Южно-Уральский государственный университет

На правах рукописи

Латипова Алина Таиховна

ПРИМЕНЕНИЕ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ К ПРОБЛЕМЕ ОПТИМИЗАЦИИ БЮДЖЕТА ПРОДАЖ ПРИ ЦЕНОВОЙ ДИВЕРСИФИКАЦИИ

Специальность: 05.13.18 «Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель –
доктор физико-математических наук,
профессор А. В. Панюков

Челябинск – 2008

ОГЛАВЛЕНИЕ

Введение

Актуальность темы. В настоящее время в России повсеместно внедряются международные стандарты финансовой отечности и управленческого учета. Правительство России со своей стороны предпринимает различные меры для ускорения этого процесса, так как это является необходимым условием для привлечения иностранного капитала [6, 57].

Важной частью такого учета является бюджетирование, которое является инструментом финансового планирования и контроля за деятельностью компании и её структурных подразделений. Ключевая цель бюджетирования - обеспечение производственно-коммерческого процесса необходимыми по объему и структуре ресурсами. Бюджеты являются ценным источником информации в планировании хозяйственных операций, способствуют координации деятельности различных подразделений, применяются для оценки эффективности [4, 8, 25, 58, 60, 62].

Отсутствие нормативов затрат и планирование продаж исходя из объемов производства ведет к снижению качества планирования и убыткам [6]. Также малоэффективен подход, когда различные службы и отделы самостоятельно разрабатывают свои бюджеты по собственным методикам без централизации, так как руководители подразделений будут отстаивать собственные интересы в ущерб целям всего предприятия [11].

Согласно современным концепциям развития бизнеса большое внимание должно уделяться спросу, поэтому среди методик бюджетирования наиболее популярной является нормативный подход, основанный на данных о спросе - бюджете продаж (об объемах и ценах реализации на различные виды продукции) [6, 57, 58]. Все остальные бюджеты рассчитываются на основе бюджета продаж и различных норм. Данный подход позволяет прогнозировать практически все аспекты деятельности фирмы.

Тем не менее, возникает проблема обоснованности бюджета продаж и его эффективности, т.к. в методике бюджетирования данные об объемах продаж являются заданными. В работах по бюджетированию [6, 22, 52, 55, 57, 58] данный вопрос рассмотрен поверхностно: основным способом улучшения качества бюджетов по мнению авторов является имитационное моделирование, которое в таких условиях применять бессмысленно в виду наличия большого числа альтернатив.

Кроме того, не учитывается возможность применения нескольких ценовых стратегий [12, 27]. Под ценовой стратегией большинство авторов [6, 22, 52, 55, 57, 58] понимают установление цены для различных видов товаров на основе расчета себестоимости единицы продукции и предполагаемой нормы рентабельности без учета сегментации спроса. Между тем, практика показывает, что одним из инструментов повышения эффективности продаж является ценовая диверсификация [12, 27]. Поэтому целесообразно разработать адекватную математическую модель оптимизации бюджета продаж и соответствующие алгоритмы поиска [36]. Исходя из этого, задача оптимизации бюджета продаж состоит в выработке эффективной структуры интенсивностей применения ценовых стратегий с учетом данных о спросе, переменных и постоянных издержек [35].

Для решения подобных экономических задач широко применяется балансовый подход, при котором доходы от продаж соизмеряются с расходами и прибылью [3, 10, 17, 18, 39, 40, 49]. Однако данные балансовые модели не учитывают ценовую диверсификацию. При этом по некоторым товарным позициям в найденном решении выручка может не покрывать переменных затрат, что приводит к повышению финансовой нестабильности. В других моделях [16, 39] рассматриваются балансовые отношения потоков материалов и продукции в натуральном выражении, в результате чего не учитывается финансовая сторона. В некоторых работах [10, 21, 24] фиксируются объемы продаж, а цена рассчитывается исходя из нормативов затрат. В этом случае из рассмотрения упускается влияние цены на объем спроса. Также ряд моделей [10, 26] рассматривают предприятия, выпускающие только один вид продукции, что является редкостью. Графические методы для нахождения оптимального бюджета продаж, предложенные в данных работах, отличаются не-точностью и мало применимы для многопродуктовых моделей.

Другим важным вопросом, который зачастую недостаточно проработан в некоторых работах [11, 29], является анализ устойчивости математической модели [48] и влияния интервальной неопределенности [19] входных данных. Под устойчивостью авторы некоторых работ понимают положительно сальдо на конец планового периода на расчетном счете, превышение общей выручки над затратами [10, 22, 24]. Однако не рассматривается такой показатель устойчивости как внутренняя норма рентабельности продаж для проекта бюджетирования [38]. Интервальная неопределенность прежде всего связана с данными о спросе, так как в ходе маркетингового исследования объем спроса при заданной цене невозможно точно определить. Это в свою очередь приводит к интервальной неопределенности данных о выручке, общих затратах и прибыли [32]. Тем не менее, эта проблема в статьях [3, 10, 17, 18, 39, 40] не рассматривается, а объемы продаж рассчитываются на основе эконометрических моделей или эластичности без указания точности расчета. В модели, предложенной в диссертационной работе, показана связь интервальной неопределенности с решением для точечных данных [30, 31].

В настоящей работе приведена методика расчета оптимального бюджета продаж, которая применима к большинству производственных предприятий на этапе стратегического планирования [33]. Найденные объемы продаж не превышают спроса, переменные затраты окупаются за счет выручки, а ценообразование строится исходя из ценовой диверсификации [34, 37]. Таким образом, в данной работе учитываются роль потребителей и параметров производства в построении бюджета продаж.

Расчет внутренней нормы рентабельности, предложенный в настоящей работе, позволяет оценить устойчивость модели, то есть определить эффективность ценовых стратегий и уровень критичности рентабельности для различных видов продукции. Выводы, полученные при анализе устойчивости, позволят предприятию выработать направления для развития рынков сбыта и снижения затрат [30].

Очевидно, что результаты, полученные в ходе оптимизации ценовой стратегии, имеют рекомендательный характер, поэтому алгоритм оптимизации удобнее реализовать в виде отдельного модуля программного обеспечения [34]. Отделение модуля оптимизации от корпоративной системы требует наличие механизма загрузки и выгрузки данных. Как правило, для решения оптимизационных задач предполагается экспорт из текстовых файлов с разделителями [8, 48], при формировании которых может быть много ошибок из-за неочевидности структуры и чувствительности к знакам пунктуации. На данный момент самыми удобными форматами данных для таких операций являются язык разметки `XML (Extensible Mark-Up Language)` и книги `Microsoft Excel`, которые поддерживаются современными корпоративными системами [1, 23, 41, 62, 64, 65]. Для облегчения экспорта-импорта нами разработан файл с правилами для структуры `XML`-файла, а также параметры для книг `Excel`. При разработке программного обеспечения необходимо уделить большое внимание обеспечению целостности базы данных [28]. Недостатком многих модулей оптимизации является проблема представления результатов [11, 50], поэтому в разработанном модуле уделяется внимание формированию выходных форм [36]. Модуль может применяться на различных уровнях управления, поэтому модуль должен быть тиражируем и прост в установке.

Цель и основные задачи диссертационной работы. Целью работы является разработка методов моделирования и оптимизации бюджета продаж, а также создание программного обеспечения оптимизации бюджетирования. Для достижения поставленной цели необходимо решить следующие задачи:

1. Построить математическую модель для решения проблемы оптимизации бюджета продаж с учетом ценовой диверсификации. Разработать методы исследования полученной модели на продуктивность и устойчивость при точечном и интервальном характере входных данных.

2. Построить и обосновать эффективные численные методы определения оптимального бюджета продаж с использованием построенной модели.

3. Разработать проблемно-ориентированный комплекс программ, реализующий данные численные методы. Провести вычислительные эксперименты и внедрить программное обеспечение на предприятии.

Методы исследования. Для исследования применяются методы математического анализа, математико-экономического моделирования.

Научная новизна состоит в построенной математической модели бюджетирования, в численных методах нахождения параметров равновесия модели Неймана и найденных новых свойствах параметров равновесия, в разработке комплекса программ:

1. Новизна построенной математической модели формирования бюджета продаж заключается в применении наряду с балансовым подходом ценовой диверсификации.

2. Новизна методов нахождения параметров равновесия модели Неймана состоит в применении подходов теории антагонистических игр.

3. Впервые предложены методы нахождения граничных значений параметров равновесия модели Неймана при интервальном задании исходных данных.

Связь работы с государственными и международными программами. Работа выполнена при финансовой поддержке Министерства образования РФ и правительства Челябинской области (грант для аспирантов номер 035.06.06-04.БМ 2004 года и грант для студентов 2002 года по секции «Естественные науки»).

Практическая значимость. Практическая значимость состоит в повышении качества разрабатываемых бюджетов продаж. Предложенные в работе численные методы могут быть использованы для анализа параметров равновесия фоннеймановских моделей общего вида. Разработанный комплекс программ (свидетельство РосПатента о регистрации № 2007613463, см. приложение Д) внедрен в Челябинском филиале ЗАО «Пронто-Уфа» (см. приложение Е) и в учебный процесс в Южно-Уральском государственном университете на специальностях «Экономика и управление на предприятии», «Прикладная информатика», «Математические методы в экономике» и «Статистика». Научные результаты исследования и разработанное программное обеспечение могут быть использованы при разработке проектов бюджетов как государственных, так и коммерческих предприятий, в частности предприятий сотовой связи, рекламных агентств, при организации Интернет-подписки, при страховании и т.д.

Апробация работы. Основные результаты и положения работы докладывались и обсуждались на следующих научных мероприятиях:

- семинар «Проблемы исследования операций» в Уфимском государственном авиационном техническом университете, 2008 г.
- семинар отдела математического программирования ИММ УрО РАН, Екатеринбург, 2008 г.
- 38-ая Региональная молодежная конференция ИММ УрО РАН «Проблемы теоретической и прикладной математики», Екатеринбург, январь-февраль 2007 г.
- 37-ая Региональная молодежная конференция ИММ УрО РАН «Проблемы теоретической и прикладной математики», Екатеринбург, январь-февраль 2006 г.
- Всероссийская конференция «Экономика и менеджмент: проблемы и перспективы», Санкт-Петербург, июнь 2005 г.
- 36-ая Региональная молодежная конференция ИММ УрО РАН «Проблемы теоретической и прикладной математики», Екатеринбург, январь-февраль 2005 г.
- Международная конференция «Стратегия развития минерально-сырьевого комплекса в XXI веке», Москва, октябрь 2004 г.
- Международная конференция «Диалог-симпозиум: наука и инновации», Томск, октябрь 2004 г.
- Всероссийская конференция с международным участием ИММ СО РАН «Дискретный анализ и исследование операций», Новосибирск, июнь 2004 г.

- 35-ая Региональная молодежная конференция ИММ УрО РАН «Проблемы теоретической и прикладной математики», Екатеринбург, январь 2004 г.

- ежегодные научно-практические конференции ЮУрГУ.

Доклады на Международной конференции «Диалог-симпозиум: наука и инновации», Томск, октябрь 2004 г. и на Всероссийской конференции «Экономика и менеджмент: проблемы и перспективы», Санкт-Петербург, июнь 2005 г. отмечены дипломами (см. приложение Д).

Публикации. По теме диссертации опубликовано 11 работ. В их числе зарегистрированных программ для ЭВМ – 1 (см. приложение Д); статей в журналах, рекомендованных ВАК – 2; статей в сборниках научных трудов – 7, тезисов докладов   1.

Первая глава посвящена описанию проблемы бюджетирования и её математической модели.

Рассматриваются виды бюджетов и этапы построения бюджетов. Описывается нормативный подход. Делается вывод о ключевой роли бюджета продаж. Построение бюджета продаж связано с выработкой ценовой стратегии, поэтому в данной главе также уделяется внимание ценовой диверсификации.

Во второй главе формулируется модель оптимизации: описываются входные и выходные параметры, ограничения и целевая функция. Показано, что полученная модель бюджетирования аналогична модели Неймана. Рассмотрены проблема нахождения параметров модели и экономическая интерпретация ряду агрегированных параметров данной задачи. Для проведения вычислительного эксперимента с использованием коммерческого программного обеспечения разработана программа генерации тестовых задач. Представлены и проанализированы результаты вычислительного эксперимента с применением пакетов коммерческого программного обеспечения, показавшие неудовлетворительное качество решения сгенерированных задач

В третьей главе рассмотрены методы и алгоритмы решения задачи, изложенной во второй главе. Предложен подход на основе теории матричных игр.

Четвертая глава посвящена решению задачи при интервальной неопределенности исходных данных. Рассматриваются некоторые аспекты интервального анализа. Показана связь интервального и точечного решения задачи бюджетирования.

В пятой главе проведено исследование проблемы автоматизации процессов бюджетирования, в ходе которого указаны основные требования к проекту, ключевые функции и основные ограничения.

В шестой главе приведен расчет бюджета продаж с использованием разработанного программного обеспечения.

В заключении подведены итоги проведенного исследования и обоснованы направления использования полученных результатов.

Нумерация разделов в настоящей работе тройная – j.k.l, где j – номер главы, k – номер раздела, l – номер подраздела.

1. Проблема бюджетирования

1.1. Методика бюджетирования

В настоящее время не существует однозначного определения понятия бюджета. В [57] предлагается следующее определение: бюджет (от англ. budget – сумка) – имеющая официальную силу, признанная или принятая роспись, таблица, ведомость доходов и расходов экономического субъекта за определенный период времени, обычно за год. Чаще всего бюджет составляется для учета количества располагаемых и расходуемых денежных средств и их взаимного соответствия; это количественное выражение централизованно устанавливаемых показателей плана предприятия на определенный период по [6]:

– использованию капитальных, товарно-материальных, финансовых ресурсов;

– привлечению источников финансирования текущей и инвестиционной деятельности;

– доходам и расходам;

– движению денежных средств;

– инвестициям (капитальным и финансовым вложениям).

Определяющими характеристиками бюджета предприятия являются формализация, централизация и системность [58]. Формализация означает, что бюджет – это прежде всего совокупность количественных показателей.

Методология бюджетирования на отдельных предприятиях может различаться [6]:

– «сверху вниз» (проекты бюджетов подразделений разрабатываются службами аппарата управления);

– «снизу вверх» (проекты бюджетов разрабатываются самими подразделениями);

– встречное планирование (проекты бюджетов разрабатываются подразделениями с последующей корректировкой службами аппарата управления).

Однако, в конечном итоге, утверждение бюджетных показателей происходит по линии «аппарат управления (директивный орган) – подразделение (центр ответственности за исполнение утвержденных бюджетных показателей)».

«Сквозной» характер бюджетирования (системность) означает, что в бюджетном процессе совокупность бюджетов отдельных центров ответственности в обязательном порядке формирует сводный бюджет компании в целом.

Бюджет предприятия всегда разрабатывается для определенного временного интервала, называемого бюджетным периодом [57]. Предприятие может одновременно составлять несколько бюджетов, различающихся по продолжительности бюджетного периода (текущий квартальный бюджет, среднесрочный годовой бюджет, долгосрочный бюджет развития на 3–5 лет и т.д.).

Бюджет – это насыщенный количественными показателями документ, в соответствии с которым предприятие ведет свою хозяйственную деятельность. Бюджетирование – это процесс составления и реализации данного документа в практической деятельности компании.

Бюджетирование – это процесс разработки и формирования плановых бюджетов, объединяющих планы руководства предприятия и в первую очередь производственные, маркетинговые и финансовые [6].

Бюджетирование – это управленческая технология, предназначенная для выработки и повышения финансовой обоснованности принимаемых управленческих решений [52].

Проанализировав точку зрения разных авторов на сущность бюджетирования, можно сделать вывод об общем характере их представлений этого процесса как **системы разработки и реализации управления ресурсами предприятия** на основе взаимосвязанных производственных и финансовых показателей.

Для принятия эффективных решений в области управления ресурсами необходим не только опыт и экономическая интуиция, но и реальное математическое обоснование этих решений. Цели обоснования принятия определенных управленческих и финансовых решений служат различные экономико-математические модели и методы.

В общем виде модель бюджетирования предприятия можно представить следующим образом (рисунок 1.1) [58].

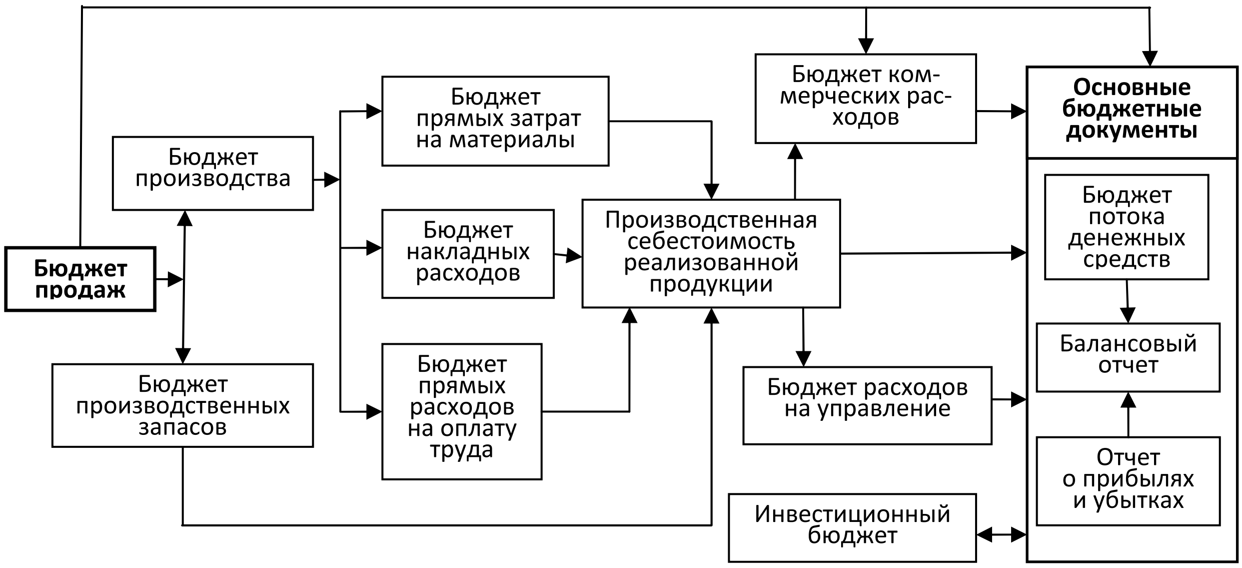

Рисунок 1.1. Блок-схема составления основных бюджетов компании

Применяемые в финансовом планировании виды бюджетов можно разделить на 4 основные группы [57].

1. **Основные бюджеты** (бюджет доходов и расходов, бюджет движения денежных средств, расчетный баланс).

2. **Операционные бюджеты** (бюджет продаж, бюджет прямых материальных затрат, бюджет прямых затрат на оплату труда, бюджет управленческих расходов, бюджет коммерческих расходов, бюджет производства, бюджет запасов).

3. **Вспомогательные бюджеты** (инвестиционный бюджет, план капитальных затрат, кредитный план, бюджеты отдельных проектов и программ).

4. **Дополнительные (специальные) бюджеты** (налоговый бюджет, план распределения чистой прибыли).

Основные бюджеты предназначены для управления финансами предприятия, оценки финансового состояния бизнеса, принятия управленческих решений.

Операционные и вспомогательные бюджеты необходимы для более точного составления основных бюджетов, определения наиболее важных пропорций, ограничений и допущений, которые следует учитывать при составлении основных бюджетов. Состав операционных и вспомогательных бюджетов может определяться руководителями предприятия прежде всего исходя из характера стоящих перед организацией целей и задач, а также степени методической, организационной и технической готовности предприятия [6].

Дополнительные (специальные) бюджеты необходимы для более точного определения целевых показателей и нормативов финансового планирования, более точного учета особенностей отраслевого или местного налогообложения. Набор специальных бюджетов может определяться самостоятельно руководителем предприятия в зависимости от специфики хозяйственной деятельности.

На базе операционных и вспомогательных бюджетов составляется сводный производственный бюджет или **мастер-бюджет**.

Итогом действия процесса бюджетирования предприятия является **сводный бюджет** – консолидированный финансовый план, разрабатываемый на основе бюджетов нескольких видов бизнесов или структурных подразделений.

Отправной точкой формирования бюджета предприятия является формирование **бюджета продаж**. В основе достижения устойчивого положения и коммерческого успеха любого предприятия лежит определение реальных ориентиров деятельности, т.е. прогнозирование изменения объема предоставляемых услуг и реализуемых товаров.

Для прогнозирования объема продаж применяют следующие математические методы [57].

1. Построение эконометрических моделей. Недостатком этого метода является значительная трудоемкость установления зависимости для прогнозирования, не всегда имеется достаточное количество статистической информации

для построения модели. Преимуществом является достаточная точность и простота расчетов.

2. Метод экстраполяции. Заключается в прогнозировании на основе предположения о сохранении тенденции развития объекта в будущем. Не всегда подобные предположения можно считать обоснованными, поэтому точность данного метода достаточно невысокая. Достоинством метода является простота расчетов.

3. Экспертный метод**.** Метод заключается в прогнозирование развития объектов по экспертным оценкам специалистов в данной области. Целесообразно применять, если нет достаточной статистической информации по развитию объекта.

Бюджет производства отражает объем производства, соответствующий запланированному объему реализации продукции и запасу готовой продукции на конец предстоящего периода [6].

Бюджет прямых материальных затрат составляется на основе производственного бюджета. Он показывает, сколько сырья и материалов требуется для производства и сколько сырья и материалов должно быть закуплено.

Существует два основных расчетных инструмента определения потребности прямых материальных затрат:

– метод технологического нормирования;

– метод сравнительного анализа счетов.

Технологическое нормирование относится, как правило, к той части бюджета основных материалов, которая расходуется на производственные цели. На большинстве предприятий существуют удельные нормы расхода по видам сырья и материалов в расчете на одну единицу отдельного вида продукции. Таким образом, при имеющейся плановой производственной программе (бюджет производства), применяя технологические нормы (стандарты), рассчитывается производственная потребность в разрезе видов товарно-материальных ресурсов на данный бюджетный период.

Метод сравнительного анализа счетов является более простым в практическом применении, но одновременно дает более грубую оценку. Он используется в основном на тех предприятиях, где отсутствует система технологического нормирования. Суть данного метода заключается в том, что за ряд прошлых бюджетных периодов по данным оперативной отчетности сопоставляются объемы производства и сбыта с динамикой расходования материальных ресурсов и на основе средневзвешенной устанавливаются нормы расхода на данный бюджетный период. Затем эти нормы применяются к плановым объемам производства и сбыта для определения потребностей в основных материалах.

Бюджет запасов является планом изменения объема и структуры запасов товарно-материальных ценностей.

Бюджет закупок – план закупок материальных оборотных средств (сырья, материалов и комплектующих) и изменения складских запасов материальных оборотных средств предприятия за бюджетный период.

Объем закупок сырья и материалов зависит от ожидаемого объема их использования, а также от предполагаемого уровня запасов. Для оптимизации запасов и закупок применяют математические модели управления запасами.

Модель управления запасами включает: выбор и обоснование критерия оптимизации, расчет издержек управления запасами, формулировку ограничений, моделирование спроса (расхода) и пополнения запасов, расчет стратегии управления. В настоящее время существует большое количество методов и моделей управления запасами, являющихся предметом изучения одного из разделов исследования операций – теории управления запасами.

Бюджет доходов и расходов, или, как его еще называют, бюджет прибыли и убытков показывает соотношение всех доходов (выручки) от реализации в плановый период со всеми видами расходов, которые предполагает понести в этот же период предприятие.

Бюджет движения денежных средств – это план движения расчетного счета и наличных денежных средств в кассе предприятия, отражающий все прогнозируемые поступления и списания денежных средств в результате хозяйственной деятельности компании.

Расчетный баланс, или прогноз по балансовому листу формируется в соответствии со сложившейся структурой активов и задолженностей и ее наиболее вероятным изменением в процессе реализации других бюджетов.

Бюджеты должны быть увязаны между собой по горизонтали и быть взаимоувязанными по вертикали (уровням управления) путем формирования сводных бюджетов структурных подразделений [58].

Таким образом, **ключевым бюджетом**, на основе которого строятся все основные бюджеты, является **бюджет продаж**. Необоснованность бюджета продаж влечет за собой необоснованность всех остальных бюджетов [36].

Бюджет продаж представляет собой данные для каждого товара об объемах продаж при заданной цене. Таким образом, для формирования бюджета продаж необходимо разработать **ценовую политику фирмы**.

1.2. Ценовая политика фирмы

Ценовая политика фирмы представляет собой установление фирмой цен на продукты при различных условиях. Среди основных целей ценовой политики является максимизация прибыли (долгосрочная и краткосрочная), а также расширение спроса и ускорение его роста [27].

При разработке ценовой политики необходимо учитывать зависимость от цены **спроса** (чувствительность спроса к цене) и оценить величину **издержек** при данном уровне спроса [12]. Для построения функциональной зависимости объема спроса от цены применяют **ценовую эластичность** (определенную при маркетинговом исследовании) или данные экспертов.

Существует несколько **способов ценообразования** [12]:

1) метод надбавок (цена равна издержкам плюс процент прибыли);
2) метод потребительской оценки;
3) метод следования за лидером конкуренции;
4) ценообразование на конкурсные проекты.

При любой цене должно выполняться **условие покрытия** (цена на товар покрывает переменные издержки на единицу данного продукта) [12].

Цена на похожие товары с различными характеристиками может быть разной. Такие отличия называются **ценовыми модификациями**. К ценовым модификациям относятся [12]:

1) модификации по географическому признаку;
2) через систему набавок и скидок;
3) модификации для стимулирования продаж;
4) ценовая диверсификация (дискриминация) и т.д.

Ценовая диверсификация – одна из самых эффективных стратегий. **Ценовая диверсификация** представляет собой приспособление цен к специфическим особенностям отдельных потребителей, модификациям товара или различиям стандартов [12]. Формы модификаций для ценовой дискриминации могут быть следующими:

1) модификация в зависимости от потребительского сегмента;
2) модификация цен в зависимости от форм продукта и его применения;
3) модификация цен в зависимости от имиджа;
4) дифференциация в зависимости от местоположения;
5) модификация цен в зависимости от времени.

Основными **ценовыми стратегиями** являются следующие виды [12]:

1) стратегия ориентации на товар;
2) стратегия ориентации на технологию;
3) стратегия ориентации на качество;
4) стратегия ориентации на сервис;
5) стратегия ориентации на покупателя.

Самой выигрышной ценовой стратегией, как показывает практика, является последняя. Таким образом, ценовая диверсификация является эффективным инструментом ценовой политики и может быть использована для формирования бюджета продаж. Для проведения диверсификации нужно проанализировать рынок и выделить в нем сектора (по уровню доходов, территориальному расположению, предпочтениям). Для каждого такого сегмента должна быть выработана ценовая стратегия (пример ценовой диверсификации - тарифные планы сотовых компаний) [12].

Важнейшим бюджетным документом для инвесторов является бюджет операционной прибыли, в котором отражаются суммы выручки, переменных и постоянных затрат. Инвесторы заинтересованы в максимизации эффективности инвестиций, т.е. в максимизации рентабельности [25].

Таким образом, при утверждении бюджетов необходимо представить инвесторам бюджет продаж, экономически обоснованный с точки зрения рентабельности и учитывающий ценовую диверсификацию. Поэтому процесс бюджетирования можно представить в виде циклической последовательности этапов [30].

1.3. Выводы и результаты

Данная глава посвящена экономической постановке задачи бюджетирования:

1) рассмотрены методики бюджетирования:

1.1) представлена классификация бюджетов;

1.2) показана ключевая роль бюджета продаж;

2) представлены варианты ценовой политики, необходимой для построения бюджет продаж:

2.1) рассмотрены способы ценообразования;

2.2) описаны виды ценовых модификаций;

2.3) показано, что наиболее эффективной модификацией является ценовая диверсификация;

2.4) рассмотрены типы ценовых стратегий.

Исходя из этого в разделе 1.2 делается вывод, что наиболее эффективной ценовой политикой является ценовая диверсификация, ориентированная на покупателя, максимизирующая рентабельность продаж. Таким образом, проведен анализ предметной области, необходимый для разработки математической модели.

2. Математическое моделирование бюджетирования с применением ценовой диверсификации

Как было показано в главе 1, наиболее эффективной ценовой политикой является ценовая диверсификация, ориентированная на покупателя, максимизирующая рентабельность продаж. В связи с этим основой математического моделирования должна быть сегментация рынка, при которой для каждого сектора вырабатывается своя ценовая стратегия и определяется его удельный объем в общих продажах.

2.1. Математическая модель

Пусть потребителей n товаров (услуг), реализуемых предприятием, можно разделить на m **секторов**. Для каждого j-того сектора компанией разработана своя **ценовая стратегия**. Тогда данные о ценовых стратегиях задаются двумя матрицами P и Q, где матрица $P = \left[p_{ij} \right]$ - **матрица цен**, p_{ij} - цена на i-й продукт без налога на добавленную стоимость (НДС) при применении стратегии j $(i = 1,2,...n; j = 1,2,...m)$; матрица $Q = \left[q_{ij} \right]$ - **матрица объемов спроса**, где q_{ij} - ожидаемый спрос на i-й продукт при использовании стратегии j с единичной интенсивностью [30]**. Интенсивность использования стратегий** задается вектором x, где x_j - интенсивность j-й стратегии, $x^T = (x_1, x_2, ..., x_j, ..., x_m)$, $0 \le x_j \le 1$.

Согласно нормативной концепции [6] при бюджетировании используется данные о различных **нормах затрат**. В рамках разных ценовых стратегий один и тот же товар может иметь разные потребительские свойства (качество, вид упаковки, вид сырья и т.д.). Кроме того, может быть различным уровень расходов на маркетинг и перевозку. Следовательно, нормы затрат при ценовой диверсификации можно задать следующими матрицами норм переменных затрат на единицу готовой продукции (ГП) в стоимостном выражении [32]:

- матрицей $M = \left[m_{ij} \right]$ норм затрат сырья и материалов без НДС;

- матрицей $L=\left[l_{ij}\right]$ норм затрат труда с начислениями по единому социальному налогу;
- матрицей $S=\left[s_{ij}\right]$ норм затрат на маркетинг и транспортные услуги без НДС.

Также задан вектор c постоянных затрат на различные виды продукции ($c_i \ge 0,\ i=1,2,...,n$).

Применение ценовой диверсификации должно быть экономически обоснованным, т.е. выручка должна покрывать все расходы. Данное ограничение можно представить в виде **внутрифирменного баланса** [34]

$$(P\circ Q)x \ge (M\circ Q)x+(L\circ Q)x+(S\circ Q)x+r(P\circ Q)x+c, \qquad (2.1.1)$$

где r - это процент прибыли в сумме выручки (до обложения налогом на прибыль), общий для всех видов продукции ($r\ge 0$); символ “$\circ$” означает поэлементное умножение матриц $P\circ Q=\left[p_{ij}\cdot q_{ij}\right]$.

Целью бюджетирования является определение вектора интенсивностей x^*, максимизирующего рентабельность r^*, т.е.

$$(x^*,r^*)=\arg\max_{(x,r)\in D} r,$$

$$D=\left\{(x^*,r^*)\middle|\begin{array}{c}(P\circ Q)x\ge(M\circ Q)x+(L\circ Q)x+(S\circ Q)x+r(P\circ Q)x+c,\\ 0\le x\le 1.\end{array}\right\} \qquad (2.1.2)$$

Полученная задача оптимизации бюджета продаж является задачей билинейного программирования с n балансовыми ограничениями и $m+1$ переменными. Исходные данные в модели бюджетирования должны удовлетворять трем условиям [35]:

1. Предполагается, что в каждой стратегии продаж используется как минимум один товар, и каждый товар применяется как минимум в одной стратегии. Т.е. в матрице Q отсутствуют нулевые строки и нулевые столбцы.

2. Цена на товар должна полностью покрывать переменные расходы (условие безубыточности): $p_{ij}\ge m_{ij}+l_{ij}+s_{ij}$.

3. Выпуск и реализация i-того товара сопровождается переменными затратами. Если для стратегии j объем реализации положительный ($q_{ij} > 0$), то $m_{ij} + l_{ij} + s_{ij} > 0$.

Матрицу $B = P \circ Q$ назовем **матрицей выручки**. Матрицу $A = (M + L + S) \circ Q$ назовем **матрицей переменных издержек**. Тогда внутрифирменный баланс (2.1.1) можно записать в таком виде: $Bx - Ax \ge c + rBx$, $x \ge 0, c \ge 0$, $r \ge 0$ [35].

2.2. Анализ модели

Полученная модель бюджетирования **аналогична модели Неймана**: матрица выручки аналогична матрице выпуска, матрица переменных издержек - матрице затрат, вектор постоянных затрат - вектору конечного спроса. Исходя из этой аналогии, для анализа модели бюджетирования можно применять подходы, которые используются для анализа продуктивности моделей неймановского типа [42].

Модель бюджетирования назовем **продуктивной**, если для системы ограничений $\left(Bx - Ax \ge c\right)$ существует решение $x \ge 0$ при любом неотрицательном векторе постоянных затрат $c \ge 0$. Доказано, что модель Неймана (A, B) продуктивна тогда и только тогда, когда фробениусово число модели $\overline{\lambda} < 1$ [30]. Вследствие отмеченной аналогии для продуктивности модели бюджетирования необходимо и достаточно выполнение неравенства $\overline{\lambda} < 1$.

Для **нахождения числа Фробениуса** $\overline{\lambda}$ используется задача нелинейного программирования [2], которая трансформирована в следующий вид [33]:

$$(\overline{\lambda}, \overline{x}, \overline{w}) = \arg \max_{\lambda, x, w} \lambda$$

$$\begin{cases} (A - \lambda B)x \le 0, (A - \lambda B)^T w \ge 0, \\ (x, e^m) = 1,\ (w, e^n) = 1, \\ x \ge 0,\ w \ge 0,\ \lambda \ge 0. \end{cases} \tag{2.2.1}$$

Её можно также записать следующим образом:

$$(\bar{\lambda},\bar{x},\bar{w}) = \arg\max_{(\lambda,x,w)\in D(A,B)} \lambda,$$

$$D(A,B) = \left\{ (\lambda,x,w) \left| \begin{matrix} (A-\lambda B)x \le 0, (A-\lambda B)^T w \ge 0, \\ (x,e^m)=1,\ (w,e^n)=1, x\ge 0,\ w\ge 0,\ \lambda \ge 0 \end{matrix} \right. \right\}. \qquad (2.2.2)$$

Величину $\bar{r} = (1-\bar{\lambda})$ можно интерпретировать как уровень рентабельности продаж, достижимый при любом неотрицательном векторе постоянных затрат $c \ge 0$, а вектор Фробениуса $\bar{x}$ как структуру интенсивностей стратегий, обеспечивающую данный уровень рентабельности.

Задача нахождения луча Неймана при фиксированном значении $\lambda = \bar{\lambda}$ является **задачей линейного программирования** [43].

Теорема о **дополняющей нежесткости** позволяет дать следующую интерпретацию ряду величин [31, 42].

Величина $-\sum_{j=1}^{m}(a_{ij} - \bar{\lambda}b_{ij})\cdot \bar{x}_j$ показывает величину сверхнормативной маржинальной прибыли от реализации i-го продукта. Величину $\bar{w}_i$ можно рассматривать как показатель критичности рентабельности i-го продукта.

Величина $\sum_{i=1}^{n}(a_{ij} - \bar{\lambda}b_{ij})\cdot \bar{w}_i$ показывает уровень скрытой убыточности j-той стратегии. Величина $(b_{ij} - a_{ij} - \bar{r}b_{ij})$ – сверхнормативная маржинальная прибыль на единицу i-го товара при единичной интенсивности j-ой стратегии.

2.3. Аналитическое исследование для задачи *m=2, n=2*

Рассмотрим аналитическое решение задачи для модели Неймана:

$$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \text{ и } B=\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

Чтобы найти число и векторы Фробениуса, согласно (2.2.2) необходимо решить следующую задачу.

$$(\bar{\lambda},\bar{x},\bar{w}) \to \arg\max_{\lambda,x,w} \lambda$$

$$\begin{cases} (a_{11}-\lambda b_{11})x_1+(a_{12}-\lambda b_{12})x_2 \le 0, \\ (a_{21}-\lambda b_{21})x_1+(a_{22}-\lambda b_{22})x_2 \le 0, \\ (a_{11}-\lambda b_{11})w_1+(a_{21}-\lambda b_{21})w_2 \ge 0, \\ (a_{12}-\lambda b_{12})w_1+(a_{22}-\lambda b_{22})w_2 \ge 0, \\ x_1+x_2=1, \\ w_1+w_2=1, \\ \lambda,x,w \ge 0. \end{cases} \qquad (2.3.1)$$

Рассмотрим случай $b_{11} \ne b_{12}$, $b_{11} \ne b_{21}$, $b_{12} \ne b_{22}$, $b_{21} \ne b_{22}$ и все элементы матрицы B положительны. Тогда задача (2.3.1) примет вид

$$(\bar{\lambda},\bar{x},\bar{w}) \to \arg\max_{\lambda,x_1,w_1} \lambda$$

$$\begin{cases} \lambda \ge ((a_{11}-a_{12})x_1+a_{12})/((b_{11}-b_{12})x_1+b_{12}), \\ \lambda \ge ((a_{21}-a_{22})x_1+a_{22})/((b_{21}-b_{22})x_1+b_{22}), \\ \lambda \le ((a_{11}-a_{21})w_1+a_{21})/((b_{11}-b_{21})w_1+b_{21}), \\ \lambda \le ((a_{12}-a_{22})w_1+a_{22})/((b_{12}-b_{22})w_1+b_{22}), \\ 0 \le x_1 \le 1, \\ 0 \le w_1 \le 1, \\ \lambda \ge 0. \end{cases} \qquad (2.3.2)$$

Введем функции

$$\lambda_1(x_1)=((a_{11}-a_{12})x_1+a_{12})/((b_{11}-b_{12})x_1+b_{12}),$$

$$\lambda_2(x_1)=((a_{21}-a_{22})x_1+a_{22})/((b_{21}-b_{22})x_1+b_{22}),$$

$$\lambda_3(w_1)=((a_{11}-a_{21})w_1+a_{21})/((b_{11}-b_{21})w_1+b_{21}),$$

$$\lambda_4(w_1)=((a_{12}-a_{22})w_1+a_{22})/((b_{12}-b_{22})w_1+b_{22}).$$

Исследуем функцию $\lambda_1(x_1)$ на возрастание и убывание при $x_1 \in [0;1]$.

$$d\lambda_1 / dx_1 = \left(\frac{(a_{11} - a_{12})x_1 + a_{12}}{(b_{11} - b_{12})x_1 + b_{12}} \right)' = \frac{a_{11}b_{12} - a_{12}b_{11}}{((b_{11} - b_{12})x_1 + b_{12})^2}.$$

Таким образом, при $a_{11}/b_{11} > a_{12}/b_{12}$ функция $\lambda_1(x_1)$ является возрастающей; при $a_{11}/b_{11} < a_{12}/b_{11}$ - убывающей; а при $a_{12}/b_{12} = a_{11}/b_{11}$ - константой. Следовательно, в случае $a_{11}/b_{11} > a_{12}/b_{12}$ функция $\lambda_1(x_1)$ достигнет максимума $\max \lambda_1(x_1) = a_{11}/b_{11}$ при $x_1 = 1$ и минимума $\min \lambda_1(x_1) = a_{12}/b_{12}$ при $x_1 = 0$. Если же $a_{11}/b_{11} < a_{12}/b_{11}$, то функция $\lambda_1(x_1)$ достигнет максимума $\max \lambda_1(x_1) = a_{12}/b_{12}$ при $x_1 = 0$ и минимума $\min \lambda_1(x_1) = a_{11}/b_{11}$ при $x_1 = 1$.

Аналогично можно исследовать функции $\lambda_2(x_1)$, $\lambda_3(w_1)$ и $\lambda_4(w_1)$. Так, если $a_{11}/b_{11} > a_{21}/b_{21}$, то функция $\lambda_3(w_1)$ достигнет максимума $\max \lambda_3(w_1) = a_{11}/b_{11}$ при $w_1 = 1$ и минимума $\min \lambda_3(w_1) = a_{12}/b_{12}$ при $w_1 = 0$. Если $a_{11}/b_{11} < a_{21}/b_{21}$, то функция $\lambda_4(w_1)$ достигнет максимума $\max \lambda_4(w_1) = a_{21}/b_{21}$ при $w_1 = 0$ и минимума $\min \lambda_4(w_1) = a_{11}/b_{11}$ при $w_1 = 0$.

Из (2.3.2) следует, что если функции $\lambda_3(w_1)$ и $\lambda_4(w_1)$ являются одновременно возрастающими или убывающими при $w_1 \in [0;1]$, то

$$\bar{\lambda} = \max_{w_1 \in \{0,1\}} \min \{ \lambda_3(w_1); \lambda_4(w_1) \}.$$

В случае одновременного возрастания $\bar{\lambda} = \min\{a_{11}/b_{11}; a_{12}/b_{12}\}$, $w_1^o = 1$,

а в случае убывания - $\bar{\lambda} = \min\{a_{21}/b_{21}; a_{22}/b_{22}\}$, $w_1^o = 0$, где $w_1^o = \arg\max_{w_1} \bar{\lambda}$.

Если же одна из функций убывает, а другая возрастает, то найдем

$$w_1^*: \quad \lambda_3(w_1^*) = \lambda_4(w_1^*).$$

В этом случае

$$\bar{\lambda} = \begin{cases} \lambda_3(w_1^*) = \lambda_4(w_1^*),\ w^0 = w_1^*, & \text{при} \quad w_1^* \in [0,1]; \\ \min\{\lambda_3(1), \lambda_4(1)\},\ w^0 = 1, & \text{при} \quad w_1^* > 1; \\ \min\{\lambda_3(0), \lambda_4(0)\},\ w^0 = 0, & \text{при} \quad w_1^* < 0; \end{cases}$$

Итак, найдено число Фробениуса $\overline{\lambda}$ и двойственный вектор Фробениуса $\overline{w} = \left(w_1^o, 1 - w_1^o \right)$. При известном значении числа Фробениуса $\overline{\lambda}$ нахождение вектора Фробениуса $\overline{x}$ сводится к решению системы линейных неравенств с одной переменной, т.е. представляет тривиальную задачу. Однако, аналитическое описание ее решения в общем случае является достаточно громоздким.

Рассмотрим аналитическое решение задачи модели Неймана со следующими числовыми данными:

$$A = \begin{pmatrix} 3 & 6 \\ 4 & 7 \end{pmatrix} \text{ и } B = \begin{pmatrix} 4 & 6 \\ 5 & 10 \end{pmatrix}.$$

Т.к. все элементы матрицы B являются ненулевыми и $b_{11} \neq b_{12}$, $b_{11} \neq b_{21}$, $b_{12} \neq b_{22}$, $b_{21} \neq b_{22}$, то рассчитаем функции

$$\lambda_1(x_1) = (-3x_1 + 6)/(-2x_1 + 6),\ \lambda_2(x_1) = (-3x_1 + 7)/(-5x_1 + 10),$$

$$\lambda_3(w_1) = (-w_1 + 4)/(-w_1 + 5),\ \lambda_4(w_1) = (-w_1 + 7)/(-4w_1 + 10).$$

В нашем случае $a_{11}/b_{11} < a_{21}/b_{21}$ ($3/4 < 4/5$), поэтому $\max \lambda_3(w_1) = 4/5$ при $w_1 = 0$ и $\min \lambda_3(w_1) = 3/4$ при $w_1 = 1$. Кроме того, $a_{12}/b_{12} > a_{22}/b_{22}$ ($1 > 7/10$). Откуда $\max \lambda_4(w_1) = 1$ при $w_1 = 1$ и $\min \lambda_4(w_1) = 7/10$ при $w_1 = 0$.

При $w_1 \in [0;1]$ функция $\lambda_3(w_1)$ является убывающей, а $\lambda_4(w_1)$ - возрастающей. Ищем точку пересечения, решив уравнение

$$(-w_1 + 4)/(-w_1 + 5) = (-w_1 + 7)/(-4w_1 + 10).$$

Решением этого уравнения (с учетом $w_1 \in [0;1]$) будет $\overline{w}_1 = (7 - \sqrt{34})/3 \approx 0.39$.

Тогда число Фробениуса $\overline{\lambda} = \dfrac{5 + \sqrt{34}}{8 + \sqrt{34}} \approx 0.783$, $\overline{w}_2 = 1 - \overline{w}_1 = (\sqrt{34} - 4)/3 \approx 0.61$.

Чтобы найти вектор Фробениуса $\overline{x}$, подставим $\overline{\lambda}$ в первые два ограничения задачи. Решением этих двух неравенств будет $\overline{x}_1 = \dfrac{18}{14 + \sqrt{34}} \approx 0.908$, откуда $\overline{x}_2 = \dfrac{\sqrt{34} - 4}{14 + \sqrt{34}} \approx 0.092$. Рассчитаем также число Неймана $\underline{\lambda}$. Т.к. $a_{11}/b_{11} < a_{12}/b_{12}$ ($3/4 < 1$), $\max \lambda_1(x_1) = 1$ при $x_1 = 0$ и $\min \lambda_1(x_1) = 3/4$ при $x_1 = 1$. Кроме того,

$a_{21}/b_{21} > a_{22}/b_{22}$ $(4/5 > 7/10)$. Откуда $\max\lambda_2(x_1) = 4/5$ при $x_1 = 1$ и $\min\lambda_2(x_1) = 7/10$ при $x_1 = 0$. При $x_1 \in [0;1]$ функция $\lambda_1(x_1)$ является убывающей, а $\lambda_2(x_1)$ - возрастающей. Ищем точку пересечения, решив уравнение

$$(-3x_1 + 6)/(-2x_1 + 6) = (-3x_1 + 7)/(-5x_1 + 10).$$

Решением этого уравнения (с учетом $x_1 \in [0;1]$) будет $\underline{x}_1 = \dfrac{18}{14+\sqrt{34}} \approx 0.908$,

$\underline{x}_2 = \dfrac{\sqrt{34}-4}{14+\sqrt{34}} \approx 0.092$, при этом оказалось, что $\underline{\lambda} = \overline{\lambda} = \dfrac{5+\sqrt{34}}{8+\sqrt{34}} \approx 0.783$.

Рассмотрим аналитическое решение задачи модели Неймана со следующими числовыми данными:

$$A = \begin{pmatrix} 4 & 0 \\ 2 & 4 \end{pmatrix} \text{ и } B = \begin{pmatrix} 9 & 0 \\ 4 & 8 \end{pmatrix}.$$

Для нахождения лучей Неймана необходимо решить следующую систему неравенств:

$$\lambda \to \max$$
$$\begin{cases} (4-9\lambda)x_1 \le 0, (2-4\lambda)x_1 + (4-8\lambda)x_2 \le 0, \\ (4-9\lambda)w_1 + (2-4\lambda)w_2 \ge 0, (4-8\lambda)w_2 \ge 0, \\ x_1 + x_2 = 1, w_1 + w_2 = 1, \lambda, x, w \ge 0. \end{cases}$$

Выразим x_1 и w_1 через x_2 и w_2 соответственно:

$$\lambda \to \max$$
$$\begin{cases} \lambda x_1 \ge 4/9 x_1, \lambda \ge 1/2, \\ \lambda \le (2w_1 + 2)/(5w_1 + 4), \\ \lambda(1 - w_1) \le (1 - w_1)/2, \\ 0 \le x_1 \le 1, 0 \le w_1 \le 1, \lambda \ge 0. \end{cases}$$

Функция $\lambda_3(w_1) = (2w_1 + 2)/(5w_1 + 4)$ будет иметь максимум $\max\lambda_3(w_1) = 1/2$ при $w_1 = 0$ и минимум $\min\lambda_3(w_1) = 4/9$ при $w_1 = 1$.

Таким образом, число Фробениуса для данной модели равно $\overline{\lambda} = 1/2$. Подставив его в первые два ограничения, получим, что $\overline{x}_1 \in [0;1]$, $\overline{x}_2 = 1 - \overline{x}_1$.

Согласно второму ограничению $\lambda \geq 1/2$, поэтому первое и второе ограничение системы будут одновременно выполняться при $\underline{x}_1 \in [0;1]$, $\underline{x}_2 = 1 - \underline{x}_1$. Откуда число Неймана совпадает с числом Фробениуса $\underline{\lambda} = \overline{\lambda} = 1/2$.

Из приведенных примеров видно, что аналитическое решение задачи плохо обозримо даже для задач размерности *m,n*=2. Поэтому в дальнейшем ограничимся рассмотрением численных методов решении задачи (2.2.2).

2.4. Применение коммерческого программного обеспечения для решения задачи

Применение стандартных пакетов программного обеспечения, например, `MS Excel,` не всегда эффективно. Рассмотрим численные примеры, иллюстрирующие эту проблему. В частности, расчет для модели Неймана с матрицами

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ и } B = \begin{pmatrix} 1.5 & 0 \\ 0 & 2 \end{pmatrix}$$

не дает правильного решения.

Действительно, легко показать, что допустимое значение для λ, удовлетворяющее всем ограничениям системы (рассчитанное аналитически), принадлежит диапазону $[0.5; 2/3]$ (число Неймана равно 0.5, а число Фробениуса - $2/3$). Стандартные программные средства при начальных значениях для поиска оптимального решения $\lambda_0 = 0$, $x_1^0 = x_2^0 = w_1^0 = w_2^0 = 1/2$, находят только допустимое решение (число Неймана) $\underline{\lambda} = 0.5$.

Легко проверить, что если числа Неймана и Фробениуса не совпадают (такое возможно, например, при большом количестве нулевых элементов в матрицах A и B), то стандартные пакеты не находят оптимальное значение числа Фробениуса. Это объясняется тем, что согласно известной теореме [2], модель Неймана может иметь несколько положений равновесия, отличающихся набором ненулевых координат соответствующих им векторов x и w. Стандартные пакеты находят допустимое решение исходя из начальных значений переменных для поиска, затем выполняют проверку чувствительности данного решения. Такая проверка не будет эффективной из-за смены набора ненулевых координат в векторах x и w для большего значения λ. Кроме того, качество решения будет сильно зависеть от начальных условий поиска.

В стандартных пакетах также существуют проблемы точности поиска и большой размерности. Например, для модели Неймана:

$$A=\begin{pmatrix} 0 & 0 & 0 & 2 & 89 & 73 & 3 & 72 \\ 0 & 0 & 0 & 0 & 56 & 84 & 39 & 56 \\ 0 & 0 & 0 & 52 & 47 & 21 & 12 & 42 \\ 0 & 0 & 0 & 21 & 31 & 19 & 98 & 60 \\ 89 & 11 & 68 & 17 & 64 & 31 & 42 & 28 \\ 74 & 92 & 47 & 30 & 62 & 0 & 0 & 0 \\ 45 & 86 & 54 & 85 & 72 & 0 & 0 & 0 \\ 3 & 0 & 56 & 99 & 49 & 0 & 0 & 0 \\ 91 & 22 & 67 & 7 & 83 & 0 & 0 & 0 \end{pmatrix},$$

$$B=\begin{pmatrix} 0 & 0 & 0 & 0 & 115 & 97 & 0 & 80 \\ 0 & 0 & 0 & 6 & 46 & 70 & 44 & 52 \\ 0 & 0 & 0 & 42 & 54 & 28 & 0 & 24 \\ 0 & 0 & 0 & 0 & 40 & 22 & 90 & 47 \\ 116 & 10 & 83 & 40 & 71 & 47 & 39 & 0 \\ 90 & 120 & 50 & 20 & 79 & 0 & 0 & 0 \\ 60 & 100 & 63 & 101 & 49 & 0 & 0 & 0 \\ 0 & 18 & 61 & 86 & 37 & 0 & 0 & 0 \\ 100 & 48 & 52 & 37 & 79 & 0 & 0 & 0 \end{pmatrix}$$

поиск решения `MS Excel` не может найти оптимальное решение (см. рисунок 2.4.1).

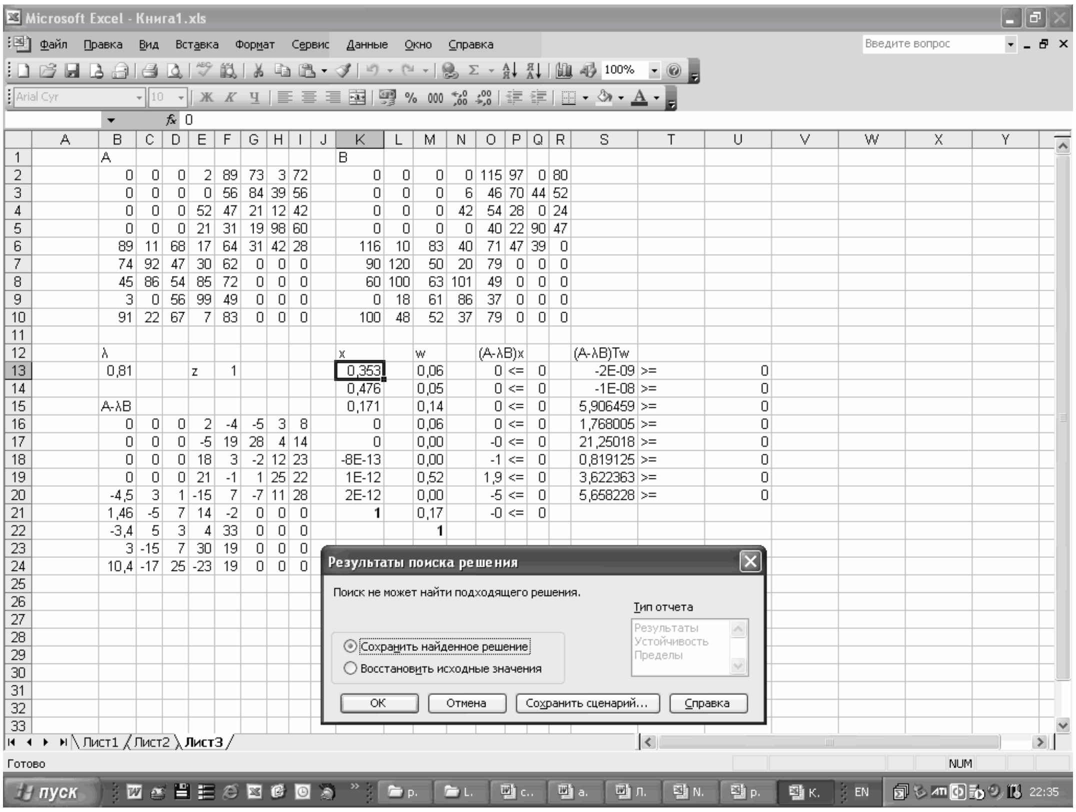

Рисунок 2.4.1. Пример расчета посредством поиска решения `MS Excel`

Для более тщательного анализа возможностей применения коммерческого программного обеспечения для решения задачи проведем вычислительный эксперимент. Варианты задач будем генерировать с помощью разработанного автором работы `Windows`-приложения [15, 56], листинг которого представлен в приложении А. Рассмотрим далее алгоритм генерации файлов.

Шаг 1. Указываются размер задачи (n и m), число файлов для генерации `num_files`, максимальное значение для элементов матрицы A для генератора случайных чисел `a_max`, предполагаемый процент нулевых компонент `proc`, оценка для числа Фробениуса $\tilde{\lambda}$. Переход к шагу 2.

Шаг 2. Инициализируем счетчик числа созданных файлов, числа строк и столбцов для матриц A и B. Переход к шагу 3.

Шаг 3. Инициализация матрицы A. Переход к шагу 3.1.

Шаг 3.1. Генерируем случайное целое число в интервале от `0` до `a_max`. Переход к шагу 3.2.

Шаг 3.2. Проверяем, является ли полученное число меньше `proc*a_max/100`. Если является, то переходим к шагу 3.3, в противном случае переходим к шагу 3.4.

Шаг 3.3. Проверяем, является ли текущий элемент матрицы A последним в строке или столбце. Если да, то переход к шагу 3.3.1, в противном случае к шагу 3.3.2.

Шаг 3.3.1. Проверяем, являются ли все элементы в данной строке (столбце) нулевыми. Если да, то запись сгенерированного случайного числа в текущий элемент матрицы A и переход к шагу 3.5. В противном случае переход к шагу 3.3.2.

Шаг 3.3.2. Запись нулевого значения в элемент матрицы A. Переход к шагу 3.5

Шаг 3.4. Запись сгенерированного случайного числа в текущий элемент матрицы A. Переход к шагу 3.5.

Шаг 3.5. Проверяем, равен ли номер текущего столбца m. Если да, то переход к шагу 3.6. В противном случае номер столбца увеличиваем на единицу, переход к шагу 3.1.

Шаг 3.6. Проверяем, равен ли номер текущей строки n. Если нет, то номер строки увеличиваем на единицу и переходим к шагу 3.1. В противном случае переход к шагу 4.

Шаг 4. Инициализация матрицы B. Переход к шагу 4.1.

Шаг 4.1. Инициализация массива коэффициентов β. Переход к шагу 4.1.1.

Шаг 4.1.1. Генерируем случайное целое число в интервале от `0` до `100`. Переход к шагу 4.1.2.

Шаг 4.1.2. Проверяем, является ли полученное число меньше `proc`. Если является, то переходим к шагу 4.1.3, в противном случае переходим к шагу 4.1.5.

Шаг 4.1.3. Проверяем, является ли текущий элемент массива β последним. Если да, то переход к шагу 4.1.4, в противном случае к шагу 4.1.5.

Шаг 4.1.4. Проверяем, являются ли все элементы массива β нулевыми. Если да, то запись единицы в текущий элемент массива β и переход к шагу 4.2. В противном случае переход к шагу 4.1.5.

Шаг 4.1.5. Запись нулевого значения в элемент массива β. Переход к шагу 4.1.6.

Шаг 4.1.6. Проверяем, равен ли m номер текущего элемента массива β. Если да, то переход к шагу 4.2. В противном случае номер элемента увеличиваем на единицу.

Шаг 4.2. Запуск расчета строки матрицы B. Переход к шагу 4.2.1.

Шаг 4.2.1. Расчет значения текущего элемента матрицы B по формуле

$b_{ij} = \beta_j / \sum_{j=1}^{m} \beta_j * \frac{\sum_{k=1}^{m} a_{ik}}{\tilde{\lambda}}$. Переход к шагу 4.2.2.

Шаг 4.2.2. Проверяем, равен ли номер текущего столбца m. Если да, то переход к шагу 4.2.3. В противном случае номер столбца увеличиваем на единицу и переходим к шагу 4.2.1.

Шаг 4.2.3. Проверяем, равен ли номер текущей строки n. Если нет, то номер строки увеличиваем на единицу и переходим к шагу 4.1. В противном случае переход к шагу 5.

Шаг 5. Запись исходных данных в новый файл. Счетчик файлов увеличиваем на единицу. Переход к шагу 6.

Шаг 6. Проверяем число созданных файлов. Если оно меньше `num_files`, то переход к шагу 3. В противном случае выход из программы.

Результаты использования надстройки «Поиск решения» `MS Excel`:

1. Размер задачи: $n = 4$, $m = 5$, число задач - 100, число решенных задач - 85.

2. Размер задачи: $n = 5$, $m = 6$, число задач - 100, число решенных задач - 84.

3. Размер задачи: $n = 8$, $m = 6$, число задач - 100, число решенных задач - 75.

4. Размер задачи: $n = 10$, $m = 10$, число задач - 100, число решенных задач - 70.

При решении задач большей размерности число решенных задач резко сокращается.

Таким образом, для вычисления луча Неймана целесообразно использовать специально разработанное программное обеспечение, которое было разработано в ходе выполнения диссертационной работы [42]. В его основе лежат теоретические результаты, полученные автором [31, 33].

2.5. Выводы и результаты

Исходя из экономической постановки, рассмотренной ранее, в данной главе предложена математическая модель оптимизации бюджет продаж:

1) рассмотрены входные и выходные параметры модели;

2) сформулировано основное ограничение (внутрифирменный баланс);

3) введен критерий оптимальности (максимизация рентабельности продаж).

Также проведен анализ полученной модели:

1) предложено понятие продуктивности для модели бюджетирования;

2) показана аналогия модели бюджетирования с моделью фоннеймановского типа;

3) рассмотрена экономическая интерпретация ряда величин используемых в модели.

4) аналитическое решение при $m, n = 2$ является многовариантным, поэтому в общем случае целесообразно численное исследование модели;

5) использование коммерческих программ для непосредственного решения задачи приводит к неправильным результатам.

Из изложенного следует необходимость разработки численных методов и алгоритмов анализа модели. Этому вопросу посвящена следующая глава.

3. Методы и алгоритмы определения параметров модели

3.1. Определение параметров равновесия модели Неймана

Анализ модели Неймана предполагает нахождение для заданных ($m\times n$) матриц A и B с неотрицательными элементами решений (λ, x, w) системы билинейных неравенств и уравнений

$$(A-\lambda B)x \le 0, \left(x, e^m\right) = 1,\ x \ge 0\,, \tag{3.1.1}$$

$$(A-\lambda B)^T w \ge 0, \left(w, e^n\right) = 1,\ w \ge 0\,. \tag{3.1.2}$$

Экстремальные допустимые значения λ (т.е. число Неймана и число Фробениуса) могут быть найдены с помощью решения задач билинейной оптимизации

$$\underline{\lambda} = \min\left\{\lambda : (A-\lambda B)x \le 0, \left(x, e^m\right) = 1,\ x \ge 0\right\}, \tag{3.1.3}$$

$$\overline{\lambda} = \max\left\{\lambda : (A-\lambda B)^T w \ge 0, \left(w, e^n\right) = 1,\ w \ge 0\right\}. \tag{3.1.4}$$

При фиксированном значении $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ соответствующие значения (x,w) могу быть найдены как решения систем линейных уравнений и неравенств (3.1.1) и (3.1.2).

Известно, что у неразложимой модели Неймана числа Неймана и Фробениуса совпадают [2]. В модели бюджетирования имеет место $a_{ij} = 0 \Leftrightarrow b_{ij} = 0$. Как правило, в матрице затрат A и соответственно в матрице выручки B имеются нулевые элементы, т.е. имеется изолированная пара. Модель неймановского типа с изолированной парой является разложимой. В разложимых моделях, в отличие от неразложимых, числа Неймана и Фробениуса могут не совпадать.

Известно, что задача Неймана с матрицами A и B без нулевых строк и столбцов имеет решение. Конструктивное доказательство данного факта, содержащее алгоритм построения решения приведен в [2].

Предложенный в [2] алгоритм состоит в нахождении корней монотонной функции

$$u(\lambda) = \min_{x:\left(x,e^m\right)=1,\, x\ge 0} \quad \max_{i=1,2,\ldots n} \sum_{j=1}^{m} \left(a_{ij} - \lambda b_{ij}\right) x_j \tag{3.1.5}$$

или

$$v(\lambda) = \max_{w:\left(w,e^n\right)=1,\, w\ge 0} \quad \min_{j=1,2,\ldots m} \sum_{i=1}^{n} \left(a_{ij} - \lambda b_{ij}\right) w_i\,. \tag{3.1.6}$$

При фиксированном λ значения функций $u(\lambda)$ (3.1.5) и $v(\lambda)$ (3.1.6) равны значениям следующих взаимно двойственных задач линейного программирования

$$\min\left\{u : \left(A - \lambda B\right) x \le u, \left(x, e^m\right) = 1, x \ge 0\right\}, \tag{3.1.7}$$

$$\max\left\{v : \left(A - \lambda B\right)^T w \ge v, \left(w, e^n\right) = 1, w \ge 0\right\}. \tag{3.1.8}$$

Таким образом, упомянутый алгоритм требует решения последовательности задач линейного программирования (3.1.7) и/или (3.1.8). Легко заметить, что при значениях λ близких к искомым, т.е. когда $u(\lambda), v(\lambda) \to 0$, соответствующие задачи становятся сильно вырожденными, что влечет невозможность их решения с помощью традиционных средств, использующих вычисления с плавающей точкой.

Утверждение 3.1.1. *Имеют место равенства*

$$\overline{\lambda} = \max_{w:(w,e^n)=1, w\ge 0} \quad \min_{j=1,2,\ldots,m} \frac{\sum_{i=1}^{n} a_{ij} w_i}{\sum_{i=1}^{n} b_{ij} w_i}, \quad \underline{\lambda} = \min_{x:(x,e^m)=1, x\ge 0} \quad \max_{i=1,2,\ldots,n} \frac{\sum_{j=1}^{m} a_{ij} x_j}{\sum_{j=1}^{m} b_{ij} x_j}. \tag{3.1.9}$$

Доказательство. В соответствии с выражением (3.1.3) справедлива следующая последовательность импликаций

$$\left(\left(A - \lambda B\right) x \le 0, \left(x, e^m\right) = 1, x \ge 0\right) \Leftrightarrow$$

$$\Leftrightarrow\left(\left(\forall i=1,2,\ldots,n\right)\left(\sum_{j=1}^{m}a_{ij}x_j\le\lambda\sum_{j=1}^{m}b_{ij}x_j\right),\left(x,e^m\right)=1,\ x\ge 0\right)\Leftrightarrow$$

$$\Leftrightarrow\left(\left(\forall i=1,2,\ldots,n\right)\left(\lambda\ge\frac{\sum_{j=1}^{m}a_{ij}x_j}{\sum_{j=1}^{m}b_{ij}x_j}\right),\left(x,e^m\right)=1,\ x\ge 0\right)\Rightarrow$$

$$\Rightarrow\left(\underline{\lambda}=\min_{x:\left(x,e^m\right)=1,\ x\ge 0}\lambda(x),\quad \lambda(x)=\max_{i=1,2,\ldots,n}\frac{\sum_{j=1}^{m}a_{ij}x_j}{\sum_{j=1}^{m}b_{ij}x_j}\right), \tag{3.1.10}$$

в которой через $\lambda(x)$ обозначено допустимое значение λ при фиксированном значении x.

Аналогично доказывается второе равенство

$$\left(\left(A-\lambda B\right)^T w\ge 0,\left(w,e^n\right)=1,\ x\ge 0\right)\Rightarrow$$

$$\Rightarrow\left(\left(\forall j=1,2,\ldots,m\right)\left(\sum_{i=1}^{n}a_{ij}w_i\ge\lambda\sum_{j=1}^{n}b_{ij}w_i\right),\left(w,e^n\right)=1,\ w\ge 0\right)\Rightarrow$$

$$\Rightarrow\left(\left(\forall i=1,2,\ldots,m\right)\left(\lambda\le\frac{\sum_{i=1}^{n}a_{ij}w_i}{\sum_{i=1}^{n}b_{ij}w_i}\right),\left(w,e^n\right)=1,\ w\ge 0\right)\Rightarrow$$

$$\Rightarrow\left(\overline{\lambda}=\max_{w:\left(w,e^n\right)=1,\ w\ge 0}\lambda(w),\quad \lambda(w)=\min_{j=1,2,\ldots,m}\frac{\sum_{i=1}^{n}a_{ij}w_i}{\sum_{i=1}^{n}b_{ij}w_i}\right). \tag{3.1.11}$$

Утверждение 3.1.1 доказано ■

Теорема 3.1.1. *Пусть* (A,B) *- модель Неймана, в которой:*

1) $(\forall i):(\sum_{j=1}^{m}b_{ij}>0)$ и $(\forall j):(\sum_{i=1}^{n}b_{ij}>0)$;

2) $\underline{\lambda}^{>} = \max\limits_{i=1,2,\ldots,n} \dfrac{\sum\limits_{j=1}^{m} a_{ij}}{\sum\limits_{j=1}^{m} b_{ij}}$ и $\overline{\lambda}^{<} = \min\limits_{j=1,2,\ldots,m} \dfrac{\sum\limits_{i=1}^{n} a_{ij}}{\sum\limits_{i=1}^{n} b_{ij}}$;

3) $\underline{\lambda}^{>} \le \overline{\lambda}^{<}$.

Тогда в модели Неймана (A,B) *существуют положения равновесия* (λ, x, w), *в которых* $\lambda \in [\underline{\lambda}^{>}, \overline{\lambda}^{<}]$.

Доказательство. Полагая $x_j = \dfrac{1}{m}$ ($j = 1,2,\ldots,m$), с учетом (3.1.10) имеем верхнюю оценку для нижней границы $\underline{\lambda}$

$$\underline{\lambda}^{>} = \max_{i=1,2,\ldots,n} \frac{\sum\limits_{j=1}^{m} a_{ij}}{\sum\limits_{j=1}^{m} b_{ij}} \ge \underline{\lambda}. \tag{3.1.12}$$

Полагая $w_i = \dfrac{1}{n}$ ($i = 1,2,\ldots,n$), с учетом (3.1.11) имеем нижнюю оценку для верхней границы $\overline{\lambda}$

$$\overline{\lambda}^{<} = \min_{j=1,2,\ldots,m} \frac{\sum\limits_{i=1}^{n} a_{ij}}{\sum\limits_{i=1}^{n} b_{ij}} \le \overline{\lambda}. \tag{3.1.13}$$

В соответствии с условием теоремы $\underline{\lambda}^{>} \le \overline{\lambda}^{<}$, поэтому $\underline{\lambda}^{>}, \overline{\lambda}^{<} \in \left[\underline{\lambda}, \overline{\lambda}\right]$. Теорема доказана ■

Не трудно показать, что из условия $\underline{\lambda}^{>} \le \overline{\lambda}^{<}$ следует $\underline{\lambda}^{>} = \overline{\lambda}^{<}$. Как показывает практика, несмотря на то, что условия данной теоремы просты, они редко выполнимы. Рассмотрим более конструктивные подходы к определению параметров равновесия.

Случай $(B > 0)$

Поскольку

$$\left\{x:\left(x,e^m\right)=1,\,x\ge 0\right\}\supset\left\{x:\left(x,e^m\right)=1,\,x_j\in\{0,1\},\,j=1,2,\ldots,m\right\},$$

то

$$\underline{\lambda}=\min_{x:\left(x,e^m\right)=1,\,x\ge 0}\ \max_{i=1,2,\ldots,n}\frac{\sum_{j=1}^{m}a_{ij}x_j}{\sum_{j=1}^{m}b_{ij}x_j}\le\min_{j=1,2,\ldots,m}\ \max_{i=1,2,\ldots,n}\frac{a_{ij}}{b_{ij}}$$

Аналогично, из включения

$$\left\{w:\left(w,e^n\right)=1,\,w\ge 0\right\}\supset\left\{w:\left(w,e^n\right)=1,\,w_i\in\{0,1\},\,i=1,2,\ldots,n\right\}$$

следует

$$\overline{\lambda}=\max_{w:\left(w,e^n\right)=1,\,w\ge 0}\ \min_{j=1,2,\ldots,m}\frac{\sum_{i=1}^{n}a_{ij}w_i}{\sum_{i=1}^{n}b_{ij}w_i}\ge\max_{i=1,2,\ldots,n}\ \min_{j=1,2,\ldots,m}\frac{a_{ij}}{b_{ij}}.$$

Утверждение 3.1.2.

$$\left(\forall j=1,2,\ldots m\right)\left(\max_{w:\left(w,e^n\right)=1,\,w\ge 0}\frac{\sum_{i=1}^{n}a_{ij}w_i}{\sum_{i=1}^{n}b_{ij}w_i}\ge\max_{w:\left(w,e^n\right)=1,\,w\ge 0}\sum_{i=1}^{n}\frac{a_{ij}}{b_{ij}}w_i\right)\quad(3.1.14)$$

$$\left(\forall\, i=1,2,\ldots,n\right)\left(\min_{x:\left(x,e^m\right)=1,\,x\ge 0}\frac{\sum_{j=1}^{m}a_{ij}x_j}{\sum_{j=1}^{n}b_{ij}x_j}\le\min_{x:\left(x,e^m\right)=1,\,x\ge 0}\sum_{j=1}^{n}\frac{a_{ij}}{b_{ij}}x_j\right)\quad(3.1.15)$$

Доказательство. Для доказательства неравенства (3.1.14) зафиксируем произвольное $j=1,2,\ldots,m$. Легко заметить, что в правой части рассматриваемого неравенства максимум достигается на векторе

$$\arg\max_{\mathbf{w}:\left(\mathbf{w},\mathbf{e}^n\right)=1,\,\mathbf{w}\ge 0}\sum_{i=1}^{n}\frac{a_{ij}}{b_{ij}}w_i=\mathbf{w}^*=\left\{w_k^*=\delta_k^{i^*},\,i^*=\arg\max_{i=1,2,\ldots,n}\left(\frac{a_{ij}}{b_{ij}}\right),\,k=1,2,\ldots,n\right\},$$

где δ_k^i – символ Кронекера. Справедливость неравенства (3.1.14) следует из того, что

$$\mathbf{w}^* \in \left\{ \mathbf{w} : \left(\mathbf{w}, \mathbf{e}^n \right) = 1, \mathbf{w} \geq 0 \right\}; \quad \frac{\sum_{i=1}^{n} a_{ij} w_i^*}{\sum_{i=1}^{n} b_{ij} w_i^*} = \frac{a_{i^* j}}{b_{i^* j}} = \max_{\mathbf{w}:\left(\mathbf{w},\mathbf{e}^n\right)=1,\, \mathbf{w}\geq 0} \sum_{i=1}^{n} \frac{a_{ij}}{b_{ij}} w_i .$$

Для доказательства неравенства (3.1.15) зафиксируем произвольное значение $i = 1, 2, \ldots, n$. В правой части рассматриваемого неравенства минимум достигается на векторе

$$\arg \min_{\mathbf{x}:\left(\mathbf{x},\mathbf{e}^m\right)=1,\, \mathbf{x}\geq 0} \sum_{j=1}^{m} \frac{a_{ij}}{b_{ij}} x_j = \mathbf{x}^* = \left\{ x_k^* = \delta_k^{j^*},\ j^* = \arg \min_{j=1,2,\ldots,m} \left(\frac{a_{ij}}{b_{ij}} \right), k = 1, 2, \ldots, m \right\},$$

Справедливость неравенства (3.1.15) следует из того, что

$$\mathbf{x}^* \in \left\{ \mathbf{x} : \left(\mathbf{x}, \mathbf{e}^m \right) = 1, \mathbf{x} \geq 0 \right\}; \quad \frac{\sum_{j=1}^{m} a_{ij} x_j^*}{\sum_{j=1}^{m} b_{ij} x_j^*} = \frac{a_{ij^*}}{b_{ij^*}} = \min_{\mathbf{x}:\left(\mathbf{x},\mathbf{e}^m\right)=1,\, \mathbf{x}\geq 0} \sum_{j=1}^{m} \frac{a_{ij}}{b_{ij}} x_j .$$

Утверждение 3.1.2 доказано ■

Утверждение 3.1.3.

$$\underline{\lambda} \leq \min_{x:\left(x,e^m\right)=1,\, x\geq 0} \ \max_{w:\left(w,e^n\right)=1,\, w\geq 0} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{a_{ij}}{b_{ij}} w_i x_j \qquad (3.1.16)$$

$$\overline{\lambda} \geq \max_{w:\left(w,e^n\right)=1,\, w\geq 0} \ \min_{x:\left(x,e^m\right)=1,\, x\geq 0} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{a_{ij}}{b_{ij}} w_i x_j \qquad (3.1.17)$$

Доказательство. Из утверждений 3.2.1 и 3.2.2 следует

$$\underline{\lambda} = \min_{x:\left(x,e^m\right)=1,\, x\geq 0} \ \max_{i=1,2,\ldots,n} \frac{\sum_{j=1}^{m} a_{ij} x_j}{\sum_{j=1}^{m} b_{ij} x_j} \leq \min_{x:\left(x,e^m\right)=1,\, x\geq 0} \ \max_{i=1,2,\ldots,n} \sum_{j=1}^{m} \frac{a_{ij}}{b_{ij}} x_j \qquad (3.1.18)$$

Поскольку

$$\left\{ w : \left(w, e^n \right) = 1,\ w \geq 0 \right\} \supset \left\{ w : \left(w, e^n \right) = 1,\ w_i \in \{0, 1\},\ i = 1, 2, \ldots, n \right\},$$

то

$$\min_{x:\left(x,e^m\right)=1,\ x\ge 0}\ \max_{i=1,2,\ldots,n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i \le \min_{x:\left(x,e^m\right)=1,\ x\ge 0}\ \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i x_j \quad (3.1.19)$$

Из сопоставления неравенств (3.1.18) и (3.1.19) следует справедливость неравенства (3.1.16).

Аналогично для значения $\overline{\lambda}$ в соответствии с (3.1.9) и (3.1.15) имеем

$$\overline{\lambda} = \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\ \min_{j=1,2,\ldots,m}\frac{\sum_{i=1}^{n}a_{ij}w_i}{\sum_{i=1}^{n}b_{ij}w_i} \ge \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\ \min_{j=1,2,\ldots,m}\sum_{i=1}^{n}\frac{a_{ij}}{b_{ij}}w_i \quad (3.1.20)$$

Из включения

$$\left\{x:\left(x,e^m\right)=1,\ x\ge 0\right\}\supset\left\{x:\left(x,e^m\right)=1,\ x_j\in\{0,1\},\ j=1,2,\ldots,m\right\}$$

следует

$$\max_{w:\left(w,e^n\right)=1,\ w\ge 0}\ \min_{j=1,2,\ldots,m}\sum_{i=1}^{n}\frac{a_{ij}}{b_{ij}}w_i \ge \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\ \min_{x:\left(x,e^m\right)=1,\ x\ge 0}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i x_j \quad (3.1.21)$$

Сопоставление неравенств (3.1.20) и (3.1.21) доказывают справедливость неравенства (3.1.17).

Утверждение 3.1.3 доказано ■

Рассмотрим матричную игру G_C с матрицей $C = [a_{ij}/b_{ij}]$, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,m$ (стратегиями первого игрока являются столбцы этой матрицы, а второго – строки). В соответствии с теоремой Неймана – Моргенштерна для матричных игр [9, 44] имеем

$$\begin{aligned}&\min_{x:\left(x,e^m\right)=1,\ x\ge 0}\ \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i x_j =\\ &= \max_{w:\left(w,e^n\right)=1,\ w\ge 0}\ \min_{x:\left(x,e^m\right)=1,\ x\ge 0}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i x_j = \qquad (3.1.22)\\ &= \lambda^* = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{a_{ij}}{b_{ij}}w_i^* x_i^*,\end{aligned}$$

где (w^*, x^*) − оптимальные смешанные стратегии игры G_C, λ^* − цена игры.

Сопоставление (3.1.16), (3.1.17) и (3.1.22) дает $\lambda^* \in \left[\underline{\lambda}, \overline{\lambda}\right]$, т.е. при значении параметра λ= λ* обе системы (3.1.1) и (3.1.2) совместны.

Итог изложенному сформулируем в виде теоремы.

Теорема 3.1.2. *Пусть* (A, B) − *модель Неймана; все элементы матрицы* B *являются ненулевыми;* G_C − *матричная игра с матрицей выигрышей* $C = [a_{ij}/b_{ij}]$, $i = 1,2,\dots,n$, $j = 1,2,\dots,m$; λ^* − *цена игры* G_C. *Тогда в модели Неймана* (A,B) *существует положение равновесия* (λ,x,w), *в котором* $\lambda = \lambda^*$.

Очевидно, что оптимальные смешанные стратегии игры G_C, т.е. векторы (w^*, x^*), в общем случае не будут решениями систем (3.1.1) и (3.1.2), соответствующими значению $\lambda = \lambda^*$. Если же решением игры G_C являются чистые стратегии, т.е. векторы

$$w^* = \left\{ w_i = \delta_i^{i_0},\, i = 1,2,\dots,n \right\}, \quad x^* = \left\{ x_j = \delta_j^{j_0},\, j = 1,2,\dots,m \right\},$$

где δ_l^k – символ Кронекера, то (w^*, x^*) являются также соответствующими решениями систем (3.1.1) и (3.1.2).

Для модели Неймана с матрицами

$$A = \begin{pmatrix} 5 & 2 & 3 & 5 \\ 3 & 1 & 4 & 2 \\ 0 & 2 & 4 & 5 \end{pmatrix} \text{ и } B = \begin{pmatrix} 8 & 3 & 4 & 5 \\ 6 & 3 & 8 & 4 \\ 6 & 4 & 3 & 6 \end{pmatrix}$$

матрица отношений C равна

$$C = \begin{pmatrix} 5/8 & 2/3 & 3/4 & 1 \\ 1/2 & 1/3 & 1/2 & 1/2 \\ 0 & 1/2 & 4/3 & 5/6 \end{pmatrix}.$$

Седловой точкой данной матрицы будет элемент первой строки, первого столбца $5/8 = 0.625$. Т.е. имеем решение матричной игры в чистых стратегиях $(w^*)^T = \{1;0;0\}$ и $(x^*)^T = \{1;0;0;0\}$ с ценой игры $G_C = 0.625$. Согласно теореме 3.1.2 $\overline{x} = x^*$, $\overline{w} = w^*$ и $G_C = \overline{\lambda}$. Действительно, число Фробениуса равно $\overline{\lambda} = 0.625$, а вектор Фробениуса $\overline{x}^T = \{1;0:0;0\}$ и $\overline{w}^T = \{1;0;0\}$

Для модели Неймана с матрицами

$$A=\begin{pmatrix}8 & 4 & 3 & 5\\ 3 & 1 & 4 & 2\\ 0 & 2 & 4 & 5\end{pmatrix} \text{ и } B=\begin{pmatrix}8 & 3 & 4 & 5\\ 6 & 3 & 8 & 4\\ 6 & 4 & 3 & 6\end{pmatrix}.$$

матрица отношений C не имеет седловых точек.

$$C=\begin{pmatrix}1 & 4/3 & 3/4 & 1\\ 1/2 & 1/3 & 1/2 & 1/2\\ 0 & 1/2 & 4/3 & 5/6\end{pmatrix}.$$

Решением матричной игры в смешанных стратегиях для данной матрицы является $G_C \cong 0.842105$, что совпадает с числом Фробениуса. Однако вектор Фробениуса $\overline{x}^T = \{0.2258; 0; 0.7742; 0\}$ не совпадает с вектором игровой стратегии $x^* = \{0.3684; 0; 0.6316; 0\}$. На практике используются модели Неймана с нулевыми элементами в матрице B, поэтому рассмотрим более универсальный подход.

Нахождение $x(\lambda)$ и $w(\lambda)$

Предварительно заметим, что обе задачи взаимно двойственной пары (3.1.7)-(3.1.8) имеют непустые допустимые множества. В частности, допустимым решением задачи (3.1.7) будет

$$x_j = \frac{1}{m},\ j = 1,2,\ldots m, \quad u = \frac{1}{m}\cdot \min_{j=1,2,\ldots,m}\left(\sum_{i=1}^{n}\left(a_{ij} - \lambda b_{ij}\right)\right),$$

а допустимым решением задачи (3.1.8)

$$w_i = \frac{1}{n},\ i = 1,2,\ldots n, \quad v = \frac{1}{n}\cdot \min_{i=1,2,\ldots,n}\left(\sum_{j=1}^{m}\left(a_{ij} - \lambda b_{ij}\right)\right).$$

Следовательно, в соответствии с теоремой двойственности в линейном программировании, задачи (3.1.7) и (3.1.8) имеют оптимальные решения и равные оптимальные значения при любом значении λ.

Из определения задач (3.1.7) и (3.1.8) следует, что если их оптимальные значения равны нулю, то оптимальные решения обеих задач будут вырожденными, т.е. в их оптимальных базисных решениях значения некоторых базисных переменных будут нулевыми. Отмеченный факт приводит к вычислительной неустойчивости алгоритмов, использующих традиционные численные методы оптимизации, в частности – симплекс-метод.

В данной работе показано, что решением задач (3.1.7) и (3.1.8) являются оптимальные смешанные стратегии соответствующей матричной игры.

Предположим, что при фиксированном значении λ оптимальные значения задач (3.1.7) и (3.1.8) положительны $u(\lambda)=v(\lambda)=\gamma>0$. В этом случае

$$\left\{\left(A-\lambda B\right)x\le u,\left(x,e^{m}\right)=1,\,x\ge 0\right\}\Leftrightarrow\left\{\left(A-\lambda B\right)\frac{x}{\gamma}\le e^{n},\left(\frac{x}{\gamma},e^{m}\right)=\frac{1}{\gamma},\frac{x}{\gamma}\ge 0\right\},$$

$$\left\{\left(A-\lambda B\right)^{T}w\ge v,\left(w,e^{n}\right)=1,\,w\ge 0\right\}\Leftrightarrow\left\{\left(A-\lambda B\right)^{T}\frac{w}{\gamma}\ge e^{m},\left(\frac{w}{\gamma},e^{n}\right)=\frac{1}{\gamma},\frac{w}{\gamma}\ge 0\right\}.$$

Введение неотрицательных переменных

$$\xi_{j}=\frac{x_{j}}{\gamma},\,j=1,2\ldots,m;\quad \pi_{i}=\frac{w_{i}}{\gamma},\,i=1,2,\ldots,n \tag{3.1.23}$$

позволяет поставить во взаимно однозначное соответствие оптимальным решениям задач (3.1.7) и (3.1.8) оптимальные решения следующей пары взаимно двойственных задач

$$\left\{\max_{\xi}\left(\xi,e^{m}\right):\quad\left(A-\lambda B\right)\xi\le e^{n},\,\xi\ge 0\right\}, \tag{3.1.24}$$

$$\left\{\min_{\pi}\left(\pi,e^{n}\right):\quad\left(A-\lambda B\right)^{T}\pi\ge e^{m},\,\pi\ge 0\right\}, \tag{3.1.25}$$

имеющих допустимые решения и положительное оптимальное значение.

Как известно [44], пара взаимно двойственных задач (3.1.24) и (3.1.25), имеющих допустимые решения и положительное оптимальное значение, эквивалентна матричной игре $G^{>}$ с платежной матрицей (A-λB), в которой множеством стратегий первого игрока являются столбцы платежной матрицы, а множеством стратегий второго игрока – строки этой же матрицы. При этом, оптимальное значение θ и оптимальные решения ξ^* и π^* задач (3.1.24) и (3.1.25), цена игры Θ и оптимальные смешанные стратегии Ξ, П удовлетворяют соотношениям

$$\Theta=\frac{1}{\theta};\quad \Xi=\frac{\xi^{*}}{\theta};\quad \Pi=\frac{\pi^{*}}{\theta}.$$

Сопоставление данных равенств с (3.1.23), (3.1.24) и (3.1.25) дает

$$u=v=\gamma=\theta,\quad x=\Xi,\quad w=\Pi.$$

Таким образом, решение матричной игры $G^{>}$ в рассматриваемом случае дает оптимальное значение и оптимальные решения пары взаимно двойственных задач (3.1.7) и (3.1.8).

Теперь предположим, что при фиксированном значении λ оптимальные значения задач (3.1.7) и (3.1.8) отрицательны $u(\lambda)=v(\lambda)=\gamma<0$. В этом случае

$$\left\{ (A-\lambda B)x \le u, \left(x,e^m\right)=1,\, x\ge 0\right\} \Leftrightarrow \left\{-(A-\lambda B)\left(-\frac{x}{\gamma}\right)\ge e^n, \left(-\frac{x}{\gamma},e^m\right)=-\frac{1}{\gamma}, \left(-\frac{x}{\gamma}\right)\ge 0\right\},$$

$$\left\{(A-\lambda B)^T w \ge v, \left(w,e^n\right)=1,\, w\ge 0\right\} \Leftrightarrow \left\{-(A-\lambda B)^T\left(-\frac{w}{\gamma}\right)\le e^m, \left(-\frac{w}{\gamma},e^n\right)=-\frac{1}{\gamma}, \left(-\frac{w}{\gamma}\right)\ge 0\right\}.$$

Введение неотрицательных переменных

$$\xi_j=-\frac{x_j}{\gamma},\ j=1,2\ldots,m;\quad \pi_i=-\frac{w_i}{\gamma},\ i=1,2,\ldots,n$$

позволяет поставить во взаимно однозначное соответствие оптимальным решениям задач (3.1.7) и (3.1.8) оптимальные решения следующей пары взаимно двойственных задач

$$\left\{\min_{\xi}\left(\xi,e^m\right):\quad -(A-\lambda B)\xi\ge e^n,\, \xi\ge 0\right\}, \tag{3.1.26}$$

$$\left\{\max_{\pi}\left(\pi,e^n\right):\quad -(A-\lambda B)^T\pi\le e^m,\, \pi\ge 0\right\}, \tag{3.1.27}$$

имеющих допустимые решения и положительное оптимальное значение.

Данная пара взаимно двойственных задач, в соответствии с [9], эквивалентна матричной игре $G^{<}$ с платежной матрицей ($-A+\lambda B$), в которой множеством стратегий первого игрока являются строки платежной матрицы, а множеством стратегий второго игрока – столбцы этой же матрицы. Сопоставление игр $G^{>}$ и $G^{<}$ показывает, что они различаются только нумерацией игроков и знаками элементов платежной матрицы.

Изложенное позволяет сделать вывод, что решение игры $G^{>}$ дает оптимальное значение и оптимальные решения пары взаимно двойственных задач (3.1.7) и (3.1.8) также и случае $u(\lambda)=v(\lambda)=\gamma<0$.

Из равенства ненулевых значений игр $G^{>}$ и $G^{<}$ оптимальному значению пары взаимно двойственных задач (3.1.7) и (3.1.8) следует, что при

$u(\lambda) = v(\lambda) = \gamma = 0$ значения этих игр также будут равны нулю, т.к. противоположное предположение приводит к противоречию. Поскольку, в соответствии с (3.1.25) и , смешанные стратегии игр $G^{>}$ и $G^{<}$ дают допустимые решения задач (3.1.7), (3.1.8), то решение игры $G^{>}$ дает оптимальное значение и оптимальные решения пары взаимно двойственных задач (3.1.7) и (3.1.8) также и случае $u(\lambda) = v(\lambda) = 0$.

Итог изложенному можно сформулировать в виде следующей теоремы.

Теорема 3.1.3. *Пусть* Г *– матричная игра с платежной матрицей* $(A\text{-}\lambda B)^{\mathrm{T}}$, *пусть также* x^*, w^* *– оптимальные смешанные стратегии первого и второго игроков соответственно*, u^* *– цена игры* Г. *Тогда* (u^*, x^*) *и* (u^*, w^*) *– оптимальные решения задач* (3.1.7) *и* (3.1.8) *соответственно.*

Следует заметить, что эквивалентность задач (3.1.7) и (3.1.8) матричной игре отмечалась и ранее в работах Д. Неймана и И.И. Еремина [14].

3.2. Описание алгоритма нахождения параметров модели

Теоремы 3.1.1, 3.1.2 и 3.1.3, доказанные в предыдущем разделе, позволяют построить вполне устойчивый алгоритм нахождения функции $u(\lambda)$, применяя аппарт теории игр [35].

Известно, что данная функция $u(\lambda)$ является непрерывной и монотонно убывающей; $u(0) > 0$ и $u(+\infty) = -\infty$ [10], поэтому задачу можно решить посредством половинного деления с применением симплекс-метода [31, 48, 54] с применением задач (3.1.26) или (3.1.27).

Диапазон поиска числа Фробениуса имеет перед половинным делением верхнюю границу $\overline{\lambda} = 1$ и нижнюю границу $\underline{\lambda} = 0$. Пусть $\varepsilon > 0$ - погрешность (малая величина, близкая к нулю) для половинного деления. Для работы алгоритма необходимо также ввести переменные: λ_+ - максимальное среди найденных половинным делением значений λ, при которых $u(\lambda) > \varepsilon$; λ_- - минимальное среди найденных половинным делением значений λ, при которых $u(\lambda) < -\varepsilon$; λ_0 - максимальное среди найденных половинным делением значений λ, при которых $|u(\lambda)| \le \varepsilon$ (см. приложение А). В ходе работы алгоритма выполняются следующие шаги (см. рисунок 3.2.1):

1. Алгоритм ищет решение у задачи (3.1.24) при $\lambda = 1$. Для этого рассчитывается матрица с положительными элементами $(A - \lambda B + \Psi)$, где Ψ -

матрица размера $n \times m$, все элементы которой равны ψ^* ($\Psi = \{\psi_{ij}\}$, $\psi_{ij} = \psi^*$), а $\psi^* = \min\limits_{i,j}(a_{ij} - b_{ij}) + \varepsilon$ ($i = 1,2,\ldots n$; $j = 1,2,\ldots m$). Если полученное с помощью симплекс-метода значение функции $|u(\lambda)| = \left|1/(\xi, e^m) - \psi^*\right| \le \varepsilon$, то модель бюджетирования (A, B) непродуктивна, поэтому следует выход из алгоритма (см. приложение А).

2. Если матрица B не содержит нулевых элементов, то рассчитывается матрица $C = \left(\dfrac{a_{ij}}{b_{ij}}\right)$ (см. раздел 3.1). Для этой матрицы вычисляется минимакс и максимин, $\overline{\lambda} := \min\limits_j \max\limits_i c_{ij}$, $\underline{\lambda} := \max\limits_i \min\limits_j c_{ij}$.

3. Проверяется наличие седловой точки для матрицы C. Если она найдена и равна c_{IJ}, $\overline{\lambda} = \underline{\lambda} = \lambda^* = c_{IJ}$, $w_I = x_J = 1$, выход из алгоритма.

4. Алгоритм ищет решение у задачи (3.1.24) при $\lambda := (\overline{\lambda} + \underline{\lambda})/2$. Если полученное с помощью симплекс-метода значение функции $u(\lambda) = 1/(\xi, e^m) - \psi^* > \varepsilon$, то $\lambda_+ := \lambda$. Если же $|u(\lambda)| = \left|1/(\xi, e^m) - \psi^*\right| \le \varepsilon$, то $\lambda_0 := \lambda$. В случае отрицательного значения $u(\lambda) = 1/(\xi, e^m) - \psi^* < -\varepsilon$ $\lambda_- := \lambda$.

5. Если радиус интервала поиска $(\overline{\lambda} - \underline{\lambda})/2 \le \varepsilon$, то выход из алгоритма. В противном случае $\lambda := (\overline{\lambda} + \underline{\lambda})/2$. Если полученное с помощью симплекс-метода значение функции $u(\lambda) = 1/(\xi, e^m) - \psi^* > \varepsilon$, то $\lambda_+ := \max\{\lambda; \lambda_+\}$ и $\underline{\lambda} := \lambda$. Если же $|u(\lambda)| = \left|1/(\xi, e^m) - \psi^*\right| \le \varepsilon$, то $\lambda_0 = \max\{\lambda; \lambda_0\}$ и $\overline{\lambda} := \lambda$. В случае отрицательного значения $u(\lambda) = 1/(\xi, e^m) - \psi^* < -\varepsilon$ $\lambda_- = \min\{\lambda; \lambda_-\}$ и $\underline{\lambda} := \lambda$.

6. Далее повторение пятого шага, пока не будет выхода из алгоритма.

Если модель продуктивна, то $\lambda^* \in (\underline{\lambda}; \overline{\lambda})$.

В результате вычислительного эксперимента с разработанным алгоритмом были решены все задачи, используемые при тестировании коммерческого программного обеспечения.

Данный алгоритм был протестирован на задачах, сгенерированных программой, описанной в главе 2. Все задачи были решены.

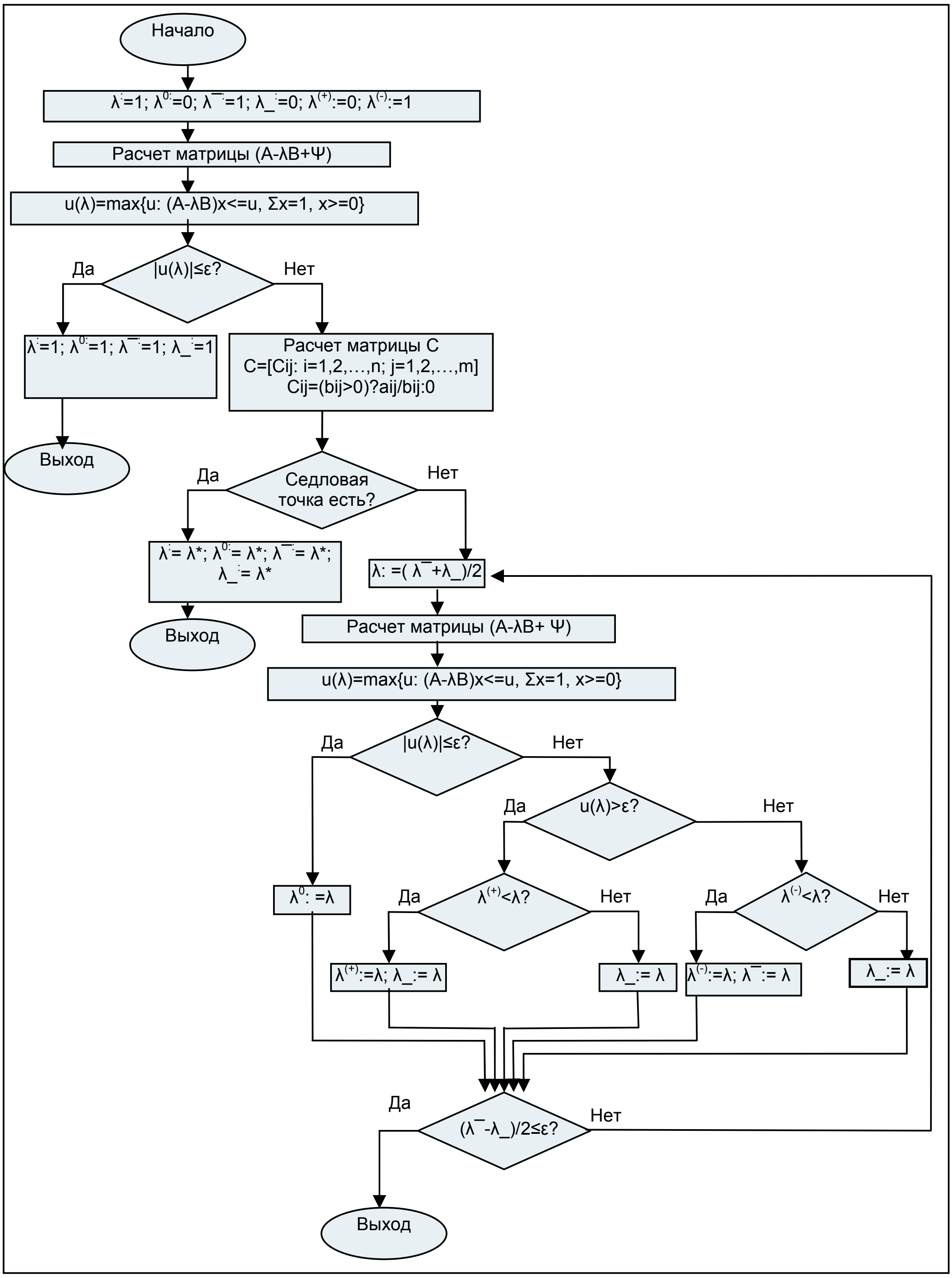

Рисунок 3.2.1. Блок-схема алгоритма половинного деления

3.3. Выводы и результаты

Данная глава рассмотрены методы и алгоритмы для решения математической модели, описанной в предыдущей главе:

1) значение параметра $\overline{\lambda}$ возможно определить, используя теоремы 3.2.1, 3.2.2 и 3.2.3;

2) для решения задачи (2.2.1) целесообразно использовать устойчивые алгоритмы теории игр;

3) построено программное обеспечение, позволяющее находить решение рассматриваемой задачи, что подтверждено вычислительным экспериментом.

4. Исследование проблемы при интервальной неопределенности

Основными исходными данными при построении бюджетов являются данные о спросе (P,Q). Эти данные, как правило, получают в результате опросов потребителей, экспертов. Следовательно, полученные данные о спросе могут иметь интервальную неопределенность [31]. Соответственно интервальными будут матрица переменных издержек $\mathbf{A}=\left[\underline{a}_{ij},\overline{a}_{ij}\right]$ ($i=\overline{1,n},\ j=\overline{1,m}$), и выручки $\mathbf{B}=\left[\underline{b}_{ij},\overline{b}_{ij}\right]$ ($i=\overline{1,n},\ j=\overline{1,m}$). Для решения данной неопределенности применяется аппарат интервального анализа [19]. В разделе 4.1 данной главы с целью целостности изложения приведены основные понятия интервального анализа. Основные результаты, относящиеся к проблеме бюджетирования приведены в разделе 4.2.

4.1. Основные понятия интервального анализа

Интервал $[x]$ - это некоторое односвязное подмножество из $\mathbb{R}$.

Нижняя граница интервала $[x]$, обозначаемая как $\underline{x}$, определяется как

$$\underline{x} \triangleq \sup\{a \in \mathbb{R} \cup \{-\infty;\infty\} \mid \forall x \in [x], a \le x\}. \tag{4.1.1}$$

Верхняя граница интервала $[x]$, обозначаемая как $\overline{x}$, определяется как

$$\overline{x} \triangleq \inf\{b \in \mathbb{R} \cup \{-\infty;\infty\} \mid \forall x \in [x], x \le b\}. \tag{4.1.2}$$

Ширина любого интервала определяется как $w([x]) \triangleq \overline{x} - \underline{x}$.

Средняя точка (или **центр**) любого ограниченного и непустого интервала $[x]$ определяется как $\mathrm{mid}([x]) \triangleq \dfrac{\underline{x}+\overline{x}}{2}$.

Прямое произведение двух интервалов есть прямоугольник $\mathbb{R}^2$.

Интервал называется **замкнутым**, если он является замкнутым подмножеством из $\mathbb{R}$. Множество всех замкнутых интервалов можно обозначить как $\mathbb{IR}$.

Интервал $[x] \in \mathbb{IR}$ называется **точечным**, когда $\underline{x}$ и $\overline{x}$ совпадают.

Интервальное объединение двух непустых замкнутых интервалов $[x]$ и $[y]$ записывается как

$$\forall[x]\in\mathbb{IR},\ \forall[y]\in\mathbb{IR},\ [x]\sqcup[y]=[\min\{\underline{x},\underline{y}\},\max\{\overline{x},\overline{y}\}].$$

Интервальное пересечени*е* двух непустых замкнутых интервалов $[x]$ и $[y]$ удовлетворяет соотношениям

$$[x]\cap[y]=\begin{cases}[\max\{\underline{x},\underline{y}\},\min\{\overline{x},\overline{y}\}], & \text{если } \max\{\underline{x},\underline{y}\}\le\min\{\overline{x},\overline{y}\},\\ 0 & \text{, в противном случае.}\end{cases}$$

Четыре классические операции над вещественными числами (сложение, вычитание, умножение и деление) могут быть расширены на действия с интервалами. **Интервальный аналог** $[f]$ **любой функции** f удовлетворяет следующему соотношению

$$[f([x])]=[\{f(x)\,|\,x\in[x]\}]. \tag{4.1.3}$$

Если α - некоторое вещественное число и $[x]$ - некоторый непустой интервал, то интервал

$$\alpha[x]\triangleq\{\alpha x\,|\,x\in[x]\}$$

задается как

$$\alpha[x]=\begin{cases}[\alpha\underline{x},\alpha\overline{x}], & \text{если } \alpha\ge 0,\\ [\alpha\overline{x},\alpha\underline{x}], & \text{если } \alpha<0.\end{cases}$$

Для непустых замкнутых интервалов

$$[x]+[y]=[\underline{x}+\underline{y},\overline{x}+\overline{y}],$$
$$[x]-[y]=[\underline{x}-\overline{y},\overline{x}-\underline{y}],$$
$$[x]*[y]=[\min\{\underline{x}\underline{y},\overline{x}\underline{y},\underline{x}\overline{y},\overline{x}\overline{y}\},\max\{\underline{x}\underline{y},\overline{x}\underline{y},\underline{x}\overline{y},\overline{x}\overline{y}\}].$$

Операция интервального деления определяется соотношениями

$$1/[y]=\begin{cases}\varnothing, & \text{если } [y]=[0,0],\\ [1/\overline{y},1/\underline{y}], & \text{если } 0\notin[y],\\ [1/\overline{y},\infty[, & \text{если } \underline{y}=0 \text{ и } \overline{y}>0,\\ ]-\infty,1/\underline{y}], & \text{если } \underline{y}<0 \text{ и } \overline{y}=0.\end{cases}$$

и

$$[x]/[y]=[x]*(1/[y]).$$

Вещественный интервальный вектор – это подмножество $\mathbb{R}^n$, которое определяется как произведение n замкнутых интервалов. Такой вектор также называют **параллелотопом** и записывают как

$$[\mathbf{x}] = [x_1] \times [x_2] \times \ldots \times [x_n], \text{ где } [x_i] = [\underline{x}_i, \overline{x}_i], \text{ для } i = 1, 2, \ldots, n .$$

Его i-ая интервальная компонента $[x_i]$ есть проекция $[\mathbf{x}]$ на i-тую ось.

Нижняя граница параллелотопа $[\mathbf{x}]$ есть точечный вектор, составленный из нижних границ его интервальных компонент:

$$\underline{\mathbf{x}} = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n)^T .$$

Верхняя граница параллелотопа $[\mathbf{x}]$ есть точечный вектор, составленный из верхних границ его интервальных компонент:

$$\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)^T .$$

Ширина параллелотопа $[\mathbf{x}] = ([x_1], [x_2], \ldots, [x_n])^T$ есть

$$w([\mathbf{x}]) \triangleq \max_{1 \le i \le n} w([x_i]) .$$

Если параллелотоп $[\mathbf{x}]$ является ограниченным и непустым, то его средняя точка (или центр) есть

$$\mathrm{mid}[\mathbf{x}] \triangleq (\mathrm{mid}([x_1]), \mathrm{mid}([x_2]), \ldots, \mathrm{mid}([x_n]))^T .$$

Пусть $\mathbb{R}^{m \times n}$ - множество всех матриц с вещественными коэффициентами из m строк и n столбцов.

Интервальная матрица размерности $(m \times n)$ является подмножеством из $\mathbb{IR}^{m \times n}$ и может быть определенная как прямое произведение mn замкнутых интервалов:

$$[\mathbf{A}] = \begin{pmatrix} [a_{11}] & \ldots & [a_{1n}] \\ \vdots & & \vdots \\ [a_{m1}] & \ldots & [a_{mn}] \end{pmatrix} = [a_{11}] \times [a_{12}] \times \ldots \times [a_{mn}] = ([a_{ij}])_{1 \le i \le m, 1 \le j \le n}, \quad (4.1.4)$$

где $[a_{ij}] = [\underline{a}_{ij}, \overline{a}_{ij}]$ является проекцией интервальной матрицы $[\mathbf{A}]$ на (i, j)-ую ось.

Множество всех интервальных матриц размерности $m \times n$ обозначается как $\mathbb{IR}^{m \times n}$.

Интервальная матрица называется **точечной**, если все её элементы являются точечными.

Нижняя граница $\underline{\mathbf{A}}$ интервальной матрицы $[\mathbf{A}]$ есть точечная матрица составленная из нижних границ её компонент:

$$\underline{\mathbf{A}} \triangleq \begin{pmatrix} \underline{a}_{11} & \cdots & \underline{a}_{1n} \\ \vdots & & \vdots \\ \underline{a}_{m1} & \cdots & \underline{a}_{mn} \end{pmatrix}. \tag{4.1.5}$$

Аналогично, **верхняя граница** $\overline{\mathbf{A}}$ интервальной матрицы есть точечная матрица

$$\overline{\mathbf{A}} \triangleq \begin{pmatrix} \overline{a}_{11} & \cdots & \overline{a}_{1n} \\ \vdots & & \vdots \\ \overline{a}_{m1} & \cdots & \overline{a}_{mn} \end{pmatrix}. \tag{4.1.6}$$

Ширина интервальной матрицы $w([\mathbf{A}])$ определяется как

$$w([\mathbf{A}]) \triangleq \max_{1 \le i \le m, 1 \le j \le n} w([a_{ij}]).$$

Если $[\mathbf{A}] \in \mathbb{IR}^{m \times n}$ ограничена и непуста, то её **средняя точка (центр)** задается как

$$\operatorname{mid}([\mathbf{A}]) = \operatorname{mid}([a_{ij}])_{1 \le i \le m, 1 \le j \le n}. \tag{4.1.7}$$

Для $[\mathbf{A}]$ и $[\mathbf{B}]$ из $\mathbb{IR}^{m \times n}$ и $\mathbf{C}$ (точечной матрицы) из $\mathbb{IR}^{m \times n}$

$$[\mathbf{A}] \subset [\mathbf{B}] \Leftrightarrow [a_{ij}] \subset [b_{ij}] \text{ для } 1 \le i \le m, 1 \le j \le n, \tag{4.1.8}$$

$$\mathbf{C} \in [\mathbf{B}] \Leftrightarrow c_{ij} \in [b_{ij}] \text{ для } 1 \le i \le m, 1 \le j \le n. \tag{4.1.9}$$

Для интервальных матриц $[\mathbf{A}],[\mathbf{B}]$ из $\mathbb{IR}^{m \times n}$, $[\mathbf{C}]$ из $\mathbb{IR}^{n \times l}$, вектора $[\mathbf{x}]$ из $\mathbb{IR}^n$ и числа $\alpha \in \mathbb{R}$ можно ввести следующие операции:

$$\alpha[\mathbf{A}] = (\alpha[a_{11}] \times \alpha[a_{12}] \times \ldots \times \alpha[a_{mn}]), \tag{4.1.10}$$

$$[\mathbf{A}] + [\mathbf{B}] = ([a_{ij}] + [b_{ij}])_{1 \le i \le m, 1 \le j \le n}, \tag{4.1.11}$$

$$[\mathbf{A}]*[\mathbf{C}] = (\sum_{k=1}^{n}[a_{ik}]*[c_{kj}])_{1\le i\le m, 1\le j\le l}, \qquad (4.1.12)$$

$$[\mathbf{A}]*[\mathbf{x}] = (\sum_{j=1}^{n}[a_{ij}]*[x_j])_{1\le i\le m}. \qquad (4.1.13)$$

Рассмотрим $n_\mathbf{x}$ переменных $x_i \in \mathbb{R}$, $i = 1,2,...,n_\mathbf{x}$, связанных $n_\mathbf{f}$ ограничениями вида:

$$f_j(x_1, x_2, ..., x_{n_\mathbf{x}}) = 0, \ j \in \{1,2,...,n_\mathbf{f}\}.$$

Обозначим вектор $\mathbf{x}$ как $\mathbf{x} = (x_1, x_2, ..., x_{n_\mathbf{x}})^T$, а соответствующую область значений вектора $\mathbf{x}$ как $[\mathbf{x}] = [x_1]\times[x_2]\times...\times[x_{n_\mathbf{x}}]$.

Пусть $\mathbf{f}$ - некоторая вектор функция, координатами которой являются функции f_j. Тогда систему ограничений можно записать $\mathbf{f}(\mathbf{x}) = 0$. Оно соответствует **задаче выполнения ограничений** $\mathcal{H}: (\mathbf{f}(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}])$.

Множество решений определяется как $\mathbb{S}: (\mathbf{f}(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}])$.

Сжатие задачи $\mathcal{H}$ означает замену параллелотопа $[\mathbf{x}]$ некоторой меньшей областью $[\mathbf{x}']$ такой, что множество решения остается неизменным, т.е. $\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$. Существует **оптимальное сжатие** $\mathcal{H}$, которое соответствует замещению $[\mathbf{x}]$ наименьшим параллелотопом, содержащим $\mathbb{S}$.

Сжимающим оператором (или оператором сжатия) для $\mathcal{H}$ является любой оператор, который может применен для сжатия $\mathcal{H}$. К таким операторам относятся $\mathcal{C}_{GE}$ (метод исключения Гаусса), $\mathcal{C}_{GS}$ (алгоритм Гаусса-Зейделя), $\mathcal{C}_K$ (метод Кравчика), $\mathcal{C}_{\uparrow\downarrow}$ (метод вперед-назад), $\mathcal{C}_{LP}$ (линейное программирование), $\mathcal{C}_{GEP}$ (метод исключения Гаусса с улучшением обусловленности), $\mathcal{C}_N$ (метод Ньютона с улучшением обусловленности), $\mathcal{C}_{||}$ (параллельная линеаризация).

К сожалению, данные методы сжатия не позволят существенно сжать область для λ, x и w в силу вырожденности задач (2.1.1) и (2.2.1). Поэтому нами был разработан другой подход.

4.2. Анализ интервальной неопределенности для модели бюджетирования

Далее будем обозначать через $\operatorname{mid}\mathbf{A}$ и $\operatorname{mid}\mathbf{B}$ точечные матрицы, элементами которых являются центры интервальных элементов матриц $\mathbf{A}$ и $\mathbf{B}$ соответственно [31].

Теорема 4.2.1. *Пусть*

1) β_1 и β_2 удовлетворяют условиям $\tilde{A} = \beta_1 \cdot \operatorname{mid}\mathbf{A} \in \mathbf{A}, \quad \tilde{B} = \beta_2 \cdot \operatorname{mid}\mathbf{B} \in \mathbf{B}$;

2) $\left(\lambda^*, x^*, w^*\right) = \arg \max\limits_{(\lambda,x,w)\in D(\operatorname{mid}\mathbf{A},\operatorname{mid}\mathbf{B})} \lambda$.

Тогда $\left(\dfrac{\lambda^*\beta_1}{\beta_2}, x^*, w^*\right) = \arg \max\limits_{(\lambda,x,w)\in D(\tilde{A},\tilde{B})} \lambda$.

Доказательство.

Для модели $(\tilde{A},\tilde{B})$ имеем

$$(\tilde{\lambda}^*,\tilde{x},\tilde{w}) = \arg \max_{(\tilde{\lambda},x,w)\in D(\tilde{A},\tilde{B})} \lambda,$$
$$D(\tilde{A},\tilde{B}) = \left\{ (\tilde{\lambda},x,w) \,\middle|\, \begin{array}{c} (\tilde{A}-\tilde{\lambda}\tilde{B})x \le 0, (\tilde{A}-\tilde{\lambda}\tilde{B})^T w \ge 0, \\ (x,e^m)=1,\ (w,e^n)=1, x\ge 0,\ w \ge 0,\ \tilde{\lambda}\ge 0 \end{array} \right\}. \tag{4.2.1}$$

Сделав в задаче (4.2.1) замену переменной $\tilde{\lambda} = \lambda\beta_1/\beta_2$, получим:

$$(\lambda^*,\tilde{x},\tilde{w}) = \arg \max_{(\lambda,x,w)\in D(\tilde{A},\tilde{B})} \lambda,$$
$$D(\tilde{A},\tilde{B}) = \left\{ (\lambda,x,w) \,\middle|\, \begin{array}{c} (\tilde{A}-\lambda\beta_1/\beta_2\tilde{B})x \le 0, (\tilde{A}-\lambda\beta_1/\beta_2\tilde{B})^T w \ge 0, \\ (x,e^m)=1, (w,e^n)=1, \\ x\ge 0,\ w\ge 0,\ \lambda \ge 0 \end{array} \right\}. \tag{4.2.2}$$

В соответствии с условием 1) полученная задача совпадает с задачей нахождения параметров (λ^*, x^*, w^*) для модели Неймана $(\operatorname{mid}\mathbf{A},\operatorname{mid}\mathbf{B})$:

$$(\lambda^*,x^*,w^*) = \arg \max_{(\lambda,x,w)\in D(\mathrm{mid}\mathbf{A},\mathrm{midB})} \lambda,$$
$$D(\mathrm{mid}\mathbf{A},\mathrm{midB}) = \left\{ (\lambda,x,w) \,\middle|\, \begin{array}{l} (\mathrm{mid}\mathbf{A}-\lambda\mathrm{midB})x \le 0, \\ (\mathrm{mid}\mathbf{A}-\lambda\mathrm{midB})^T w \ge 0, \\ (x,e^m)=1, (w,e^n)=1, \\ x\ge 0,\ w\ge 0,\ \lambda\ge 0 \end{array} \right\}. \tag{4.2.3}$$

Тогда, решением для модели $(\tilde{A},\tilde{B})$ будет $(\lambda^*\beta_1/\beta_2, x^*, w^*)$ ■

Как следует из теоремы для моделей вида $(\tilde{A},\tilde{B})$ коэффициенты β_1 и β_2 влияют на параметр

$$\tilde{r} = 1-\tilde{\lambda} = 1-\lambda^*\beta_1/\beta_2 = 1-(1-r^*)\beta_1/\beta_2 = \frac{\beta_1}{\beta_2}\left(r^*-1\right)+1$$

и не влияют на векторы x^* и w^*, т.е. политика диверсификации цен полностью определяется матрицами центров интервалов.

Пусть $\underline{\mathbf{A}}$ и $\underline{\mathbf{B}}$ представляют собой матрицы нижних границ интервалов матриц $\mathbf{A}$ и $\mathbf{B}$ соответственно. Пусть $\overline{\mathbf{A}}$ и $\overline{\mathbf{B}}$ представляют собой матрицы верхних границ интервалов матриц $\mathbf{A}$ и $\mathbf{B}$ соответственно.

Теорема 4.2.2. *Пусть*

1) *точечные матрицы* $(\tilde{A},\tilde{B})$ *удовлетворяют условиям* $\tilde{A}\in\mathbf{A}$ *и* $\tilde{B}\in\mathbf{B}$;

2)

$$(\tilde{\lambda},\tilde{x},\tilde{w}) = \arg\max_{(\lambda,x,w)\in D(\tilde{A},\tilde{B})} \lambda,$$

$$D(\tilde{A},\tilde{B}) = \left\{(\lambda,x,w)\left|\begin{array}{c}(\tilde{A}-\lambda\tilde{B})x\le 0,(\tilde{A}-\lambda\tilde{B})^T w\ge 0,\\ (x,e^m)=1,\ (w,e^n)=1, x\ge 0,\ w\ge 0,\ \lambda\ge 0\end{array}\right.\right\}. \quad (4.2.4)$$

3)

$$(\overline{\lambda},\overline{x},\overline{w}) = \arg\max_{(\lambda,x,w)\in D(\overline{\mathbf{A}},\underline{\mathbf{B}})} \lambda,$$

$$D(\overline{\mathbf{A}},\underline{\mathbf{B}}) = \left\{(\lambda,x,w)\left|\begin{array}{c}(\overline{\mathbf{A}}-\lambda\underline{\mathbf{B}})x\le 0,(\overline{\mathbf{A}}-\lambda\underline{\mathbf{B}})^T w\ge 0,\\ (x,e^m)=1,(w,e^n)=1,\\ x\ge 0,\ w\ge 0,\ \lambda\ge 0\end{array}\right.\right\}. \quad (4.2.5)$$

4)

$$(\underline{\lambda},\underline{x},\underline{w}) = \arg\max_{(\lambda,x,w)\in D(\underline{\mathbf{A}},\overline{\mathrm{B}})} \lambda,$$

$$D(\underline{\mathbf{A}},\overline{\mathrm{B}}) = \left\{(\lambda,x,w)\left|\begin{array}{c}(\underline{\mathbf{A}}-\lambda\overline{\mathrm{B}})x\le 0,(\underline{\mathbf{A}}-\lambda\overline{\mathrm{B}})^T w\ge 0,\\ (x,e^m)=1,(w,e^n)=1,\\ x\ge 0,\ w\ge 0,\ \lambda\ge 0\end{array}\right.\right\}. \quad (4.2.6)$$

Тогда $\underline{\lambda}\le\tilde{\lambda}\le\overline{\lambda}$.

Доказательство.

Из условия 1) следует, что для любых i и j $(i=1,2,...n; j=1,2,...m)$ выполняется:

$$\overline{a}_{ij} \geq \tilde{a}_{ij},\ \overline{b}_{ij} \geq \tilde{b}_{ij};\ \underline{a}_{ij} \leq \tilde{a}_{ij},\ \underline{b}_{ij} \leq \tilde{b}_{ij}. \tag{4.2.7}$$

Откуда $\tilde{A} = \underline{\mathbf{A}} + A' = \overline{\mathbf{A}} - A''$, $\tilde{B} = \underline{\mathbf{B}} + B' = \tilde{\mathbf{B}} - B''$, где $A' = \left(a'_{ij}\right) = \left(\tilde{a}_{ij} - \underline{a}_{ij}\right)$, $B' = \left(b'_{ij}\right) = \left(\tilde{b}_{ij} - \underline{b}_{ij}\right)$, $A'' = \left(a''_{ij}\right) = \left(\overline{a}_{ij} - \tilde{a}_{ij}\right)$, $B'' = \left(b''_{ij}\right) = \left(\overline{b}_{ij} - \tilde{b}_{ij}\right)$. Из (4.2.7), что все элементы матриц A', B', A'' и B'' неотрицательны.

Заменим матрицы $\tilde{A} = \overline{\mathbf{A}} - A''$ и $\tilde{B} = \underline{\mathbf{B}} + B'$ для задачи (4.2.4):

$$(\tilde{\lambda}, \tilde{x}, \tilde{w}) = \arg \max_{(\lambda, x, w) \in D(\tilde{A}, \tilde{B})} \lambda,$$

$$D(\tilde{A}, \tilde{B}) = \left\{ (\lambda, x, w) \middle| \begin{array}{c} (\overline{\mathbf{A}} - A'' - \lambda(\underline{\mathbf{B}} + B'))x \leq 0, \\ (\overline{\mathbf{A}} - A'' - \lambda(\underline{\mathbf{B}} + B'))^T w \geq 0, \\ (x, e^m) = 1,\ (w, e^n) = 1, x \geq 0,\ w \geq 0,\ \lambda \geq 0 \end{array} \right\}. \tag{4.2.8}$$

В соответствии с (4.2.8) имеет место неравенство:

$$(\overline{\mathbf{A}} - \tilde{\lambda}\underline{\mathbf{B}})^T \tilde{w} \geq 0.$$

Откуда следует, что для любого $j = 1, 2, \ldots, m$

$$\tilde{\lambda} \leq \frac{\sum_{i=1}^{n} \overline{a}_{ij} \tilde{w}_i}{\sum_{i=1}^{n} \underline{b}_{ij} \tilde{w}_i} \leq \frac{\sum_{i=1}^{n} \overline{a}_{ij} \overline{w}_i}{\sum_{i=1}^{n} \underline{b}_{ij} \overline{w}_i}. \tag{4.2.9}$$

Последнее неравенство в данной цепочке следует из условия 3) теоремы, в соответствии с которым

$$\overline{w} = \arg \max_{w:(w, e^n) = 1, w \geq 0} \min_{j = 1, 2, \ldots, m} \frac{\sum_{i=1}^{n} \overline{a}_{ij} w_i}{\sum_{i=1}^{n} \underline{b}_{ij} w_i}.$$

Поскольку $\overline{\lambda} = \max\limits_{w:(w,e^n)=1,w\geq 0} \min\limits_{j=1,2,\ldots,m} \dfrac{\sum\limits_{i=1}^{n} \overline{a}_{ij} w_i}{\sum\limits_{i=1}^{n} \underline{b}_{ij} w_i}$, то имеем: $\tilde{\lambda} \leq \overline{\lambda}$.

Неравенство $\tilde{\lambda} \geq \underline{\lambda}$ доказывается аналогично. Действительно, после замены матриц $\tilde{A} = \underline{\mathbf{A}} + A'$ и $\tilde{B} = \overline{\mathbf{B}} - B''$ в задаче (4.2.4) имеем задачу

$$(\tilde{\lambda}, \tilde{x}, \tilde{w}) = \arg \max_{(\lambda,x,w)\in D(\tilde{A},\tilde{B})} \lambda,$$

$$D(\tilde{A},\tilde{B}) = \left\{ (\lambda, x, w) \left| \begin{array}{l} (\underline{\mathbf{A}} + A' - \lambda(\overline{\mathbf{B}} - B''))x \leq 0, \\ (\underline{\mathbf{A}} + A' - \lambda(\overline{\mathbf{B}} - B''))^T w \geq 0, \\ (x, e^m) = 1,\ (w, e^n) = 1, x \geq 0, \\ w \geq 0,\ \lambda \geq 0 \end{array} \right. \right\}. \qquad (4.2.10)$$

Из условия (4.2.10) следует $(\underline{\mathbf{A}} - \tilde{\lambda}\overline{\mathbf{B}})\tilde{x} \leq 0$, поэтому для любого $i = 1,2,\ldots,n$ имеет место неравенство:

$$\tilde{\lambda} \geq \frac{\sum\limits_{j=1}^{m} \underline{a}_{ij} \tilde{x}_j}{\sum\limits_{j=1}^{m} \overline{b}_{ij} \tilde{x}_j} \geq \frac{\sum\limits_{j=1}^{m} \underline{a}_{ij} \underline{x}_j}{\sum\limits_{j=1}^{m} \overline{b}_{ij} \underline{x}_j}. \qquad (4.2.11)$$

Последнее неравенство в данной цепочке следует из того, что

$$\underline{x} = \arg \min_{x:(x,e^m)=1,x\geq 0} \max_{i=1,2,\ldots,n} \frac{\sum\limits_{j=1}^{m} \underline{a}_{ij} x_j}{\sum\limits_{j=1}^{m} \overline{b}_{ij} x_j}.$$

Поскольку $\underline{\lambda} = \min\limits_{x:(x,e^m)=1,x\geq 0} \max\limits_{i=1,2,\ldots,n} \dfrac{\sum\limits_{j=1}^{m} \underline{a}_{ij} x_j}{\sum\limits_{j=1}^{m} \overline{b}_{ij} x_j}$, то $\tilde{\lambda} \geq \underline{\lambda}$.

Теорема доказана ■

4.3. Выводы и результаты

В данной главе изучено влияние интервальной неопределенности исходных данных модели Неймана на параметры равновесия. В частности показано, что:

1) при пропорциональности (с коэффициентами β_A и β_B) элементов матриц затрат A и выпуска B матрицам центров интервалов $\mathrm{mid}\mathbf{A}$ и $\mathrm{mid}\mathrm{B}$ соответственно, матрицы $\mathrm{mid}\mathbf{A}$ и $\mathrm{mid}\mathrm{B}$ однозначно определяют интенсивность применения стратегий, при этом значение числа Фробениуса оказывается зависимой от коэффициентов β_A и β_B;

2) число Фробениуса ограничено сверху числом Фробениуса для модели $(\overline{\mathbf{A}},\underline{\mathbf{B}})$, снизу – числом Фробениуса для модели $(\underline{\mathbf{A}},\overline{\mathbf{B}})$, где $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$ - точечные матрицы верхних границ интервальных матриц $\mathbf{A}$ и $\mathbf{B}$ соответственно, $\underline{\mathbf{A}}$, $\underline{\mathbf{B}}$ - точечные матрицы нижних границ.

5. Программное обеспечение

В данной главе предложена архитектура системы оптимизации бюджетирования при применении ценовой диверсификации. В работе приведено описание и функции отдельных блоков, рассмотрены связи между ними. Очерчены перспективы развития системы и возможности интеграции с существующими.

В качестве цели проекта была поставлены задача разработки модуля программно-аппаратного обеспечения, позволяющего [38, 42]:

1) проводить численную оптимизацию ценовой стратегии;

2) вести базу данных о проектах бюджетирования и предоставлять отчеты по проектам в формате `MS Excel` и `xml`;

3) осуществлять загрузку в модуль и выгрузку в корпоративные системы данных в формате `xml`.

Для реализации поставленных задач был разработан однопользовательский модуль, предназначенный для оптимизации ценовой стратегии (МОЦС), с возможностью интеграции с корпоративными информационными системами [63].

5.1. Технология разработки МОЦС

Было принято решение разрабатывать программное обеспечение МЦОС на основе именно объектно-ориентированного подхода, для чего сначала необходимо провести объектную декомпозицию рассматриваемой задачи с помощью методологии объектно-ориентированного анализа и проектирования (ООАП). Перед проектированием программной системы сначала была создана модель оптимизации [7]. На основе этого был сформирован набор объектов свойственных данной задачи [45].

5.2. Анализ функциональных требований

Проектирование данной прикладной программной системы было начато с анализа требований, которым она должна будет удовлетворять [51, 59, 62] для того, чтобы понять назначение и условия эксплуатации системы настолько, чтобы суметь составить ее предварительный проект.

Проведем анализ функциональных требований к разрабатываемой системе, с помощью выделения основных целей и задач создания системы, ограничения круга потребителей и формулировки функций, реализуемых ею, и перечисления ее атрибутов [50, 61].

5.2.1. Цели и задачи

Главной задачей проекта является создание модуля оптимизации ценовой стратегии с возможностью импорта-экспорта данных и формирования отчетов в формате `Excel` и `xml`.

Потребителями информации будут финансовые менеджеры, руководители маркетинговых служб и других подразделений, занимающихся бюджетированием.

Основная цель разработки системы – это создание системы оптимизации бюджетирования, предназначенной для определения эффективной интенсивности применения ценовой стратегии и оценки внутренней рентабельности для бюджета продаж. Вторичными целями разработки являются:

1) повышение уровня автоматизации сбора данных для бюджетирования;

2) предоставление доступа пользователям как к оперативной информации о бюджетах, так и к архивной;

3) оценка продуктивности бюджетов;

4) ведение баз данных о проектах бюджетов.

5.2.2. Функции системы

Функции МОЦС делятся на следующие категории:

1) очевидные – их выполнение очевидно для пользователя;

2) скрытые – необходимы, но пользователю не видны;

3) дополнительные – необязательны.

В результате проведенного анализа предметной области был сформирован следующий список необходимых функций системы (по умолчанию функция является скрытой), сгруппированных по смысловому назначению.

1. Оптимизация ценовой стратегии.

1.1. Импорт данных о проекте бюджетирования в базу данных из файлов в формате `Excel` и `xml`.

1.2. Оптимизация ценовой стратегии и расчет параметров продуктивности.

1.3. Сохранение результатов оптимизации и расчетов в подсистеме хранения данных.

1.4. При поступлении новых данных обновление соответствующей информации и перерасчет с учетом новых данных.

2. Хранение результатов оптимизации.

2.1. Хранение исходных данных для каждого проекта бюджетов в базе данных.

2.2. Хранение результатов оптимизации и анализа продуктивности.

2.3. Резервное копирование базы данных на съемные носители (дополнительная).

2.4. Проведение работ по переиндексации и контролю целостности базы данных (дополнительная).

3. Предоставление пользователю информации о проектах бюджетирования.

3.1. Определение прав пользователя на доступ к необходимой информации (очевидная).

3.2. Получение от пользователя критериев для выборки данных (очевидная).

3.3. Запрос к подсистеме хранения данных в соответствии с критериями пользователя.

3.4. Получение результатов запроса от подсистемы хранения данных.

3.5. Запрос расчета к подсистеме оптимизации и анализа продуктивности.

3.6. Предоставление результатов работы пользователю (очевидная).

4. Тестирование и поддержание работоспособности системы (дополнительная).

4.1. Проверка составляющих частей системы по требованию администратора системы.

4.2. Настройка конфигурации системы.

4.3. Запуск и остановка работы системы.

5. Импорт-экспорт данных в формате `Excel` и `xml`

5.1. Импорт данных в формате `Excel`: настройка импорта и загрузка данных.

5.2. Импорт данных в формате `xml`: настройка импорта и загрузка данных.

5.3. Экспорт данных в формате `Excel`: настройка экспорта и выгрузка данных.

5.4. Экспорт данных в формате `xml`: настройка экспорта и выгрузка данных.

5.2.3. Архитектура системы

На основании составленных функций проектируемой системы был составлен эскиз архитектуры системы оптимизации ценовой стратегии, приведенный на рисунке 5.2.3.1.

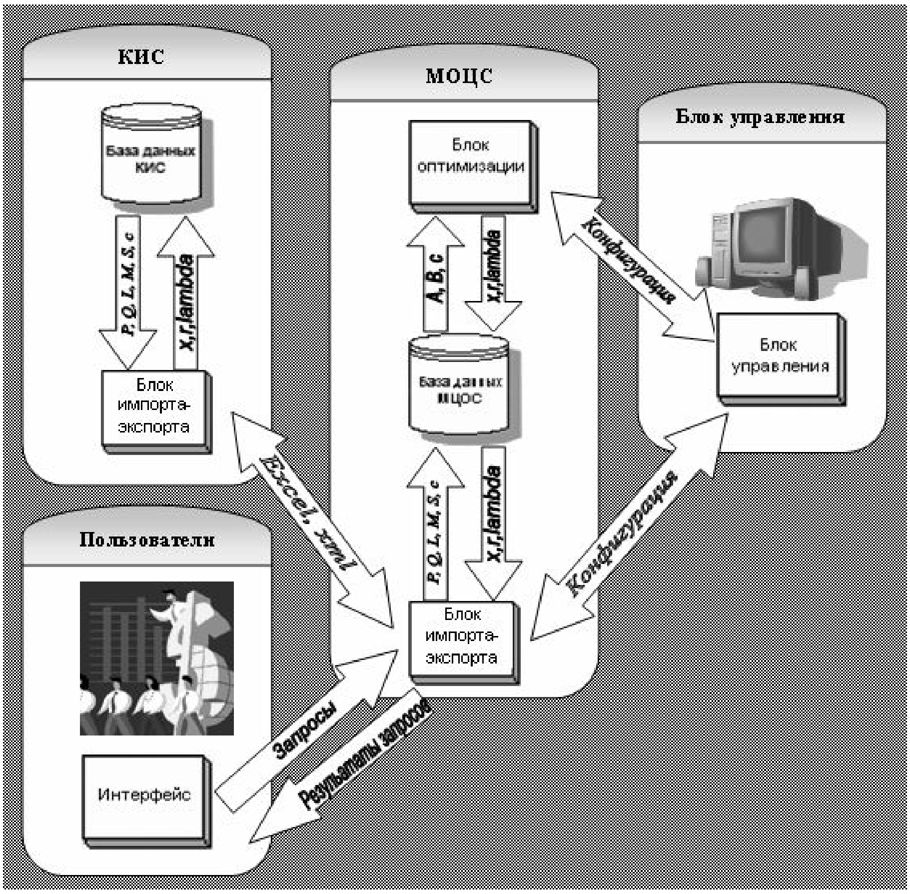

Рисунок 5.2.3.1. Архитектура системы оптимизации

В состав системы входит КИС (корпоративная информационная система) с блоком загрузки-выгрузки данных для файлов формата `Excel` и `xml`; модуль оптимизации ценовой стратегии, состоящий из блока загрузки-выгрузки данных для файлов формата `Excel` и `xml`. Доступ пользователей к модулю осуществляется посредством интерфейса. Конфигурация системы производится с помощью блока управления. Данный блок позволяет администратору системы задавать конфигурацию файлов для импорта-экспорта и управлять другими параметрами системы, устанавливать права доступа пользователей к данным.

Информация, полученная из КИС, обрабатывается блоком загрузки-выгрузки модуля. Данный блок формирует на основе файлов импорта соответствующие `SQL`-запросы на добавление или изменение к базе данных, где основной информацией являются значения матриц P, Q, L, M, S и c. Блок оптимизации получает из базы данных значения матриц A, B и c. Результаты оптимизации и анализа продуктивности записываются в базу данных. Пользователь работает с системой посредством интерфейса. Он может вносить изменения в закачанные из КИС информацию, а также получать необходимые отчеты. Закачанные в КИС результаты работы модуля могут быть далее изменены менеджером при необходимости.

При запуске программы пользователь вводит свой логин и пароль. Работа пользователей регистрируется в журнале операций. В системе поддерживается каскадное удаление данных по каждой операции.

5.3. Программа оптимизации

Основным классом, используемым при оптимизации является класс TNeumann (см. приложение А). Его свойствами являются:

- **int nRows_N -** число продуктов *n;*
- **int nCols_N -** число стратегий *m;*
- **TMatrix *A** - матрица затрат *A*;
- **TMatrix *B** - матрица выпуска *B*;
- **TMatrix *x** - вектор интенсивностей *x*;
- **TMatrix *p** - вектор двойственных переменных;
- **double lambda_up** - текущая верхняя граница интервала поиска числа Фробениуса;
- **double lambda_low** - текущая нижняя граница интервала поиска числа Фробениуса;
- **double lambda_tmp** - текущая оценка числа Фробениуса;
- **int iteration** - количество итераций, сделанных на данный момент;
- **int slvng_N** – флаг выполнения, имеет значение `-1`, если число Фробениуса равно `1;` имеет значение `1,` если число Фробениуса меньше `1;` имеет значение `0,` если оптимизация не была проведена; имеет значение `-2`, если число Фробениуса больше `1` или меньше `0;` имеет значение `3`, если существует седловая точка;
- **double precision** – точность расчета

У данного класса следующие методы:

1) конструкторы:

1.1) **`TNeumann(void)`** - конструктор без параметров;

1.2) **`TNeumann(int,int)`** - конструктор с заданным числом продуктов и стратегий;

1.3) **`TNeumann(int,int,TMatrix&,TMatrix&)`** - конструктор с заданным числом продуктов и стратегий и матрицами A и B;

1.4) **`TNeumann(char*)`** - конструктор с загрузкой из текстового файла с табуляциями;

2) **`TNeumann(void)`** - деструктор;

3) функции вывода:

3.1) **`friend int GetNRows(const TNeumann&)`** - функция для вывода числа продуктов;

3.2) **`friend int GetNCols(const TNeumann&)`** - функция для вывода числа продуктов;

3.3) **`double& GetLambdaUp(void)`** - функция для вывода верхней текущей границ интервала для числа Фробениуса;

3.4) **`double& GetLambdaLow(void)`** - функция для вывода нижней текущей границ интервала для числа Фробениуса;

3.5) **`double& GetA(int,int)`** - функция для вывода матрицы A;

3.6) **`TMatrix& GetA(void)`** - функция для вывода матрицы A;

3.7) **`double& GetB(int,int)`** - функция для вывода матрицы B;

3.8) **`TMatrix& GetB(void)`** - функция для вывода матрицы B;

3.9) **`double& GetX(int)`** - функция для вывода вектора x;

3.10) **`TMatrix& GetX(void)`** - функция для вывода вектора x;

3.11) **`double& GetP(int)`** - функция для вывода вектора двойственных оценок продуктов;

3.12) **`TMatrix& GetP(void)`** - функция для вывода вектора двойственных оценок продуктов;

4) функции ввода:

4.1) **void PutVariables(double*, double*)** - функция для присваивания значений переменным;

4.2) **void PutA(double, int, int)** - функция для ввода матрицы *A*;

4.3) **void PutB(double, int, int)** - функция для ввода матрицы *B*;

5) **void MakeGame(double const)** – основной метод оптимизации;

6) **void MinIntervalLambda(void)** – метод расчета для матрицы отношений.

Метод **MakeGame** является программной реализацией алгоритма представленного в разделе 3.2. В ходе работы он может вызвать метод **MinIntervalLambda.**

Класс **TMatrix** позволяет производить различные операции с матрицами, такими как сложение, умножение, транспонирование, вычисление максимального элемента, вывод элементов матриц.

Класс **TLinProb** применяется для решения задач линейного программирования. Он используется внутри метода **MakeGame**.

5.4. Структура базы данных

Таблицы `var_type`, `oper_type`, `access_type`, `oper`, `result`, `rent` являются системными, т.е. из приложения пользователь данные этих таблиц не может изменять. Таблицы `strategy`, `product`, `ed_izm` являются справочниками, которые могут вводиться вручную или закачиваться с помощью операций импорта. Таблица `users` хранит данные о пользователях (пароль, права доступа) [28].

Таблицы `var_cost`, `const_cost`, `sales` содержит данные о нормах прямых затрат, данные о постоянных затратах и объемах продаж соответственно. В них данные можно вводить после заполнения справочников. Также возможен импорт.

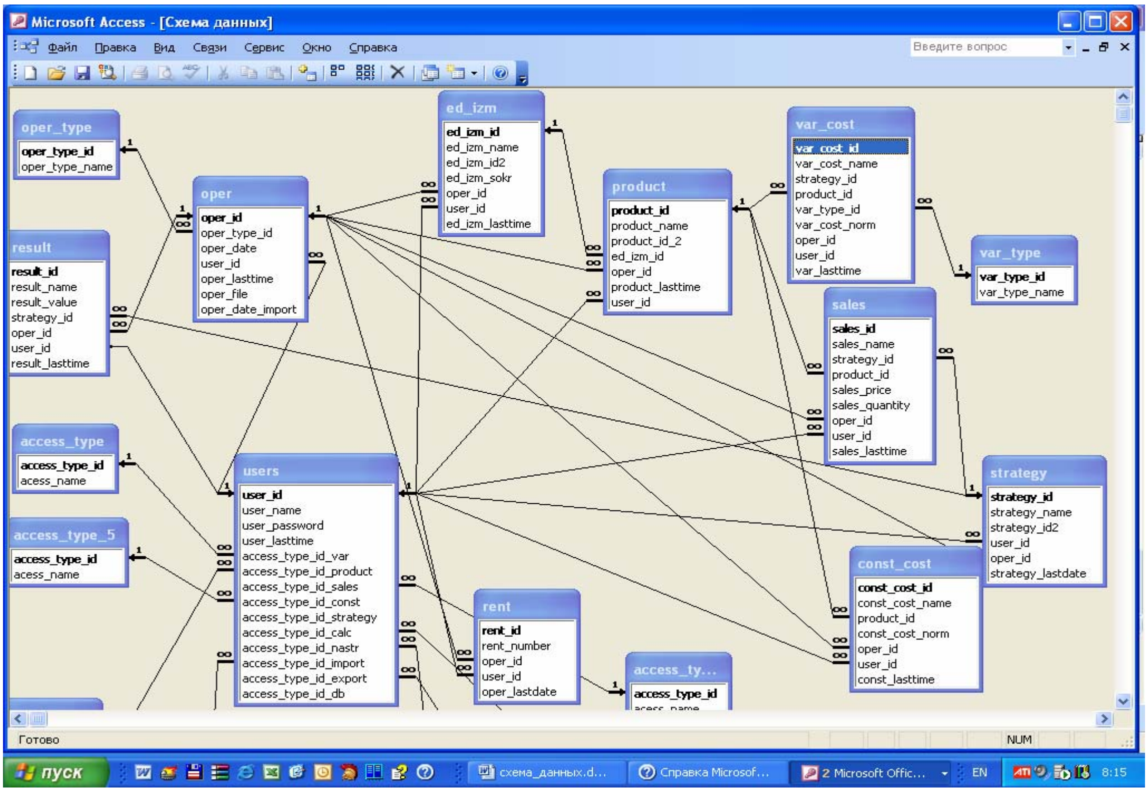

Рис. 5.4.1. Схема базы данных.

Информация о полях таблиц базы данных [47] представлена ниже (см. табл. 5.4.1).

Табл. 5.4.1. Информация о полях таблиц базы данных бюджетирования

Таблица	Поле	Тип данных	Описание
var_type **(тип прямых затрат)**	var_type_id	Счетчик (ключевое поле)	Номер типа прямых затрат
	var_type_name	Строка (255 символов)	Наименование типа прямых затрат
oper_type **(тип операций)**	oper_type_id	Счетчик (ключевое поле)	Номер типа операций
	oper_type_name	Строка (255 символов)	Наименование типа операций
access_type **(права доступа)**	access_type_id	Счетчик (ключевое поле)	Номер права доступа
	access_name	Строка (255 символов)	Наименование права доступа

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
oper **(журнал регистрации операций)**	oper_id	Счетчик (ключевое поле)	Номер операции
	oper_type_id	Числовое	Идентификатор типа операции (ключевое поле из таблицы oper_type)
	oper_date	Дата/время	Дата создания операции
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	oper_lasttime	Дата/время	Дата последнего изменения операции
	oper_file	Строка (255 символов)	Название файла импорта
	oper_date_import	Дата/время	Дата импорта
result **(результаты оптимизации)**	result_id	Счетчик (ключевое поле)	Номер результата
	result_name	Строка (255 символов)	Наименование результата
	result_value	Числовое	Значение результата
	strategy_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы strategy)
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	result_lasttime	Дата/время	Дата последнего изменения результата

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
rent (данные о рентабельности)	rent_id	Счетчик (ключевое поле)	Номер рассчитанного значения
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	oper_lastdate	Дата/время	Дата последнего изменения рентабельности
const_cost (данные о постоянных затратах)	const_cost_id	Счетчик (ключевое поле)	Номер нормы для постоянных затрат
	const_cost_name	Строка (255 символов)	Наименование нормы для постоянных затрат
	product_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы product)
	const_cost_norm	Числовое	Значение нормы с точностью до 2 знаков после запятой
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	const_lasttime	Дата/время	Дата последнего изменения нормы постоянных затрат

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
strategy **(справочник стратегий)**	strategy_id	Счетчик (ключевое поле)	Номер стратегии
	strategy_name	Строка (255 символов)	Наименование стратегии
	strategy_id2	Строка (255 символов)	Номер стратегии в файле импорта
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	strategy_lastdate	Дата/время	Дата последнего изменения данных о стратегии
product **(справочник готовой продукции и услуг)**	product_id	Счетчик (ключевое поле)	Номер продукта
	product_name	Строка (255 символов)	Наименование продукта
	product_id2	Строка (255 символов)	Номер продукта в файле импорта
	ed_izm_id	Числовое	Идентификатор единицы измерения (ключевое поле из таблицы ed_izm)
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	product_lasttime	Дата/время	Дата последнего изменения данных о продукте
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
ed_izm (справочник единиц измерения)	ed_izm_id	Счетчик (ключевое поле)	Номер единицы измерения
	ed_izm_name	Строка (255 символов)	Наименование единицы измерения
	ed_izm_id2	Строка (255 символов)	Номер единицы измерения в файле импорта
	ed_izm_sokr	Строка (255 символов)	Сокращение для единицы измерения
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	ed_izm_lasttime	Дата/время	Дата последнего изменения данных о единицах измерения
users (данные о пользователях)	user_id	Счетчик (ключевое поле)	Номер пользователя
	user_name	Строка (255 символов)	Наименование пользователя
	user_password	Строка (255 символов)	Пароль (закодированный)
	access_type_id_var	Числовое	Идентификатор доступа к данным о прямых затратах (ключевое поле из таблицы access_type)

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
users (данные о пользователях)	access_type_id_product	Числовое	Идентификатор доступа к справочнику готовой продукции (ключевое поле из таблицы access_type)
	access_type_id_sales	Числовое	Идентификатор доступа к данным о спросе (ключевое поле из таблицы access_type)
	access_type_id_const	Числовое	Идентификатор доступа к нормам постоянных затрат (ключевое поле из таблицы access_type)
	access_type_id_strategy	Числовое	Идентификатор доступа к справочнику стартегий (ключевое поле из таблицы access_type)
	access_type_id_calc	Числовое	Идентификатор доступа к данным оптимизации (ключевое поле из таблицы access)
	access_type_id_nastr	Числовое	Идентификатор доступа к настройкам (ключевое поле из таблицы access)
	access_type_id_import	Числовое	Идентификатор доступа к операциям импорта (ключевое поле из таблицы access_type)
	access_type_id_export	Числовое	Идентификатор доступа к операциям экспорта (ключевое поле из таблицы access_type)

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
users (данные о пользователях)	access_type_id_db	Числовое	Идентификатор доступа к операциям работы с файлом БД (ключевое поле из таблицы access_type)
var_cost (данные о прямых затратах)	var_cost_id	Счетчик (ключевое поле)	Номер нормы для прямых затрат
	var_cost_name	Строка (255 символов)	Наименование нормы для прямых затрат
	strategy_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы strategy)
	product_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы product)
	var_type_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы var_type)
	var_cost_norm	Числовое	Значение нормы с точностью до 2 знаков после запятой
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	var_lasttime	Дата/время	Дата последнего изменения нормы прямых затрат

Продолжение табл. 5.4.1

Таблица	Поле	Тип данных	Описание
sales (данные об объемах продаж и ценах)	sales_id	Счетчик (ключевое поле)	Номер данных по продажам
	sales_name	Строка (255 символов)	Наименование данных по продажам
	strategy_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы strategy)
	product_id	Числовое	Идентификатор стратегии (ключевое поле из таблицы product)
	sales_price	Числовое	Значение цены с точностью до 2 знаков после запятой
	sales_quantity	Числовое	Значение объема продаж с точностью до 2 знаков после запятой
	oper_id	Числовое	Идентификатор операции (ключевое поле из таблицы oper)
	user_id	Числовое	Идентификатор пользователя (ключевое поле из таблицы user)
	sales_lasttime	Дата/время	Дата последнего изменения данных о продажах

5.5. Импорт-экспорт данных

В МЦОС предусмотрено два формата для автоматического импорта-экспорта:

1) файлы XML;

2) книга Microsoft Excel.

Далее рассмотрим особенности и программную реализацию для данных форматов.

5.5.1. Формат XML

Разработка XSD-схемы для XML-файлов

В состав инструментов Microsoft.NET/SDK входит утилита xsd.exe, позволяющая создавать xsd-файл на основе эталонного xml-файла. Затем полученный файл структуры xsd дорабатывается [66].

Например, для справочника готовой продукции (см. рисунок 5.5.1.1) файл spr_GP.xsd будет иметь вид (см. рисунок 5.5.1.2).

```
<?xml version="1.0" encoding="Windows-1251"?>
<spr vid="product"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="spr_gp.xsd">
    <spr_item rewrite="true">
                <KLS_NUM>1</KLS_NUM>
                <KLS_NAME>Платные объявления</KLS_NAME>
                <USER_ID>1</USER_ID>
    </spr_item>
    <spr_item rewrite="true">
                <KLS_NUM>2</KLS_NUM>
                <KLS_NAME>Объявления по
купонам</KLS_NAME>
                <USER_ID>1</USER_ID>
                <ED_IZM>1</ED_IZM>
    </spr_item>
</spr>
```

Рисунок 5.5.1.1. Пример XML-файла для справочника ГП

```
я╗┐<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="spr">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="spr_item" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
         <xs:all>
              <xs:element name="KLS_NAME" type="xs:string" />
              <xs:element name="KLS_NUM"  minOccurs="0" type="xs:string" />
              <xs:element name="USER_ID"  minOccurs="0" type="xs:string" />
              <xs:element name="ED_IZM"  minOccurs="0" type="xs:string" />
         </xs:all>
           <xs:attribute name="rewrite" type="xs:boolean" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="vid" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
         <xs:enumeration value="product"/>
       </xs:restriction>
      </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Рисунок 5.5.1.2. Файл `spr_GP.xsd` (схема данных для файла справочника готовой продукции).

Первые две строчки содержат информацию о стандарте xsd-файла.

```
я╗┐<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
```

В теге `<xs:element>` указывается узел дерева для описания, атрибут `type` показывает простой (примитивный) тип данных (если узел содержит текст) или тип данных описанных в другом месте пространства имен. Для текстовых узлов type может иметь значение `xs:string`, `xs:boolean`, `xs:float`, `xs:dateTime` и др.

Тег `<xs:complexType>` применяется, если данный элемент xml-дерева имеет детей. Тег `<xs:sequence>` определяет строгую последовательность дочерних элементов (в данном случае для `spr`):

```
<xs:sequence>
  <xs:element name="spr_item" minOccurs="1" maxOc-
curs="unbounded">
```

В нашем случае дочерним элементом для spr может быть только spr_item. Атрибут minOccurs определяет минимально возможное повторений данного дочернего узла в родительском; maxOccurs – максимально возможное (если он равен "unbounded", то число неограниченно).

Тег <xs:all> показывает, что в данном узле будут все описываемые ниже дочерние элементы в количестве от minOccurs до maxOccurs (если эти атрибуты не указаны, то они по умолчанию равны 1). В нашем примере строки

```
    <xs:all>
          <xs:element name="KLS_NAME" type="xs:string" />
          <xs:element name="KLS_NUM"  minOccurs="0"
type="xs:string" />
          <xs:element name="USER_ID"  minOccurs="0"
type="xs:string" />
          <xs:element name="ED_IZM"  minOccurs="0"
type="xs:string" />
    </xs:all>
```

показывают, что узел spr_item должен обязательно иметь один дочерний узел KLS_NUM и может содержать (и не содержать) узлы KLS_NAME, USER_ID, ED_IZM.

Тег <xs:attribute> применяется для описания атрибута данного узла. Параметр type задается так же, как для <xs:element>. В противном случае после тега <xs:attribute> вводится тег <xs:simpleType>. Например, тег <xs:attribute name="rewrite" type="xs:boolean" /> показывает, что атрибут rewrite для spr_item может принимать значения «true» и «false». Однако для корневого узла spr тип атрибута vid

```
<xs:attribute name="vid" use="required">
<xs:simpleType>
 <xs:restriction base="xs:string">
   <xs:enumeration value="product"/>
 </xs:restriction>
</xs:simpleType>
</xs:attribute>
```

не будет примитивным. Его значение будет не только строковым, оно будет строго равно «product» для справочника ГП. Атрибут use имеет значение "required", т.к. наличие данного атрибута для spr является обязательным. Внутри тега <xs:simpleType> должен находиться один из трех тегов <xs:restriction> (ограниченный тип данных), <xs:list> (список значений), <xs:annotation> (документация к описанию) и <xs:union> (объединение ограниченных типов). В нашем примере используется тег <xs:restriction>. Атрибут base описывает тип данных для атрибута, который будет ограниваться посредством тега <xs:restriction>. Тег <xs:enumeration> применяется, если необходимо перечислить, какие значения может принимать атрибут (примеры других xsd-схем см. в приложении Г).

Чтобы была проведена проверка структуры xml-файла с использованием xsd-файла:

1) если информация об xsd-файле не указана в xml-файле, указать расположение данного файла, присвоив свойству SchemaRef объекту TXMLDocument строку с путем;

2) для свойства ParseOptions (настройки для парсера) присвоить значение true для полей:

   2.1) poResolveExternals – внешние структуры данных (описания пространств имен, внешние DTD- и xsd-схемы и др. описания для проверки целостности) будут подключены во время загрузки xml-структуры парсером;

   2.2) poValidateOnParse – парсер анализирует xml-структуру в соответствии с его схемой с учетом корректности xml-структуры как таковой;

3) необходимо активировать объект класса TXMLDocument (привоить свойству Active значение true).

Если парсер обнаружит ошибку в структуре xml-файла, то он сгенерирует исключение типа EXMLDocError. В сообщении будет указана ошибка (в каком теге произошел сбой чтения). Для того чтобы перехватить данное исключение, целесообразно применить блок try…catch (см. рисунок 5.5.1.3 и приложение В) [15].

```
    try
    {
XMLDocument1->Active=true;
// попытка активировать xml-документ
// во время активации парсер анализирует структуру данного файла
    }
    catch(Exception &e)
// если произошла ошибка в анализе документа,
// то генерируется исключение, которое перехватывается
    {
       MessageBox( 0, StrCat("Ошибка в выборе файла XML:
\n",e.Message.c_str()), "Ошибка", MB_OK );
       // в сообщении выводится текст, описывающий ошибку
    }
```

Рисунок 5.5.1.3. Структура обработки ошибки структуры XML-файла

Если произойдет хотя бы одна ошибка при загрузке парсером, то XML-документ не будет активирован. Сообщение о первой встретившейся ошибке будет выглядеть следующим образом (см. рисунок 5.5.1.4).

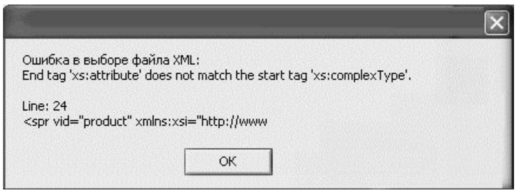

Рисунок 5.5.1.4. Сообщение об ошибке загрузки XML-файла парсером

Работа с XML-документами в проекте C++ Builder 2007

В `C++ Builder 2007` для работы с `XML` применяется класс `TXMLDocument` (группа компонент `Internet`).

Используя `TXMLDocument`, можно прочитать данные `XML` из существующего файла, загрузить `XML`-документ из строки или создать новый, пустой `XML`-документ [1]. Можно напрямую применять этот класс для загрузки `XML`-документа, его чтения, изменения и сохранения в нем изменений. Свойство `DocumentElement` дает доступ к корневому элементу `XML`-документа, с помощью чего получить доступ к дочерним элементам (узлам). Дочерние узлы затем можно добавлять, удалять или изменять. Класс `TXMLDocument` реализует интерфейс `IXMLDocument`. Приложения получают доступ к этому интерфейсу через свойства и методы класса `TXMLDocument` или через узлы `XML`-документа.

TXMLDocument использует внешний парсер DOM (Document Object Model) для анализа структуры. Парсер как правило назначется посредством свойства DOMVendor. Когда свойство Active имеет значение true, TXMLDocument применяет данный парсер для анализа и представления структуры XML-документа для его проверки и модификации.

Технология DOM (Document Object Model) представляет собой набор стандартных интерфейсов для представления XML-документа и работы с ним. Эти интерфейсы разработаны различными сторонними поставщиками. Если пользователь не хотел бы использовать поставщика по умолчанию, то в C ++ Builder можно зарегистрировать дополнительных разработчиков DOM-интерфейсов. Файл XMLDOM содержит описания всех DOM-интерфейсов, описанных на 2 уровне XML DOM спецификации W3C [20, 53, 65, 69]. Каждый разработчик программного обеспечения для DOM должен внедрить все данные интерфейсы.

Для того чтобы использовать продукцию разработчиков DOM, которая поддерживается C++ Builder, необходимо найти заголовочный файл, с помощью которого внедряются DOM-интерфейсы. Название такого файла обычно заканчивается строкой 'xmldom.' Например, для разработок Microsoft используется файл MSXMLDOM. Если подключить нужный заголовочный файл к проекту, то DOM-интерфейсы автоматически регистрируются и их можно применять в программе.

Для работы XML-документом нужно выполнить следующие шаги [56]:

- Добавить компонент TXMLDocument (вкладка Internet панели инструментов) в форму или модуль данных.
- Присвоить значение свойству DOMVendor для определения метода анализа структуры XML.
- Если XML-документ уже существует, то указать месторасположение файла в свойстве FileName; или если он представляет собой строку, задать эту строку в качестве значения свойства XML.

- Присвоить свойству Active значение true.

Если объект класса TXMLDocument активен, то можно обходить узлы XML-дерева, считывая и меняя их значения. Для возврата к корневому элементу можно использовать DocumentElement.

После загрузки структуры посредством DOM-парсера, данные XML-документа будут представлять собой иерархическую структуру в виде дерева, состоящего из узлов. Для парсера все размеченные элементы внутренними узлами. Т.е. текстовые значения расположенные между тэгами будут восприниматься как дочерние узлы элемента, к которому относится данный тэг.

Доступ к каждому узлу осуществляется через интерфейс XMLIntf.IXMLNode, начиная с коревого узла, доступ к которому осуществляется посредством свойства TXMLDocument.DocumentElement. Интерфейс IXMLNode позволяет определить, имеет ли узел дочерние узлы или он является конечным с текстовым значением, проверяя значение свойства IXMLNode.IsTextElement. Если это конечный текстовый элемент, то можно прочитать или изменить его значение, используя свойство IXMLNode.Text. В противном случае можно получить доступ к его дочерним узлам, используя свойство IXMLNode.ChildNodes. Это свойство типа _di_IXMLNodeList возвращает указатель на список дочерних узлов. Таким образом, для того чтобы вывести текстовые данные XML-документа, необходимо провести обход XML-дерева, представленный на рисунке 5.5.1.5 и 5.5.1.6 (листинг см. в приложении В).

Чтобы узнать, сколько дочерних узлов у текущего узла, можно использовать свойство Count (типа int) класса IXMLNodeList. Доступ к дочернему узлу дерева с необходимым номером можно осуществить через функцию GetNode(), аргументом которой является индекс дочернего узла типа int (первый узел будет иметь индекс 0, второй – 1 и т.д.). Если заранее известна правильная структура XML-файла, то для поиска дочернего узла с необходимым именем можно использовать функцию FindNode() класса IXMLNodeList. Существует три варианта для состава аргументов для этой функции. В двух из них в качестве первого аргумента типа `WideString` выступает имя узла. Эту функцию удобно применять, когда порядок следования дочерних узлов не является строгим (например, для `XML`-файла с данными по видам готовой продукции неважно, в каком порядке следуют теги `KLS_NUM`, `KLS_NAME`, `USER_ID`, `ED_IZM_ID` внутри тега `spr_item`). Исходя из этого процеес импорта справочника ГП будет иметь вид (см. рисунки 5.5.1.7 и 5.5.1.8).

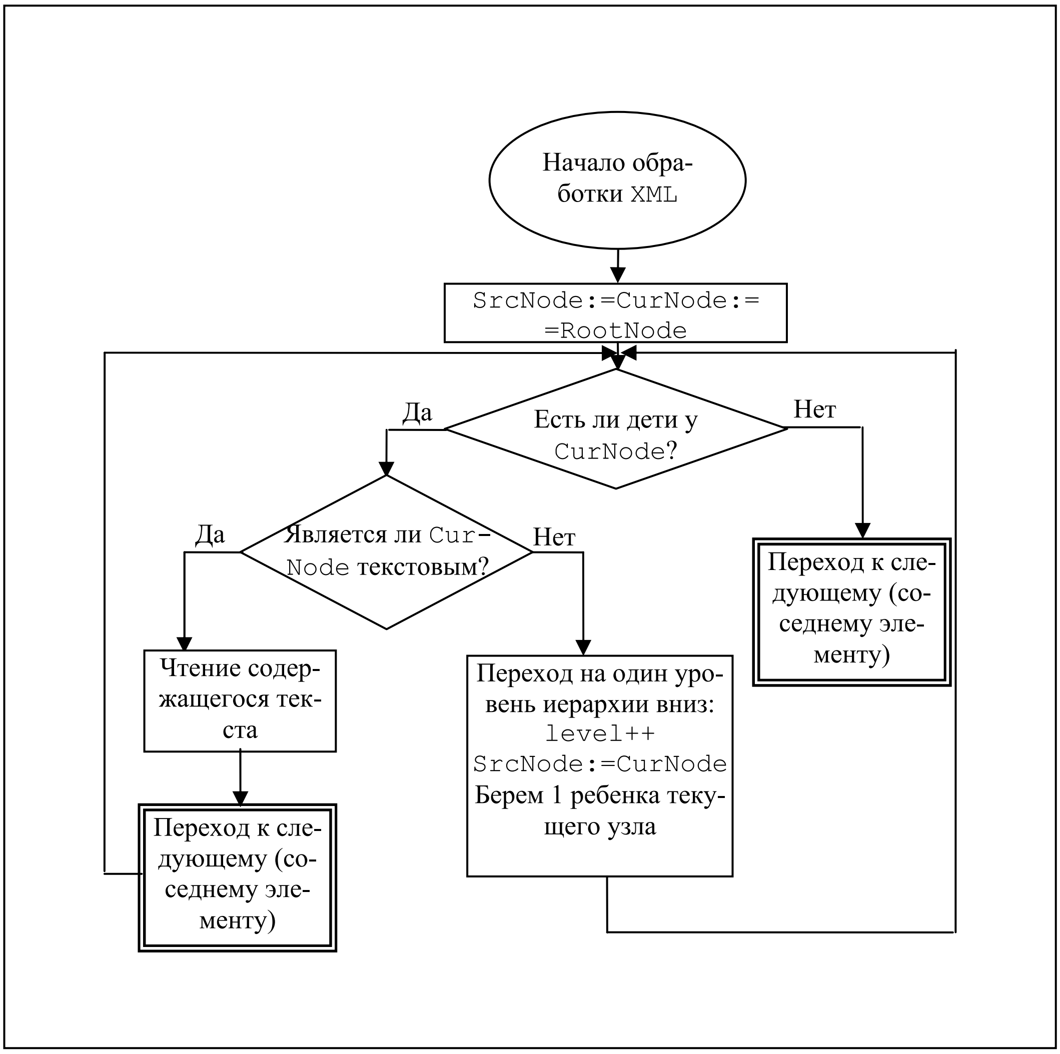

Рисунок 5.5.1.5. Схема обхода дерева для чтения текста узла

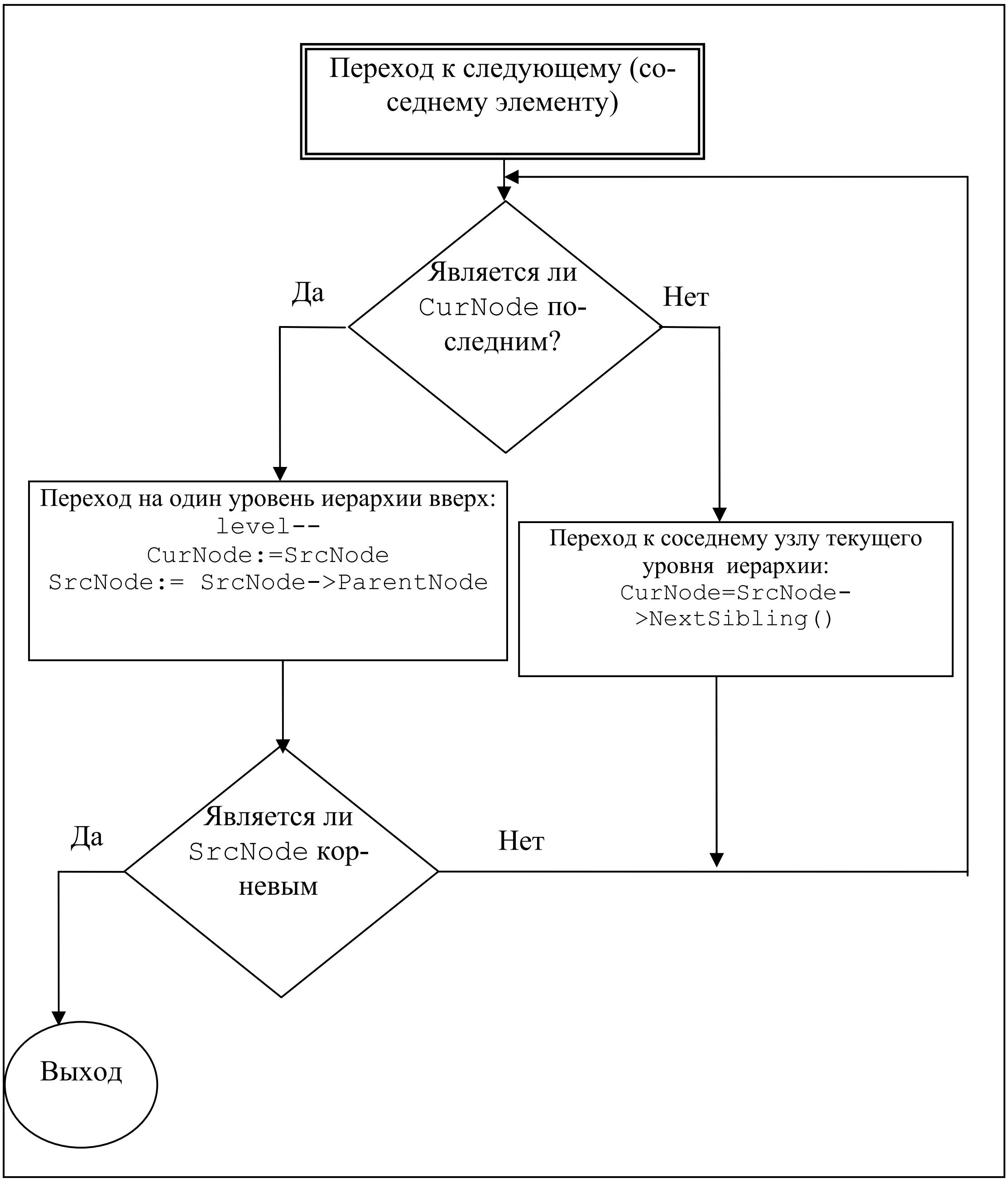

Рисунок 5.5.1.6. Схема алгоритма поиска ближайшего соседа

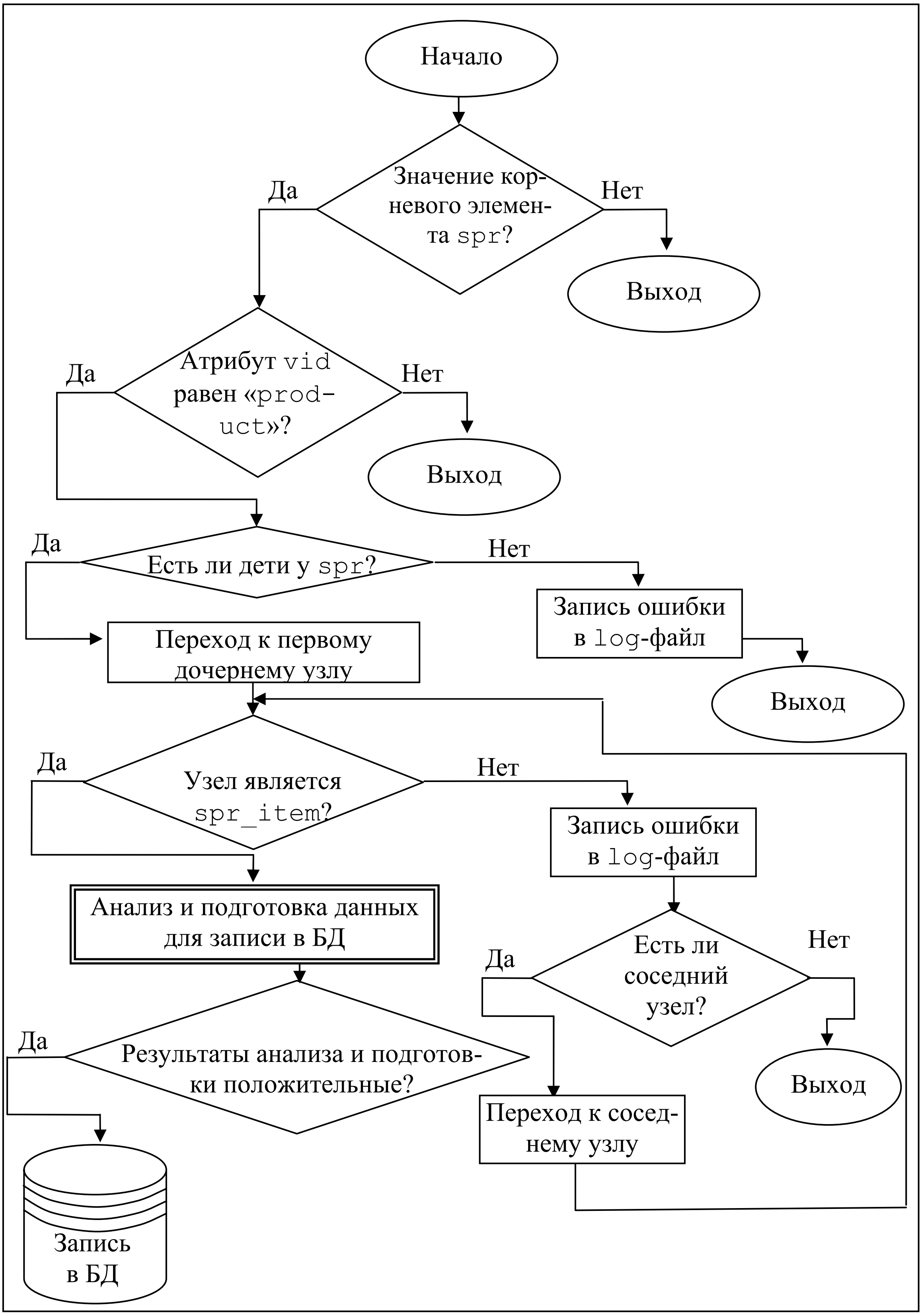

Рисунок 5.5.1.7. Схема импорта из xml – файла для справочника ГП

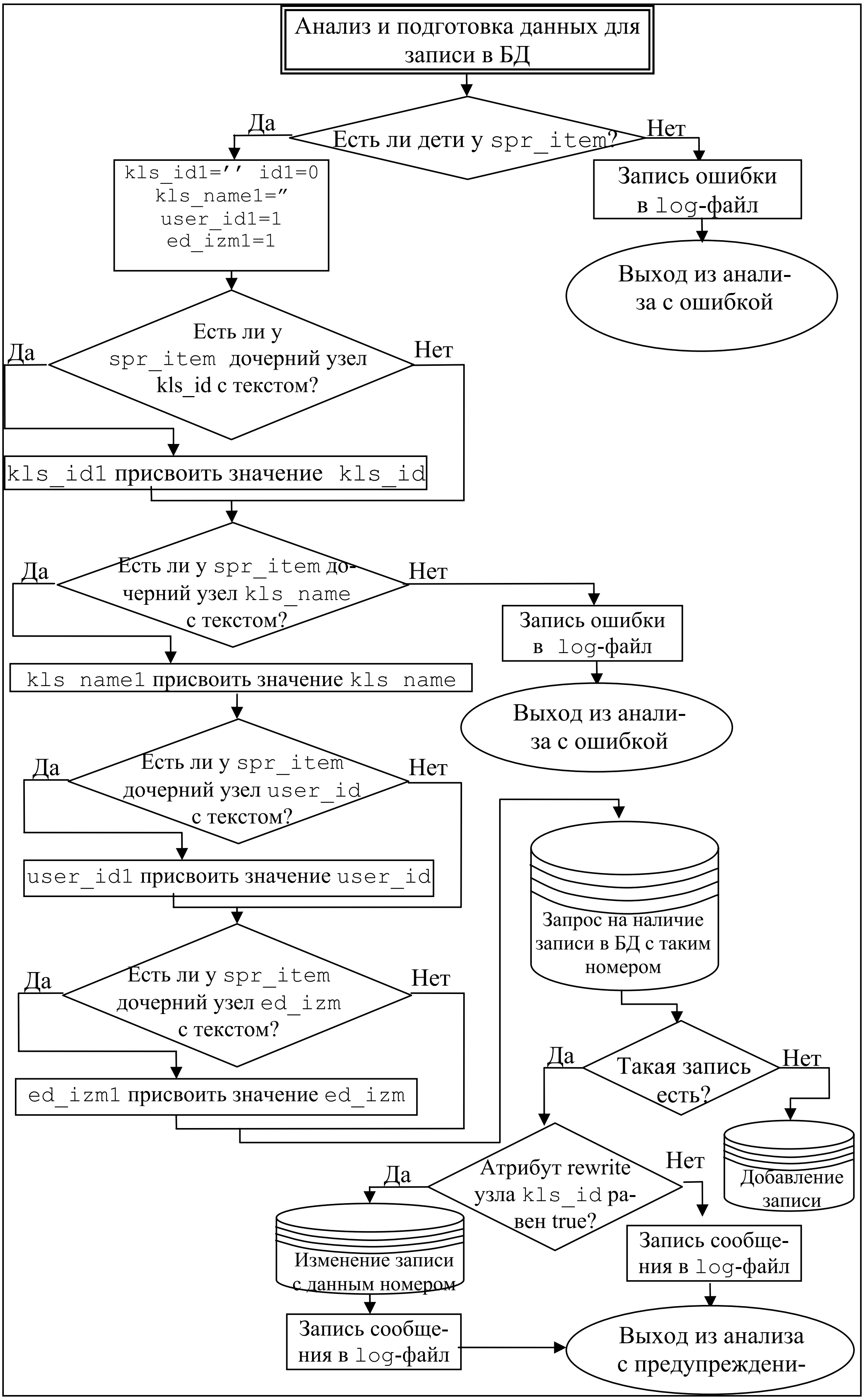

Рисунок 5.5.1.8.Схема анализа и подготовки данных для записи в БД

5.5.2. Формат Microsoft Excel

В C++ Builder 2006 реализованы два основных способа для работы с файлами Microsoft Excel [13, 46]:

- посредством раннего связывания, применяя объект класса TExcelApplication (из закладки Office XP на панели инструментов);
- посредством позднего связывания - подключение к книге Microsoft Excel через **OLE**-технологию (OLE – Object Linking & Embedding).

Первый способ разработчики из корпорации Borland считают более эффективным. Именно этот подход по их словам получит дальнейшее развитие в следующих версиях C++ Builder [64]. Однако он имеет ряд недостатков:

- предназначен для работы с конкретной версией Microsoft Office (например, в C++ Builder 2006 поддерживается версия Office XP, а в C++ Builder 6 – версия Office 2000);
- сложность в работе с функциями: необходимо указывать значения для всех параметров функции (для неиспользуемых параметров используется переменная EmptyParam);
- применение специфических типов данных, несвойственных стандартным типам данных, использующихся в проектах C++ Builder для Windows-приложений (например, wchar);
- существенное замедление работы редактора исходного кода и снижение скорости компиляции.

На практике OLE-технология применяется программистами гораздо чаще, чем первая в силу следующих преимуществ [46, 67]:

- более стабильное подключение к книгам Excel;
- более удобный способ работы с методами и свойствами приложения Excel посредством методов OlePropertyGet, OlePropertySet, Exec;
- применение стандартных типов данных;
- более быстрая компиляция.

В большинстве тиражируемых программ используется технология OLE.

Единственным недостатком второго способа является более медленная работа приложения в режиме Runtime, т.к. приложение постоянно посылает вызовы GetIDsOfNames к серверу Excel для получения идентификаторов методов и свойств по их именам, в то время как при использовании технологии COM вызовы GetIDsOfNames не используются, вместо имен методов и свойств уже поставлены их идентификаторы (см. раздел 5.5.3) [5].

Таким образом, первый способ наиболее подходит при сложной обработке книги Excel, а второй – для рутинных операций импорта-экспорта [23].

В обеих технологиях используется класс Variant. Объекты данного класса могут содержать значения различного типа (числовые, строковые, даты), т.е. используются как универсальные контейнеры. В частности они могут применяться для хранения объектов типа OLE Automation.

Для того, чтобы работать с OLE-объектами в C++ Builder 2006, необходимо подключить файл OleServer.hpp (в более ранних версиях ComObj.hpp). Работа с Excel посредством COM-технологии в C++ Builder 2006 требует подключения Excel_XP_srvr.h.

Так как в проекте бюджетирования импорт в Excel и экспорт являются важными операциями, целесообразно реализовать их с помощью класса TExcelServer (см. листинг в приложении Б). Свойства данного класса являются закрытыми (private), относятся к типу данных Variant:

- **VarExcel** – контейнер OLE-объекта, связанного с экземпляром сервера Excel, с которым работает основная программа;
- **VarBook** - контейнер OLE-объекта, связанного с текущей книгой, открытой на данном сервере Excel;
- **VarSheet** - контейнер OLE-объекта, связанного с текущим листом данной книги;
- **VarCell** - контейнер OLE-объекта, связанного с текущей ячейкой данного листа.

Кроме того, есть два свойства, показывающие характер работы с книгой:

- **`bool write`** - `true` если для записи, `false` для чтения;
- **`int flag`** - текущее состояние подключения к `Excel`.

Класс `TExcelServer` описывается следующими методами:

1) `bool ExcelOpen(AnsiString &BookName, bool write_mode)` - открывает Excel и рабочую книгу по имени (на чтение или на запись);

2) `bool ExcelClose()` - закрывает Excel;

3) `bool SheetOpen(AnsiString &SheetName)` - открывает лист рабочей книги по названию;

4) `bool SheetOpen(int SheetNum)` - открывает лист рабочей книги по названию;

5) `bool CellSet(int CellRow, int CellCol, float CellValue)` - заносит числовое значение в ячейку с указанным адресом;

6) `bool CellSet(int CellRow, int CellCol, AnsiString &CellText)` - заносит строку в ячейку с указанным адресом;

7) `bool CellSet(int CellRow, int CellCol, AnsiString& CellTextDate)` - заносит дату в ячейку с указанным адресом;

8) `AnsiString CellGet(int CellRow, int CellCol)` – возвращает текстовое значение ячейки с указанным именем;

9) `bool CellBorderSet(int CellRow, int CellCol, int Border_Code, int Line_Style )` - рисует рамку ячейки;

10) `bool CellFontSet(int CellRow, int CellCol,, int Font_Size)` - изменяет размер шрифта;

11) `bool CellFontSet(int CellRow, int CellCol, AnsiString Font_name )` - изменяет тип шрифта;

12) `bool CellFontStyleSet(int CellRow, int CellCol, int Font_Style)` - изменяет стиль шрифта;

13) `bool CellWrapTextSet(int CellRow, int CellCol)` - сворачивает текст в ячейке;

14) `bool CellVAlignSet(int CellRow, int CellCol, int VAlign_Style)` - вертикальное выравнивание в ячейке;

15) `bool CellHAlignSet(int CellRow, int CellCol, int HAlign_Style)` - горизонтальное выравнивание в ячейке;

16) `bool RangeMerge(int CellRow1, int CellCol1,int CellRow2, int CellCol2)` - объединение ячеек с указанным адресом;

17) `bool CellFormatSet(int CellRow, int CellCol, AnsiString& CellFormat)` – установка формата для выбранной ячейки.

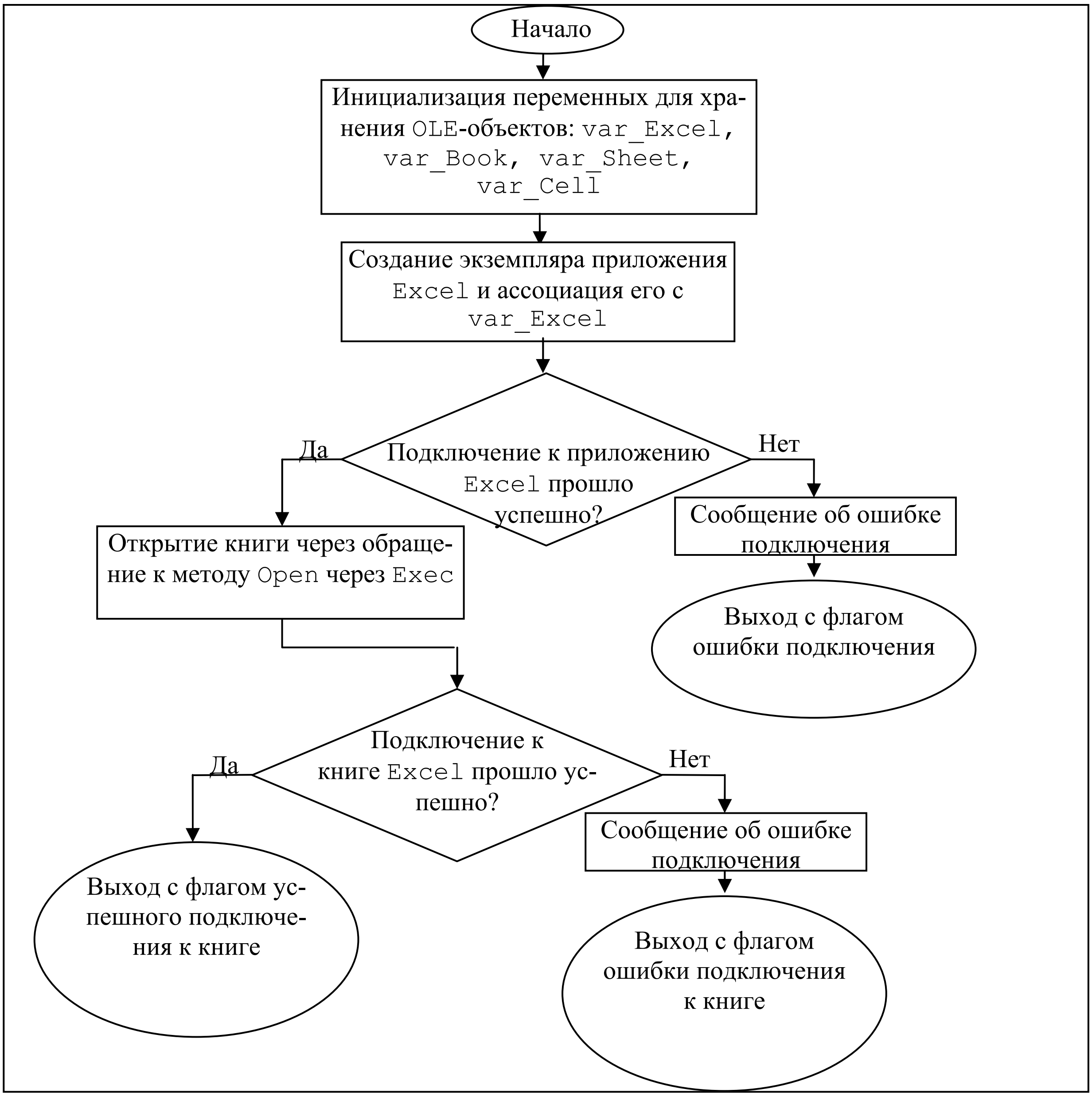

Рисунок 5.5.2.1. Схема подключения к приложению Excel

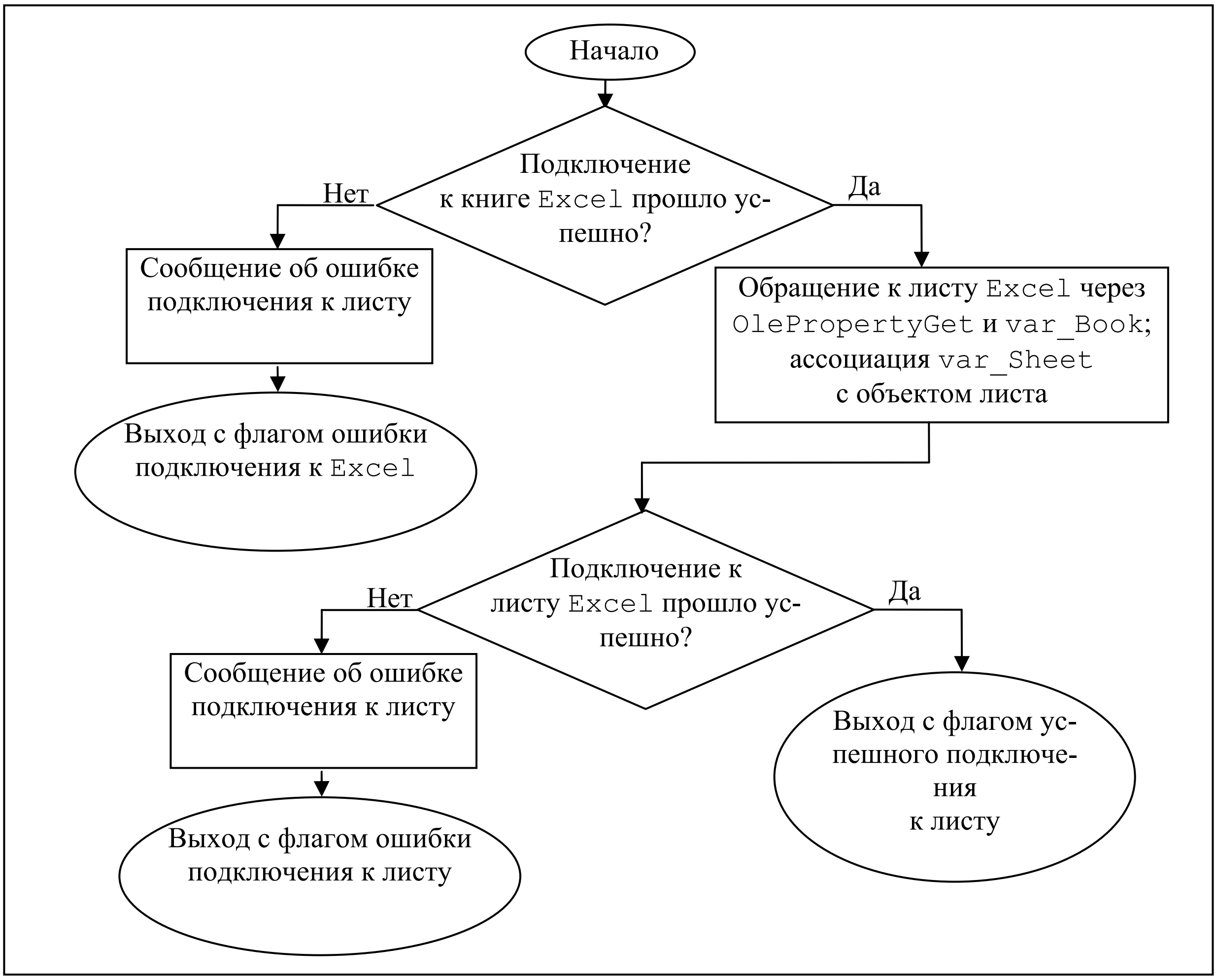

Рисунок 5.5.2.2. Операция подключения к листу

На рисунках 5.5.2.1 и 5.5.2.2 схематично представлены алгоритмы подключения к книге и листу `Excel`. При создании экземпляра приложения запускается сервер `Excel` в скрытом режиме. Затем открывается книга, после чего можно перейти к конкретному листу. Работа с внешним приложением может быть неустойчивой, поэтому необходимо перехватывать структурой `try…catch` исключения, возникающие при сбоях (см. рисунок 5.5.2.3).

```
ExcelServer::ExcelServer()
{
write=false;
try
        {
        flag=0;//начали работу с сервером
        var_Excel = CreateOleObject("Excel.Application");//создаем OLE объект
        flag=1;//если все ОК, флаг на успешное подключение с сервером
        }
catch(...)//ловим исключения
        {
           MessageBox( 0, "Ошибка при запуске сервера Excel", "Ошибка", MB_OK );
           flag=-1; // сообщение об ошибке, флаг на ошибку
        }
}
```

Рисунок 5.5.2.3. Операция создания экземпляра приложения `Excel`

5.6. Выводы и результаты

Разработанные численные методы и алгоритмы использованы в созданном комплексе программ, который позволяет:

1) вести базу данных проекта бюджетирования с обеспечением её целостности;

2) обеспечивать связь (импорт-экспорт данных) с внешними программами в частности с программами `Microsoft Office` и `XML`-парсерами;

3) находить оптимальные параметры проекта бюджетирования;

4) формировать отчетные формы с интерпретацией результатов.

6. Расчет бюджета продаж для предприятия ЧФ ЗАО «Пронто-Уфа»

ЗАО «Пронто-Уфа» (группа газет «Из рук в руки») занимается печатью изданий рекламного характера (газеты частных объявлений, справочника по недвижимости, журнала по строительству и сайта в сети Интернет http://www.irr.ru).

Данное предприятие занимается внедрением системы бюджетирования, поэтому предложенная нами модель оптимизации вызвала интерес.

Перед выполнением данной оптимизации было необходимо:

- провести анализ спроса и предложения на товары и услуги предприятия;
- сегментировать спрос и разработать для каждого сектора ценовую стратегию;
- классифицировать затраты и разработать плановые нормативы;
- выявить ограничения на интенсивности стратегий.

В результате проведенного исследования по данным вопросам была сформирована следующая прогнозная информация для июня 2007 года.

Реклама и объявления в данных изданиях являются платными (см. рисунок 6.1). Объявления от физических лиц в газету «Из рук в руки» принимаются на купонах, которые публикуются в газете. При отсутствии купона или невыполнении условий для частных объявлений физическое лицо также может приобрести место по тарифу для предприятий.

Предприятия заказывают рекламу через менеджеров ЗАО «Пронто-Уфа». Стоимость рекламы зависит от площади, оформления и расположения в печатном издании. Оплата такой рекламы осуществляется либо сразу при оформлении заказа, либо после. В случае просрочки оплаты более чем на месяц к предприятию предъявляются штрафные санкции.

Объем реализации рекламы в газете «Из рук в руки», журнале «Строительство», справочнике «Недвижимость» и на портале в сети Интернет зависит напрямую от площади блока, его характеристик и косвенно от тиража издания [68].

ТАРИФЫ

(без учета НДС, НП и МНР, начало действия - 15.02.07)

ВЫДЕЛЕННЫЕ ОБЪЯВЛЕНИЯ

Выделенные слова

Тип I

Объявления, выделенные курсивом, словом и "звездочкой"

До 20 слов	**250 руб.**
До 40 слов	**360 руб.**

Тип II

Объявления, выделенные словом и "точкой"

До 20 слов	**200 руб.**
До 40 слов	**300 руб.**

Рамки	простая	на черном фоне
до 20 слов	**280 руб.**	**330 руб.**
до 40 слов	**450 руб.**	**400 руб.**

МОДУЛЬНАЯ РЕКЛАМА

Минимальный формат - 3 модуля. Объем одной страницы 108 модулей.

Базовый модуль 40 х 20 мм

закрытый заказ	открытый заказ
300 руб.	**320 руб.**

Изменения в закрытом заказе не допускаются

Изменения в открытом заказе допускаются не позднее 6 дней до публикации.

ИНФОРМАЦИОННЫЕ СТРОКИ В ВИТРИНАХ

(стоимость рекламной строки, одна публикация)

Недвижимости	**120 руб.**
стройматериалов, услуг, автогрузовых перевозок	до 5 слов* **100 руб.**

* При условии единовременногозаказа не менее 5 строк или 5 публикаций

ИНФОРМАЦИОННЫЕ КАРТЫ (не менее 12 публикаций)**400 руб.**

ИНФОРМАЦИОННАЯ БЛОЧНАЯ СТРОКА (не менее 12 публикаций)

"Строительство" (2,25х26,2 см.)	**750 руб.**
"Средства транспорта" (4х12,8 см.)	**800 руб.**
"Дом, сад, огород","Услуги","Трудоустройство"	**500 руб.**
"Путешествия", "Мир отдыха и развлечений","Красота и здоровье"	**600 руб.**
"Сотовая связь"	**600 руб.**
"Компьютерные фирмы", "Мебель"	**700 руб.**
"Оптовая витрина"	**700 руб.**
"Юристы"	**800 руб.**
"Ритуальные услуги"	**400 руб.**
"Металл"	**800 руб.**

НАЦЕНКИ

Размещение на 1-й странице (мин. формат 12 модулей)	**+60%**
Размещение на последней странице (мин. формат 6 модулей)	**+50%**
Размещение вне рубрики по желанию заказчика	**+5%**
Выделение цветом на первой и последней страницах	**+10%**
На 1 стр. внутренних частей газеты	**+20%**

РЕКЛАМА НА САЙТЕ

Вид размещения	**Размер баннера**	**Тариф, долл.США/в неделю** (включая НДС)
В правом поле страницы	100 x 100	**200**
Внизу страницы	610 x 55	**250**
Внизу страницы	200 x 55	**100**

II позиция - Размещение баннеров на первой странице тематического раздела с формой поиска объявлений:

Вид размещения	**Статический баннер**		**Баннер в динамическом блоке**	
Размер баннера	610 x 55	200 x 55	610 x 55	200 x 55
Рубрики:	**Тариф, долл. США/в неделю** (включая НДС)			
Категория "А"	**80**	**50**	**40**	**20**
Категория "В"	**70**	**40**	**30**	**10**
Категория "С"	**35**	**20**	**15**	**8**

Оплата осуществляется в рублях по курсу ЦБ РФ.

Рисунок 6.1. Прайс ЗАО «Пронто-Уфа».

Объем реализации газеты «Из рук в руки» зависит от цены и объема спроса при заданной цене. Тираж газеты должен быть приблизительно равен объему спроса (выходит два раза в неделю). Излишки газеты возвращаются. При большом уровне возврата объем заказов от рекламодателей снижается из-за падения рейтинга газеты. С другой стороны нехватка газеты может привести к повышению недовольства со стороны покупателей. Поэтому желательно, чтобы процент возврата был на уровне 1-3%.

Кроме того, высокий объем реализации газет может привести к значительному росту прямых расходов из-за увеличения числа объявлений. При этом цена на газету может не покрыть этих расходов. Перенасыщение газеты рекламой фирм может снизить привлекательность газеты для её покупателей. Основной доход ЗАО «Пронто-Уфа» получает от продажи газеты и рекламы в ней.

Справочник «Недвижимость» также является платным (выходит раз в месяц). Однако покупка данного справочника не дает право на бесплатный купон. Целевой группой покупателей данного справочника являются физические лица, риэлтерские фирмы и др. юридические лица, заинтересованные в покупке, продаже или обмене квартир.

Цена на рекламу в журнале (выход раз в месяц) определяется спросом на данную продукцию среди рекламодателей и издержками. Бесплатные издания распространяются в точках наибольшей концентрации потенциальных потребителей рекламируемой продукции; платные – через сеть «Роспечать», рекламные компании и кассы ЗАО «Пронто-Уфа».

Затраты данной компании (полный перечень приведен на рис. 6.2) можно разделить на два вида:

- прямые (условно-переменные);
- косвенные (условно-постоянные).

К первому виду относятся затраты:

- на зарплату с ЕСН менеджеров по рекламе, наборщиков текстов, дизайнеров, редакторов, программистов по издательским системам и других сотрудников, связанных с выпуском продукции (без учета оклада);
- на бумагу и др. полиграфические материалы;
- на типографские услуги (печатные издания издаются в еманжелинской типографии);
- на Интернет-трафик (оригинал-макеты пересылаются из филиалов в головную компанию);
- на оформление заказа (затраты на печать отчетных документов – заказов, макетов рекламных блоков, актов и счетов-фактур);
- на рекламу;
- на роялти.

Ко второму виду относятся затраты:

- на зарплату (с учетом ЕСН) административно-управленческого аппарата (директора, бухгалтерии, заместителя директора, офис-менеджера, водителей, охранников, кассиров), окладную часть (с учетом ЕСН) работников, связанных с выпуском продукции;
- на аренду помещений для офиса и приемных пунктов;
- на офисную компьютерную технику;
- на услуги телефонных компаний;
- на Интернет-траффик;
- на консультационные услуги;
- на канцелярские товары общего назначения;
- на транспорт;
- на ремонт помещения;
- на электроэнергию и другие коммунальные платежи;
- на бензин;
- прочие затраты.

Прямые затраты на оплату труда можно разбить на постоянную (сравнительно небольшой оклад) и переменную части. Зарплата агентов в основном состоит из агентского вознаграждения, т.е. зависит от объема реализованной продукции и ставки агентского процента, определяемого исходя из уровня квалификации менеджера и процента оплаченных заказов этого менеджера. В среднем процент агентских составляет 10% с учетом единого социального налога (ЕСН). Зарплата наборщиков текста и редакторской группы зависит главным образом от количества печатных знаков и сложности набираемого текста. По расчетам данные расходы (с учетом ЕСН) будут составлять на одно объявление:

1) газеты «Из рук в руки» – 1 руб.;
2) журнала «Строительство» - 1,5 руб.;
3) справочника «Недвижимость» – 1,5 руб.
4) на сайте – 2 руб.

Система оплаты труда всех остальных работников является окладной. Фонд заработной платы (ФОТ) по окладам определяется штатным расписанием. Структура этого фонда практически не меняется. Как правило, рост данного фонда связан с индексацией окладов, вызванной инфляцией и ростом средней заработной платы для соответствующих профессий. В плановом году постоянная часть ФОТ с ЕСН будет составлять примерно 810 тыс. руб. в месяц.

Расходы на бумагу и типографскую пленку на печать одного стандартного экземпляра газеты «Из рук в руки» составляют 5 руб. Типографские материалы для остальных печатных изданий включены в стоимость типографских услуг. Размер платы за типографские услуги определяется прайсом поставщика этих услуг. Общая сумма затрат на типографские услуги зависит от цены печати одного экземпляра и тиража.

1	ЗАРПЛАТА
2	Отчисления от ФОТ (ЕСН)
3	Командиров.расходы 264/1/12
4	Аренда помещения 264/1/10
5	Хозрасходы 264/1/7
6	УСЛУГИ ГТС 264/1/25
7	УСЛУГИ МТС 264/1/25
8	Материалы по комп.техн.254/1/2
9	Юридич.услуги 264/1/14
10	РЕКЛАМА 264/1/28
11	Ремонтные работы 264/1/7
12	Доставка 264/1/6
13	Нотариал.услуги 264/1/16
14	Материалы 254/1/1
15	Ремонт осн.средств 260(проч)
16	Переподготовка кадров 264/1/23
17	Информационные услуги 264/1/14
18	Канцтовары 264/1/24
19	Материалы 254/1/1
20	Отчисления 14%
21	Расходы на сод.а/м
22	Автоуслуги/Транспорт.264/1/6
25	Налоги и сборы по зак.264/1/1
26	Почтово-телегр.расходы264/1/25
30	INTERNET 264/1/25
33	Износ по основным средствам
34	Страхование 263/1/
35	Инкассация 264/1/25
43	Представ.расходы 264/1/22
101	Типографские услуги 254/1/6
102	Бумага Пр 254/1/1
104	Экспедирование 264/1/49
105	Обновление ПП 264/1/26
106	Пленка 254/1/1
107	Прием объявлений 254/1/6
110	Прочие расходы 264/1/49
118	Стоянка а/трансп-та 264/1/11
120	Материалы на сод.а/тр 254/1/2
121	ГСМ 254/1/2
128	Роялти 254/1/6
129	Бумага газетная 254/1/1
130	Расчетное обсл.банка 264/1/25
145	Пейджер-услуги связи 264/1/25
151	Охрана 264/1/6
155	ККМ(обслуж..расх.м-лы) 254/1/2
159	Консульт.услуги 264/1/15
171	БУ-износ по ОС от 2 до 10
172	АМОРТИЗАЦИЯ по ОС>10 тыс.2002г
173	Износ для НУ
175	МАТРАСХОДЫ-ОСдо10тыс. 254/1/3
181	Расх.на печать журнала 254/1/6
182	Регистр.сбор за журнал
184	Отчисления в ФСтНСл
187	Аренда автомоб 264/1/10
188	Маркетинг/услуги 264/1/27
189	Коммунальные услуги
190	Обслужив. мини-АТС
191	Диагностика оборудование(тех обслуж)
192	Услуги по содержанию помещения
193	Печать блан. продукции
194	Монтаж оборудован.
195	Полиграфические услуги

Рисунок 6.2. Справочник статей затрат.

Плановая цена типографии составляет (без учета НДС):

- для газеты «Из рук в руки» - 20 руб. за экземпляр;
- для журнала «Строительство» - 14 руб. за экземпляр;
- для справочника «Недвижимость» - 15 руб. за экземпляр.

Расходы на рекламу составляют 0,1% от предполагаемой выручки.

Плата за Интернет-траффик, зависящая от количества страниц издания (размера сайта), планируется в расчете на один оригинал-макет:

- для газеты «Из рук в руки» - 500 руб.;
- для журнала «Строительство» - 3000 руб.;
- для справочника «Недвижимость» - 2500 руб.
- для одного среднего объявления на сайте – 2 руб.

Остальные затраты на пользование Интернетом являются косвенными и плановый уровень этих издержек составляет 10 тыс. руб. Затраты на оформление заказа зависят от его сложности. Плановый норматив равен 0,1% от стоимости заказа. Лицензионный сбор (роялти) платится за газету «Из рук в руки» и составляет около 14% от выручки.

Прогнозная величина остальных затрат за месяц составляет:

1) на аренду помещений для офиса и приемных пунктов – 150 тыс. руб.
2) на офисную компьютерную технику – 100 тыс. руб.;
3) на услуги телефонных компаний – 30 тыс. руб.;
4) на консультационные услуги (услуги по сопровождению бухгалтерских, управленческих и правовых информационных систем) – 10 тыс. руб.;
5) на канцелярские товары общего назначения – 60 тыс. руб.;
6) на транспорт – 25 тыс. руб.;
7) на ремонт– 5 тыс. руб.;
8) на электроэнергию и другие коммунальные платежи – 40 тыс. руб.;
9) на ГСМ – 25 тыс. руб.;
10) прочие затраты – 140 тыс. руб.

Постоянные затраты распределяются по видам продукции с помощью коэффициентов, рассчитываемых исходя из данных о выручке и трудовых затрат прошлых периодов. Значения этих коэффициентов в плановом периоде равны:

1) для тиража газеты «Из рук в руки» - 10%;

2) для рекламы в газете «Из рук в руки» - 74%;

3) для журнала «Строительство» - 6%;

4) для тиража справочника «Недвижимость» - 1%;

5) для рекламы в справочнике «Недвижимость» - 4%;

6) для сайта объявлений – 5%.

Спрос на рекламную продукцию компании «Пронто-Уфа» можно разбить на три сектора:

- частные лица, размещающие объявления в газете «»Из рук в руки» по купонам и за плату;
- фирмы с разовыми заказами;
- фирмы - постоянные клиенты;

Ценовая политика для заказчиков с купонами самая демократичная: стоимость объявления включена в цену газеты. Это связано с тем, что данные заказчики являются одновременно покупателями газеты и потребителями рекламы в газете. Чрезмерно высокая цена на газету, уменьшит объем спроса на газету, что отрицательно скажется на объеме рекламы, размещаемой другими клиентами. Таким образом, выручка от продажи газет должна покрывать расходы на печать и распространение частных объявлений.

В Челябинске помимо газеты «Из рук в руки» существуют другие издания подобного характера («Тумба», «Я по объявлению»). Более высокий тираж газета «Из рук в руки» по сравнению с основными конкурентами обеспечивается высоким качеством. Тем не менее цена на газету не должна быть слишком высокой, чтобы покупатели газеты не переключились на другие издания.

Стоимость рекламы для физических лиц без купонов выше. К таким клиентам относятся частные лица – заказчики:

- не располагающие купоном;
- готовые заплатить более высокую цену для повышения качества объявления (выделение рамкой, более крупный шрифт, вставка фотографии или рисунка и т.д.);
- размещающие объявления не в газете «Из рук в руки»;

- размещающие объявления коммерческого характера.

Тарифы на разовые заказы выше из-за риска неоплаты или отказа от подписания акта. Чаще всего такие заказы размещают мелкие фирмы и индивидуальные предприниматели. Они могут размещать объявления во всех изданиях ЗАО «Пронто-Уфа».

Для постоянных клиентов ценовая политика мягче, чем для третьей категории. Эти клиенты обеспечивают более 20% выручки от рекламы, которую они могут размещать во всех изданиях. Постоянные клиенты как правило имеют четко выработанную рекламную компанию, которую они реализовывают с помощью различных изданий ЗАО «Пронто-Уфа» (причем один клиент может размещать заказ в нескольких изданиях одновременно).

Т.о. число стратегий ценовой диверсификации равно трем ($m = 3$). Сумма интенсивностей стратегий равна 1, а $x_2 \geq 0,2$. Всего видов продукции шесть: продажа газеты «Из рук в руки» и справочника «Недвижимость», размещение рекламы в газете, справочнике, журнале «Строительство», сайте в Интернет ($n = 6$). Согласно данным маркетолога в плановом периоде спрос характеризуется следующими ценами P в руб. на ед. без НДС и объемами выпуска Q в тыс. ед. при единичной интенсивности каждой ценовой стратегии

$$P = \begin{pmatrix} 18,18 & 19,09 & 13,64 \\ 30 & 27,27 & 22,73 \\ 409,09 & 409,09 & 363,64 \\ 109,09 & 109,09 & 90,91 \\ 181,82 & 272,73 & 227,27 \\ 363,64 & 454,55 & 409,09 \end{pmatrix}, \; Q = \begin{pmatrix} 90 & 5 & 5 \\ 10 & 2 & 5 \\ 8 & 3 & 6 \\ 1 & 1,5 & 1,7 \\ 0,25 & 0,45 & 1 \\ 0,6 & 0,8 & 0,5 \end{pmatrix}.$$

На основе изложенного выше:

1) плановые прямые трудовые затрат (матрица $L \circ Q$) в тыс. руб.

$$L \circ Q = \begin{pmatrix} 90 & 5 & 5 \\ 15 & 3 & 7,5 \\ 335,27 & 125,73 & 224,18 \\ 12,41 & 18,61 & 18 \\ 4,92 & 12,95 & 24,23 \\ 23,02 & 37,96 & 21,45 \end{pmatrix};$$

2) плановые прямые материальные затраты (затраты на типографию; матрица $M \circ Q$) в тыс. руб.

$$M \circ Q = \begin{pmatrix} 900 & 50 & 50 \\ 137 & 27 & 67,5 \\ 1350 & 75 & 75 \\ 14 & 21 & 23,8 \\ 15 & 3 & 7,5 \\ 0 & 0 & 0 \end{pmatrix};$$

3) плановые прочие прямые затраты (затраты на Интернет, роялти, рекламу и оформление заказа; матрица $S \circ Q$) в тыс. руб.:

$$S \circ Q = \begin{pmatrix} 235,9 & 13,8 & 9,9 \\ 2,1 & 0,4 & 0,9 \\ 6,5 & 2,5 & 4,4 \\ 0,9 & 1,4 & 1,5 \\ 0,1 & 0,2 & 0,5 \\ 1,6 & 2,3 & 1,4 \end{pmatrix}.$$

Вектор постоянных затрат c будет иметь вид

$$c^T = (140,5;14;1039,7;84,3;56,2;70,3).$$

В результате работы программы с точностью до 10^{-8} были найдены оптимальное значение параметра r, равное 10,1%; оптимальный вектор интенсивностей стратегий $x^T = (0,5586;0,2414;0,2)$. При такой структуре продаж маржинальная рентабельность будет превышать 10,1%. Число Фробениуса $\overline{\lambda}$ для данной модели равно $0,7732$ при $\overline{x}^T = (0;1;0)$ и $\overline{w}^T = (1;0;0;0;0;0)$. Достижимый уровень маржинальной рентабельности составляет 22,68%. Полученные результаты показывают, что наиболее эффективной является вторая стратегия (фирмы с разовыми заказами), а первый продукт (газета «Из рук в руки») обладает критическим уровнем рентабельности. С экономической точки зрения это вызвано тем, что вторая стратегия более рискованная, а значит, и более прибыльная; цена на первый продукт не может значительно превышать переменные затраты из-за большой чувствительности спроса к цене.

Заключение

В ходе выполнения диссертационной работы получены следующие результаты.

- Разработаны методы математического моделирования процесса бюджетирования, позволяющие провести оптимизацию бюджета продаж с учетом ценовой диверсификации.
- Разработаны средства математического и численного анализа проблемы продуктивности проекта бюджетирования.
- Предложен численно устойчивый алгоритм определения параметров равновесия модели Неймана.
- Доказано, что при интервальной неопределенности исходных данных политика диверсификации цен определяется матрицами центров интервалов, а границы интервала для рентабельности определяются матрицами верхних и нижних границ интервалов. Даны способы их вычисления.
- Разработан комплекс программ (свидетельство РосПатента о регистрации № 2007613463), позволяющий строить оптимальные бюджеты продаж с применением ценовой диверсификации. Данный комплекс имеет развитый интерфейс (механизм импорта-экспорта в форматах xml и Excel) и внедрен на предприятии ЧФ ЗАО «Пронто-Уфа» и в учебный процесс ЮУрГУ.

Достоверность результатов работы подтверждается корректным использованием математического аппарата, теоретических и экспериментальных методов обоснования полученных результатов, результатами внедрения. Положения теории основываются на известных достижениях математической экономики, теории антагонистических игр, методов объектно-ориентированного программирования и проектирования.

Научная новизна состоит в построенной математической модели бюджетирования, в численных методах нахождения параметров равновесия модели Неймана и найденных новых свойствах параметров равновесия, в разработке комплекса программ:

1. Новизна построенной математической модели формирования бюджета продаж заключается в том, что наряду с балансовым подходом впервые применена ценовая диверсификация.

2. Новизна параметров равновесия модели Неймана состоит в применении методов теории антагонистических игр.

3. Впервые предложены методы нахождения граничных значений параметров равновесия модели Неймана при интервальном задании исходных данных.

Практическая значимость состоит в повышении качества разрабатываемых бюджетов продаж. Предложенные в работе численные методы могут быть использованы для анализа параметров равновесия фоннеймановских моделей общего вида. Разработанный комплекс программ (свидетельство РосПатента о регистрации № 2007613463, см. приложение Д) внедрен в Челябинском филиале ЗАО «Пронто-Уфа» (см. приложение Е) и в учебный процесс в Южно-Уральском государственном университете на специальностях «Экономика и управление на предприятии», «Прикладная информатика», «Математические методы в экономике» и «Статистика».

Сферой практического использования данного программного обеспечения, в частности, являются: предприятия сотовой связи, информационных услуг, компании по производству мебели и другие фирмы, которые используют ценовую диверсификацию. Подходы, примененные при разработке и анализе математической модели, могут быть применены для решения других задач (например, методы анализа при интервальной неопределенности).

Литература

1. Архангельский, А.Я. Программирование в C++Builder 6 и 2006 / А.Я. Архангельский. – М.: Бином, 2007. – 1181 с.
2. Ашманов, С.А. Введение в математическую экономику : Учеб. пособие для спец."Прикл. математика" / С.А. Ашманов. – М.: Наука, 1984. – 293 с.
3. Банин, А.А. Применение балансовой модели в анализе деятельности предприятия / А.А. Банин, М.И Летавин // Экономика и математические методы. – 2002. – Т.38, №4. – С. 23-25.
4. Башарин, Г.П. Начала финансовой математики / Г.П. Башарин. – М.: ИНФРА-М, 1997. – 160 с.
5. Бокс, Д. Сущность технологии COM / Д. Бокс; пер. с англ. А. Шкадова. – СПб. и др. : Питер Бук , 2001. – 397 с.
6. Бочаров, В.В. Коммерческое бюджетирование / В.В. Бочаров. – СПб.: Питер, 2003. – 368 с.
7. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Г. Буч; пер. с англ.; под ред. И. Романовского, Ф. Андреева. – М.; СПб.: Бином: Нев. диалект, 2001.
8. Васильков, Ю.В. Компьютерные технологии вычислений в математическом моделировании: учебное пособие / Ю.В. Васильков, Н.Н. Василькова.– М.: Финансы и статистика, 1999. – 256 с.
9. Васин, А.А. Теория игр и модели математической экономики: Учеб. пособие по направлению 510200 - Прикладная математика и информатика и по специальности 010200 - Прикладная математика и информатика / А. А. Васин, В. В. Морозов. – М. : МАКС Пресс , 2005. – 271 с.
10. Войнов, И.В. Оптимальное управление экономическими системами: учебное пособие / И.В. Войнов, А.И. Телегин. – Челябинск: Изд-во ЮУрГУ, 1999. – 65 с.

11.Гаспариан, М.С. Некоторые вопросы практического применения информационных технологий в экономике и управлении: методическое пособие / М.С Гаспариан. – М.: МЭСИ, 2000. – 89 с.
12.Герасименко, В.В. Ценовая политика фирмы: Европейский опыт. Российские перспективы. Модели и методы. Варианты и тесты / В.В. Герасименко. – М. : Финстатинформ , 1995. – 186 с.
13.Добрынин, В. Ю. Технологии компонентного программирования: Учеб. пособие. – СПб.: Издательство Санкт-Петербургского государственного университета, 2004. – 214 с.
14.Еремин, И.И. Введение в теорию линейного и выпуклого программирования / И.И. Еремин, Н.Н. Астафьев. – М.: Гл. ред. физ.-мат. лит. изд-ва «Наука», 1976. – 191 с.
15.Ермолаев, В. С++ Builder: Книга рецептов / В. Ермолаев, Т. Сорока. – М.: КУДИЦ-ОБРАЗ, 2006. – 204 с.
16.Жданов, С.А. Механизмы экономического управления предприятием / С.А. Жданов. – М.: Юнити-Дана, 2002. – 320 с.
17.Жданов, С.А. Основы теории экономического управления предприятием / С.А. Жданов. – М.: Финпресс, 2000. – 384 с.
18.Жданов, С.А. Экономические модели и методы в управлении / С.А. Жданов. – М.: Изд-во «Дело и Сервис», 1998. – 176 с.
19.Жолен, Л. Прикладной интервальный анализ / Л. Жолен, М. Кифер, О. Дидри, Э. Вальтер; под ред. Б.Т. Поляка; пер. с англ. С.И. Кумкова. – М.; Ижевск: Институт компьютерных исследований, 2005. – 468 с.
20.Зайден, М. XML в электронной коммерции / М. Зайден. – М.: БИНОМ-Лаборатория знаний, 2003. – 480 с.
21.Замков, О.О. Математические методы в экономике : Учеб. / Моск. гос. ун-т им. М.В. Ломоносова; О.О. Замков, А.В. Толстопятенко, Ю.Н. Черемных. – М. : Изд-во "ДИС" , 1997. – 365 с.
22.Казанцев, А. Моделирование бизнес-процессов при постановке бюджетирования / К.А. Казанцев, В.Г. Кунщиков //Справочник экономиста. – 2004. – N 12. – С. 54-60.

23. Карлберг, К. Управление данными с помощью Microsoft Excel / К. Карлберг; пер. с англ. – М. и др.: Вильямс , 2005. – 446 с.
24. Кобелев, Н.Б. Основы имитационного моделирования сложных экономических систем : Учеб. пособие для вузов по специальности "Мат. методы в экономике" и др. экон. специальностям / Н.Б. Кобелев. – М.: Дело, 2003 . – 335 с.
25. Ковалев, В.В. Финансовый анализ: методы и процедуры / В.В.Ковалев. – М.: Финансы и статистика, 2006. – 559 с.
26. Колемаев, В.А. Математическая экономика : Учеб. для вузов по экон. Специальностям / В.А, Колемаев. – М.: ЮНИТИ-Дана, 2002. – 398 с.
27. Котлер, Ф. Маркетинг. Менеджмент / Ф. Котлер; под общ. ред. Л.А. Волковой, Ю.Н. Каптуревского; пер. с англ. Т. Виноградовой и др. – 10-е изд. – СПб. и др.: Питер, 2001. – (Теория и практика менеджмента). – 749 с.
28. Кренке, Д. Теория и практика построения баз данных / Д. Кренке; пер. с англ. А. Вахитова. – 9- е изд. – СПб. и др.: Питер, 2005. – (Классика computer science). – 858 с.
29. Лагоша, Б.А. Оптимальное управление в экономике: Учеб. пособие для вузов по специальности 061800 "Мат. методы в экономике" и др. специальностям / Б.А. Лагоша. – М.: Финансы и статистика , 2003. – 191 с.
30. Латипова, А.Т. Оптимизация бюджета продаж / А.Т. Латипова, А.В. Панюков // Вестник Южно-Уральского государственного университета. Серия: Рынок: Теория и практика. – 2006. – Вып. 4. – №15 (170). – С.116–120.
31. Латипова, А.Т. Анализ проблемы продуктивности модели бюджетирования / А.Т. Латипова // Проблемы теоретической и прикладной математики: Труды 38-й Региональной молодежной конференции. – Екатеринбург: УрО РАН, 2007. – С. 335-339.
32. Латипова, А.Т. Оптимизация бюджета продаж в условиях ценовой диверсификации / А.Т. Латипова, А.В. Панюков // Управление изменениями и инновации в экономических системах / под ред. д.э.н., проф. В.В. Глухова,

д.э.н., проф. А.В.Бабкина: Межвуз. сб. науч. тр. – СПб.: Изд-во Политехн. ун-та, 2006. – С. 647-654.

33. Латипова, А.Т. Математическая модель бюджетирования / А.Т. Латипова // Проблемы теоретической и прикладной математики: Труды 37-й региональной молодежной конференции. – Екатеринбург: УрО РАН, 2006. – С. 391-397.
34. Латипова, А.Т. Ценовая диверсификация в бюджетировании / А.Т. Латипова // Экономика и менеджмент: проблемы и перспективы: Труды Международной научно-практической конференции. 6-11 июня 2005 года. – СПб: Изд-во Политехн. ун-та,2005. – С. 562-566.
35. Латипова, А.Т. Разработка и исследование математических моделей бюджетирования / А.Т. Латипова // Проблемы теоретической и прикладной математики: Труды 36-й Региональной молодежной конференции. – Екатеринбург: УрО РАН, 2005. – С. 310-313.
36. Латипова, А.Т. Применение модели Неймана при решении задач бюджетирования / А.Т. Латипова // Проблемы теоретической и прикладной математики: Труды 35-й Региональной молодежной конференции. – Екатеринбург: УрО РАН, 2004. – С. 284-288.
37. Латипова, А.Т. Модель оптимизации бюджетирования для предприятий минерально-сырьевого комплекса // Стратегия развития минерально-сырьевого комплекса в XXI веке: Материалы международной конференции. Москва-Бишкек. – М: Изд-во РУДН, 2004. – С. 206-208.
38. Латипова, А.Т. Модель оптимизации ценовой стратегии для задач бюджетирования / А.Т. Латипова // Дискретный анализ и исследование операции: Материалы российской конференции (Новосибирск, 28 июня – 2 июля 2004). – Новосибирск: Изд-во Ин-та математики, 2004. – С. 206.
39. Леонтьев, В.В. Межотраслевая экономика / В.В. Леонтьев; науч. ред. и авт. предисл. А. Г. Гранберг. – М.: Экономика , 1997. – 477 с.
40. Мажукин, В.И. Математическое моделирование в экономике Ч.3: Экономические приложения: Учеб. пособие для вузов по направлению 521500 – Ме-

неджмент / В.И. Мажукин, О.Н. Королева; Рос. акад. образования, Моск. психол.-социал. ин-т. – М. : Флинта: Издательство МПСИ, 2005. – 174 с.

41. Овчинников, С.М. XML: Язык форматирования документов World Wide Web / С.М. Овчинников. – М.: Майор: Осипенко , 2001. – 154 с.
42. Панюков, А.В. Программное обеспечение системы оптимизации бюджета продаж с использованием ценовой диверсификации / А. В. Панюков, А. Т. Латипова // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. –2008. – Вып. 3. – №7 (103). – С.16–20.
43. Панюков, А.В. Математическое моделирование экономических процессов: Конспект лекций / А.В. Панюков. – Челябинск: ЧГТУ, 1997. – 125 с.
44. Петросян, Л.А. Теория игр: Учеб. пособие для ун-тов по специальности "Математика" / Л.А. Петросян, Н.А. Зенкевич, Е.А. Семина. – М.: Высшая школа: Кн. Дом "Университет", 1998. – 299 с.
45. Подбельский, В.В. Язык Си++: Учеб. пособие для вузов по направлениям «Приклад. мат.», «Вычисл. машины, комплексы, системы и сети» / В.В. Подбельский. – М.: Финансы и статистика, 2005. – 559 с.
46. Роджерсон, Д. Основы COM / Д. Роджерсон; пер. с англ. – М. : Изд. отдел "Рус. ред.": ТОО "Channel Trading Ltd.", 1997. – 352 с.
47. Сеннов, А. Access 2003: Практическая разработка баз данных / А. Сеннов. – СПб. и др.: Питер,2005. – (Учебный курс). – 255 с.
48. Схрейвер, А. Теория линейного и целочисленного программирования: В 2-х т. / А. Схрейвер; пер. с англ. С. А. Тарасова и др.; под ред. Л. Г. Хачияна. – М.:Мир, 1991. – Т. 1. – 364 с.
49. Проблемы экономики: Труды XXI Российской школы по проблемам науки и технологий (Миасс, 26-28 июня 2001 г.). – М.: Российская академия наук, 2001. – 310 с.
50. Уткин, В.Б. Информационные системы и технологии в экономике: Учеб. для вузов по специальности 351400 "Прикладная информатика и др. меж-

дисциплинар. специальностям / В.Б. Уткин, К.В. Балдин. - М.: ЮНИТИ-Дана,2003. - (Информатика). - (Профессиональный учебник). – 334 с.

51. Фаулер, М. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт; пер. с англ. А. М. Вендрова; под ред. Л. А. Калиниченко – М.: Мир, 1999. – 191 с.
52. Фомин, П.А. Бюджетирование - теория и практика производственно- финансового планирования и анализа / П.А. Фомин // Финансы и кредит. – 2003. – N 1. – С. 55-60.
53. Фридман, А.Л. Основы объектно-ориентированной разработки программынх систем / А.Л. Фридман. – М.: Финансы и статистика, 2000. – 192 с.
54. Хачиян, Л.Г. Полиномиальный алгоритм в линейном программировании / Л.Г. Хачиян // Доклады АН СССР – М.: АН ССС, 1979. – Т. 244 – с. 1093-1096.
55. Хруцкий, В. Е. Внутрифирменное бюджетирование: Настольная книга по постановке финансового планирования / В.Е. Хруцкий, В.В. Гамаюнов. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 456 с.
56. Шамис, В. А. C++ Builder Borland Developer Studio 2006 / В. А. Шамис. – СПб. и др.: Питер, 2007. - (Для профессионалов). – 780 с.
57. Щиборщ, К.В. Анализ хозяйственной деятельности предприятий России / К.В. Щиборщ. – М.: Дело и Сервис,2003. – 318 с.
58. Financial management. – М.: Carana corporation: USAID: RPC, 1998. – (Business Toolkit). – 290 с.
59. Jacobson, I. Object-Oriented Software Engineering / I. Jacoibson. – New York: ASM press, 1992. – 528 p.
60. Miller, A. Strategic Management / A. Miller. – 3rd ed. – Boston etc.: Irwin: McGraw-Hill, 1998. – 1120 p.
61. Maimon, O. Optimal Flow Control in Manufacturing Systems: Production Planning and Scheduling / O. Maimon, E. Khmelnitsky, K. Kogan. – Dordrecht etc.: Kluwer, 1998. – (Applied Optimization). – 346 p.

62. Vollmann, T. E. Manufacturing Planning and Control Systems/ T.E. Vollmann, W.L. Berry, D.C. Whybark. – 4th ed. – Boston etc. : Irwin: McGraw-Hill, 1997. – 836 p.
63. Свидетельство Роспатента о государственной регистрации программы для ЭВМ № 2007613463 / Программа оптимизации бюджета продаж при ценовой диверсификации / А.Т. Латипова, А.В. Панюков. – №2007612433, заявл. 19.06.2007 (зарегистр. 15.08.2007); опубл. 20.12.2007, Бюл. №4 (61) (I часть). – 1 с.
64. Елманова, Н. Использование COM-технологии в C++Builder / Н. Елманова. – М.: Центр Информационных Технологий, 1999. – http://citforum.ru/programming/cpp/com_4.shtml
65. Печерский, А. Язык XML - практическое введение / А. Печерский. – М.: Центр Информационных Технологий, 1999. – http://citforum.ru/internet/xml2/index.shtml
66. Официальный сайт компании Microsoft. – http://microsoft.com
67. http://www.cbuilder.ru
68. Официальный сайт ЗАО «Пронто-Уфа» http://www.irr.ru
69. Официальный сайт корпорации World Wide Web Consortium. – http://www.w3.org
70. http://www.xml.com

Приложения

Приложение А

Файл TNeumann.h

```
//---------------------------------------------------------------------------
#ifndef TNeumannH
#define TNeumannH
#endif
#include "TMatrix.h"
#include "TLinProgProb.h"
//---------------------------------------------------------------------------
class TNeumann
{
	private:
			int nRows_N; // число продуктов
			int nCols_N; // число стратегий
			TMatrix *A; // матрица A
			TMatrix *B; // матрица B
			TMatrix *x; // вектор x
			TMatrix *p; // вектор p (двойтсвенных переменных)
			double lambda_up; // верхняя граница для lambda
			double lambda_low; // нижняя граница для lambda
			double lambda_tmp; // текущее значение для lambda
			int iteration; // количество итераций
			int slvng_N;// имеет значение -1, если макс. lambda = 1,
			//1, если макс. lambda < 1
			//0, если оптимизация не была проведена
			//-2, если макс. lambda > 1 или < 0
			//3, если существует седловая точка
			double precision; // приближение
	public:
			TNeumann(void); // конструктор без параметров
			TNeumann(int,int); // конструктор с заданным числом
			// продуктов и стратегий
			TNeumann(int,int,TMatrix&,TMatrix&); // конструктор
			//с заданным числом продуктов и стратегий и матрицами A и B
			TNeumann(char*); // конструктор с загрузкой
			// из текстового файла с табуляцими
			~TNeumann(void); // деструктор
			TNeumann& operator=(const TNeumann&); // перегруженный
			//оператор
			// присваивания
			friend int GetNRows(const TNeumann&); // функция для вывода
			//числа продуктов
			friend int GetNCols(const TNeumann&);  // функция для вывода
			//числа продуктов
			double& GetLambdaUp(void); // функция для вывода
			//верхней оценки lambda
			double& GetLambdaLow(void); // функция для вывода
			//нижней оценки lambda
			void PutVariables(double*, double*);  // функция
			// для присваивания значений переменным
			void PutA(double, int, int);// функция для ввода матрицы A
			void PutB(double, int, int); // функция для ввода матрицы B
			double& GetA(int,int); // функция для вывода матрицы A
			TMatrix& GetA(void); // функция для вывода матрицы A
			double& GetB(int,int); // функция для вывода матрицы B
			TMatrix& GetB(void); // функция для вывода матрицы B
			double& GetX(int); // функция для вывода вектора x
			TMatrix& GetX(void); // функция для вывода вектора x
			double& GetP(int); // функция для вывода вектора p
			TMatrix& GetP(void); // функция для вывода вектора p
			void MakeGame(double const); // метод оптимизации
			void MinIntervalLambda(void);
};
```

П.А

Файл TNeumann.cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>

#pragma hdrstop

#include "TNeumann.h"

//---------------------------------------------------------------------------

int GetNCols(const TNeumann& a1)
{
      return a1.nCols_N;
};
//---------------------------------------------------------------------------
int GetNRows(const TNeumann& a1)
{
      return a1.nRows_N;
};
//---------------------------------------------------------------------------
TNeumann::TNeumann(void)
{
      nCols_N=nRows_N=0;
      A=NULL;
      B=NULL;
      x=NULL;
      p=NULL;
      lambda_up=1;
      lambda_low=0;
      lambda_tmp=1;
      slvng_N=0;
      precision=0.000001;
      iteration=0;
};
//---------------------------------------------------------------------------

TNeumann::TNeumann(int nRows1, int nCols1)
{
      nRows_N=nRows1;
      nCols_N=nCols1;
      A=new TMatrix(nRows_N,nCols_N);
      B=new TMatrix(nRows_N,nCols_N);
      x=new TMatrix(nCols_N,1);
      p=new TMatrix(nRows_N,1);
      lambda_up=1;
      lambda_low=0;
      lambda_tmp=1;
      slvng_N=0;
      precision=0.000001;
      iteration=0;

};
//---------------------------------------------------------------------------
TNeumann::~TNeumann(void)
{
            A->~TMatrix();
            B->~TMatrix();
            x->~TMatrix();
            p->~TMatrix();
};
//---------------------------------------------------------------------------
```

```
TNeumann& TNeumann::operator=(const TNeumann& a1)
{
      if (this==&a1) return (*this);
      nCols_N=a1.nCols_N;
      nRows_N=a1.nRows_N;
      if (this!=NULL) {
             A->~TMatrix();
             B->~TMatrix();
             x->~TMatrix();
             p->~TMatrix();
                              }
      A=new TMatrix(a1.nRows_N,a1.nCols_N);
      B=new TMatrix(a1.nRows_N,a1.nCols_N);
      x=new TMatrix(a1.nCols_N,1);
      p=new TMatrix(a1.nRows_N,1);
      for(int i=0; i<nRows_N;i++)
            {
                  for(int j=0; j<nCols_N;j++) A->put(i,j,a1.A->get(i,j));
            };
      for(int i=0; i<nRows_N;i++)
            {
                  for(int j=0; j<nCols_N;j++) B->put(i,j,a1.B->get(i,j));
            };
      for(int i=0; i<nCols_N;i++)
            x->put(i,0,a1.x->get(i,0));
      for(int i=0; i<nRows_N;i++)
            p->put(i,0,a1.p->get(i,0));
      lambda_up=a1.lambda_up;
      lambda_low=a1.lambda_low;
      lambda_tmp=a1.lambda_tmp;
      precision=a1.precision;
      slvng_N=a1.slvng_N;
      return (*this);
};
//-----------------------------------------------------------------------------
TMatrix& TNeumann::GetA(void)
{
      return (*A);
};
//-----------------------------------------------------------------------------
TMatrix& TNeumann::GetB(void)
{
      return (*B);
};
//-----------------------------------------------------------------------------
TMatrix& TNeumann::GetX(void)
{
      return (*x);
};
//-----------------------------------------------------------------------------
TMatrix& TNeumann::GetP(void)
{
      return (*p);
};
//-----------------------------------------------------------------------------
double& TNeumann::GetA(int nRow, int nCol)
{
      return (A->get(nRow, nCol));
};
//-----------------------------------------------------------------------------
double& TNeumann::GetB(int nRow, int nCol)
{
```

```
	return (B->get(nRow, nCol));
};
//-----------------------------------------------------------------------------

double& TNeumann::GetX(int m)
{
	return (x->get(m,0));
};
//-----------------------------------------------------------------------------
double& TNeumann::GetP(int n)
{
	return (p->get(n,0));
};
//-----------------------------------------------------------------------------
double& TNeumann::GetLambdaUp(void)
{
	return (lambda_up);
};
//-----------------------------------------------------------------------------
double& TNeumann::GetLambdaLow(void)
{
	return (lambda_low);
};
//-----------------------------------------------------------------------------
void TNeumann::PutVariables(double *x1,double *p1)
{
   for(int i=0;i<nCols_N;i++)
        x->put(i,0,x1[i]);
   for(int i=0;i<nRows_N;i++)
        p->put(i,0,p1[i]);
};
//-----------------------------------------------------------------------------
void TNeumann::MakeGame(double const prec)
{
	// инициализация вспомогательных переменных
	precision=prec;
	slvng_N=0;
	TMatrix m_x=TMatrix(nRows_N,nCols_N);
	int temp_num_ogr, temp_num_perem;
	temp_num_ogr=nRows_N;
	temp_num_perem=nCols_N;
	double **a1,u;
	double max_lambda_u_pos=0; // максимально вычисленное lambda где u>0
	double min_lambda_u_neg=1; // минимально вычисленное lambda где u<0
	double min_lambda_u_zero=-1; // минимально вычисленное lambda где u=0

	// инициализация массива для коэффициентов в ограничениях
	a1=new double *[nRows_N];
	for(int i=0;i<nRows_N;i++)
	  a1[i]=new double [nCols_N];

	for(int i=0;i<nRows_N;i++)
		for(int j=0;j<nCols_N;j++)
			      a1[i][j]=0;
	double psi; // расчет положительной платежной матрицы (A-λB+Ψ)
	psi=Max((*B)-(*A))+1;
	m_x=(*A)-(lambda_tmp*(*B))+precision;
	m_x=m_x+psi*Norm(nRows_N,nCols_N);

	// присваивание массиву **a1 значений платежной матрицы
	for(int i=0;i<nRows_N;i++)
		for(int j=0;j<nCols_N;j++)
```

```
                a1[i][j]=m_x.get(i,j);

        // инициализация вектора коэффициентов целевой функции
        double *c1;
        c1=new double[temp_num_perem];

        // присваивание вектора c1
        for(int i=0; i<temp_num_perem;i++)
        {
                c1[i]=1; // коэффициенты при ξ - единичный вектор
        }

        // инициализация вектора правой части ограничений
        double *b1;
        b1=new double[temp_num_ogr];

        // присваивание вектора b1
        for(int i=0; i<temp_num_ogr;i++)
        {
                b1[i]=1;// (A-λB+Ψ)ξ<=1
        }

        // инициализация вектора знаков переменных
        int *pos1;
        pos1=new int[temp_num_perem];

        // присваивание вектора pos1
        for(int i=0; i<temp_num_perem;i++)
        {
                {pos1[i]=1;}// ξ>=0
        }

        // инициализация вектора знаков ограничений
        int *neq1;
        neq1=new int[temp_num_ogr];

        // присваивание вектора neq1
        for(int i=0; i<temp_num_ogr;i++)
        {
                neq1[i]=-1;// (A-lambda*B)x<=1
        }

        //инициализация объекта задачи линейного программирования
        TLinProgProb* LinNeumann=new TLinProgProb(temp_num_perem,
temp_num_ogr,1,a1,c1,b1,pos1,neq1);

        //решение при lambda=1
        iteration++; // увеличиваем счетчик итераций
         double* tmp=LinNeumann->getPrimalSltn(); // расчет задачи ЛП
         if (tmp!=NULL) // Если найдено какое-то решение
        {
              u=(1/LinNeumann->get_value()-psi); // расчет значения u
              if (lambda_tmp==1 && u>=0) // если для lambda=1 u>=0
              {
              slvng_N=-1; // то выход с флагом для числа Фробениуса=1
              }
        }
        else
        {
              slvng_N=-2; // то выход с флагом для числа Фробениуса<1
        }
```

```
MinIntervalLambda(); // сжатие интервала для поиска lambda
                     // через вычисление матрицы C

lambda_tmp=(lambda_up+lambda_low)/2; // половинное деление интервала
// цикл половинного деления
while ((lambda_up-lambda_low) >= prec && slvng_N!=3)
// до тех пор пока интервал меньше точности и если нет седловой точки
{
      iteration++; // увеличиваем счетчик итерации

      // расчет платежной матрицы
      m_x=(*A)-(lambda_tmp*(*B));
      m_x=m_x+psi*Norm(nRows_N,nCols_N);

      // изменение массива ограничений
      for(int i=0;i<nRows_N;i++)
            for(int j=0;j<nCols_N;j++)
                        a1[i][j]=m_x.get(i,j);

      //инициализация объекта задачи линейного программирования
      TLinProgProb *LinNeumann1=new TLinProgProb(temp_num_perem,
temp_num_ogr,1,a1,c1,b1,pos1,neq1);
      tmp=LinNeumann1->getPrimalSltn(); // расчет задачи ЛП
      slvng_N=1; // флаг, что макс. lambda <= 1

      if (tmp!=NULL)
      {
            u=(1/LinNeumann1->get_value()-psi); // вычисляем u через
                                                // цену игры и psi

            if (u<-precision) // u меньше 0
            {
            lambda_up=lambda_tmp;  // сдвигаем верхнюю границу интервала
                                   // для макс. lambda
            min_lambda_u_neg=min(min_lambda_u_neg,lambda_tmp);
            //вычисляем новое значение для min_lambda_u_neg
            lambda_tmp=(lambda_up+lambda_low)/2;
            // делим интервал поиска пополам
            }

            if (u<=precision && u>=-precision)// если u приблизительно
                                              // равно 0
            {
            lambda_low=lambda_tmp; //сдвигаем нижнюю границу интервала
                                   // для макс. lambda

                  if (min_lambda_u_zero!=-1) // если min_lambda_u_zero
                                             // не вычислялось
                  {min_lambda_u_zero=min(min_lambda_u_zero,lambda_tmp);}
                  //вычисляем новое значение для min_lambda_u_zero
                  else
                  {min_lambda_u_zero=lambda_tmp;}
                  //вычисляем новое значение для min_lambda_u_zero

            lambda_up=min(min_lambda_u_neg,lambda_up);
            // сдвигаем верхнюю границу интервала для макс. lambda
            lambda_tmp=(lambda_up+lambda_low)/2;
            // делим интервал поиска пополам
            }

            if (u>precision) // если u больше 0
            {
```

```
                lambda_low=lambda_tmp; //сдвигаем нижнюю границу интервала
                                       // для макс. lambda
                max_lambda_u_pos=max(max_lambda_u_pos,lambda_tmp);
                //вычисляем новое значение для max_lambda_u_pos
                lambda_tmp=(lambda_up+lambda_low)/2;
                // делим интервал поиска пополам
                }
          }
      // расчет вектора x
      for(int j=0;j<nCols_N;j++)
            if(slvng_N==1 && iteration>=3)
            {
            double temp_x=tmp[j]*psi;// x=ξ*ψ
            x->put(j,0,temp_x);
            }
}
//-----------------------------------------------------------------------------
void TNeumann::MinIntervalLambda(void)
{
      double *fmin;  // минимумы строк
      double fmaxmin,fminmax; // переменные для максимина и минимакса
      int *imaxmin,*jmaxmin; // координаты максимина
      int *fminindex; // индексы столбцов для минимумов строк
      double *fmax; // максимумы столбцов
      int *fmaxindex; // индексы строк для максимумов столбцов


      // выделение памяти под массивы
      fmin=new double [nRows_N];
      imaxmin=new int [nRows_N*nCols_N];
      fminindex=new int [nRows_N];
      fmax=new double [nCols_N];
      fmaxindex=new int [nCols_N];
      jmaxmin=new int [nRows_N*nCols_N];
      fmaxmin=0;
      fminmax=1;

      // первоначальное присваивание для массивов fmin и fmax
      for (int i = 0; i < nRows_N; i++)
      {
            if (B->get(i,0)!=0) fmin[i]=A->get(i,0)/B->get(i,0);
            else fmin[i]=0;
      }
      for (int j = 0; j < nCols_N; j++)
      {
            if (B->get(0,j)!=0) fmax[j]=A->get(0,j)/B->get(0,j);
            else fmax[j]=1;
      }

      // вычисление минимальных элементов для строк матрицы C
      for(int i=0;i<nRows_N;i++)
            for(int j=0;j<nCols_N;j++)
              {if(B->get(i,j)!=0)
                 {
                       if(fmin[i]>A->get(i,j)/B->get(i,j))
                       {
                             fmin[i]=A->get(i,j)/B->get(i,j);
                             fminindex[i]=j;
                       }
                  }
              else
```

```
                {
                        fmin[i]=0;
                        fminindex[i]=j;
                }
        }
  // вычисление максимальных элементов для строк матрицы C
  for(int j=0;j<nCols_N;j++)
        for(int i=0;i<nRows_N;i++)
          {
          if(B->get(i,j)!=0)
              {
                    if(fmax[j]<A->get(i,j)/B->get(i,j))
                    {
                          fmax[j]=A->get(i,j)/B->get(i,j);
                          fmaxindex[j]=i;
                    }
              }
                    else
                {
                          fmax[j]=1;
                          fmaxindex[j]=i;
                }
           }


  // поиск максимина
  for(int i=0;i<nRows_N;i++)
    if (fmaxmin<fmin[i])
    {
         fmaxmin=fmin[i];
    }

  lambda_low=fmaxmin; // сдвиг нижней границы для поиска  макс. lambda


  // поиск минимакса
  for(int j=0;j<nCols_N;j++)
    if (fminmax>fmax[j])
    {
         fminmax=fmax[j];
    }

  lambda_up=fminmax; // сдвиг верхней границы для поиска  макс. lambda

  if(fmaxmin==fminmax) // если минимакс и максимин совпадают
        {
        lambda_tmp=fmaxmin; // то седловая точка найдена
        slvng_N=3; // флаг на нахождение седловой точки
        iteration++; // увеличение счетчика итераций
        // вычисляем координаты седловой точки
        for(int i=0;i<nRows_N;i++)
              for(int j=0;j<nCols_N;j++)
                    if (fmax[j]==fmaxmin && fmin[i]==fmaxmin && fmaxin-
dex[j]==i && fminindex[i]==j)
                          {
                                imaxmin[tmp1]=i;
                                jmaxmin[tmp1]=j;
                          }
        // рассчитываем вектор x
        for(int j=0;j<nCols_N;j++)
              if(j==jmaxmin[tmp1]) x->put(j,0,1);
              else x->put(j,0,0);
```

```
            // рассчитываем вектор p
            for(int i=0;i<nRows_N;i++)
                  if(i==imaxmin[tmp1]) p->put(i,0,1);
                  else p->put(i,0,0);
            }
}
```

Программа генератор

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
int t, num_files1, n=10, m=10, num_files=100, proc=30, a_max=100;
// t - переменная для записи случайного значения
// n, m - число строк и столбцов в матрицах A и B
// num_files1 - счетчик числа сгенерированных файлов
// num_files  - максимальное количество создаваемых файлов
// proc - ограничение на процент нулевых элементов в матрицах
// a_max - максимальное возможное значение для a
      int *a_sum, *a_sum2, *beta, sum_beta=0,*sum_beta2;
// a_sum - массив для хранения сумм строк матрицы A
// a_sum2 - массив для хранения сумм столбцов матрицы A
// beta - массив для хранения коэффициентов β, используемых
// для генерации строки матрицы B
// sum_beta2 - массив для хранения сумм столбцов β для матрицы B
// sum_beta - сумма коэффициентов β для текущей строки матрицы B
      double lambda, lambda_min=0.8;
// переменные для генерации прогнозного значения λ
      AnsiString tmp=""; a_sum=new int[n]; beta=new int[m];
// tmp - строковая переменная, используемая для записи в файл
// инициализация динамических массивов
      a_sum2=new int[m]; sum_beta2=new int[m];
      for (int i = 0; i < n; i++) a_sum[i]=0;
      for (int j = 0; j < m; j++)
        {a_sum2[j]=0;
        sum_beta2[j]=0;}
      for (num_files1=1; num_files1 <= num_files; num_files1++)
// начало цикла генерации файлов
        {RandSeed=15+num_files1*10; t=Random(20);
// генерация случайного числа для λ
        lambda=lambda_min+t/100.0;
// инициализация объекта TFileStream для работы с файлами
        TFileStream *f1;
        tmp="n_"+AnsiString(n) + "_m_" + AnsiString(m) +
"_f_"+AnsiString(num_files1) + ".txt";
// задание имени файла
        f1=new TFileStream(tmp.c_str(),fmCreate);
        tmp=AnsiString(n)+"\t"+AnsiString(m);
        f1->Write(tmp.c_str(),tmp.Length());
// запись значений n и m в файл через табуляцию
        f1->Write("\r\n",2);
// перевод строки
        RandSeed=25+num_files1; tmp="";
// запуск генерации матрицы A
        for (int i = 1; i <= n; i++)
            {for (int j=1; j<= m; j++)
              {t=Random(a_max); // получение случайного a[i][j]
              if (t<=a_max*proc/100.0) t=0;
// обнуление некоторых элементов матрицы A на основе proc
// проверка на отсутствие нулевых строк и столбцов в матрице A
              if(j==m && a_sum[i-1]==0 && t==0) t=1;
              if(i==n && a_sum2[j-1]==0 && t==0) t=1;
              if(j!=1) tmp=tmp+"\t";
```

П.А

```
            tmp=tmp+AnsiString(t);
            // формирование строки матрицы A
            a_sum[i-1]+=t; a_sum2[j-1]+=t;}
          // суммирование для строк и столбцов матрицы A
            f1->Write(StrCat(tmp.c_str(),"\r\n"),tmp.Length()+2);
          // запись строки в файл
            tmp="";}
      RandSeed=30+num_files1;
// запуск генерации матрицы β и B
      for (int i = 0; i < n; i++)
          {sum_beta=0;
          // обнуление суммы для новой строки коэффициентов β
          for (int j = 0; j < m; j++)
            {beta[j]=Random(100);
            // генерация β
            if(beta[j]<=proc) beta[j]=0;
            // обнуление некоторых коэффициентов β на основе proc
// проверка на отсутствие нулевых строк и столбцов в матрице
// коэффициентов β
            if (j==m-1 && sum_beta==0) beta[j]=1;
            if (i==n-1 && sum_beta2[j]==0) beta[j]=1;
            // суммирование для строк и столбцов матрицы β
            sum_beta+=beta[j];
            sum_beta2[j]+=beta[j];}
// расчет строки матрицы B
          for (int j=0; j< m; j++)
            {t=ceil(a_sum[i]*beta[j]/lambda/sum_beta);
// расчет b[i][j]: b[i][j] = ∑j a[i][j]* β[j] / ∑j β[j] / lambda;
            if(j!=0) tmp=tmp+"\t";
            // формирование строки матрицы B и запись в файл
            tmp=tmp+AnsiString(t);}
            f1->Write(StrCat(tmp.c_str(),"\r\n"),tmp.Length()+2);
            tmp="";}
    // обнуление массивов для хранения сумм
      for (int i = 0; i < n; i++){a_sum[i]=0;}
      for (int j = 0; j < m; j++){a_sum2[j]=0; sum_beta2[j]=0;}
    // удаление объекта для работы с файлом
      delete f1;
    }
    // освобождение памяти от динамических массивов
    delete a_sum; delete a_sum2;
    delete sum_beta2; delete beta;
```

Приложение Б

Файл ExcelServer.h

```
//---------------------------------------------------------------------------

#ifndef ExcelServerH
#define ExcelServerH
//---------------------------------------------------------------------------



class ExcelServer
{
public:

ExcelServer();
~ExcelServer();

      bool ExcelOpen(AnsiString& BookName, bool write_mode);  //открывает Ex-
cel
      bool ExcelClose(void); //закрывает Excel
      bool SheetOpen(AnsiString& SheetName);//открывает лист рабочей книги
                                                          //по названию
      bool SheetOpen(int SheetNum);//открывает лист рабочей лист по номеру
      bool CellSet(int CellRow, int CellCol, float CellValue);
      //заносит числовое значеие в ячейку
      bool CellSet(int CellRow, int CellCol, AnsiString& CellText);
      //заносит строку
      //в ячейку
      bool CellSetDate(int CellRow, int CellCol, AnsiString& CellTextDate);
      //заносит дату
      //в ячейку
      AnsiString CellGet(int CellRow, int CellCol);
      //возвращает текстовое значение
      bool CellFormatSet(int CellRow, int CellCol,AnsiString&
CellFormat);//установка формата для выбранной ячейки
      bool CellBorderSet(int CellRow, int CellCol, int Border_Code, int
Line_Style );
                                                       //рисует рамку ячейки
                                                       //1 - слева   (7)
                                                       //2 - сверху  (8)
                                                       //4 - снизу   (9)
                                                       //8 - справа  (10)
                              // или сумма когда рамок несколько (от 1 до 15)
      bool CellFontSizeSet(int CellRow, int CellCol, int Font_Size );
                                                      //изменяет размер шрифта
      bool CellFontSet(int CellRow, int CellCol, AnsiString Font_name );
                                                      //изменяет вид шрифта
      bool CellFontStyleSet(int CellRow, int CellCol, int Font_Style);
                                                       //изменяет стиль шрифта
                                                       //0 - обычный
                                                       //1 - Жирный
                                                       //2  - курсив
                                                       //4 - подчеркивание
      bool CellWrapTextSet(int CellRow, int CellCol);
                                                //сворачивает текст в ячейке
      bool CellVAlignSet(int CellRow, int CellCol, int VAlign_Style);
            //вертикальное выравнивание в ячейке
            //VAlignBottom  -4107
            //VAlignCenter  -4108
            //VAlignTop  -4160
```

П.Б

```
      bool CellHAlignSet(int CellRow, int CellCol, int HAlign_Style);
                  //горизонтальное выравнивание в ячейке
                  //HAlignCenter  -4108
                  //HAlignLeft  -4131
                  //HAlignRight  -4152 +
      bool RangeMerge(int CellRow1, int CellCol1,int CellRow2, int CellCol2);

private:

      Variant var_Excel,var_Book,   var_Sheet,var_Cell;
      bool write;//true если для записи, false для чтения
      int flag;//текущее состояние подключения к Excel
      //0 - сервер Excel не запущен
      //1 - сервер Excel запущен
      //2 - сервер Excel запущен и открыта книга Excel
      //3 - сервер Excel запущен и открыта книга Excel с нужным листом
//4 - сервер Excel запущен и открыта книга Excel с нужными листом и яейкой
      //-1 - сбой при запуске сервера Excel
      //-2 - сбой при запуске книги  Excel
      //-3 - сбой при запуске листа  Excel
      //-4 - сбой при работе с ячейками
      //-5 - сбой при закрытии сервера Excel
};
#endif
//---------------------------------------------------------------------------
```

Файл ExcelServer.cpp

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "ExcelServer.h"

//---------------------------------------------------------------------------

#pragma package(smart_init)
 //код наличия рамки ячейки
#define BLeft 1      //слева
#define BRight 8     //справа
#define BTop 2       //сверху
#define BBottom 4   //снизу

//выравнивание
//по горизонтали
#define HAlignCenter  -4108 //по центру
#define HAlignLeft  -4131//по левому краю
#define HAlignRight  -4152//по правому краю

//выравнивание
//по вертикали
#define VAlignBottom  -4107  //по нижнему краю
#define VAlignCenter  -4108 //по центру
#define VAlignTop  -4160 //по высоте


 //линии рамки
#define LNone 0      //рамка отсутствует
#define LSingle 1    //одинарная тонкая
```

П.Б

```
#define LDouble 9    //двойная тонкая
#define LBold 7      //жирная
ExcelServer::ExcelServer()
{
write=false;
try
      {
      flag=0;//начали работу с сервером
      var_Excel = CreateOleObject("Excel.Application");//создаем OLE объект
      flag=1;//если все ОК, флаг на успешное подключение с сервером
      }
catch(...)//ловим исключения
      {
         MessageBox( 0, "Ошибка при запуске сервера Excel", "Ошибка", MB_OK );
         flag=-1; // сообщение об ошибке, флаг на ошибку
      }
}
bool ExcelServer::ExcelOpen(AnsiString& BookName, bool write_mode)
{
write=write_mode;
if(flag==1)//сервер Excel запущен
{
      try
      {
            Variant books;
            books = var_Excel.OlePropertyGet("Workbooks");
            if(!write)

      books.Exec(Procedure("Open")<<BookName<<"ReadOnly:=true"<<"CorruptLoad:=
xlExtractData");
            else
                  books.Exec(Procedure("Open")<<BookName);
            var_Book = books.OlePropertyGet("item",1);
            flag=2;//если все ОК, флаг на успешное открытии книги
            SheetOpen(1);
            return true;
      }
      catch(...)
      {
         MessageBox( 0, StrCat("Ошибка при открытии книги Excel
",BookName.c_str()), "Ошибка", MB_OK );
         flag=-2; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}
bool ExcelServer::SheetOpen(AnsiString& SheetName)
{
if(flag==2)//если книга запущена
{
      try
      {
            var_Sheet= var_Book.OlePropertyGet("WorkSheets",SheetName);
            var_Excel.OlePropertySet("Visible", 1);

      var_Excel.OlePropertyGet("Application").OlePropertySet("WindowState",
"2");
            flag=3;//если все ОК, флаг на успешное открытии листа
            return true;
      }
```

П.Б

```
      catch(...)
      {
         MessageBox( 0, StrCat("Ошибка при открытии листа Excel
",SheetName.c_str()), "Ошибка", MB_OK );
         flag=-3; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}

bool ExcelServer::SheetOpen(int SheetNum)
{
if(flag==2)//если книга запущена
{
      try
      {
            var_Sheet= var_Book.OlePropertyGet("WorkSheets",SheetNum);
            var_Excel.OlePropertySet("Visible", 1);

      var_Excel.OlePropertyGet("Application").OlePropertySet("WindowState",
"2");
            flag=3;//если все ОК, флаг на успешное открытии листа
            return true;
      }
      catch(...)
      {

         AnsiString tmp=AnsiString(SheetNum);
         MessageBox( 0, StrCat("Ошибка при открытии листа Excel №",
tmp.c_str()), "Ошибка", MB_OK );
         flag=-3; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}

bool ExcelServer::CellSet(int CellRow, int CellCol, float CellValue)
{
if(flag==3)//если лист запущен
{
      try
      {
            var_Cell=var_Sheet.OlePropertyGet("Cells",CellRow,CellCol);
            //проверяем язык работы для правильного имени формата
            if
(var_Excel.OlePropertyGet("LanguageSettings").OlePropertyGet("LanguageID",2)=
=1049)
            {
            var_Cell.OlePropertySet("NumberFormat","Основной");
            }
            else
            {
            var_Cell.OlePropertySet("NumberFormat","General");
            }
            var_Cell.OlePropertySet("Value",CellValue);
            //Variant str=var_Cell.OlePropertyGet("NumberFormat");
            flag=4;//если все ОК, флаг на успешное открытии листа
            return true;
```

П.Б

```
      }
      catch(...)
      {
         AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
         MessageBox( 0, StrCat("Ошибка при работе с ячейкой Excel
",tmp.c_str()), "Ошибка", MB_OK );
         flag=-4; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}

bool ExcelServer::CellSet(int CellRow, int CellCol, AnsiString& CellText)
{
if(flag==3)//если лист запущен
{
      try
      {
            var_Cell=var_Sheet.OlePropertyGet("Cells",CellRow,CellCol);
            //проверяем язык работы для правильного имени формата
            var_Cell.OlePropertySet("NumberFormat","@");
            var_Cell.OlePropertySet("Value",CellText.c_str());
            //Variant str=var_Cell.OlePropertyGet("NumberFormat");
            flag=4;//если все ОК, флаг на успешное открытии листа
            return true;
      }
      catch(...)
      {
         AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
         MessageBox( 0, StrCat("Ошибка при работе с ячейкой Excel
",tmp.c_str()), "Ошибка", MB_OK );
         flag=-4; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}
bool ExcelServer::CellSetDate(int CellRow, int CellCol, AnsiString& Cell-
TextDate)
{
if(flag==3)//если лист запущен
{
      try
      {
            var_Cell=var_Sheet.OlePropertyGet("Cells",CellRow,CellCol);
            //проверяем язык работы для правильного имени формата
            if
(var_Excel.OlePropertyGet("LanguageSettings").OlePropertyGet("LanguageID",2)=
=1049)
            {
            var_Cell.OlePropertySet("NumberFormat","Основной");
            var_Cell.OlePropertySet("Value",CellTextDate.c_str());
            var_Cell.OlePropertySet("NumberFormat","ДД.ММ.ГГГГ");
            }
            else
            {
            var_Cell.OlePropertySet("NumberFormat","General");
```

```
            var_Cell.OlePropertySet("Value",CellTextDate.c_str());
            var_Cell.OlePropertySet("NumberFormat","DD.MM.YYYY");
            }
            //Variant str=var_Cell.OlePropertyGet("NumberFormat");
            flag=4;//если все ОК, флаг на успешное открытии листа
            return true;
      }
      catch(...)
      {
         AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
         MessageBox( 0, StrCat("Ошибка при работе с ячейкой Excel
",tmp.c_str()), "Ошибка", MB_OK );
         flag=-4; // сообщение об ошибке, флаг на ошибку
         return false;
      }
}
else
   return false;
}
bool ExcelServer::CellBorderSet( int CellRow, int CellCol, int Border_Code,
                                            int Line_Style )
{
  if(write)
  {
      try
      {
            var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
            switch(Line_Style)
            {
            //одинарная тонкая линия
            case 1:
            {
            //проверим в цикле необходимость рисования
            //рамки с каждой стороны ячейки
            for( int i=1; i <= 4; i++ )
                  if( (Border_Code*2) & (2<<(i-1)))
                        var_Cell.OlePropertyGet("Borders",
i+6).OlePropertySet("LineStyle", LSingle);
            return true;
            }
            //жирная линия
            case 7:
            {
            for( int i=1; i <= 4; i++ )
                  if( (Border_Code*2) & (2<<(i-1)))
                  {
                  var_Cell.OlePropertyGet("Borders",
i+6).OlePropertySet("LineStyle", LSingle);
                  var_Cell.OlePropertyGet("Borders",
i+6).OlePropertySet("LineStyle", LBold);
                  }
            return true;
            }
            //двойная тонкая
            case 9:
            {
            for( int i=1; i <= 4; i++ )
                  if((Border_Code*2) & (2<<(i-1)))
                  var_Cell.OlePropertyGet("Borders",
i+6).OlePropertySet("LineStyle", LDouble);
            return true;
```

П.Б

```
            }

            //если задан другой стиль линии рамки
            default:
            {
            for( int i=1; i <= 4; i++ )
                  if( (Border_Code*2) & (2<<(i-1)))
                  var_Cell.OlePropertyGet("Borders",
i+6).OlePropertySet("LineStyle", Line_Style);
            return true;
            }
            }
      }
      catch(...)
      {
         AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
         MessageBox( 0, StrCat("Ошибка при рисовании рамки ячейки
",tmp.c_str()), "Ошибка", MB_OK );
           return false;
      }
  }
  else return false;
}

bool ExcelServer::CellFontSizeSet(int CellRow, int CellCol, int Font_Size)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);

      var_Cell.OlePropertyGet("Font").OlePropertySet("Size",Font_Size);
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
                  MessageBox( 0, StrCat("Ошибка при изменении размера шрифта
ячейки ",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;
}
bool ExcelServer::CellFontSet(int CellRow, int CellCol, AnsiString Font_Name)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);

      var_Cell.OlePropertyGet("Font").OlePropertySet("Name",Font_Name.c_str())
;
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
```

```
                    MessageBox( 0, StrCat("Ошибка при изменении типа шрифта
ячейки ",tmp.c_str()), "Ошибка", MB_OK );
                    return false;
                    }
      }
      else return false;
}

bool ExcelServer::CellFontStyleSet(int CellRow, int CellCol, int Font_Style)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
                  if (Font_Style==0)
                  {
                  var_Cell.OlePropertyGet("Font").OlePropertySet("Bold",0);
                  var_Cell.OlePropertyGet("Font").OlePropertySet("Italic",0);

      var_Cell.OlePropertyGet("Font").OlePropertySet("Underline",0);
                  }
                  if (Font_Style & 1)
                  {
                  var_Cell.OlePropertyGet("Font").OlePropertySet("Bold",1);
                  }
                  if (Font_Style & 2)
                  {
                  var_Cell.OlePropertyGet("Font").OlePropertySet("Italic",1);
                  }
                  if (Font_Style & 4)
                  {

      var_Cell.OlePropertyGet("Font").OlePropertySet("Underline",1);
                  }
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
                  MessageBox( 0, StrCat("Ошибка при изменении стиля шрифта
ячейки ",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;
}

 bool ExcelServer::CellWrapTextSet(int CellRow, int CellCol)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
                  var_Cell.OlePropertySet("WrapText",1);
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+"
Столбец"+AnsiString(CellCol);
```

```
                  MessageBox( 0, StrCat("Ошибка при  сворачивании текста 
ячейки ",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;
}
bool ExcelServer::CellVAlignSet(int CellRow, int CellCol, int VAlign_Style)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
                  var_Cell.OlePropertySet("VerticalAlignment",VAlign_Style);
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+" 
Столбец"+AnsiString(CellCol);
                  MessageBox( 0, StrCat("Ошибка при верт.выравнивании ячейки 
",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;
}

bool ExcelServer::CellHAlignSet(int CellRow, int CellCol, int HAlign_Style)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
                  var_Cell.OlePropertySet("HorizontalAlignment",HAlign_Style);
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="Строка "+AnsiString(CellRow)+" 
Столбец"+AnsiString(CellCol);
                  MessageBox( 0, StrCat("Ошибка при гориз.выравнивании ячейки 
",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;
}
bool ExcelServer::RangeMerge(int CellRow1, int CellCol1,int CellRow2, int 
CellCol2)
{
      if (write)
      {
            try {
                  var_Cell=var_Sheet.OlePropertyGet("Cells", Cell-
Row1,CellCol1);
                  Variant var_Cell2;
                  var_Cell2=var_Sheet.OlePropertyGet("Cells", Cell-
Row2,CellCol2);

      var_Sheet.OlePropertyGet("Range",var_Cell,var_Cell2).Exec(Procedure("Mer
ge"));
```

```
                  return true;
                  }
            catch(...)
                  {
                  AnsiString tmp="1 ячейка Строка "+AnsiString(CellRow1)+"
Столбец"+AnsiString(CellCol1)+" 2 ячейка Строка "+AnsiString(CellRow2)+"
Столбец"+AnsiString(CellCol2);
                  MessageBox( 0, StrCat("Ошибка  при объединении ячеек
",tmp.c_str()), "Ошибка", MB_OK );
                  return false;
                  }
      }
      else return false;

}
bool ExcelServer::ExcelClose(void)
{
if(flag!=0)
      try
      {
      if(!write)
var_Excel.OlePropertyGet("Application").OlePropertySet("DisplayAlerts",0);
      var_Excel.Exec(Procedure("Quit"));
      flag=0;
      return true;
      }
      catch(...)
      {
      MessageBox( 0, "Ошибка при закрытии приложения Excel", "Ошибка", MB_OK
);
      flag=-5;
      return false;
      }
else
      return false;
}
bool ExcelServer::CellFormatSet(int CellRow, int CellCol,AnsiString& CellFor-
mat)
{
 if(flag!=0)
      try
      {
      var_Cell=var_Sheet.OlePropertyGet("Cells", CellRow,CellCol);
      AnsiString tmpFormat;
      if
(var_Excel.OlePropertyGet("LanguageSettings").OlePropertyGet("LanguageID",2)=
=1049)
            {
            if (CellFormat=="")
                  {
                  var_Cell.OlePropertySet("NumberFormat","Основной");
                  flag=4;
                  return true;
                  }
            if(AnsiPos(AnsiString("0"),CellFormat)!=0 || Ansi-
Pos(AnsiString("#"),CellFormat)!=0) //если формат числовой
                  {
                  tmpFormat=AnsiReplaceStr(CellFormat,".",",");
                  var_Cell.OlePropertySet("NumberFormat",tmpFormat.c_str());
                  flag=4;
                  return true;
                  }
```

П.Б

```
            if(AnsiPos(AnsiString("d"),CellFormat)!=0 || Ansi-
Pos(AnsiString("m"),CellFormat)!=0 || AnsiPos(AnsiString("y"),CellFormat)!=0
|| AnsiPos(AnsiString("M"),CellFormat)!=0 ||  Ansi-
Pos(AnsiString("s"),CellFormat)!=0 ||  Ansi-
Pos(AnsiString("h"),CellFormat)!=0) //если формат даты
                  {
                  tmpFormat=AnsiReplaceStr(CellFormat,"d","д");
                  tmpFormat=AnsiReplaceStr(tmpFormat,"y","г");
                  tmpFormat=AnsiReplaceStr(tmpFormat,"m","м");
                  tmpFormat=AnsiReplaceStr(tmpFormat,"M","М");
                  tmpFormat=AnsiReplaceStr(tmpFormat,"h","ч");
                  tmpFormat=AnsiReplaceStr(tmpFormat,"s","с");
                  var_Cell.OlePropertySet("NumberFormat",tmpFormat.c_str());
                  flag=4;
                  return true;
                  }
            var_Cell.OlePropertySet("NumberFormat",CellFormat.c_str());
            flag=4;
            return true;
            }
      else
            {
                  if(CellFormat=="")
                  {
                        var_Cell.OlePropertySet("NumberFormat","General");
                        flag=4;
                        return true;
                  }
                  else
                  {

      var_Cell.OlePropertySet("NumberFormat",CellFormat.c_str());
                        flag=4;
                        return true;
                  }
            }
      }
      catch(...)
      {
      AnsiString tmp=AnsiString("Строка ")+AnsiString(CellRow)+AnsiString("
Столбец")+AnsiString(CellCol);
      MessageBox( 0, StrCat("Ошибка при изменении числового формата
",tmp.c_str()), "Ошибка", MB_OK );
      return false;
      }
else
      return false;

}


ExcelServer::~ExcelServer()
{
 if (!write) {
 ExcelClose();
 }
}
```

Приложение В

Функция чтения текстовых данных из файла XML

```
void __fastcall TSDIAppForm::ReadXML(void)
{
	// Работа с корневым элементом
	_di_IXMLNode RootNode = XMLDocument1->DocumentElement;
	// Доступ к корневому элементу
	_di_IXMLNode CurNode,SrcNode; // врменные переменные для указателей
	                                // на дочерний и родительский узел
	int level=1; // уровень загрузки дерева 1
	Memo1->Lines->Add("Начала дерева - "+AnsiString(RootNode->GetNodeName()));
	// вывод в поле Memo информации о начале XML-дерева
	CurNode=SrcNode=RootNode; // указатели узлов на корневой
	while(level!=0) // пока уровень иерархии не нулевой
	{
	if(!CurNode->HasChildNodes || CurNode->IsTextElement)
	// если узел текстовый или не имеет дочерних узлов
	{
		while(CurNode==SrcNode->ChildNodes->GetNode(SrcNode->ChildNodes->Count-
1)) // если мы обошли все дочерние узлы для SrcNode
		{
			if(CurNode!=RootNode) // Если поиск не начальном этапе
			{
				// Переход на один уровень иерархии вверх CurNode=SrcNode;
				SrcNode=SrcNode->ParentNode;
				level--;
			}
			else
			{
				return; // в противном случае выход из цикла
			}
		}
		CurNode=CurNode->NextSibling(); // переход к следующему
		                                 // дочернему узлу для SrcNode
		if(CurNode->IsTextElement) // этот узел текстовый
		{
			Memo1->Lines->Add("Уровень - "+AnsiString(level)+"
"+AnsiString(CurNode->GetNodeName())+" "+AnsiString(CurNode->GetNodeValue()));
			// если да, то печать его текста и названия в поле Memo
		}
		else
		Memo1->Lines->Add("Уровень - "+AnsiString(level)+" "+AnsiString(CurNode-
>GetNodeName()));
			// в противном случае вывести название и уровень CurNode
	}
	else // если узел CurNode имеет дочерние узлы
	{
		// переход на один уровень иерархии вниз
		SrcNode=CurNode;
		CurNode=CurNode->ChildNodes->GetNode(0);
		level++;
		if(CurNode->IsTextElement) //Если узел CurNode текстовый
		{
			Memo1->Lines->Add("Уровень - "+AnsiString(level)+"
"+AnsiString(CurNode->GetNodeName())+" "+AnsiString(CurNode->GetNodeValue()));// если
да, то печать его текста и названия в поле Memo
		}
		else //в противном случае	{
		{	Memo1->Lines->Add("Уровень - "+AnsiString(level)+"
"+AnsiString(CurNode->GetNodeName()));
			// если да, то печать его текста и названия в поле Memo
		}
	}// окончание перехода на один уровень по ирархии вниз
	}// окончание цикла обхода дерева
}
```

П.В

Функции импорта данных из файла XML

```
void __fastcall TSDIAppForm::ImportXML(void)
{
      _di_IXMLNode RootNode = XMLDocument1->DocumentElement;
      Variant tmp="product";
      AnsiString tmpstr;
      if(RootNode->GetNodeName()=="spr" && RootNode->GetAttribute("vid")==tmp)
            {
                  ImportXMLSpr();
                  return;
            }
      tmp="strategy";
      if(RootNode->GetNodeName()=="spr" && RootNode->GetAttribute("vid")==tmp)
            {
                  ImportXMLStrategy();
                  return;
            }
      if(RootNode->GetNodeName()=="sales")
            {
                  ImportXMLSales();
                  return;
            }
      if(RootNode->GetNodeName()=="costs")
            {
                  ImportXMLCosts();
                  return;
            }
      tmpstr=AnsiString(XMLDocument1->FileName);
      MessageBox( 0, StrCat("Неверный формат корневого узла файла XML ",
tmpstr.c_str()), "Ошибка", MB_OK);
 }
void __fastcall TSDIAppForm::ImportXMLSpr(void)
{
      AnsiString tmp;
      DateTimeToString(tmp,AnsiString("dd_mm_hh_nn_ss"),Now());
      tmp="xml_sprGP"+tmp+".log";
      int fHandle=-1;
      try
      {
            if(!FileExists(tmp)) fHandle=FileCreate(tmp);
            else
            {fHandle=FileOpen(tmp,fmOpenWrite);
            }
            if(fHandle==-1)
            MessageBox( 0, "Ошибка создания или открытия log-файла ", "Ошиб-
ка", MB_OK);
            AnsiString buff="Начало экспорта";
            if(fHandle!=-1)
                  FileWrite(fHandle,buff.c_str(),buff.Length());
            _di_IXMLNode RootNode = XMLDocument1->DocumentElement;
            _di_IXMLNode SrcNode,CurNode;
            if (RootNode->HasChildNodes)
            {
            ParseXMLSpr(fHandle);
            }
            FileClose(fHandle);}
      catch(...)
      { MessageBox( 0, "Ошибка создания или открытия log-файла ", "Ошибка",
MB_OK);}
}
```

Приложение Г

Схема XML-данных для данных о продажах

```
я╗┐<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="sales" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="sales">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="sales_item" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="KLS_NUM" type="xs:string" minOccurs="0" maxOccurs="1"
/>
              <xs:element name="NAME" type="xs:string" minOccurs="0" maxOccurs="1" />
              <xs:element name="STRATEGY_ID" type="xs:string" minOccurs="1" maxOc-
curs="1" />
              <xs:element name="PRODUCT_ID" type="xs:string" minOccurs="1" maxOc-
curs="1"/>
              <xs:element name="PRICE" minOccurs="1" maxOccurs="1" >
             <xs:simpleType>
                   <xs:restriction base="xs:decimal">
                         <xs:minInclusive value="0"/>
                         <xs:fractionDigits value="2"/>
                   </xs:restriction>
             </xs:simpleType>
              </xs:element>
              <xs:element name="QUANTITY" minOccurs="1" maxOccurs="1">
             <xs:simpleType>
                   <xs:restriction base="xs:decimal">
                         <xs:minInclusive value="0"/>
                         <xs:fractionDigits value="5"/>
                   </xs:restriction>
             </xs:simpleType>
              </xs:element>
              <xs:element name="USER_ID" type="xs:string" minOccurs="0" maxOc-
curs="1"/>
            </xs:all>
            <xs:attribute name="rewrite" type="xs:boolean" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

П.Г

Схема XML-данных для данных о нормах затрат

```
я╗┐<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="costs" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="costs">
    <xs:complexType>
      <xs:all>
        <xs:element name="var_costs" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="KLS_NUM" type="xs:string" minOccurs="0" maxOccurs="1"
/>
              <xs:element name="NAME" type="xs:string" minOccurs="0" maxOccurs="1" />
              <xs:element name="STRATEGY_ID" type="xs:string" minOccurs="1" maxOc-
curs="1" />
              <xs:element name="PRODUCT_ID" type="xs:string" minOccurs="1" maxOc-
curs="1"/>
              <xs:element name="NORM" minOccurs="1" maxOccurs="1">
             <xs:simpleType>
                    <xs:restriction base="xs:decimal">
                           <xs:minInclusive value="0"/>
                           <xs:fractionDigits value="5"/>
                    </xs:restriction>
             </xs:simpleType>
              </xs:element>
              <xs:element name="TYPE_ID" minOccurs="1" maxOccurs="1">
             <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="1"/>
                        <xs:enumeration value="2"/>
                        <xs:enumeration value="3"/>
                    </xs:restriction>
             </xs:simpleType>
              </xs:element>
              <xs:element name="USER_ID" type="xs:string" minOccurs="0" maxOc-
curs="1"/>
            </xs:all>
            <xs:attribute name="rewrite" type="xs:boolean" />
          </xs:complexType>
        </xs:element>
        <xs:element name="fixed_costs" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="KLS_NUM" type="xs:string" minOccurs="0" maxOccurs="1"
/>
              <xs:element name="NAME" type="xs:string" minOccurs="0" maxOccurs="1" />
              <xs:element name="PRODUCT_ID" type="xs:string" minOccurs="1" maxOc-
curs="1"/>
              <xs:element name="NORM" minOccurs="1" maxOccurs="1">
             <xs:simpleType>
                    <xs:restriction base="xs:decimal">
                           <xs:minInclusive value="0"/>
                           <xs:fractionDigits value="2"/>
                    </xs:restriction>
             </xs:simpleType>
              </xs:element>
              <xs:element name="USER_ID" type="xs:string" minOccurs="0" maxOc-
curs="1"/>
            </xs:all>
            <xs:attribute name="rewrite" type="xs:boolean" />
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Приложение Д

Дипломы


dialog e.V. Ассоциация экономистов Германии и России
Vereinigung Deutscher und Russischer Ökonomen диалог +

ДИПЛОМ

Настоящий диплом вручается

Латиповой Алине Таиховне,

занявшему(шей) 2е место в конкурсе докладов на конференции молодых ученых, проходившей в рамках Международного Диалог-симпозиума

«НАУКА И ИННОВАЦИИ: ТРАНСФЕР ЗНАНИЙ И ТЕХНОЛОГИЙ ЧЕРЕЗ ТЕХНОЛОГИЧЕСКИЕ И БИЗНЕС-ИНКУБАТОРЫ В РОССИИ И ГЕРМАНИИ»

Диалог+ Томск
8-9 Октября 2004 г.
г. Томск

400 ТОМСК

Председатель НП "Диалог+ Томск"
Д.М. Хлопцов

П.Д

IME

АНХ

Министерство образования и науки Российской Федерации
Институт проблем региональной экономики РАН
Академия народного хозяйства при Правительстве РФ
Санкт-Петербургский государственный университет
Санкт-Петербургский государственный университет экономики и финансов
Санкт-Петербургский государственный инженерно-экономический университет
Санкт-Петербургский государственный университет телекоммуникаций
Санкт-Петербургская военная академия связи
Академия проблем военной экономики и финансов
Международная академия менеджмента
Учебно-методическое объединение вузов России в области менеджмента
Комитет экономики, промышленной политики и торговли Санкт-Петербурга

ИНСТИТУТ УПРАВЛЕНИЯ И ЭКОНОМИКИ (САНКТ-ПЕТЕРБУРГ)
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ДИПЛОМ

УЧАСТНИКА IV-й МЕЖДУНАРОДНОЙ
НАУЧНО-ПРАКТИЧЕСКОЙ КОНФЕРЕНЦИИ
«ЭКОНОМИКА И МЕНЕДЖМЕНТ: ПРОБЛЕМЫ
И ПЕРСПЕКТИВЫ»

6 – 11 ИЮНЯ 2005 ГОДА

Санкт-Петербург

Изд-во СПбГПУ. Зак. 244-Б. Т. 100. 2005.

П.Д

Свидетельство о государственной регистрации программы

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

об официальной регистрации программы для ЭВМ

№ 2007613463

Программа оптимизации бюджета продаж при ценовой диверсификации

Правообладатель(ли): *Латипова Алина Таиховна (RU), Панюков Анатолий Васильевич (RU)*

Автор(ы): *Латипова Алина Таиховна, Панюков Анатолий Васильевич (RU)*

Заявка № 2007612433

Дата поступления 19 июня 2007 г.

Зарегистрировано в Реестре программ для ЭВМ

15 августа 2007 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

Приложение Е

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ И ОГРАНИЧЕННАЯ ГАРАНТИЯ

на программный продукт

«Программа оптимизации бюджета продаж при ценовой диверсификации»

(1 рабочее место)

Настоящее лицензионное соглашение заключается между Челябинский филиалом ЗАО «Пронто-Уфа» (далее «Покупатель»), Латиповой Алиной Таиховной и Панюковым Анатолием Васильевичем (далее «Авторы») относительно указанного выше программного продукта (далее «программа») Авторов, включающего в себя программное обеспечение, записанное на соответствующих носителях и электронную документацию. Устанавливая, копируя или иным образом используя программу, Покупатель тем самым принимает на себя условия настоящего соглашения. Если Покупатель не принимает условий данного соглашения, то он не имеет права использовать данную программу.

Настоящее соглашение дает Покупателю нижеследующие права.

Разрешается установка на компьютере и использование одной копии программы или любой предыдущей версии.

Не разрешается осуществлять вскрытие технологии, декомпиляцию и дизассемблирование программы, за исключением и только в той степени, в которой такие действия явно разрешены действующим законодательством, несмотря на наличие в соглашении данного ограничения.

Программа лицензируется как единое целое. Не разрешается предоставлять программу в прокат или во временное пользование. Все права собственности и авторские права на программу сопровождающие ее печатные материалы и любые копии программы принадлежат Авторам. Все права Авторов на программу защищены законами и международными соглашениями об авторских правах, а также другими законами и договорами, регулирующими отношения авторского права.

Программа может поставляться на нескольких видах носителей.

Программа распространяется по принципу "AS IS" (как есть). Авторы не несут никакой ответственности за любой ущерб, нанесенный программой, любую упущенную выгоду в результате или процессе использования программы и не дают никаких гарантий в отношении выхода новых версий.

АВТОРЫ

______ (Латипова А. Т.) 29 октября 2007 год

______ (Панюков А.В.) 29.10.2007

ПОКУПАТЕЛЬ

ЧФ ЗАО «Пронто-Уфа»

ИНН-0278009824, КПП-745303001

Адрес: 454080, г. Челябинск, пр-т Ленина, 81-410

Директор ЧФ ЗАО «Пронто-Уфа» ______ А.В. Самохвалов

29 октября 2007 год

ПРОНТО-УФА