix[j, i];

    return matrixT;
}

private void toolStripButton1_Click_1(object sender,
EventArgs e)
{
    _matrix.Clear();

    for (int i = 0; i < kGrid.Rows.Count; i++)
    {
        int m = 0;
        for (int j = 0; j < aGrid.Rows.Count; j++)
            {
                if (_a[j, i] == 1) m++;
            }
    }
}

}

var matrix = new double[m, m];
for (int j = 0; j < m; j++)
    for (int k = 0; k < m; k++)
        matrix[j, k] = 1;

_matrix.Add(matrix);
toolStripButton2.Enabled = true;
toolStripButton1.Enabled = true;
}

}

public string write(double[,] matrix, int m, int n)
{
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
            {
                string x = Math.Round(matrix[i, j],
3).ToString();
                sb.Append(x);
                sb.Append(' ', (8 - x.Length) * 2);
            }
        sb.AppendLine();
    }

    return sb.ToString();
}

private void анализToolStripMenuItem_Click(object
sender, EventArgs e)
{
    toolStripButton3_Click(sender, e);
}
}

}

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

//Data Source=.;Initial Catalog=Air;Integrated Security=True

namespace atp_kra
{
    public partial class Form3 : Form
    {
        SqlConnection SqlConnection = new SqlConnection();
    }
}

```

```

bool n = false;
public Form3()
{
    InitializeComponent();
}

public void BaseAccess() //подключение
{
    SqlConnection.ConnectionString = "Data Source =.;
Initial Catalog = Air; Integrated Security = True";
    SqlConnection.Open();
}

private void Form3_Load(object sender, EventArgs e)
{
    данные в таблицу "airDataSet1.Dolgnost". При необходи-
мости она может быть перемещена или удалена.
    this.dolgnostTableAdapter.Fill(this.air-
DataSet1.Dolgnost);
    // TODO: данная строка кода позволяет загрузи-
ть данные в таблицу "airDataSet.View_Trud_dgvr".
При необходимости она может быть перемещена или
удалена.
    this.view_Trud_dgvrTableAdapter.Fill(this.air-
DataSet.View_Trud_dgvr);
    // TODO: данная строка кода позволяет загрузи-
ть данные в таблицу "airDataSet.Fiz_lit". При необходи-
мости она может быть перемещена или удалена.
    this.fiz_litTableAdapter.Fill(this.airDataSet.Fiz_lit);
    // TODO: данная строка кода позволяет загрузи-
ть данные в таблицу "airDataSet.Dolgnost". При необ-
ходимости она может быть перемещена или удалена.
    this.dolgnostTableAdapter.Fill(this.airDataSet.Dol-
gnost);
    dolgnostDataGridView.Columns[0].Visible =
false;//скрыли столбец
    button3.Visible = false; // скрыла удаление
    tip_AirDataGridView.Columns[0].Visible =
false;//скрыли столбец
    per_BPDataGridView.Columns[0].Visible =
false;//скрыли столбец
}

// ДОЛЖНОСТЬ СЧЕТЧИК
private void button1_Click(object sender, EventArgs
e) //кнопка добавить
{
    groupBox1.Visible = true;
    groupBox1.Text = "Добавление должности";
    n = false;
    textBox1.Clear();
    textBox2.Clear();
}

private void button4_Click(object sender, EventArgs
e) // ок
{
    SqlConnection.Close();
    BaseAccess();
    if (n == false)

```

```

{
    if (!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrWhiteSpace(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Text) &&
!string.IsNullOrWhiteSpace(textBox2.Text))
    {
        SqlCommand command1 = new SqlCom-
mand("INSERT INTO Dolgnost (Nazv, Kr_nazv) VAL-
UES(@nazv, @Kr_nazv)", SqlConnection);
        command1.Parameters.AddWith-
Value("@Nazv", textBox1.Text);
        command1.Parameters.AddWith-
Value("@Kr_nazv", textBox2.Text);
        command1.ExecuteNonQuery();
        textBox1.Clear();
        textBox2.Clear();
        SqlConnection.Close();
        this.dolgnostTableAdapter.Fill(this.air-
DataSet.Dolgnost);
    }
    else
    {
        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }
    groupBox1.Visible = false;
}
else
{
    if (!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrWhiteSpace(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Text) &&
!string.IsNullOrWhiteSpace(textBox2.Text))
    {
        dolgnostDataGridView.Columns[0].Visible =
true;// открвли столбец
        dolgnostDataGridView.CurrentRow = dol-
gnostDataGridView[0, dolgnostBindingSource.Position];
        SqlCommand command1 = new SqlCom-
mand("UPDATE Dolgnost SET Nazv = @Nazv,
Kr_nazv=@Kr_nazv WHERE Kod=@Kod", SqlConne-
ction);
        command1.Parameters.AddWith-
Value("@Nazv", textBox1.Text);
        command1.Parameters.AddWith-
Value("@Kr_nazv", textBox2.Text);
        command1.Parameters.AddWith-
Value("@Kod", dolgnostDataGridView.Cur-
rentCell.Value);
        command1.ExecuteNonQuery();
        n = false;
        textBox1.Clear();
        textBox2.Clear();
        SqlConnection.Close();
        dolgnostDataGridView.Columns[0].Visible =
false;// открвли столбец
        this.dolgnostTableAdapter.Fill(this.air-
DataSet.Dolgnost);
    }
    else
    {

```

```

        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }
    groupBox1.Visible = false;
}
}

private void textBox1_KeyPress(object sender, Key-
PressEventArgs e)
{
    if (Char.IsDigit(e.KeyChar) == true)
    {
        label18.Text = "Допускается ввод только
букв!";
        e.Handled = true;
    }
    else
    {
        label18.Text = "";
    }
    char ch = e.KeyChar;
}

private void textBox2_KeyPress(object sender, Key-
PressEventArgs e)
{
    if (Char.IsDigit(e.KeyChar) == true)
    {
        label18.Text = "Допускается ввод только
букв!";
        e.Handled = true;
    }
    else
    {
        label18.Text = "";
    }
    char ch = e.KeyChar;
}

private void button2_Click(object sender, EventArgs
e) // кнопка редактировать
{
    groupBox1.Visible = true;
    groupBox1.Text = "Редактирование должности";
    n = true;
}

private void button3_Click(object sender, EventArgs
e) // кнопка удалить
{
    SqlConnection.Close();
    BaseAccess();
    dolgnostDataGridView.Columns[0].Visible = true;//
открывли столбец
    dolgnostDataGridView.CurrentCell = dolgnostData-
GridView[0, dolgnostBindingSource.Position];
    SqlCommand command3 = new SqlCommand("DE-
LETE FROM Dolgnost WHERE Kod=@Kod", SqlConnec-
tion);
    command3.Parameters.AddWithValue("@Kod",
dolgnostDataGridView.CurrentCell.Value);
    command3.ExecuteNonQuery();

    SqlConnection.Close();
    dolgnostDataGridView.Columns[0].Visible = false;
    this.dolgnostTableAdapter.Fill(this.airDataSet.Dol-
gnost);
}

private void button5_Click(object sender, EventArgs
e) // отмена
{
    textBox6.Clear();
    textBox8.Clear();
}

private void button7_Click(object sender, EventArgs
e) //редактирование
{
    groupBox2.Visible = true;
    groupBox2.Text = "Редактирование физического
лица";
    n = true;
    textBox8.ReadOnly = true;
}

private void button9_Click(object sender, EventArgs
e) // ok
{
    SqlConnection.Close();
    BaseAccess();
    if (n == false)
    {
        if (!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrEmpty(textBox6.Text) &&
!string.IsNullOrEmpty(textBox6.Text) &&
!string.IsNullOrEmpty(textBox8.Text) &&
!string.IsNullOrEmpty(textBox8.Text))
        {
            SqlCommand command1 = new SqlCom-
mand("INSERT INTO Fiz_lit (Tab_Nom, Familia, Imya,
Otch) VALUES(@Tab_Nom, @Familia, @Imya, @Otch)",
SqlConnection);

            command1.Parameters.AddWithValue("@Fa-
milia", textBox4.Text);
            command1.Parameters.AddWith-
Value("@Imya", textBox5.Text);
            command1.Parameters.AddWith-
Value("@Otch", textBox6.Text);
            command1.Parameters.AddWith-
Value("@Tab_Nom", textBox8.Text);
            command1.ExecuteNonQuery();
            textBox8.ReadOnly = false;
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
            textBox8.Clear();

            SqlConnection.Close();

```

```

        this.fiz_litTableAdapter.Fill(this.air-
DataSet.Fiz_lit);
    }
    else
    {
        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }
    groupBox2.Visible = false;
}
else
{
    if (!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrEmpty(textBox6.Text) &&
!string.IsNullOrEmpty(textBox6.Text) &&
!string.IsNullOrEmpty(textBox8.Text) &&
!string.IsNullOrEmpty(textBox8.Text))
    {
        SqlCommand command1 = new SqlCom-
mand("UPDATE Fiz_lit SET Familia = @Familia,
Imya=@Imya, Otch=@Otch WHERE
Tab_Nom=@Tab_Nom", SqlConnection);
        command1.Parameters.AddWithValue("@Fa-
milia", textBox4.Text);
        command1.Parameters.AddWith-
Value("@Imya", textBox5.Text);
        command1.Parameters.AddWith-
Value("@Otch", textBox6.Text);
        command1.Parameters.AddWith-
Value("@Tab_Nom", textBox8.Text);
        command1.ExecuteNonQuery();
        n = false;
        textBox8.ReadOnly = false;
        textBox4.Clear();
        textBox5.Clear();
        textBox6.Clear();
        textBox8.Clear();
        SqlConnection.Close();
        this.fiz_litTableAdapter.Fill(this.air-
DataSet.Fiz_lit);
    }
    else
    {
        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }
    groupBox2.Visible = false;
}
}

private void Form3_FormClosing(object sender,
FormClosingEventArgs e)
{
    if (SqlConnection != null && SqlConnection.State
!= ConnectionState.Closed)
        SqlConnection.Close();
}

```

```

private void button10_Click(object sender, EventArgs
e) // отмена
{
    groupBox2.Visible = false;
    n = false;
    textBox8.ReadOnly = false;
    textBox4.Clear();
    textBox5.Clear();
    textBox6.Clear();
    textBox8.Clear();
    this.fiz_litTableAdapter.Fill(this.airDataSet.Fiz_lit);
}

// ВКЛАДКА ТРУДОВОЙ ДОГОВОР без счетчика
private void button8_Click(object sender, EventArgs
e) // добавлять
{
    groupBox3.Visible = true;
    groupBox3.Text = "Добавление трудового дого-
вора";
    n = false;
    textBox3.Clear();
    dateTimePicker1.Value = DateTime.Today;
    dateTimePicker2.Value = DateTime.Today;
}

private void button12_Click(object sender, EventArgs
e) // OK
{
    SqlConnection.Close();
    BaseAccess();
    if (n == false)
    {
        if (!string.IsNullOrEmpty(textBox3.Text) &&
!string.IsNullOrEmpty(textBox3.Text))
        {
            SqlCommand command1 = new SqlCom-
mand("INSERT INTO Trud_dgvr (Kod, Data_n, Data_o,
Kod_FL, Kod_dol) VALUES(@Kod, @Data_n, @Data_o,
@Kod_FL, @Kod_dol)", SqlConnection);
            command1.Parameters.AddWith-
Value("@Kod", textBox3.Text);
            command1.Parameters.AddWith-
Value("@Data_n", dateTimePicker1.Value);//////////
            command1.Parameters.AddWith-
Value("@Data_o", dateTimePicker2.Value);//////////
            command1.Parameters.AddWith-
Value("@Kod_FL", comboBox2.SelectedValue);
            command1.Parameters.AddWith-
Value("@Kod_dol", comboBox1.SelectedValue);
            command1.ExecuteNonQuery();
            textBox3.ReadOnly = false;
            textBox3.Clear();
            dateTimePicker1.Value = DateTime.Today;
            dateTimePicker2.Value = DateTime.Today;
            SqlConnection.Close();
        }
    }
}

```

```

        this.view_Trud_dgvrTableAdapter.Fill(this.air-
DataSet.View_Trud_dgvr);
    }
    else
    {
        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }
    _n = j;
    if (_m > 0 && _n > 0)
    {
        _a = new double[_m, _n];
        toolStripButton1.Enabled = true;
    }
}
}
}

```

```

private void aGrid_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)

```

```

{
    var form = new Alterna-
tiveForm(@"\Struct\kritery.xml", false, kGrid) { Text =
"Выбор критериев" };
    for (int i = 0; i < form.grid.Rows.Count; i++)
    {
        DataGridViewCheckBoxCell cell = (Data-
GridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
        if (_a[e.RowIndex, i] > 0)
            cell.Value = true;
    }
}

```

```

if (form.ShowDialog() == DialogResult.OK)

```

```

{
    for (int i = 0; i < form.grid.Rows.Count; i++)
    {
        DataGridViewCheckBoxCell cell = (Data-
GridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
        if ((bool)cell.EditedFormattedValue == true)
            _a[e.RowIndex, i] = 1;
        else _a[e.RowIndex, i] = 0;
    }
}

```

```

    groupBox3.Visible = false;
}
else

```

```

{
    if (!string.IsNullOrEmpty(textBox3.Text) &&
!string.IsNullOrWhiteSpace(textBox3.Text))
    {

```

```

        SqlCommand command1 = new SqlCom-
mand("UPDATE Trud_dgvr SET Data_n=@Data_n,
Data_o=@Data_o, Kod_FL=@Kod_FL,
Kod_dol=@Kod_dol WHERE Kod=@Kod", SqlConnec-
tion);

```

```

        command1.Parameters.AddWithValue-
Value("@Kod", textBox3.Text);
        command1.Parameters.AddWithValue-
Value("@Data_n", dateTimePicker1.Value);//////////
        command1.Parameters.AddWithValue-
Value("@Data_o", dateTimePicker2.Value);//////////

```

```

        command1.Parameters.AddWith-
Value("@Kod_FL", comboBox2.SelectedValue);
        command1.Parameters.AddWith-
Value("@Kod_dol", comboBox1.SelectedValue);
        command1.ExecuteNonQuery();
        textBox3.ReadOnly = false;
        textBox3.Clear();
        dateTimePicker1.Value = DateTime.Today;
        dateTimePicker2.Value = DateTime.Today;
        SqlConnection.Close();
        this.view_Trud_dgvrTableAdapter.Fill(this.air-
DataSet.View_Trud_dgvr);

```

```

    }

```

```

    else
    {
        MessageBox.Show("Поля должны быть за-
полнены!", "Внимание");
    }

```

```

        groupBox3.Visible = false;
    }
}

```

```

private void button11_Click(object sender, EventArgs
e) //редактирование

```

```

{
    n = false;
    textBox12.ReadOnly = false;
    textBox7.Clear();
    textBox9.Clear();
    textBox12.Clear();
}

```

```

private void button17_Click(object sender, EventArgs
e) // редактирование

```

```

{
    _n = j;
    if (_m > 0 && _n > 0)
    {
        _a = new double[_m, _n];
        toolStripButton1.Enabled = true;
    }
}

```

```

private void aGrid_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)

```

```

{
    var form = new Alterna-
tiveForm(@"\Struct\kritery.xml", false, kGrid) { Text =
"Выбор критериев" };
    for (int i = 0; i < form.grid.Rows.Count; i++)
    {
        DataGridViewCheckBoxCell cell = (Data-
GridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
        if (_a[e.RowIndex, i] > 0)
            cell.Value = true;
    }
}

```

```

if (form.ShowDialog() == DialogResult.OK)

```

```

{
    for (int i = 0; i < form.grid.Rows.Count; i++)
    {

```



```
DataGridViewCheckBoxCell cell = (Data-
GridViewCheckBoxCell)form.grid.Rows[i].Cells[0];
if ((bool)cell.EditedFormattedValue == true)
    _a[e.RowIndex, i] = 1;
else _a[e.RowIndex, i] = 0;
}
```

```
groupBox4.Visible = true;
groupBox4.Text = "Редактирование аэропорта";
n = true;
textBox12.ReadOnly = true;
```

```
}
private void button15_Click(object sender, EventArgs
e) //ok
{
    SqlConnection.Close();
    BaseAccess();
    if (n == false)
    {
        if (!string.IsNullOrEmpty(textBox7.Text) &&
!string.IsNullOrWhiteSpace(textBox7.Text) &&
```