

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания
кафедра

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

09.03.02 Информационные системы и технологии
код и наименование направления

**ПРОЕКТИРОВАНИЕ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЯ НА ОСНОВЕ
ОЦЕНКИ РЕЙТИНГА ЗАЕМЩИКОВ КРЕДИТНОЙ ОРГАНИЗАЦИИ**
тема

Руководитель


_____ подпись

Е.В. Киргизова
инициалы, фамилия

Выпускник


_____ подпись

Д.В. Погодин
инициалы, фамилия

Лесосибирск 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания
кафедра

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

09.03.02 Информационные системы и технологии
код и наименование направления

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЯ НА ОСНОВЕ
ОЦЕНКИ РЕЙТИНГА ЗАЕМЩИКОВ КРЕДИТНОЙ ОРГАНИЗАЦИИ
тема

Работа защищена « 24 » июня 2019 г. с оценкой «отлично»

Председатель ГЭК


подпись

Е.Г. Кучумов
инициалы, фамилия

Члены ГЭК


подпись

Т.В. Захарова
инициалы, фамилия


подпись

Е.В. Киргизова
инициалы, фамилия


подпись

А.А. Степанов
инициалы, фамилия


подпись

В.В. Фирер
инициалы, фамилия

Руководитель


подпись

Е.В. Киргизова
инициалы, фамилия

Выпускник


подпись

Д.В. Погодин
инициалы, фамилия

Лесосибирск 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Проектирование системы принятия решения на основе оценки рейтинга заемщиков кредитной организации» содержит 76 страниц текстового документа, 31 рисунок, 6 таблиц, 3 приложения, 44 использованных источника.

Ключевые слова: ПРОЕКТИРОВАНИЕ, СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ, ОЦЕНКА РЕЙТИНГА, АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА

Работа актуальна тем, что на исследуемом предприятии в настоящее время оценка кредитоспособности потенциальных заемщиков осуществляется без использования средств поддержки принятия решений, что делает данный процесс очень трудоемким. Внедрение автоматизированной системы позволит снизить число рутинных операций менеджера кредитного отдела.

Цель работы: разработать автоматизированную информационную систему поддержки принятия решения на основе рейтинга заемщиков кредитной организации ООО «Норд».

Объект исследования: системы поддержки принятия решения.

Предмет исследования: особенности проектирования систем поддержки принятия решения на основе оценки рейтинга заемщиков кредитной организации.

Основные задачи исследования:

- рассмотреть основные понятия, связанные с системами поддержки принятия решения, кредитным риском и процессом кредитования;
- обосновать выбор средств для проектирования автоматизированной информационной системы поддержки принятия решений;
- выполнить реализацию, а также тестирование информационной системы принятия решения на основе оценки рейтинга заемщиков кредитной организации.

В процессе исследования спроектирована и разработана автоматизированная информационная система поддержки принятия решения на основе оценки рейтинга заемщиков кредитной организации ООО «Норд».

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области	7
1.1 Основные характеристики систем поддержки принятия решений	7
1.2 Основные закономерности построения систем управления рисками	15
1.3 Методики оценки кредитоспособности заемщиков	16
1.4 Характеристика предприятия	24
2 Проектирование информационной системы принятия решения на основе оценки рейтинга заемщиков кредитной организации	27
2.1 Постановка задачи на проектирование	27
2.2 Анализ существующего программного обеспечения.....	28
2.3 Выбор прикладных инструментов проектирования.....	30
2.4 Проектирование базы данных.....	39
2.5 Проектирование интерфейса пользователя	42
3 Реализация, тестирование и оценка экономической эффективности	46
3.1 Разработка программного кода приложения.....	46
3.2 Тестирование приложения	54
Заключение	61
Список использованных источников	63
Приложения А SQL скрипты инициализации базы данных.....	67
Приложение Б Листинги программных модулей	69
Приложение В Сертификат научной статьи.....	76

ВВЕДЕНИЕ

Автоматизация бизнес-процессов – это процесс управления информацией, данными и процессами для снижения затрат, ресурсов и инвестиций. Это повышает производительность за счет автоматизации ключевых бизнес-процессов с помощью компьютерных технологий [29].

Все это направлено на внедрение программных приложений для облегчения рутинных бизнес-задач путем инициации, выполнения и завершения, а также для достижения эффективности рабочего процесса в масштабах всего предприятия.

Автоматизация предназначена для поддержания эффективности и повышения стабильности и операционной производительности недостаточно загруженных работников путем интеграции критически важных для бизнеса программных приложений [31].

При наличии правильных инструментов автоматизация бизнес-процессов может быть достаточно легкой и может принести значительные преимущества для предприятия. Основными преимуществами автоматизации операций, которые чаще всего указываются, являются: снижение затрат, производительность, доступность, надежность.

Работа актуальна тем, что на исследуемом предприятии в настоящее время оценка кредитоспособности потенциальных заемщиков осуществляется без использования средств поддержки принятия решений, что делает данный процесс очень трудоемким. Внедрение автоматизированной системы позволит снизить число рутинных операций менеджера кредитного отдела.

Цель работы: разработать автоматизированную информационную систему поддержки принятия решения на основе рейтинга заемщиков кредитной организации ООО «Норд».

Объект исследования: системы поддержки принятия решения.

Предмет исследования: особенности проектирования систем поддержки принятия решения на основе оценки рейтинга заемщиков кредитной организации.

Для достижения поставленной в выпускной квалификационной работе цели необходимо решить следующие задачи:

- рассмотреть основные понятия, связанные с системами поддержки принятия решения, кредитным риском и процессом кредитования;
- обосновать выбор средств для проектирования автоматизированной информационной системы поддержки принятия решений;
- выполнить реализацию, а также тестирование информационной системы принятия решения на основе оценки рейтинга заемщиков кредитной организации.

Методологической основой исследования выступили труды отечественных и зарубежных ученых Е. А. Буглак, А.А. Волкова, Д. Тидвелла.

Методологическая база исследования: в процессе исследования использовался комплексный подход к объекту и предмету исследования, применялись общенаучные методы познания (индукция, дедукция, сравнение, анализ, синтез, обобщение), а также методы сравнительного статистического и динамического анализа, абстрактно-логический метод.

Практическая значимость заключается во внедрении автоматизированной информационной системы поддержки принятия решений на основе оценки рейтинга заемщиков кредитной организации ООО «Норд». Выпускная квалификационная работа может быть полезна студентам, обучающимся по направлению 09.03.02 Информационные системы и технологии.

По результатам исследования опубликована статья «Проектирование системы принятия решения на основе оценки рейтинга заемщиков кредитной организации» в сборнике XLII студенческой международной научно-практической конференции «Молодежный научный форум» №12(42), 2019.

Выпускная квалификационная работа содержит 76 страниц текстового документа, 31 рисунок, 6 таблиц, 2 приложения, 44 использованных источника.

1 Анализ предметной области

1.1 Основные характеристики систем поддержки принятия решений

Система поддержки принятия решений (СППР) (англ. Decision Support System, DSS) – компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности. СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных.

Для анализа и выработок предложений в СППР используются разные методы. Это могут быть: информационный поиск, интеллектуальный анализ данных, поиск знаний в базах данных, рассуждение на основе прецедентов, имитационное моделирование, эволюционные вычисления и генетические алгоритмы, нейронные сети, ситуационный анализ, когнитивное моделирование и др. Некоторые из этих методов были разработаны в рамках искусственного интеллекта. Если в основе работы СППР лежат методы искусственного интеллекта, то говорят об интеллектуальной СППР, или ИСППР [14].

Сформулировать общепринятое определение СППР трудно. Это объясняется тем, что ее конструкция существенно зависит от вида задач, для решения которых она разрабатывается, видов данных, возможностей программного обеспечения, а также от пользователей системы.

Тем не менее, можно выделить некоторые общепризнанные элементы и характеристики СППР. Прежде всего, СППР – это интерактивная автоматизированная система, которая помогает ЛПР (лицу принимающему решение) использовать данные и модели для принятия решений. Система должна обладать возможностью работать с интерактивными запросами, при этом язык формирования запросов должен быть достаточно простым для изучения.

Система поддержки принятия решения:

- оперирует слабоструктурированными задачами;
- предназначена для ЛПР различных уровней;
- может быть адаптирована для группового и индивидуального использования;
- поддерживает как взаимозависимые, так и последовательные решения;
- поддерживает три фазы процесса решения: интеллектуальную часть, проектирование и выбор;
- поддерживает различные методы решения, что может быть полезно при решении задачи группой ЛПР;
- является гибкой и адаптируется к изменениям как внутренней среды организации, так и ее окружения;
- проста в использовании и модификации;
- повышает эффективность процесса принятия решений;
- позволяет человеку управлять процессом принятия решений с помощью компьютера, а не наоборот;
- поддерживает эволюционное использование и легко адаптируется к изменяющимся требованиям;
- может быть легко построена, если может быть сформулирована логика конструкции СППР;
- поддерживает моделирование;
- позволяет использовать знания.

Данные есть результат наблюдения за физическим объектом или явлением: например, ежедневным объемом производства, ежедневным объемом продаж или уровнем запасов продукта. База данных есть совокупность взаимосвязанных файлов. Системы управления базами данных представляют собой компьютерные программы, связанные с управлением большим количеством данных в физическом хранилище, а также с созданием и обновлением запросов к базам данных. СУБД могут представлять собой как «стороннее» программное обеспечение, так и быть встроенными в СППР [20].

Математические модели обычно встроены в СППР, а пользователи могут создавать, редактировать, обновлять или удалять модели. Современные СППР представляют пользователю довольно широкий выбор режимов работы: на основе интерфейсных меню, языка команд, вопросов и ответов, а также взаимодействие на основе форм, систем распознавания речи и графического пользовательского интерфейса. В частности, графический пользовательский интерфейс (graphic users interface) предусматривает использование иконок, кнопок, выпадающих меню и панелей. В последние годы эти элементы стали наиболее распространенным способом общения пользователей с информационными системами. Простейшая архитектура СППР представлена на рисунке 1, а ее место в комплексной информационной системе предприятия – на рисунке 2.

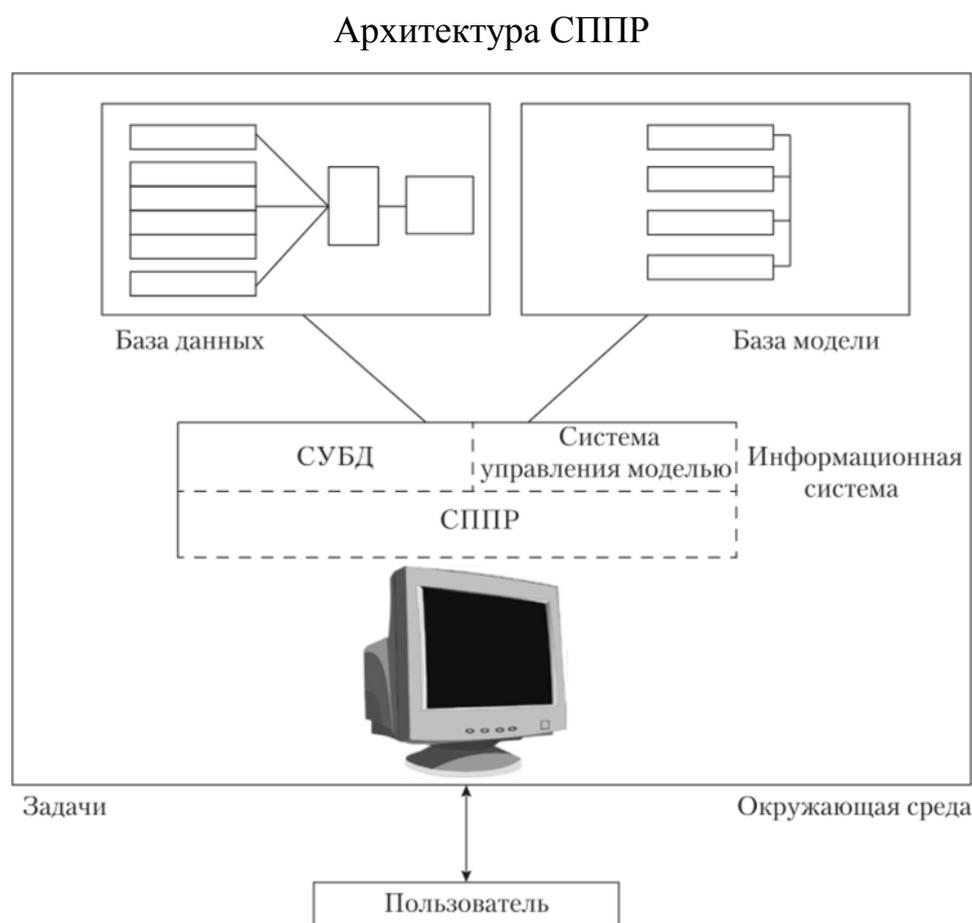


Рисунок 1 – Архитектура СППР



Рисунок 2 – Место СПДР в комплексной информационной системе предприятия

Для развертывания приложений широко используются средства пакетирования файлов в клиент/серверной среде.

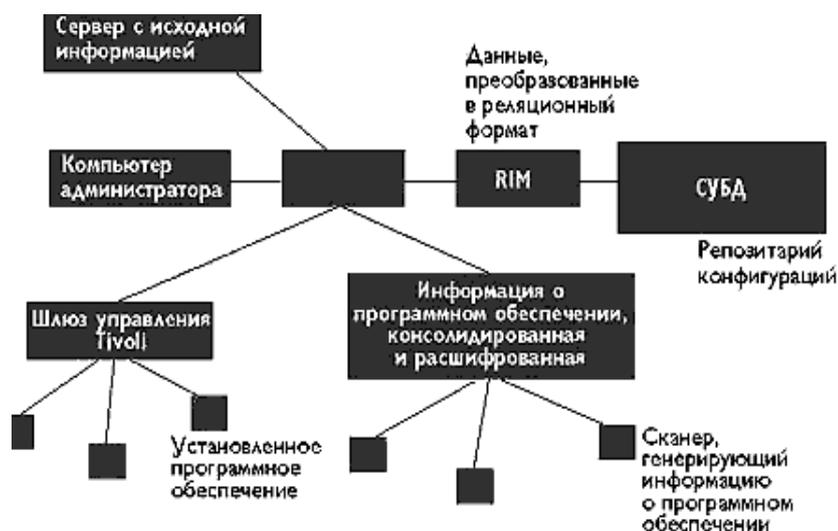


Рисунок 3 – Развертывание приложений по сети

Исследователи СППР также занимаются выявлением связей между работой людей и конечным результатом. Человеческий фактор включает в себя когнитивные вопросы (характерные особенности ЛПР в части обработки и использования информации для решения проблемы), личные вопросы, демографические характеристики (возраст, пол), а также персональные особенности (образование, уровень подготовки, наличие опыта, вовлеченность пользователя) [13].

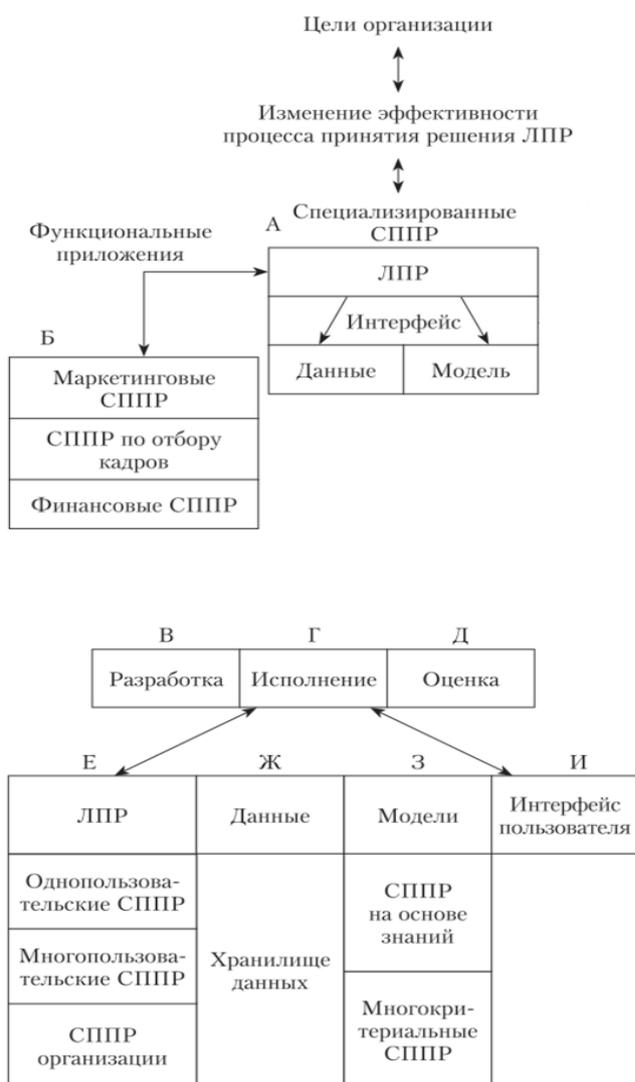


Рисунок 4 – процесс организации СППР

Интерактивность СППР означает, что система реагирует на различные действия, посредством которых человек влияет на вычислительный процесс, в частности, в диалоговом режиме. Человек и система обмениваются

информацией в темпе, который сравним со скоростью обработки информации человеком. Однако практика показывает, что лишь немногие руководители хотят и умеют вести прямой диалог с компьютером: многие предпочитают взаимодействие с системой через посредника или в режиме косвенного доступа, с возможностью пакетной обработки данных. Вместе с тем свойство интерактивности необходимо для исследования новых проблем и ситуаций при адаптивном проектировании прикладных СППР [24].

Интегрированность СППР – это совместимость составляющих системы управления данными и средств общения с пользователями в процессе поддержки принятия решений.

Мощность означает способность системы отвечать на существенные вопросы.

Доступность – это способность обеспечить предоставление ответов на запросы пользователя в определенной форме и в нужное время.

Гибкость характеризует возможность системы адаптироваться к изменениям потребностей и ситуаций.

Надежность означает способность системы выполнять требуемые функции в течение длительного периода.

Робастность – это способность системы восстанавливаться в случае возникновения ошибочных ситуаций как внешнего, так и внутреннего происхождения. Хотя между надежностью и робастностью существует определенная связь, это две разные характеристики: система, которая никогда не восстанавливается в случае возникновения ошибочных ситуаций, может быть надежной, не будучи робастной. Однако система с высоким уровнем робастности, которая может восстанавливаться и продолжать работу во многих ошибочных ситуациях, может быть отнесена к ненадежным, поскольку она может не обладать способностью выполнять необходимые процедуры [11].

Управляемость означает, что пользователь может контролировать действия системы, вмешиваясь в ход решения задачи.

Современные компьютерные системы поддержки принятия решений:

- предоставляют руководителю помощь в процессе принятия решений и обеспечивают поддержку во всем диапазоне структурированных, полуструктурированных и неструктурированных задач;

- поддерживают и делают более обоснованными соображения и оценки руководителя, но не заменяют и не отменяют их (контроль остается за человеком). Благодаря удобному интерфейсу пользователь чувствует себя комфортно и не боится работать с системой;

- повышают эффективность принимаемых решений. В отличие от административных информационных систем, в которых акцент делается на максимальную производительность аналитического процесса, в СППР значительно более весомыми являются эффективность процесса принятия решений и самих решений;

- интегрируют модели и аналитические методы с доступом к данным и выборкой данных. Для оказания помощи в принятии решений активизируются одна или несколько моделей (математических, статистических, имитационных, количественных, качественных или комбинированных). Содержание баз и хранилищ данных охватывает историю текущих и предшествующих операций, а также информацию внутреннего характера и информацию о среде;

- просты в использовании даже для лиц, которые не имеют богатого опыта работы с компьютером. Системы являются «дружественными» для пользователей, не требуют глубоких знаний в области вычислительной техники и обеспечивают простую навигацию, диалоговую документацию, встроенные средства обучения и другие атрибуты программных интерфейсных систем;

- построены по принципу интерактивного решения задач. Пользователь имеет возможность поддерживать диалог с СППР в непрерывном режиме, не ограничиваясь вводом отдельных команд с последующим ожиданием результатов;

- ориентированы на гибкость и адаптивность к изменениям как во внешней среде, так и в подходах к решению задач, которые выбирает пользователь;

- не навязывают пользователю какой-либо определенный процесс принятия решений. Пользователь имеет ряд возможностей и может выбирать их в форме и последовательности, которые соответствуют стилю «воображаемых моделей» его познавательной деятельности.

Оценка эффективности применения СППР связана с анализом затрат и выгод, получаемых от их внедрения. Уникальность данных систем состоит в том, что, хотя они обеспечивают значительную экономию средств и рост прибыли предприятия, оценка эффективности их применения представляется весьма проблематичной. Поэтому исследователи используют анализ последствий принимаемых решений, изменений в процессе принятия решений, концептуальных изменений в видении проблемы руководством, изменений в процедурах, а также анализ затрат и выгод, изменений в обслуживании, а также управленческие оценки системы [34].

Идеальная СППР:

- оперирует со слабоструктурированными решениями;
- предназначена для ЛПР различного уровня;
- может быть адаптирована для группового и индивидуального использования;
- поддерживает как взаимозависимые, так и последовательные решения;
- поддерживает три фазы процесса решения: интеллектуальную часть, проектирование и выбор;
- поддерживает разнообразные стили и методы решения, что может быть полезно при решении задачи группой ЛПР;
- является гибкой и адаптируется к изменениям как организации, так и ее окружения;
- проста в использовании и модификации;
- улучшает эффективность процесса принятия решений;
- позволяет человеку управлять процессом принятия решений с помощью компьютера, а не наоборот;

- поддерживает эволюционное использование и легко адаптируется к изменяющимся требованиям;
- может быть легко построена, если может быть сформулирована логика конструкции СППР;
- поддерживает моделирование;
- позволяет использовать знания.

1.2 Основные закономерности построения систем управления рисками

Согласно отчету исследовательской компании «Meridien Research» все разработчики программных обеспечений по управлению рисками в банке могут быть поделены на две группы [10]:

1. «Лидеры», предлагающие универсальные программные средства, позволяющие оценивать несколько видов рисков одновременно и рассчитывать возможные потери.
2. «Последователи», которые предлагают системы по выявлению рисков, которые подходят под одну определенную бизнес задачу банка.

К первой группе разработчиков можно отнести такие компании как: Algorithmics, SunGard Trading and Risk и SAS Risk Management.

Системы управления рисками компании Algorithmics внедрили и применяют 27 крупнейших банков во всем мире. Также продукты компании применяют страховые и биржевые компании, такие как: Allianz Group, Bluecrest, HSBC, Nedbank и другие.

Ко второй группе можно отнести таких разработчиков как: BARRA, IQ Financial, IRIS, Kamakura, MKI Risk и другие [30].

В настоящее время в мировой практике одной из распространенных технологий по управлению банковскими рисками является VaR-система. VaR выражает суммарную меру риска, а также VaR-система способна произвести сравнение рисков по разным и по различным финансовым инструментам. VaR-

система позволяет банкам получить конкретный ответ на вопрос, который возникает при проведении финансовых операций: какой максимальный убыток рискует понести банк за конкретный период времени и какова вероятность их наступления.

1.3 Методики оценки кредитоспособности заемщиков

При оформлении любого вида кредита каждый банк или иное финансовое учреждение всегда проводит анализ кредитоспособности заёмщика. Это важный аспект, без которого не обходится ни одна заявка на получение кредита. Банк объективно оценивает заемщика по различным параметрам и делает выводы относительно того можно ли ему выдать займ и какой лимит кредитования для него оптимален [11].

Единого эталона оценки кредитоспособности нет, каждый банк сам определяет важные для себя критерии и методы проведения анализа. Кто-то использует специальные программы с математическими формулами, кто-то обладает специальным отделом, где заявки рассматриваются реальными людьми.

В любом случае всё сводится к тому, что банк определяет конкретные критерии, которым должен соответствовать потенциальный заёмщик для получения одобрения.

Как уже говорилось, каждый кредитор разрабатывает свою систему анализа кредитоспособности заёмщика, устанавливая конкретные критерии.

Представим существующие методики оценки кредитоспособности юридических лиц.

Важнейшим этапом кредитования коммерческими банками юридических лиц является оценка кредитоспособности. Данные о кредитоспособности заемщика важны как для кредитора, так и для самого заемщика.

Для коммерческого банка данная информация позволяет снизить риск невозврата средств из-за неплатежеспособности заемщика, а для последнего – обеспечить финансовую устойчивость и стабильное финансовое состояние.

Основными целями оценки кредитоспособности является определение способности заемщика вернуть денежные средства, оценка риска кредитования конкретного юридического лица, а также определение оптимально возможного размера кредита, предоставляемого заемщику [12].

В настоящее время в РФ и за рубежом существует множество методов оценки кредитоспособности заемщиков банка. Поэтому возникает необходимость сравнительного анализа содержания и возможности использования этих методов на практике.

Методы оценки кредитоспособности во многих странах различаются, что связано, в первую очередь, с различиями законодательства, обычаями делового оборота, культурой и традициями.

Многообразие подходов к оценке кредитоспособности вызвано различной степенью доверия к качественным и количественным критериям оценки, а также сложившейся исторической практикой кредитования.

Методы и модели оценки, используемые за рубежом, можно разделить на классификационные модели и модели комплексного анализа:

– классификационные модели, позволяющие разбить заемщиков на определенные классы, включают в себя рейтинговые и прогнозные методы.

– рейтинговая оценка строится на основе полученных значений финансовых коэффициентов и выражается в баллах, которые рассчитываются путем умножения каждого показателя на его вес в рейтинге.

Простота определения кредитоспособности путем рейтинговой оценки определяет преимущества метода. В международной практике существуют рейтинговые оценки заемщика, проводимые специальными агентствами, например, Standard & Poor's, Modis и другими [21].

Следующим видом классификационных моделей являются прогнозные модели, получившие широкое распространение в США и Великобритании, основывающиеся на статистических методах, наиболее популярным из которых является множественный дискриминантный анализ (MDA), основанный на дискриминационной функции Z (модель Альтмана и модель Чессера).

«Z-анализ» позволяет установить различия между кредитоспособными и некредитоспособными группами клиентов банка. Методика расчета в соответствии с данным методом предполагает перемножение факторов, характеризующих финансовое состояние заемщика, и коэффициентов, которые определены в соответствии со статистическими данными о состоянии организаций. Полученный результат позволяет отнести оцениваемого заемщика либо к банкротам, либо к успешно действующим организациям [13].

Применение данного метода российскими банками проблематично. Это связано, во-первых, с отсутствием официальной статистики банкротств, во-вторых, с влиянием множества факторов на финансовое состояние организаций, что не позволит составить достоверный прогноз вероятного банкротства организации.

Помимо моделей MDA, к которым относятся модели Альтмана и Чессера, существуют более упрощенные прогнозные модели, которые основаны на системе показателей, например, система коэффициентов У. Бивера. Данная система включает коэффициенты Бивера и текущей ликвидности, экономической рентабельности и финансового левериджа [14].

Преимуществом модели, основанной на системе показателей, является отсутствие взаимосвязки коэффициентов и весовых значений, что дает возможность получить более точный результат. Недостатком такой модели является игнорирование специфики организаций и экономической ситуации в стране [15].

К прогнозным моделям также относится модель CART, что расшифровывается «классификационные и регрессионные деревья». Это непараметрическая модель, которая возможна для широкого использования, доступна для понимания, а также отличается несложностью расчетов [16].

Недостатками классификационных моделей являются преобладание количественных факторов и переоценка их роли, а также добровольный выбор количественных показателей, участвующих в оценке.

Следующей группой методов оценки кредитоспособности заемщиков, после классификационных моделей, являются модели комплексного анализа («Шести Си», CAMPARI, PARTS и оценочная система анализа), которые позволяют агрегировать количественные и качественные характеристики заемщика.

В США широко применяется правило «Шести Си», согласно которому критериями отбора клиентов являются такие экономические категории, как характер заемщика, способность заимствовать средства, денежные средства, обеспечение, условия, контроль. Каждому из критериев придается вес, разрабатывается таблица степени рискованности и определяется совокупная оценка кредитного риска по конкретному заемщику [17].

Данная методика включает критерии, которые не всегда возможно использовать в России. Например, в России не используется показатель репутации заемщика. Также в условиях нестабильной экономики сомнительным оказывается параметр экономической конъюнктуры и его числовое выражение.

В Англии термин «PARTS» сосредотачивает в себе требования при оценке кредитоспособности заемщиков, включающий в себя цель, размер, срок и обеспечение кредита, а также возврат долга и процентов [18].

Модель «PARTS» нецелесообразна для применения российскими банками, поскольку не учитывают прогнозирование финансового состояния заемщика, его ликвидность и платежеспособность. Оценка ведется только в базовых параметрах текущего времени. Английские банки также производят оценку возможного риска невозврата кредита с использованием методики CAMPARI, включающей такие критерии, как репутация и оценка бизнеса заемщика, анализ необходимости обращения за ссудой, цель кредита и ее обоснование, возможность погашения и способ страхования кредитного риска [19].

Данная методика включает наибольшее количество параметров из рассматриваемых, что позволяет наиболее точно определять

кредитоспособность заемщика. Однако часть параметров не может быть использована при оценке кредитоспособности российских организаций из-за отсутствия данных показателей. Например, в России отсутствуют оценки репутации заемщика, либо используются лишь крупными корпорациями, имеющими иностранный капитал в структуре уставного капитала.

Таким образом, в зарубежной практике банковского кредитования существует множество методик, основанных, преимущественно, на финансовом анализе. В то же время важную информацию о заемщике несут нефинансовые методы оценки.

Зарубежные банки особое внимание уделяют финансовой устойчивости и надежности компаний. Российским банкам приходится основываться на зарубежном опыте в связи с тем, что история оценки кредитоспособности в России не продолжительна. Однако в «чистом» виде применение перечисленных методик осложняется объективными межгосударственными различиями в деятельности заемщиков [36].

Кроме того, для осуществления проверки кредитоспособности клиента по зарубежным методикам необходим большой объем разнообразных данных, начиная от характеристики качества управления и заканчивая содержанием аудиторского заключения. Средние, а тем более мелкие российские предприятия не в состоянии предоставить такой объем информации.

Использование зарубежного опыта в оценке кредитоспособности заемщиков в России возможно только в дифференцированном виде, основанном на получении среднеотраслевых значений, рейтингах отраслей и предприятий.

Далее рассмотрим методы оценки кредитоспособности в РФ. Единой методики оценки кредитоспособности заемщика у российских банков не существует. Банк имеет право ориентироваться на широко используемый международный и отечественный опыт либо разработать собственный подход.

В оценке кредитоспособности заемщика российскими банками большое значение имеет финансовый анализ. Он проводится разными способами [20]:

- на основе системы финансовых показателей;
- на основе анализа денежных потоков;
- на основе анализа делового риска.

Система финансовых коэффициентов оценки кредитоспособности включает множество показателей, основными среди которых являются коэффициенты ликвидности, оборачиваемости, финансового левериджа, рентабельности, обеспечения долга, значения которых сравниваются с нормативными [21].

Данный метод ориентирует банк на оценку конечного финансового результата по данным отчетности предыдущих периодов, а не на анализ процесса осуществления деятельности и прогноз дальнейшего состояния организации. К недостаткам метода можно отнести игнорирование политических и общеэкономических изменений в стране, изменение организационной структуры управления организацией, смены форм собственности и т. д.

Альтернативным методом оценки кредитоспособности заемщиков коммерческих банков является анализ денежного потока, позволяющего сделать вывод о слабых местах управления организацией, в основе которого лежит использование фактических показателей, характеризующих оборот средств клиента в отчетном периоде.

Для анализа денежного потока берут данные как минимум за три истекших года. Применение данного метода повышает достоверность оценки кредитоспособности, т. к. позволяет определить способность организации погашать обязательства собственными средствами. Однако такой метод имеет высокую трудоемкость.

Еще одним методом оценки кредитоспособности заемщика является анализ делового риска, который позволяет прогнозировать достаточность источников погашения ссуды. Применение метода связано с проблемой доступа к внутренней информации заемщика при оценке внутренних рисков в деятельности организации. Каждый из приведенных выше способов оценки

кредитоспособности российскими банками взаимно дополняют друг друга. В отечественной практике анализа кредитоспособности мало внимания уделяется качественным характеристикам заемщика [29].

Методики, применяемые российскими банками в отношении юридических лиц, основаны на анализе финансовой отчетности. Возможности анализа качественных показателей ограничены из-за отсутствия единой нормативной базы по отраслям экономики. Нет и отраслевых справочников или классификаторов, позволяющих достоверно отнести ту или иную организацию-заемщика к определенному классу кредитоспособности с учетом ее отраслевых особенностей, а также дающих банкам возможность оценивать свой риск при предоставлении кредитных ресурсов.

Российские коммерческие банки вынуждены опираться в основном на собственную информационную базу. Таким образом, был проведен сравнительный анализ методического содержания подходов российских и зарубежных банков к оценке кредитоспособности заемщиков.

Основными отличиями принципов работы отечественных банков от иностранных являются менее жесткие требования к заемщикам и учет меньшего числа критериев при оценке кредитоспособности.

Методики оценки кредитоспособности российскими банками основаны на финансовом анализе, в то время, как зарубежные банки основное внимание уделяют качественным характеристикам [22].

Действующие методические подходы к оценке кредитоспособности в РФ нуждаются в развитии и совершенствовании на основе практики зарубежных банков. Методы оценки, кроме анализа финансово-хозяйственной деятельности заемщиков юридических лиц, оценки имущественного положения, анализа структуры капитала, ликвидности и платежеспособности, должны содержать также и кредитную историю заемщика, его рыночные позиции, качество управления, перспективы развития бизнеса и другое [23].

Необходимо осуществлять анализ вероятности банкротства заемщика, что позволит определить способность заемщика погасить задолженность. Для этого

представляется целесообразным разработать и внедрить единую нормативную базу для определения финансового состояния организаций и публикации рейтингов кредитоспособности организаций, что, в свою очередь, позволит упростить процесс определения кредитоспособности и снизить риски [24].

Кроме того, основываясь на зарубежном опыте, российским банкам необходимо включать в методику анализ качественных характеристик, что приведет к более точной оценке кредитоспособности заемщиков.

Подобные дополнения методик позволят приобщить российские банки к зарубежным стандартам оценки кредитоспособности заемщиков юридических лиц.

Были выделены показатели предприятия, на основе которого кредитная организация может произвести оценки кредитоспособности потенциального заемщика (таблица 1).

Таблица 1 – Показатели для оценки кредитоспособности

№	Показатель
1	Годовая выручка (в тыс.руб.)
2	Динамика выручки
3	Чистые активы (в тыс.руб.)
4	Отношение совокупного кредитного портфеля к среднемесячной выручке
5	Соотношение Долг/ЕБИТДА
6	Краткосрочная долговая нагрузка
7	Отношение краткосрочных кредитных обязательств к валюте баланса
8	Отношение ЕБИТ к сумме процентов к уплате
9	Соотношение собственных и заемных средств
10	Коэффициент автономии (обеспеченности собственным капиталом)
11	Коэффициент маневренности собственного капитала
12	Коэффициент текущей ликвидности
13	Валовая рентабельность (в процентах)
14	Чистая рентабельность (в процентах)
15	Рентабельность активов (в процентах)
16	Оборачиваемость оборотных активов (в днях)
17	Оборачиваемость краткосрочных обязательств (в днях)
18	Длительность финансового цикла (в днях)
19	Качество дебиторской задолженности (в процентах)
20	Качество кредиторской задолженности (в процентах)
21	Основные виды деятельности заемщика

Продолжение таблицы 1

22	Количество сотрудников
23	Срок работы данной организационной структуры (в годах)
24	Опыт генерального директора на своем посту (в годах)
25	Опыт работы топ-менеджеров в данной отрасли (в годах)
26	Срок работы с банками (в годах)
27	Количество банков-кредиторов
28	Максимальная продолжительность непрерывной просрочки за последний год (в днях)
29	Количество просрочек за последний год
30	Позиция на рынке
31	Прозрачность и репутация акционеров / участников, бенефициаров, Членов Совета
32	Формат информации, используемой при анализе
33	Информационная открытость и степень владения информацией
34	Внешний аудитор
35	Уровень конкуренции и барьеры для входа на рынок
36	Степень зависимости основной деятельности клиента от покупателей
37	Степень зависимости основной деятельности клиента от поставщиков

1.4 Характеристика предприятия

Кредитная организация ООО «Норд» предоставляет следующие виды услуг:

- полный спектр расчетно-кассового обслуживания юридических и физических лиц;

- прием вкладов от юридических и физических лиц;

- кредитование юридических и физических лиц;

- предоставление в аренду банковских ячеек;

Стратегические цели ООО «Норд»:

- Поддержка важнейших предприятий ведущих отраслей российской экономики, представленных в различных регионах страны

- Реализация масштабных инвестиционных проектов

- Внедрение самых высоких стандартов клиентского обслуживания

- Развитие ключевых отраслей отечественной экономики

Кредитную организацию отличают высокое качество и технологичность услуг. Коммерческие продукты, предлагаемые корпоративным и частным

клиентам – приоритетное направление бизнеса. Расчетно-кассовое обслуживание, основанное на современных информационных технологиях, соответствует самым высоким стандартам качества и способно удовлетворить запросы самого взыскательного клиента.

Кредитная организация занимает достойное место на рынке финансовых услуг благодаря:

- взвешенной стратегии развития;
- человеческим и профессиональным качествам своих сотрудников;
- постоянству в отношениях с Клиентами;
- своевременному использованию новых технологий бизнеса.

Организационная структура кредитной организации представлена на рисунке 4.



Рисунок 4 – Организационная структура кредитной организации

Организационная структура кредитной организации является линейно-функциональной.

Линейно-функциональная структура подходит средним и крупным компаниям [25].

Линейно-функциональный (комбинированный) тип организационной структуры служит для устранения недостатков отдельных линейных и функциональных типов структур управления.

Такие функциональные единицы лишены прямого воздействия на исполнителей: они готовят решение для линейного руководителя, который оказывает непосредственное административное влияние на исполнителей.

Предполагается, что первому (линейному) руководителю в разработке конкретных вопросов и подготовке соответствующих решений, программ, планов помогает специальный аппарат, состоящий из функциональных подразделений (отделов, групп, отделений).

Роль функциональных агентств (служб) зависит от масштабов экономической деятельности и структуры управления предприятием. Линейно-функциональная структура управления обладает преимуществами линейной организации, но чем выше уровень управления, тем больше роль функционального управления [43].

Функциональные службы осуществляют всю техническую подготовку к деятельности, готовят решение вопросов, связанных с управлением производственным процессом, освобождают линейных руководителей от планирования, финансовых расчетов и т.д.

Преимущества и недостатки линейно-функциональной оргструктуры представлены в таблице 2.

Таблица 2 – Преимущества и недостатки линейно-функциональной оргструктуры

Преимущества	Недостатки
1. Высокая компетентность специалистов, ответственных за выполнение конкретных функций. 2. Соответствие структуры выбранной стратегии предприятия. 3. Сочетание принципа специализации управления с принципом единоначалия.	1. Отсутствие гибкости при решении новых задач. 2. Препятствует координации деятельности функциональных подразделений по реализации новых программ. 3. Затрудняется внедрение промышленного дизайна и технологических инноваций без привлечения руководителей. 4. Чрезмерное развитие вертикальной составляющей системы управления.

2 Проектирование информационной системы принятия решения на основе оценки рейтинга заемщиков кредитной организации

2.1 Постановка задачи на проектирование

Требуется разработать программную систему управленческого учета операций, производимых кредитной организацией с целью выявления кредитоспособных потенциальных заемщиков.

Важной особенностью системы является ее распределенность, то есть возможность работы на нескольких территориально удаленных подразделениях. Данная характеристика системы должна быть реализована при помощи механизма распределенной СУБД, к которой клиентские приложения должны будут подключаться по сети предприятия.

Информационная система должна выполнять такие функции:

- ввод и редактирование различных справочников;
- вывод данных в формат Excel;
- просмотр данных (формирование форм и отчетов);
- выполнение операций анализа над данными.

Информационная система работает с входными данными, преобразуя их в выходные данные.

Входными данными являются:

- информация о заемщиках и их параметрах.

Выходными данными являются:

- кредитный рейтинг заемщиков;
- формы выгрузки в Excel.

К системе также предъявляется ряд требований:

- Надежность
- предусмотреть контроль вводимой информации.
- предусмотреть защиту от некорректных действий пользователя.
- обеспечить целостность информации в базе данных.

– обеспечить отказоустойчивость системы.

2) Минимальные требования для работы клиентского компьютера: IBM-совместимый персональный компьютер с процессором AMD Opteron или AMD Athlon 64 или Intel Xeon с поддержкой Intel EM64T или Intel Pentium IV с поддержкой EM64T с тактовой частотой 1,4 GHz, 1 ГБ оперативной памяти, жесткий диск 4Gb., печатающее устройство, VGA-совместимый дисплей (рекомендуется SVGA-дисплей).

3) Минимальные требования для работы сервера: процессоры AMD Opteron или AMD Athlon 64 или Intel Xeon с поддержкой Intel EM64T или Intel Pentium IV с поддержкой EM64T с тактовой частотой 1,4 GHz, 1 ГБ оперативной памяти, жесткий диск 4Gb.

4) Средой функционирования ИС должна являться Microsoft SQL Server 2008 R2.

5) Операционной системой должна быть Windows 7.

2.2 Анализ существующего программного обеспечения

Продукты, которые предлагают отечественные поставщики для оценки и управления банковских рисков, преимущественно востребованы и используются небольшими и средними банками для построения системы управления по отдельным видам рисков.

Сравнительный анализ основных продуктов, предлагаемых отечественными фирмами в области управления банковскими рисками, представлен в таблице 3.

Таблица 3 – Отечественные системы управления банковскими рисками

Общая информация	1	2	3	4	5	6
а) наименование компании-вендора	EGAR Technology	Прогноз	Intersoft Lab	РЭА-Риск-менеджмент	СофтВэл	ИНЭК [

Продолжение таблицы 3

б) линейка продуктов	EGAR Trading & Risk Systems	«ПРОГНОЗ. Управление рисками»	ВРМ-платформа «Контур»	«Контур Корпорация. Операционные риски»	NAVIGATOR	«Финансовый риск-менеджер»
Функциональность системы						
1. Рыночный риск	+	+	+		+	+
2. Кредитный риск	+	+	+			+
3. Операционный риск	+			+		+
4. Процентный риск	+	+	+		+	+
5. Риск потери ликвидности	+	+	+		+	+
6. Управление лимитами	+	+	+		+	Расчет лимитов

Отечественными системами управления банковскими рисками являются: EGAR Technology, Прогноз, Intersoft Lab, РЭА-Риск-менеджмент, СофтВел, ИНЭК.

В таблице 4 представлены банки, использующие программные средства по управлению банковскими рисками отечественных разработчиков.

Таблица 4 – Банки, использующие программы по управлению банковскими рисками отечественных разработчиков

Банки, использующие программы по управлению банковскими рисками отечественных разработчиков Компания	Банки, которые используют системы управления рисками данной компании
Компания «Intersoft Lab»	«Альянс-банк», Меткомбанк (ОАО), Камский коммерческий банк и другие.
Компания «СофтВэл»	МДМ банк, банк «Петрокоммерц», банк «Русский стандарт», Банк Москвы, и другие.

Продолжение таблицы 4

Компания «Прогноз»	ВТБ 24, Внешэкономбанк, Газпромбанк, ЮниКредит Банк и другие.
--------------------	---

Банкам России на данном этапе требуется надлежащее методическое обеспечение и организация внутренних процедур достаточности капитала [34].

Требования Банка России к организации процедур оценки достаточности капитала наиболее полно отражают подходы к построению системы рисков в коммерческом банке и интегрируют имеющуюся международную практику оценки достаточности капитала и формирования многоуровневой системы лимитов. Однако требования регулятора носят в основном концептуальный характер, и кредитная организация, опираясь на рекомендованные принципы, должна самостоятельно разрабатывать модели и методики оценки рисков, в том числе совокупного риска, методы установления стратегических лимитов. Для эффективного управления банковскими рисками в российских банковских учреждениях необходимо создать специализированное подразделение по управлению рисками.

Для совершенствования процедур управления и оценки банковских рисков, Банку России рекомендуется разработать и закрепить во внутренних нормативных документах:

- организационные процедуры, касающиеся утверждения лимитов и принятия стратегических решений;
- организационные процедуры, касающиеся распределения капитала по направлениям деятельности/подразделениям/видам рисков и оценки эффективности направлений деятельности/подразделений с учетом риска;
- организационные процедуры, направленные на введение в практику такого метода оценки банковских рисков как «стресс-тестирование».

2.3 Выбор прикладных инструментов проектирования

Приведем краткое описание СУБД и их основные параметры.

Теперь следует перейти к выбору СУБД. Так как ранее выбрана религионная модель, то рассмотрим такие бесплатные РСУБД:

- SQLite [26];
- MySQL [27];
- PostgreSQL [28].

СУБД SQLite.

SQLite – это встроенная файловая СУБД, которая не требует установки или настройки. Это, в свою очередь, означает, что приложение не запускается в рамках отдельного серверного процесса. Безсерверная архитектура обеспечивает совместимость базы данных [40].

Вся база данных SQL содержится в одном файле на диске, и все операции чтения и записи происходят непосредственно в этом файле на диске. Поскольку данные непосредственно записываются обратно в файл диска, SQLite придерживается свойств ACID для защиты транзакций от сбоев выделения памяти и ошибок ввода-вывода диска, которые могут возникнуть в результате непредвиденных сбоев системы или сбоев питания.

Преимущества и варианты использования.

Библиотека SQLite является одной из самых компактных библиотек, размер библиотеки может быть меньше 600 КБ. Из-за своего небольшого размера и природы СУБД, она очень хорошо подходит для IoT (интернет вещей) и встраиваемых устройств.

Одним из основных преимуществ SQLite является то, что она может выступать в качестве дополнительного решения для клиент-серверных корпоративных СУБД. Например, она может локально кэшировать данные из клиент-серверной СУБД и, таким образом, уменьшить задержку для запросов и поддерживать работоспособность конечного приложения в случае перерывов в работе корпоративных СУБД.

Недостатки SQLite.

Одним из основных недостатков системы SQLite является отсутствие многопользовательских возможностей, которые можно найти в полноценных

системах СУБД, таких как MySQL и PostgreSQL. Это приводит к отсутствию детального контроля доступа, удобной системы управления пользователями и возможностей безопасности, помимо шифрования самого файла базы данных. Это является основным недостатком при разработке многопользовательских приложений.

Другим большим недостатком SQLite является обработка операций записи, которые сериализуются. Это может стать основным узким местом для приложений, требующих параллелизма. Поскольку SQLite является файловой СУБД, она может вызывать проблемы с производительностью больших наборов данных из-за ограничений файловой системы. В этих случаях было бы целесообразно выбирать клиент-серверные базы данных, такие как MySQL и PostgreSQL, особенно при работе с большими наборами данных [39].

Таким образом, SQLite – это база данных с открытым исходным кодом, поддерживаемая группой разработчиков. Некоторые из ключевых клиентов SQLite – это Facebook, Google и Apple.

СУБД MySQL.

Обзор и особенности

MySQL – одна из самых популярных и широко распространенных СУБД с открытым исходным кодом. В отличие от SQLite, она использует архитектуру клиент-сервер, которая состоит из многопоточного сервера SQL. Эта многопоточная природа MySQL обеспечивает большую производительность, поскольку потоки ядра могут легко использовать несколько процессоров. База данных написана на C и C++ и поддерживает различные платформы, такие как операционные системы Windows Server и дистрибутивы Linux (RHEL 7 и Ubuntu). СУБД также придерживается системы ACID для обеспечения согласованности транзакций и предоставляет различные Connectors и API, такие как C, C++, Java, PHP и т. Д.

Масштабируемость, безопасность и репликация – это только некоторые ключевые функции, которые делают MySQL одним из самых популярных решений в корпоративных приложениях.

Функции безопасности включают систему привилегий доступа MySQL, которая обеспечивает аутентификацию пользователя, систему управления учетными записями пользователей и зашифрованные соединения с использованием протокола SSL.

MySQL предлагает репликацию от ведущего к подчиненному и от ведущего к ведущему, что может оказаться полезным при масштабировании операций чтения, полезных в качестве решения для резервного копирования или даже в сценариях отработки отказа в случае простоя. MySQL также имеет коммерческие версии, которые имеют более широкие возможности, чем бесплатная [38].

Например, версия MySQL Enterprise имеет дополнительные функции, такие как MySQL Transparent Data Encryption (TDE), MySQL Enterprise Backup и хранилище документов MySQL.

MySQL также предлагает встроенную многопоточную библиотеку, которая обеспечивает меньшую площадь для использования во встроенных и IoT-системах.

Преимущества и варианты использования.

Помимо наличия нескольких корпоративных функций, еще одним важным отличием MySQL от SQLite является поддержка многопользовательскими функциями MySQL. Это, наряду с корпоративными функциями и масштабируемостью, делает его идеальным кандидатом для распределенных приложений.

MySQL имеет преимущество перед PostgreSQL для простых операций чтения, когда речь идет о пропускной способности и производительности. СУБД также намного проще в установке и использовании и имеет более широкое сообщество по сравнению с PostgreSQL.

Простота использования и управляемость делает ее отличным инструментом для веб-сайтов, веб-приложений, приложений OLTP и является важной частью стека LAMP. MySQL 8 также представила функции NoSQL, что делает его наравне с PostgreSQL.

В то же время имеются и недостатки.

Поскольку MySQL перемещает старые данные в отдельную область, называемую сегментами отката, массовые INSERT могут отрицательно повлиять на производительность [43].

СУБД также не очень хорошо работает с долгосрочными SELECT запросами и лучше всего подходит для небольших запросов на выборку.

Также недостатками является отсутствие полнотекстового поиска и медленных одновременных операций чтения-записи.

MySQL принадлежит и поддерживается Oracle. Поддержка сообщества осуществляется через форумы, а лучшую поддержку можно получить, приобретая коммерческие продукты. Некоторые из основных клиентов MySQL – это Facebook, GitHub и YouTube.

СУБД PostgreSQL.

PostgreSQL – это СУБД с открытым исходным кодом, в которой особое внимание уделяется расширяемости и соответствию стандартам. Как и MySQL, PostgreSQL использует модель базы данных клиент- сервер, а процесс сервера, который обрабатывает связь с клиентом, управляет файлами и операциями базы данных, известен как процесс *postgres* .

PostgreSQL обрабатывает параллельные клиентские сессии, создавая («разветвляя») новый процесс для каждого соединения. Этот процесс отделен от основного процесса *postgres* и создается и уничтожается в течение времени жизни клиентского соединения. Написанный на C, Postgres совместим с ACID и поддерживает функции и хранимые процедуры. В отличие от MySQL, PostgreSQL поддерживает материализованные представления (кэшированные представления), что приводит к более быстрому частому доступу к большим и активным таблицам.

Как и MySQL, PostgreSQL также имеет несколько расширенных функций, таких как безопасность и репликация. PostgreSQL использует синхронную репликацию между главной и подчиненной базами данных. Помимо обеспечения контроля доступа пользователей, контроля доступа на основе

хоста и аутентификации пользователей, PostgreSQL также предоставляет возможность шифровать связь клиент-сервер с использованием SSL.

PostgreSQL использует технологию, известную как Multiversion Concurrency Control или MVCC, для обеспечения согласованности данных при одновременном доступе к данным. Эта технология лучше, чем простое использование блокировок, поскольку она минимизирует конфликт блокировок в многопользовательских средах, тем самым значительно улучшая производительность [44].

Для обратной совместимости или приложений, которым нужна классическая технология блокировки, PostgreSQL также позволяет использовать технологии блокировки таблиц и строк для обеспечения параллелизма.

Преимущества и варианты использования.

Благодаря превосходным возможностям параллельной обработки PostgreSQL выходит на первое место (по сравнению с MySQL) при выполнении длинных SELECT запросов. Расширяемость базы данных PostgreSQL также делает ее идеальным кандидатом для исследований и научных проектов.

Недостатки СУБД.

Хотя различия между PostgreSQL и MySQL не так велики, но стоит отметить, что механизм хранения данных в PostgreSQL все еще требует большой работы.

По сравнению с MySQL, PostgreSQL более энергоемкий. Поскольку PostgreSQL разветвляет процесс для новых клиентских соединений, эта операция может занять до 10 МБ на каждое соединение.

Еще один большой недостаток можно увидеть во время частых обновлений, когда PostgreSQL из-за отсутствия поддержки кластерных индексов может оказать огромное негативное влияние на производительность по сравнению с базами данных MySQL.

PostgreSQL является СУБД с открытым исходным кодом и поддерживается группой глобального развития PostgreSQL.

Выбор пал на MySQL, поскольку по уровню популярности она значительно опережает PostgreSQL и SQLite. К тому же она надежна, быстрая в работе.

В качестве СУБД была выбрана mysql, поскольку она самая распространенная полноценная серверная СУБД. Также MySQL очень функциональная, свободно распространяемая СУБД, которая к тому же успешно работает с различными сайтами и веб-приложениями, то есть в будущем возможна связь системы с интернет-сайтом. Обучиться использованию этой СУБД также довольно просто.

Рассмотрим технические характеристики MySQL:

- Размер таблиц ограничен максимальным размером файла используемой операционной системы.

- Максимальный размер единого табличного пространства может достигать 64 терабайт.

- Размер страницы по умолчанию составляет 64 килобайт.

Стоит заметить, что благодаря популярности этой СУБД, существует огромное количество различных плагинов и расширений, облегчающих работу с системой.

Теперь перейдем к выбору языка программирования.

Выполним анализ языков программирования [30, 32].

Язык C.

C – это язык программирования общего назначения, изначально созданный для систем Unix. Он широко используется в кроссплатформенных системах, системе Unix и разработке игр. Его часто выбирают, потому что он более компактен, чем C ++, и работает быстрее. Это второй по распространенности язык программирования после Java. C является предшественником многих других языков программирования, включая C #, Java, JavaScript, Perl, PHP и Python.

Язык C ++.

C ++ – это язык программирования среднего уровня, который является объектно-ориентированным. Он происходит от C, однако имеет дополнения и улучшения, чтобы сделать его более многогранным языком программирования. Он хорошо подходит для крупных проектов, поскольку его можно разбить на части, что облегчает совместную работу. Он используется некоторыми из самых известных в мире технологических компаний, включая Adobe, Google, Mozilla и Microsoft [41].

Язык Objective-C.

Как и большинство языков программирования, он является производным от C. Это высокоуровневый код общего назначения, в который добавлена функция передачи сообщений. Он известен тем, что предпочитал язык программирования для приложений Apple OS X и iOS, пока его не заменил Swift.

Язык Java.

В настоящее время Java является самым популярным и широко используемым языком в мире. Хотя он изначально был создан для интерактивного телевидения, он стал известен как язык выбора для устройств Android. Это также язык программирования программного обеспечения уровня предприятия. Это хороший многоцелевой язык программирования, потому что он может использоваться кросс-платформенным (то есть он так же легко используется в приложениях для смартфонов, как и в настольных приложениях). Он похож на C ++ по синтаксису и структуре, что облегчает его использование.

Язык JavaScript.

JavaScript был создан как дополнительный язык для расширения функциональности веб-страниц. Он добавляет динамические функции, такие как формы отправки, интерактивность, анимация, отслеживание пользователей и т.д. Он в основном используется для внешней разработки или для решений по программированию, с которыми взаимодействуют клиенты. Он совместим со

всеми браузерами, что делает его хорошим универсальным кодом для веб-разработки, хотя его также трудно отладить [42].

Язык Swift.

Давно провозглашенный заменой Objective-C, когда речь заходит о программах Apple, в последнее время он набирает популярность, потому что он легко читается, прост в обслуживании и быстрее, чем Objective-C. Хотя Objective-C все еще используется, Swift быстро становится предпочтительным языком программирования для программистов, создающих программы для устройств Apple.

Язык C #.

Универсальный объектно-ориентированный язык C # (произносится си-шарп) был создан Microsoft в 2001 году. Хотя он назван в честь семейства языков программирования C, он имеет больше общего с Java, чем с другими языками C. C# в основном используется для внутренних / корпоративных решений и реже встречается в коммерческом программном обеспечении. В то же время он имеет множество преимуществ, среди которых можно отдельно выделить автоматическую сборку мусора. К тому же среды разработки C# имеют систему подсказок написания кода, а сам язык имеет множество библиотек, которые можно использовать, ускорив, тем самым, разработку готового продукта.

Язык PHP.

PHP – это открытый язык, в основном используемый для веб-разработки (или создания веб-страниц). PHP был создан для упрощения создания веб-страниц. Это довольно простой язык программирования, который можно быстро подобрать. Он используется многими веб-компаниями, включая Facebook, Wikipedia и WordPress.

Язык Ruby.

Ruby – это объектно-ориентированный язык программирования общего назначения, разработанный в середине 90-х годов в Японии. Это один из самых простых в изучении языков программирования, который часто используется в

качестве основы для фреймворка Ruby on Rails. Он известен тем, что его легко читать, и как результат многие программисты рекомендуют изучать Ruby в качестве первого языка программирования [40].

Язык Python.

Еще один универсальный высокоуровневый код Python является фаворитом начинающих программистов. Он используется в основном для веб-приложений и информационной безопасности, хотя он также популярен в академическом сообществе для анализа данных. Он используется техническими гигантами, такими как Google, Dropbox, Pinterest и Spotify.

В качестве языка программирования выбран C#, которому соответствует среда разработки Visual Studio. Этот выбор обусловлен тем, что C# фактически интегрировал в себя все плюсы языков Java и C++, а также имеет такие дополнительные особенности, как автоматический сборщик мусора, продвинутую систему поддержки разработки кода, встроенную в IDE, множество удобных и полезных типов данных, значительно ускоряющих процесс разработки. К тому же IDE Visual C# Express является бесплатной средой разработки приложений на C#.

2.4 Проектирование базы данных

Проектирование базы данных может быть разбито на такие этапы:

- инфологическое описание предметной области;
- даталогическая модель;
- физическая модель;
- генерация скрипта создания базы данных;
- импорт скрипта в СУБД и создание схемы данных.

Описание предметной области можно представить в виде трех уровней абстракции (три уровня описания элементов данных). Это внешний, концептуальный и внутренний (инфологический, даталогический и физический

уровни) уровни, которые формируют так называемую трехуровневую архитектуру (рисунок 5).

Инфологическая модель отображает реальный мир в некую совокупность концепций, полностью независимую от параметров среды хранения данных [34].

Даталогическое моделирование – это описание предметной области в рамках конкретной модели данных (в нашем случае реляционной).

Физическое проектирование подразумевает соответствие данных форматам и типам конкретной СУБД (в данном случае MySql).

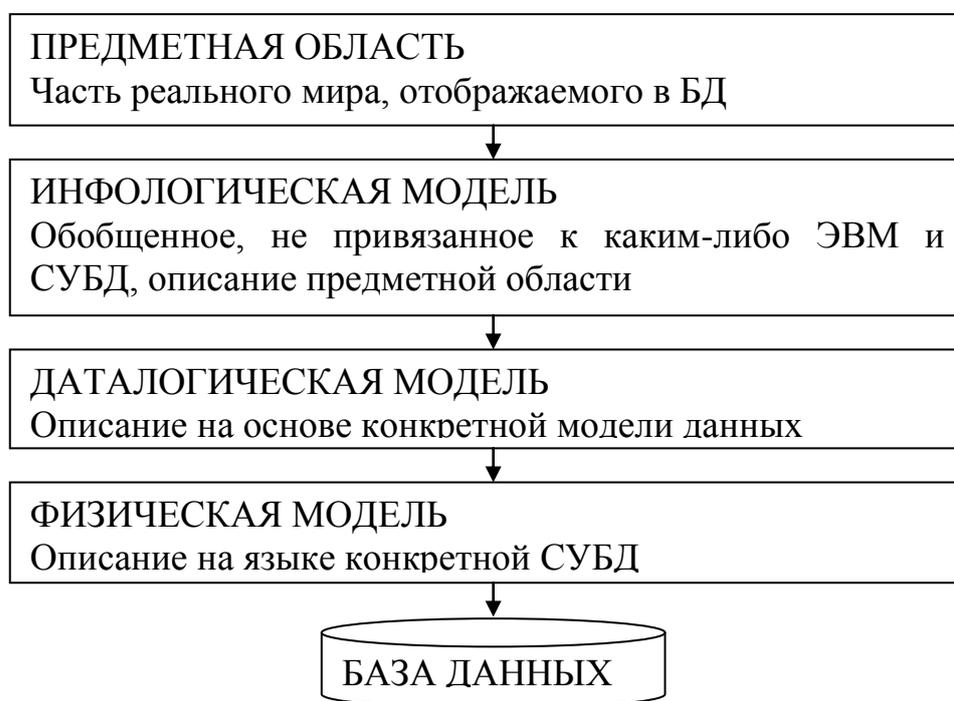


Рисунок 5 – Уровни моделей данных

В рамках инфологического проектирования выделены такие сущности:

- Заемщик
- Параметр
- Пользователь системы

Между этими сущностями присутствует связь М:М.

Атрибутами сущности «Заемщик» являются:

- ФИО контактного лица
- Название предприятия

- Форма собственности (тип)
- Страна
- Город
- Индекс
- Адрес телефон
- Описание
- Информация
- Кредитный рейтинг

Атрибутами сущности «Параметр» являются:

- Тип параметра (количественный или качественный)
- Описание
- Важность параметра
- Значение параметра
- Направление ранжирования

Поскольку в качестве модели данных будет использована реляционная, то необходимо разбить связь М:М на две 1:М. Также при переходе к даталогической модели следует учесть принципы нормализации до 3НФ [33].

При переходе к даталогической модели будем руководствоваться такими соображениями. Параметры заемщика надо разделить на качественные и количественные, поскольку они по-разному ранжируются и описываются.

Для классификации заемщиков следует ввести отдельную таблицу «Тип заемщика», которая будет описывать различные формы собственности.

Следует ввести дополнительную таблицу «Значения качественных параметров», где будет даваться текстовые значения и важность каждого конкретного значения параметра с точки зрения кредитоспособности [36].

В результате получим даталогическую модель (рисунок 6).

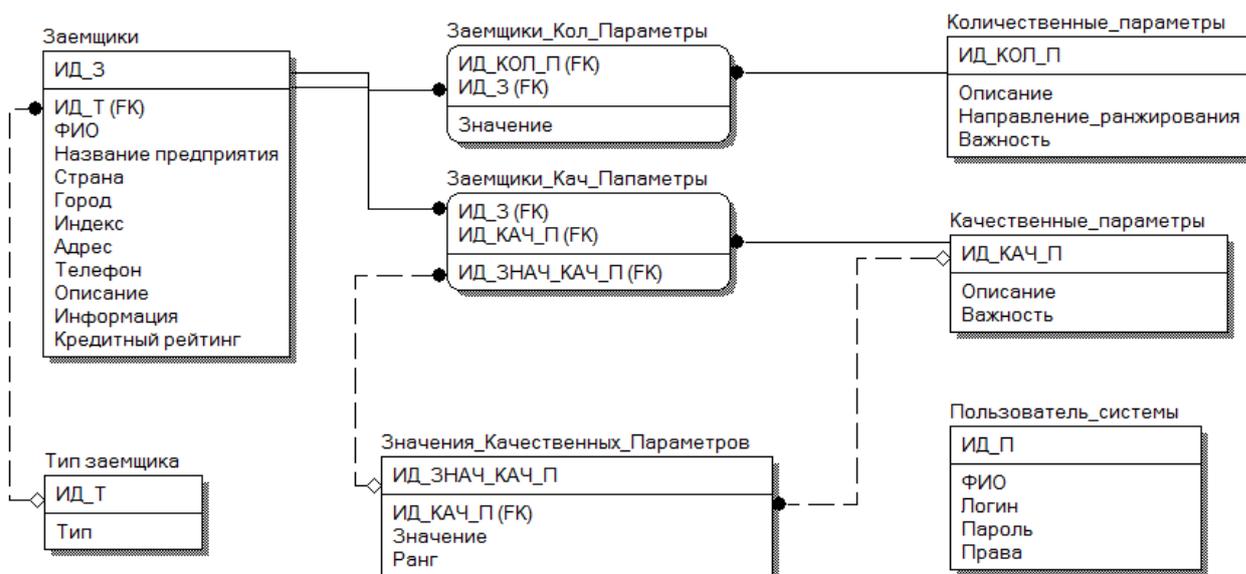


Рисунок 6 – Дatalogическая модель

Скрипт инициализации базы данных представлен в приложении А.

2.5 Проектирование интерфейса пользователя

В широком смысле интерфейс представляет собой определенную стандартами границу между независимыми объектами, обменивающимися между собой некоторой информацией. Интерфейс задает параметры объектов, а также их процедуры и характеристики взаимодействия [35].

Существуют различные средства визуального проектирования пользовательских интерфейсов: Visual Studio, Microsoft Visio, Axure RP Pro, Smart Draw, Gliffy.com, gomockingbird.com и другие. Поскольку в качестве среды разработки выбрана Visual Studio, то целесообразно для разработки интерфейса воспользоваться встроенным в неё средством визуального проектирования пользовательских интерфейсов.

Макеты форм приложения представлены на рисунках 7.

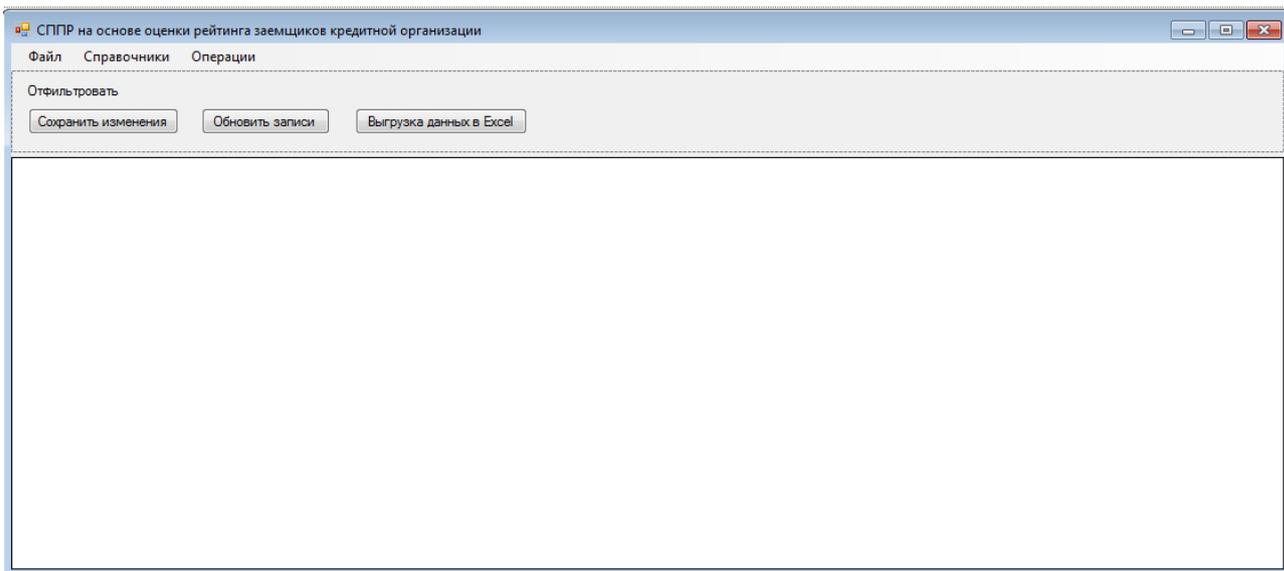


Рисунок 7 – Макет пользовательского интерфейса главной формы

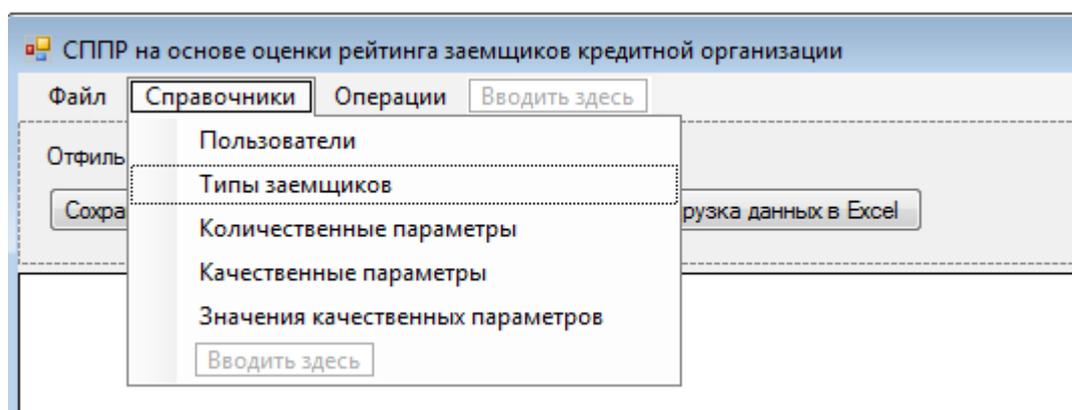


Рисунок 8 – Справочники системы

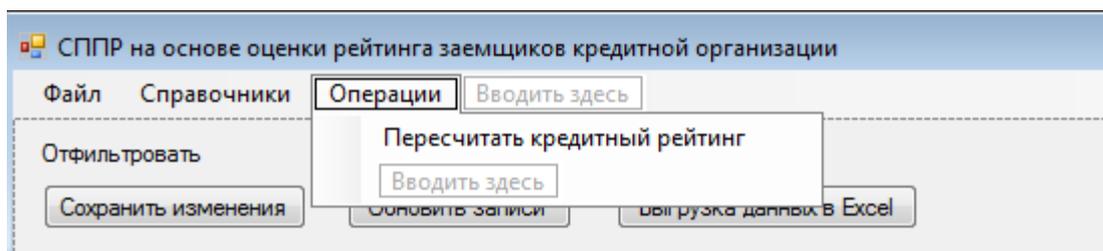


Рисунок 9 – Доступные операции

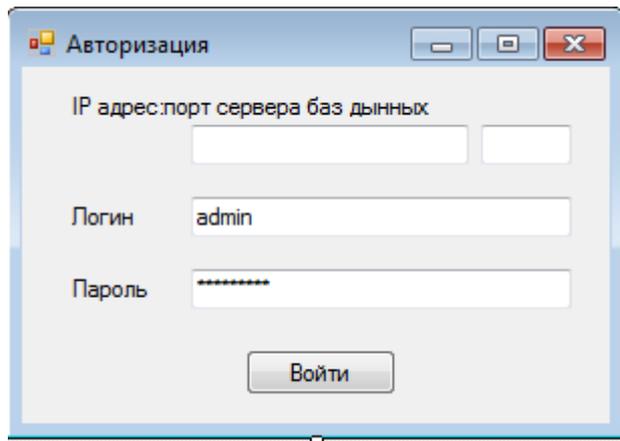


Рисунок 10 – Форма авторизации

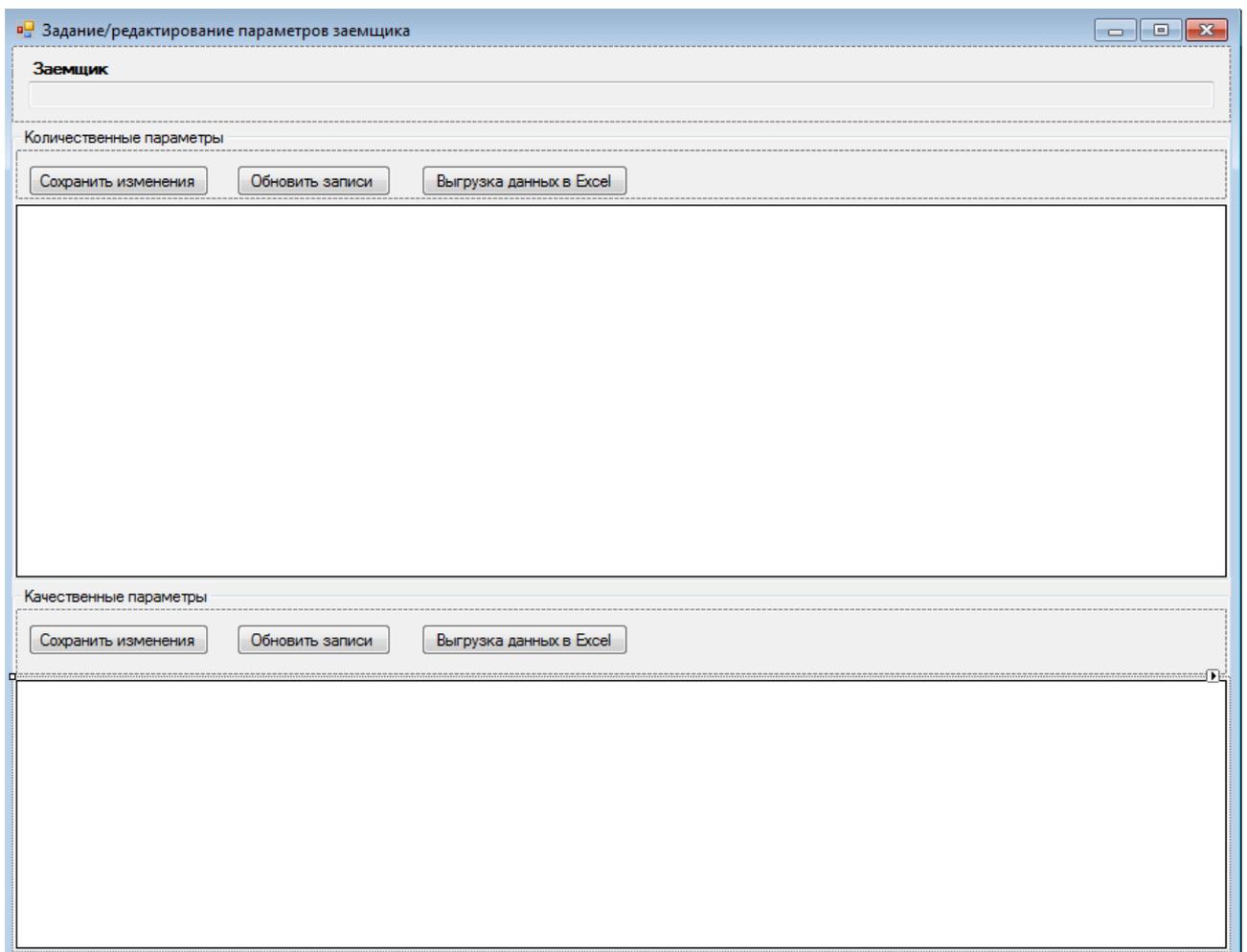


Рисунок 11 – Форма задания/редактирования параметров заемщика

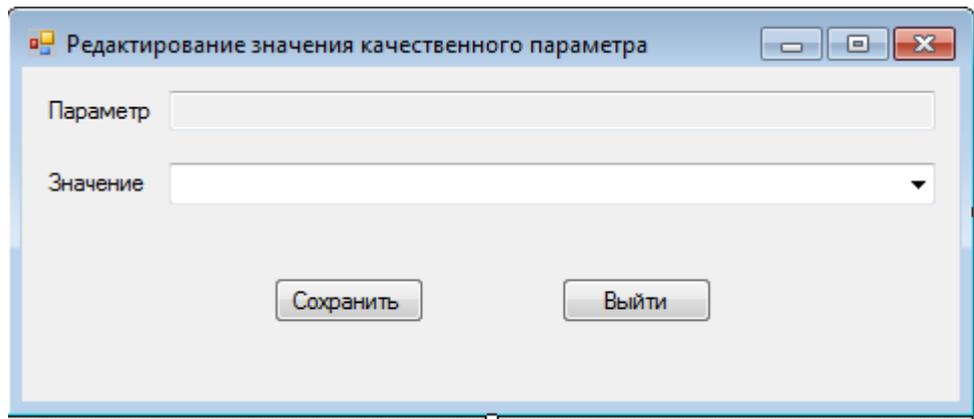


Рисунок 12 – Форма редактирования значения качественного параметра

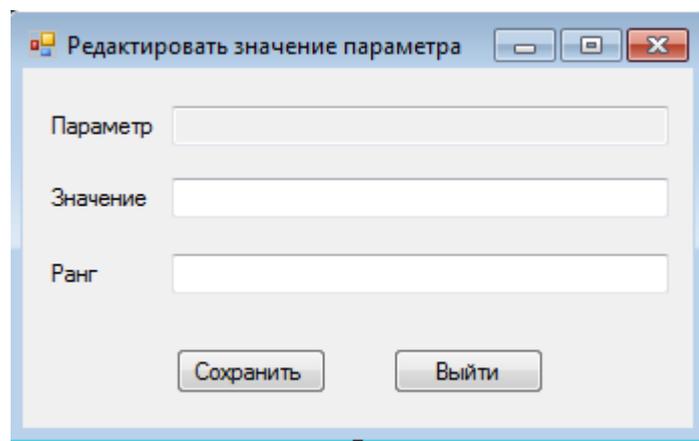


Рисунок 13 – Форма редактирования значения параметра

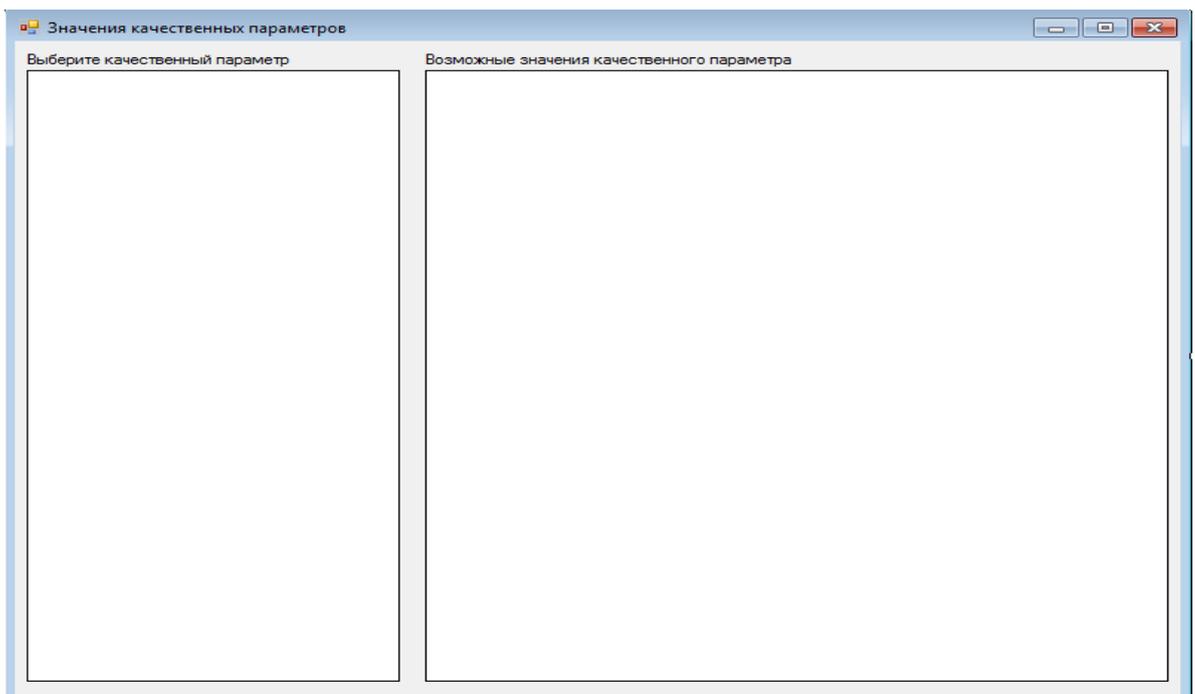


Рисунок 14 – Форма редактирования качественных параметров

3 Реализация, тестирование и оценка экономической эффективности

3.1 Разработка программного кода приложения

Представим таблицу описания функций модулей (таблица 5).

Таблица 5 – Описание функций модулей

№ п/п	Наименование модуля	Функции модуля
2	Модуль инициализации	Содержит функции по первоначальной настройке ИС. Инициализацию вспомогательных подмодулей.
3	Модуль системной логики	Содержит в себе иерархию подчиняющихся форм приложения, что обеспечивает логичную и интуитивно понятную программную навигацию
4	Модуль подключения к БД	Отвечает за обеспечение связи приложения с БД MySQL.
5	Модуль взаимодействия с БД	Отвечает за выполнение запросов к БД и предоставление для других модулей ИС унифицированных процедур для выполнения основных операций над данными: чтение, запись, изменение, удаление.
6	Модуль пользовательского интерфейса	Содержит функции, включающие: Формирование пользовательского интерфейса согласно хранимым шаблонам

Вообще, данное разделение больше логическое, чем физическое, поскольку в реальности присутствуют несколько десятков файлов, каждый из которых отвечает за определенный набор специфических функций. Однако все эти файлы могут быть разделены на логические группы, представленные в таблице выше.

В качестве языка программирования выбран C# – соответственно, средой программирования является MS Visual Studio.

Теперь перейдем к физическому представлению модулей. ИС состоит из 13 модулей (рисунок 15). Спецификация всех модулей ИС приведена в таблице 6.

Таблица 6 – Спецификация модулей

Спецификация модулей Название модуля	Назначение модуля	Вызывающий модуль
1	2	3
Program.cs	Инициализация приложения	При запуске программы
auth.cs	Главная страница	Program.cs
main.cs	Авторизация пользователей	auth.cs
FormTpl.cs	Редактирование справочников	main.cs
Excel.cs	Вывод данных в excel	analysis.cs
EditCreditParams.cs	Редактирование значений количественных и качественных параметров заемщика	main.cs
ZnKachParam.cs	Редактирование справочника «Качественные параметры»	EditCreditParams .cs
EditZnKachParam.cs	Редактирование значения качественного параметра для определенного заемщика	EditCreditParams .cs
NewOrEditKachParam.cs	Задание значения и ранга качественному параметру	EditCreditParams .cs

Модуль Program.cs – является точкой входа приложения и служит для первоначальной инициализации приложения в целом.

Модуль auth.cs – модуль авторизации, который позволяет ограничить попытки входа в приложение до 3-х при неправильном вводе пароля, а также производит аутентификацию, а первоначальную установку прав доступа для вновь идентифицированного пользователя.

Модуль main.cs – главное окно приложения, отображает список клиентов.

Модуль FormTpl – создает и отображает справочники с возможностью их редактирования, если текущий пользователь ИС обладает правами доступа.

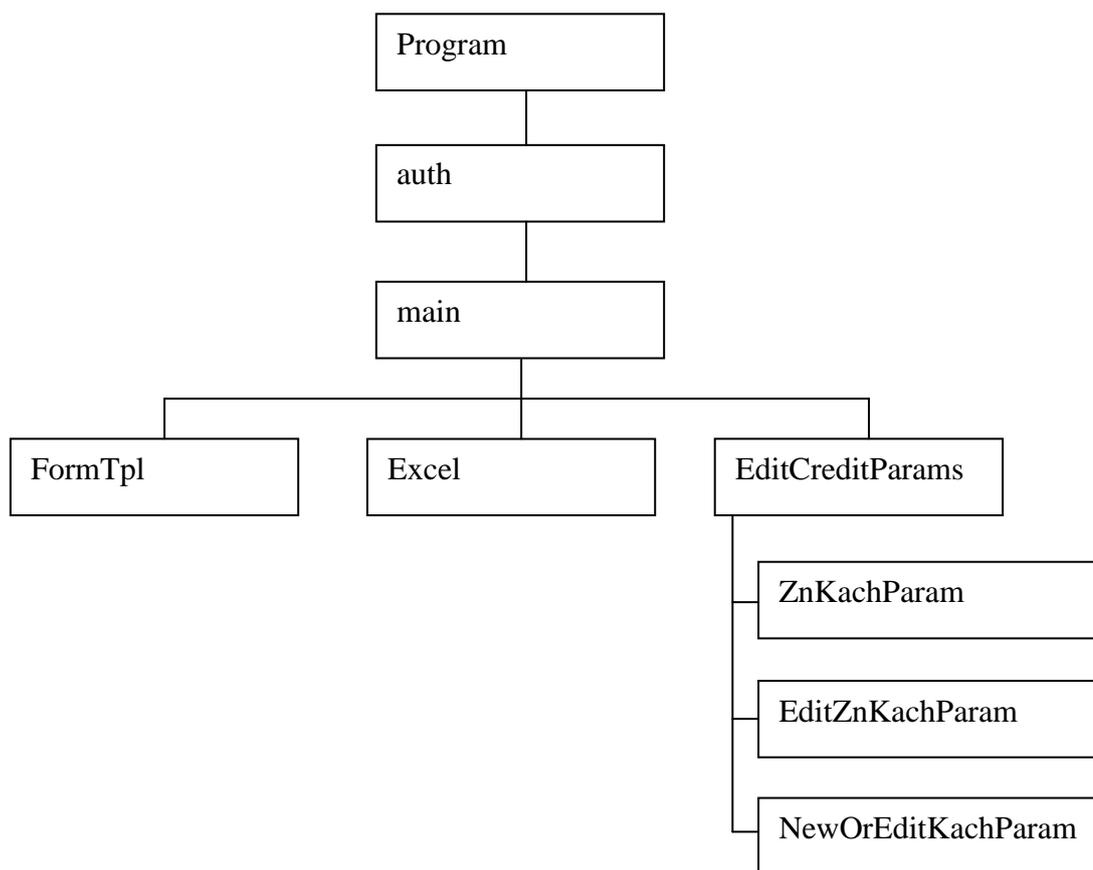


Рисунок 15 – Иерархическая схема модулей ИС

Модуль Excel.cs – служит для распечатки результатов регрессионного анализа в файл Excel.

Модуль EditCreditParams – позволяет редактировать/задавать качественные и количественные параметры для заемщика.

Модуль ZnKachParam – модуль справочника, позволяющего редактировать качественные параметры и их возможные значения.

Модуль EditZnKachParam – редактирование конкретного значения качественного параметра из справочника.

Модуль NewOrEditKachParam – задает непосредственное значение качественного параметра из списка.

Для более полной информации о структурной схеме пакета приведем структуру каталогов (рисунок 16).

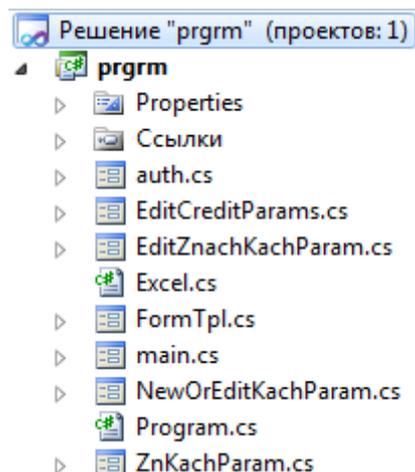


Рисунок 16 – Структурная схема пакета

Описание программных модулей

Отметим вначале, что существуют стандартные операции над данными, так называемый CRUD [36]:

- создание;
- чтение;
- изменение;
- удаление.

Данные для ИС хранятся в базе данных. Интерфейс взаимодействия с БД предоставляет СУБД MySQL.

Задача заключается в том, чтобы сформировать некие стандартизированные механизмы управления потоками данных в приложении, чтобы, во-первых, не дублировать код, во-вторых, обеспечить логичную цепочку передачи сообщений по каналам управления.

В рамках разработки ПО одним из основных моментов является взаимодействие с базой данных. Были разработаны универсальные функции выполнения запросов к БД:

- SELECT
- INSERT
- Другие запросы

Вначале следует установить соединение с БД:

```

public MySqlConnection getConnection() // Получить соединение с БД
{
    try
    {
        if (cn == null || cn.State != ConnectionState.Open)
        {
            cn = new MySqlConnection();
            cn.ConnectionString = connString;
            cn.Open();
            connected = true;
            return cn;
        }
        else
        return cn;
    }
    catch(MySqlException ex)
    {
        if(DEBUG)
            MessageBox.Show(«Ошибка подключения к БД:\r\n»
+ ex.Message);
        return conn;
    }
}

```

Ниже представлены функции выполнения MYSQL запросов:

```

public bool MySqlSelect(String MySql, ref List<List<String>>
MySqlresult) // Выполнить запрос к БД
{
    try
    {
        conn = getConnection();
        MySqlCommand myCommand = new
MySqlCommand(MySql, conn);
        MySqlDataReader rd = myCommand.ExecuteReader();
        MySqlresult.Clear();
        while(rd.Read())
        {
            List<String> row = new List<String>();
            for (int i = 0; i < rd.FieldCount; i++)
            {
                row.Add(rd.GetValue(i).ToString());
            }
            MySqlresult.Add(row);
        }
        closeConnection(conn);
    }
    catch (Exception ex)
    {
        if (DEBUG)

```

```

        MessageBox.Show («Ошибка выполнения MYSQL
запроса:\r\n»+MySQL+»\r\n»+ex.Message);
        return false;
    }
    return true;
}

```

Для выполнения запросов используется компонент `MySqlDataReader`.

Результат записывается в коллекцию `List<List<String>>` `MySQLresult`, которая представляет собой список списков строк и позволяет хранить возвращенную в результате запроса таблицу.

Важным является функция подсчета статистики по рейтингу потенциальных заемщиков. Алгоритм работы такой функции можно описать:

1. Получение списка заемщиков.

2. Ранжирование заемщиков по каждому из количественных показателей и умножение позиции в отсортированном списке заемщиков на важность показателя. Получится рейтинг каждого заемщика по каждому показателю. Вычисление общего рейтинга каждого заемщика по количественным показателям.

3. Ранжирование заемщиков по каждому из качественных показателей и умножение позиции в отсортированном списке заемщиков на важность показателя. Получится рейтинг каждого заемщика по каждому показателю. Вычисление общего рейтинга каждого заемщика по качественным показателям.

4. Отображение рейтингов в таблице.

Ниже представлен код данной функции.

```

// (1) Получение списка заемщиков
        List<List<string>> sqlresult = new
List<List<string>>();
        List<List<string>> sqlresult1 = new
List<List<string>>();
        string sql, sql1;
        bool res, res1;
        Dictionary<string, double> stat = new
Dictionary<string, double>(); // Рейтинг заемщиков
        int zaemCnt;

        sql = "select id_za from zaemshiki";

```

```

res = sqlSelect(sql, ref sqlresult);
if (res && sqlresult.Count > 0)
{
    for (int i = 0; i < sqlresult.Count; i++)
        stat.Add(sqlresult[i][0], -1.0);
    zaemCnt = sqlresult.Count;
}
else
{
    MessageBox.Show("Ошибка на этапе (1.1)");
    return;
}

// (2) Ранжирование заемщиков по каждому из
количественных показателей
// и умножение позиции в отсортированном списке
заемщиков на важность показателя
// Получится рейтинг каждого заемщика по каждому
показателю
// Вычисление общего рейтинга каждого заемщика по
количественным показателям

sql = "select id_ko, napravl, vagnost from kol_param";
res = sqlSelect(sql, ref sqlresult);
if (res)
{
    string id_ko, napravl, vagnost;
    for (int i = 0; i < sqlresult.Count; i++)
    {
        id_ko = sqlresult[i][0];
        napravl = sqlresult[i][1];
        vagnost = sqlresult[i][2];
        // Получаем список заказчиков по показателю
        sql1 = "select distinct(id_za), znach from
zaem_kol where id_ko=" + id_ko + " order by znach " +
(nappravl=="0" ? "asc" : "desc");
        res1 = sqlSelect(sql1, ref sqlresult1);
        if (res1 && sqlresult1.Count > 0)
        {
            int place = 1;
            double lastZnach =
double.Parse(sqlresult1[0][1]);
            double vag = double.Parse(vagnost);
            for (int j = 0; j < sqlresult1.Count; j++)
            {
                double currZnach =
double.Parse(sqlresult1[j][1]);
                if (currZnach != lastZnach)
                    place++;
                // Добавляем часть рейтинга по новому
параметру к заемщику
                stat[sqlresult1[j][0]] += place * vag;
            }
        }
    }
}

```

```

        }
        else
        {
            MessageBox.Show("Ошибка на этапе (2.2)");
            return;
        }
    }
else
{
    MessageBox.Show("Ошибка на этапе (2.1)");
    return;
}

// (3) Ранжирование заемщиков по каждому из
качественных показателей
// и умножение позиции в отсортированном списке
заемщиков на важность показателя
// Получится рейтинг каждого заемщика по каждому
показателю
// Вычисление общего рейтинга каждого заемщика по
качественным показателям

sql = "select id_ka, vagnost from kach_param";
res = sqlSelect(sql, ref sqlresult);
if (res)
{
    string id_ka, vagnost;
    for (int i = 0; i < sqlresult.Count; i++)
    {
        id_ka = sqlresult[i][0];
        vagnost = sqlresult[i][1];
        // Получаем список заказчиков по показателю
        sql1 = "select distinct(a.id_za), b.rang from
zaem_kach as a inner join zn_kach_param as b on a.id_zn=b.id_zn
where a.id_ka=" + id_ka + " order by b.rang desc";
        res1 = sqlSelect(sql1, ref sqlresult1);
        if (res1 && sqlresult1.Count > 0)
        {
            int place = 1;
            double lastZnach =
double.Parse(sqlresult1[0][1]);
            double vag = double.Parse(vagnost);
            for (int j = 0; j < sqlresult1.Count; j++)
            {
                double currZnach =
double.Parse(sqlresult1[j][1]);
                if (currZnach != lastZnach)
                    place++;
                // Добавляем часть рейтинга по новому
параметру к заемщику
                stat[sqlresult1[j][0]] += place * vag;
            }
        }
    }
}

```

```

    }
    else
    {
        MessageBox.Show("Ошибка на этапе (3.2)");
        return;
    }
}
else
{
    MessageBox.Show("Ошибка на этапе (3.1)");
    return;
}

// Выставление рейтинга
for (int i = 0; i < mainGrd.RowCount; i++)
{
    mainGrd.Rows[i].Cells[12].Value =
stat[mainGrd.Rows[i].Cells[0].Value.ToString()];
    //mainGrd.NotifyCurrentCellDirty(true);
}
btn1.Enabled = true;

```

3.2 Тестирование приложения

При запуске приложение появляется окно авторизации (рисунок 17).

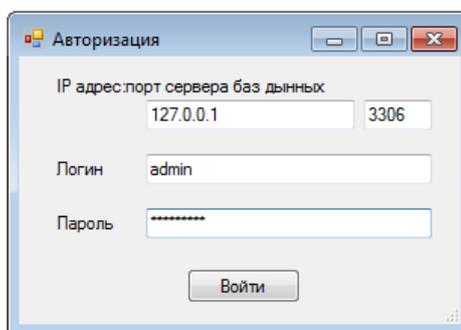


Рисунок 17 – Авторизация

При успешной авторизации загружается главное окно приложения, которое содержит информацию о потенциальных заемщиках (рисунок 18). Поле рейтинга пока пустое.

СППР на основе оценки рейтинга заемщиков кредитной организации: Вы вошли как "Администратор"

Файл Справочники Операции

Отфильтровать

Сохранить изменения Обновить записи Выгрузка данных в Excel

	Тип	ФИО	Организация	Страна	Город	Индекс	Адрес	Телефон	Описание	Информация	Рейтинг
1	ООО	Усенко Владимир Никола...	12-ПРЕСС	Российская Федер...	Красноярск	7164399	ПОДПЕСНАЯ 21	7(391)001-40-28			
2	ООО	Алферова Вера Ивановна	A. AHLSTROM OY	Российская Федер...	Красноярск	9932430	КИРОВА 71-67	7(391)627-92-14			
3	ООО	Полянская Полина Андре...	ACCENT/АКЛЕНТ	Российская Федер...	Красноярск	8570157	ВОРОШИЛОВА 3-21	7(391)464-59-42			
4	ООО	Фадеева Светлана Борис...	ALLIGNO MASCHIN...	Российская Федер...	Красноярск	8776496	НАБЕРЕЖНАЯ СЛЕНИНО УЛИЦА ...	7(391)681-55-08			
5	ООО	Складова Светлана Кири...	BIESSE S.P.A. ПРЕ...	Российская Федер...	Красноярск	7485539	ПУШАНИНА 20-24	7(391)091-12-93			
6	ООО	Балашова Алла Ивановна	Besse	Российская Федер...	Красноярск	7478721	ПУШАНИНА 20-14	7(391)109-42-41			
7	ООО	Подойницын Олег Степан...	BMH KLOECKNER S...	Российская Федер...	Красноярск	7450918	СТРОИТЕЛЕЙ 72-46	7(391)052-09-46			
8	ООО	Ильина Анастасия Алекс...	С.С.М.Л. (ИТАЛИЯ) ...	Российская Федер...	Красноярск	9900791	МИРА 9-59	7(391)370-98-86			
9	ООО	Муреева Калерия Никола...	CANON (ЯПОНИЯ)	Российская Федер...	Красноярск	8910375	ПОБАЧЕВСКОГО 1-Й 9-79	7(391)870-07-30			
10	ООО	Качанов Венезианен Васил...	Сapon	Российская Федер...	Красноярск	7953035	ПОБЕДЫ 89	7(391)494-73-54			
11	ООО	Кузнецов Юрий Иванович	COSTA HANDLING ...	Российская Федер...	Красноярск	8887376	ЧААДАЕВА 107-34	7(391)065-01-15			
12	ООО	Федорова Галина Абрам...	Dieffenbacher & co.	Российская Федер...	Красноярск	7641278	ТОПСТОГО 34-16	7(391)955-74-16			
13	ООО	Махаева Людмила Дмитр...	DIEFFENBACHER G...	Российская Федер...	Красноярск	9584536	КАРПИНСКОГО 36-2	7(391)973-70-46			
14	ООО	Князев Геннадий Иванович	DIK & FELA GRAPHI...	Российская Федер...	Красноярск	8814363	ПОБЕДЫ 95-24	7(391)312-77-61			
15	ООО	Гольяткина Мария Никол...	DIMAR W.P.W.	Российская Федер...	Красноярск	6704611	ЛАДОЖСКАЯ 29-55	7(391)378-48-40			

Рисунок 18 – Главное окно приложения

В системе доступны справочники (рисунок 19).

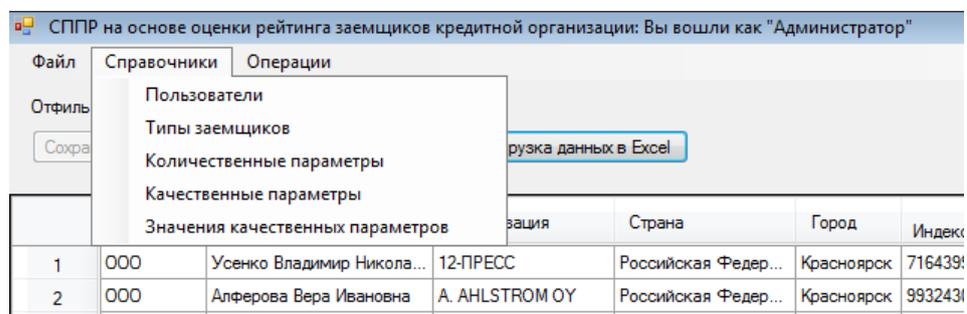


Рисунок 19 – Справочники системы

Пример редактирования справочника «Количественные параметры» представлен на рисунке 20.

Количественные параметры

Сохранить Обновить записи Выгрузка в Excel

Для записи изменений в БД нажмите "Сохранить в БД".
Для удаления записи кликните мышкой по черной стрелке слева от нужной строки и затем нажмите кнопку Delete на клавиатуре.

	Описание	Направление
1	Годовая выручка (в тыс.руб.)	1
2	Чистые активы (в тыс.руб.)	1
3	Отношение совокупного кредитного портфеля к среднемесячной выручке	0
4	Соотношение Долг/ЕБИТДА	0
5	Краткосрочная долговая нагрузка	0
6	Отношение краткосрочных кредитных обязательств к валюте баланса	0
7	Отношение ЕБИТ к сумме процентов к уплате	0
8	Соотношение собственных и заемных средств	1
9	Коэффициент автономии (обеспеченности собственным капиталом)	1
10	Коэффициент маневренности собственного капитала	1
11	Коэффициент текущей ликвидности	1
12	Валовая рентабельность (в процентах)	1
13	Чистая рентабельность (в процентах)	1
14	Рентабельность активов (в процентах)	1
15	Оборачиваемость оборотных активов (в днях)	0
16	Оборачиваемость краткосрочных обязательств (в днях)	0

Рисунок 20 – Редактирование справочника «Количественные параметры»

Пример редактирования справочника «Качественные параметры» представлен на рисунке 21.

Качественные параметры

Сохранить Обновить записи Выгрузка в Excel

Для записи изменений в БД нажмите "Сохранить в БД".
Для удаления записи кликните мышкой по черной стрелке слева от нужной строки и затем нажмите кнопку Delete на клавиатуре.

	Описание	Важность (от 1 до 100)
▶ 1	Динамика выручки	70
2	Основные виды деятельности заемщика	50
3	Позиция на рынке	60
4	Доступность информации о заемщике	70
5	Репутация акционеров / участников, бенефициаров, Членов Совета	70
6	Уровень конкуренции	85
7	Барьеры для входа на рынок	70
8	Степень зависимости основной деятельности клиента от покупателей	70
9	Степень зависимости основной деятельности клиента от поставщиков	70
10	Внешний рейтинг	90

Рисунок 21 – Редактирование справочника «Качественные параметры»

Пример редактирования справочника «Значения качественных параметров» представлен на рисунке 22.

Значения качественных параметров

Выберите качественный параметр

Описание

Динамика выручки

Основные виды деятельности заемщика

Позиция на рынке

Доступность информации о заемщике

Репутация акционеров / участников, бенефициаров, Членов Совета

Уровень конкуренции

Барьеры для входа на рынок

Степень зависимости основной деятельности клиента от покупателей

Степень зависимости основной деятельности клиента от поставщиков

Внешний рейтинг

Возможные значения качественного параметра

Значение	Ранг
AAA	100
AA	90
A	80
BBB	70
BB	60
B	50
ССС	40
СС	
С	
Дефолт	
Рейтинг не установлен	0

Добавить

Изменить

Удалить

Рисунок 22 – Редактирование справочника «Значения качественных параметров»

Значения можно изменять. Для этого надо вызвать команда «Изменить» в контекстном меню. Откроется форма редактирования значения (рисунок 23).

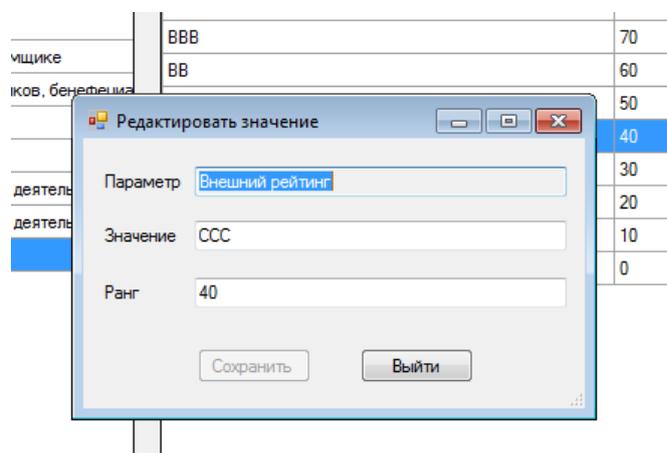


Рисунок 23 – Редактирование значения качественного параметра

Для выполнения действий над заемщиком позволяет создать запись, редактировать параметры, а также удалить запись (рисунок 24).

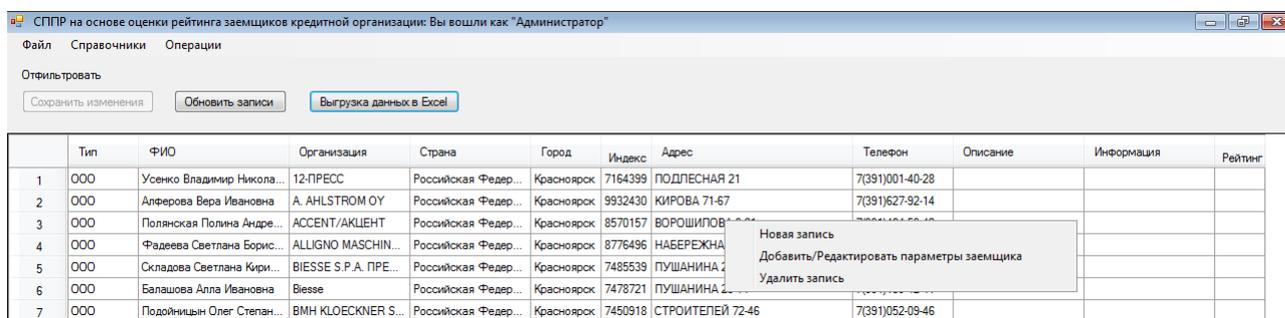


Рисунок 24 – Контекстное меню действий над заемщиком

При редактировании записи открывается форма (рисунок 25).

На ней содержатся значения количественных и качественных параметров для выбранного заемщика.

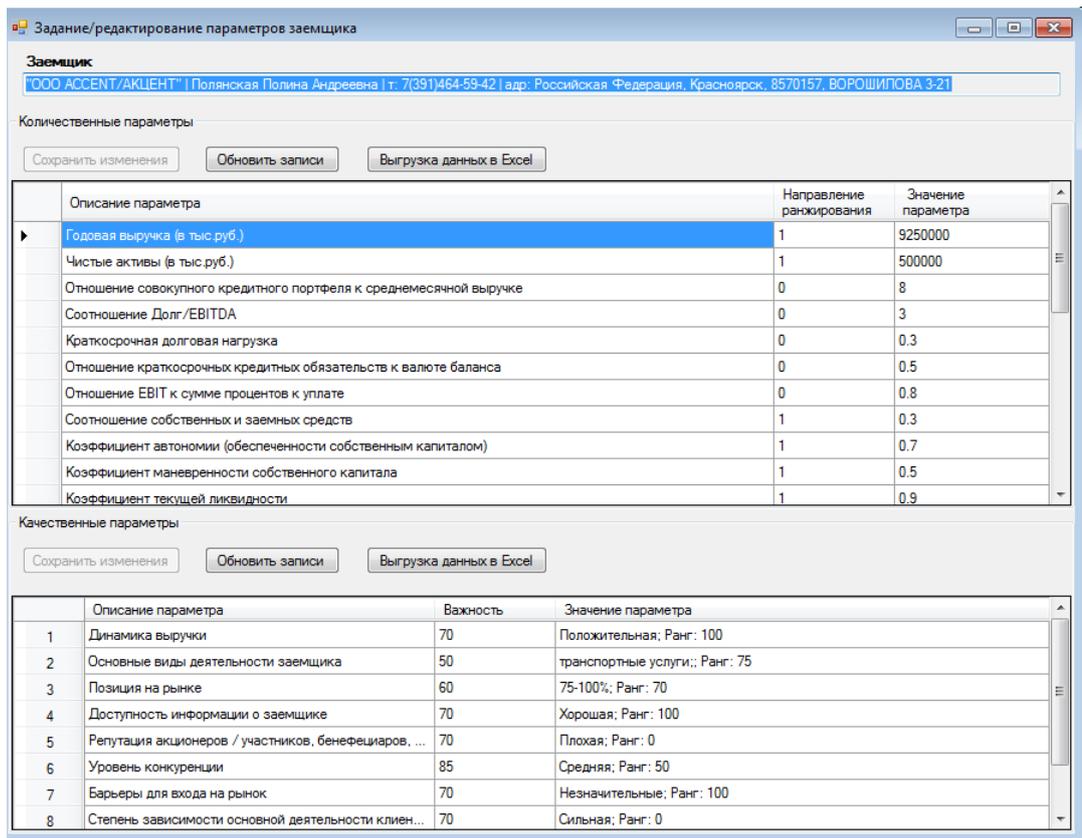


Рисунок 25 – Редактирование количественных и качественных параметров заемщика

Для выбора количественного параметра достаточно просто изменить значение в таблице. Но для выбора качественного параметра существует специальная форма, которая вызывается из контекстного меню (рисунок 26-27).



Рисунок 26 – Контекстное меню выбора нового значения параметра

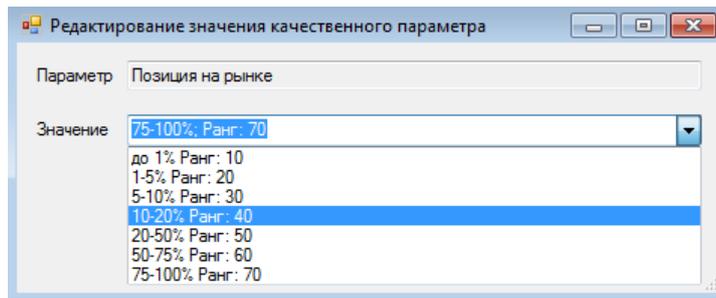


Рисунок 27 – Форма выбора нового значения параметра

В программе доступно меню операций (рисунок 28).

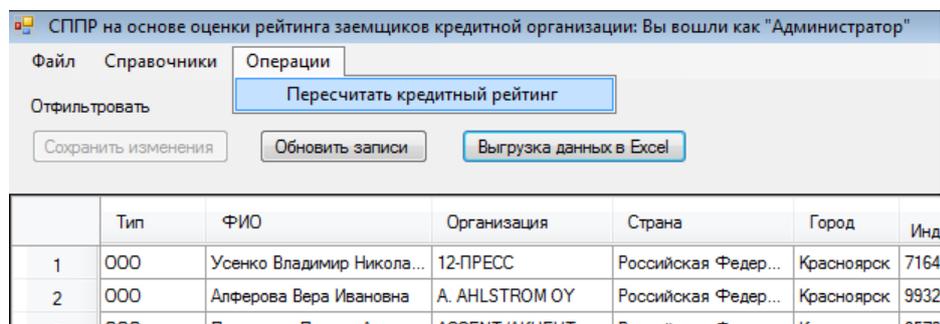


Рисунок 28 – Меню операций

В результате расчета рейтинга происходит заполнение последнего столбца в таблице с заемщиками, который потом можно отсортировать, кликая мышью на заголовок столбца (рисунок 29).

	Тип	ФИО	Организация	Страна	Город	Индекс	Адрес	Телефон	Описание	Информация	Рейтинг
1	ООО	Усенко Владимир Никола...	12-ПРЕСС	Российская Федер...	Красноярск	7164399	ПОДЛЕСНАЯ 21	7(391)001-40-28			16939
2	ООО	Алферова Вера Ивановна	A. AHLSTROM OY	Российская Федер...	Красноярск	9932430	КИРОВА 71-67	7(391)627-92-14			18039
3	ООО	Полянская Полина Андре...	ACCENT/АКЦЕНТ	Российская Федер...	Красноярск	8570157	ВОРОШИЛОВА 3-21	7(391)464-59-42			14784
4	ООО	Фадеева Светлана Борис...	ALLIGNO MASCHIN...	Российская Федер...	Красноярск	8776496	НАБЕРЕЖНАЯ С. ЛЕНИНО УЛИЦА ...	7(391)581-55-08			17169
5	ООО	Складова Светлана Кири...	BIESSE S.P.A. ПРЕ...	Российская Федер...	Красноярск	7485539	ПУШАНИНА 20-24	7(391)091-12-93			20844
6	ООО	Балашова Алла Ивановна	Biesse	Российская Федер...	Красноярск	7478721	ПУШАНИНА 20-14	7(391)109-42-41			18259
7	ООО	Подойницын Олег Степан...	BMH KLOECKNER S...	Российская Федер...	Красноярск	7450918	СТРОИТЕЛЕЙ 72-46	7(391)052-09-46			16164
8	ООО	Ильина Анастасия Алекс...	S.C.M.L. (ИТАЛИЯ) ...	Российская Федер...	Красноярск	9900791	МИРА 8-59	7(391)370-98-86			18164
9	ООО	Муреева Калерия Никола...	CANON (ЯПОНИЯ)	Российская Федер...	Красноярск	8910375	ЛОБАЧЕВСКОГО 1-Й 9-79	7(391)870-07-30			17214
10	ООО	Качанов Вениамин Васил...	Canon	Российская Федер...	Красноярск	7953035	ПОБЕДЫ 89	7(391)494-73-54			21079
11	ООО	Кузнецов Юрий Иванович	COSTA HANDLING ...	Российская Федер...	Красноярск	8887376	ЧААДАЕВА 107-34	7(391)065-01-15			18569
12	ООО	Федорова Галина Абрам...	Dieffenbacher & co.	Российская Федер...	Красноярск	7641278	ТОПСТОГО 34-16	7(391)955-74-16			18154
13	ООО	Махаева Людмила Дмитр...	DIEFFENBACHER G...	Российская Федер...	Красноярск	9584536	КАРПИНСКОГО 36-2	7(391)973-70-46			22394
14	ООО	Князев Геннадий Иванович	DIK & FELA GRAPHI...	Российская Федер...	Красноярск	8814363	ПОБЕДЫ 95-24	7(391)312-77-61			21419
15	ООО	Гольягина Мария Никол...	DIMAR W.P.W.	Российская Федер...	Красноярск	6704611	ЛАДОЖСКАЯ 29-55	7(391)378-48-40			18974

Рисунок 29 – Результат ранжирования заказчиков по кредитному рейтингу

Данную таблицу заемщиков можно вывести в Excel (рисунок 30-31).

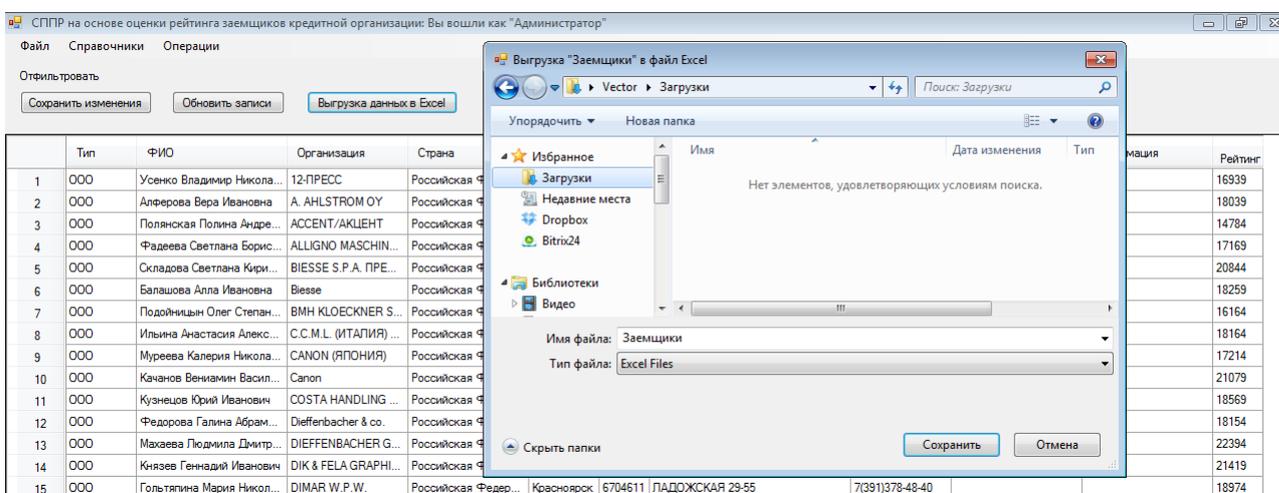


Рисунок 30 – выгрузка данных в Excel

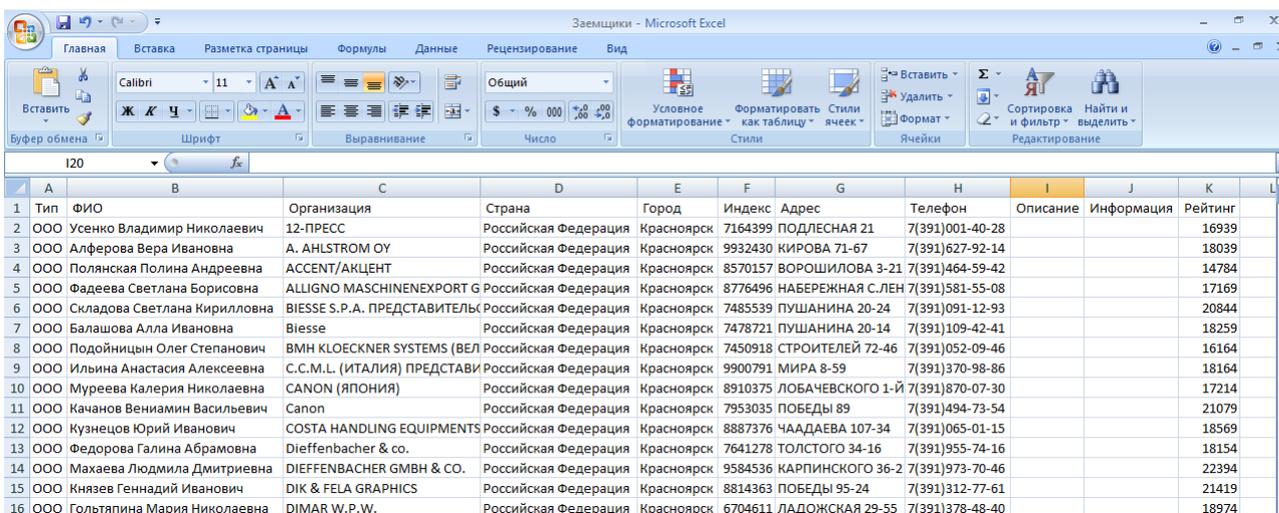


Рисунок 31 – Данные в Excel

ЗАКЛЮЧЕНИЕ

В данной работе выполнено построение и тестирование автоматизированной информационной системы поддержки принятия решения на основе оценки рейтинга заемщиков кредитной организации.

Первоначально произведен анализ предметной области. Рассмотрены такие вопросы, как:

- понятие и виды кредитного риска, управления кредитным риском;
- анализ систем управления рисками;
- обзор методик оценки кредитоспособности заемщиков.

В качестве предприятия для исследования выступила кредитная организация. Выполнена характеристика предприятия и анализ его оргструктуры.

В рамках проектирования информационной системы принятия решения на основе оценки рейтинга заемщиков выполнен анализ существующего программного обеспечения. Отечественными системами управления банковскими рисками являются: EGAR Technology, Прогноз, Intersoft Lab, РЭА-Риск-менеджмент, СофтВел, ИНЭК.

В качестве СУБД была выбрана mysql, поскольку она самая распространенная полноценная серверная СУБД. Также MySQL очень функциональная, свободно распространяемая СУБД.

В качестве языка программирования выбран C#, которому соответствует среда разработки Visual Studio. Этот выбор обусловлен тем, что C# фактически интегрировал в себя все плюсы языков Java и C++, а также имеет такие дополнительные особенности, как автоматический сборщик мусора, продвинутую систему поддержки разработки кода, встроенную в IDE, множество удобных и полезных типов данных, значительно ускоряющих процесс разработки. К тому же IDE Visual C# Express является бесплатной средой разработки приложений на C#.

В рамках проектирования базы данных выполнено описание сущностей, построена даталогическая модель, по которой сформирован скрипт инициализации БД.

Проектирование интерфейса произведено средствами графического редактора интерфейсов MS Visual Studio.

В рамках тестирования разработанного приложения произведено заполнение ее данными. Тестирование показало работоспособность системы.

Таким образом, задачи, поставленные нами в выпускной квалификационной работе, успешно выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Балдин, К. В. Риск-менеджмент : учебное пособие / К. В. Балдин. – Москва : ЭКСМО, 2008. – 368 с.
2. Банковское дело : учебник / под ред. Г. Н. Белоглазовой, Л. П. Кроливецкой. – 2-е изд. – Санкт-Петербург : Питер, 2011. – 400 с.
3. Буглак, Е. А. Современные подходы к регулированию банковских рисков / Е. А. Буглак // Молодой ученый. – 2011. – № 6. – С. 147–151.
4. Васильев, Ю. А. Работа с PostgreSQL: настройка и масштабирование [Электронный ресурс] : справочное пособие / Ю. А. Васильев. – Режим доступа: <https://knasys.ru/wp-content/uploads/2014/10/postgresql.pdf>
5. Винс, Р. Математика управления капиталом. Методы анализа риска / Р. Винс. – Москва : Альпина Паблишер, 2012. – 408 с.
6. Волков, А. А. Управление рисками в коммерческом банке / А. А. Волков. – Москва : Омега-Л, 2012. – 160 с.
7. Воробьев, С. Н. Системный анализ и управление рисками в организации / С. Н. Воробьев, К. В. Балдин. – Москва : МОДЕК, 2009. – 760 с.
8. Димитриади, Г. Г. Риски управления банком / Г. Г. Димитриади. – Санкт-Петербург : ЛКИ, 2010. – 240 с.
9. Емельянова, Н. З. Проектирование информационных систем / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. – Москва : Форум, НИЦ ИНФРА-М 2014. – 432 с.
10. Ендовицкий, Д. А. Анализ и оценка кредитоспособности заемщика: учебно-практическое пособие / Д. А. Ендовицкий, И. В. Бочарова. – Москва : КНОРУС, 2015. – 234 с.
11. Ермасова, Н. Б. Риск менеджмент организации / Н. Б. Ермасова. – Москва : Научная книга, 2011. – 120 с.
12. Жариков, В. В., Управление кредитными рисками / В. В. Жариков, М. В. Жарикова, А. И. Евсейчев. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2012. – 244 с.

13. Жибурт, Н. Л. Характеристика основных подходов к управлению кредитными рисками в мировой банковской практике / Н. Л. Жибурт // Вестник Волгоградского Государственного Университета. Серия 9: Исследования Молодых Ученых. – 2010. – № 8-2. – С. 109–113.
14. Зеленков, Ю. А. Введение в базы данных [Электронный ресурс]: учебный курс / Ю. А. Зеленков. – Режим доступа: <http://www.mstu.edu.ru/study/materials/zelenkov/toc.html>
15. Кабушкин, С. Н. Управление банковским кредитным риском : учебное пособие. – 4-е изд., стер. – Минск : Новое знание, 2009. – 336 с.
16. Кабушкин, С. Н. Управление банковским кредитным риском: учебное пособие / С. Н. Кабушкин. – Минск : Новое знание, 2012. – 254 с.
17. Каменева, О. С., Кредитный риск банка. Проблемы оценки и управления / О. С. Каменева, А. А. Гарпунов. – Санкт-Петербург : Питер. 2013. – 139 с.
18. Кричевский, М. Л. Финансовые риски / М. Л. Кричевский. – Москва : КНОРУС, 2012. – 248 с.
19. Кузнецов, С. Д. Основы баз данных / С. Д. Кузнецов. – 2-е изд. – Москва : Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2007. – 488 с.
20. Кутафьева, Л. В. Классификация банковских рисков / Л. В. Кутафьева // Молодой ученый. – 2013. – № 10. – С. 324–326.
21. Лаврушин, О. И. Банковское дело: современная система кредитования: учебное пособие / О. И. Лаврушин, О. Н. Афанасьева, С. Л. Корниенко; под ред. О. И. Лаврушина. – 3-е изд., доп. – Москва : КНОРУС, 2012. – 264 с.
22. Леонович, Л. И. Управление рисками в банковской деятельности / Л. И. Леонович, В. М. Петрушина. – Москва : Дикта, 2012. – 136 с.
23. Макконнел, С. Профессиональная разработка программного обеспечения / С. Макконнел. – Символ-Плюс, 2007. – 240 с.

24. Маклафлин, Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафлин. – Санкт-Петербург : Питер, 2016. – 544 с.
25. Помазанов, М. В. Продвинутый подход к управлению кредитным риском в банке: методология, практика, рекомендации / М. В. Помазанов. – Москва : Регламент, 2010.
26. Попов, И. В. Практика управления кредитным риском в российских коммерческих банках / И. В. Попов. – Москва : МАКС Пресс, 2010. – 20 с.
27. Пример линейно-функциональной структуры [Электронный ресурс]. – Режим доступа: <http://rthl.ru/f/primer-lineyno-funktsionalnoy-strukturi.html>
28. Сравнение языков программирования [Электронный ресурс] // Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Сравнение_языков_программирования
29. Тидвелл, Д. Разработка пользовательских интерфейсов / Д. Тидвелл. – 2-е изд. – Санкт-Петербург : Питер, 2011. – 480 с.
30. Шаталова, Е. П. Оценка кредитоспособности заемщиков в банковском риск-менеджменте : учебное пособие / Е. П. Шаталова. – 2-е изд., стер. – Москва : КНОРУС, 2012. – 168 с.
31. CRUD [Электронный ресурс] // Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/CRUD>
32. Top 10 Programming Languages [Электронный ресурс]. – Режим доступа: <http://spectrum.ieee.org/computing/software/top-10-programming-languages>
33. Vikash, K. Android Sqlite Essentials. – Изд. Книга по Требованию, 2014. – 157 с.
34. Агапов, В. Профессиональная разработка программного обеспечения : учебник / В. Агапов. – Москва : Символ-Плюс, 2007. – 240 с.
35. Мальцев, Э. В. Скоринговые системы в кредитовании физических лиц Банковский ритейл : учебник / Э. В. Мальцев. – Москва : Юрайт, 2006. – 134 с.

36. Наумов, М. Ф. Кредитный скоринг : учебник / М. Ф. Наумов. – Москва : Финтайм, 2007. – 176 с.
37. Стародубцева, Е. Б. Современный экономический словарь : учебное пособие / Е.Б Стародувцев. – Москва : ИНФРА-М, 2007. – 212 с.
38. Романов, М. Н. Основные подходы к оценки кредитного риска в РФ Банковское дело : учебное пособие / М. Н. Романов. – Москва : ПРОМТА, 2011. – 103 с.
39. Рыкова, И. Н. Методика оценки кредитоспособности заемщиков Банковское кредитование : учебник / И. Н. Рыков. – Москва : Юнайт, 2007. – 231 с.
40. Сальников, К. Кредитная политика банка : учебное пособие / К. Сальников. – Москва : Лепта, 2009. – 321 с.
41. Тавасиев, А. М. Банковское дело. Управление кредитной организацией : учебное пособие / А. М. Тавасиев. – Москва : Дашков и К., 2011. – 216 с.
42. Триф, А. А. Инвестиционная и кредитная деятельность коммерческих банков : учебное пособие / А.А. Триф. – Москва : Экономика, 2005. – 222 с.
43. Сенчагов, А. И. Финансы, денежное обращение и кредит : учебник / В. К. Сенчагов. – Москва : Проспект, 2009. – 231 с.
44. Шевченко, И. В. Совершенствование качества обслуживания клиентов кредитными организациями путем внедрения новейших банковских технологий : учебное пособие / И. В. Шевченко. – Москва : Финансы и кредит. - 2007. – 163 с.

Приложения А

SQL скрипты инициализации базы данных

```
DROP DATABASE IF EXISTS credit_rating;
CREATE DATABASE IF NOT EXISTS credit_rating;
USE credit_rating;

DROP TABLE IF EXISTS zaemshiki;
CREATE TABLE zaemshiki (
  id_za INTEGER NOT NULL AUTO_INCREMENT,
  id_ti INTEGER NOT NULL,
  fio VARCHAR(255) NOT NULL DEFAULT '',
  organizaciya VARCHAR(255) NOT NULL DEFAULT '',
  strana VARCHAR(255) NOT NULL DEFAULT '',
  gorod VARCHAR(255) NOT NULL DEFAULT '',
  indeks VARCHAR(255) NOT NULL DEFAULT '',
  adres VARCHAR(255) NOT NULL DEFAULT '',
  telefon VARCHAR(255) NOT NULL DEFAULT '',
  opisanie VARCHAR(255) NULL,
  informaciya VARCHAR(255) NULL,
  reiting FLOAT DEFAULT null,
  PRIMARY KEY (id_za)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

DROP TABLE IF EXISTS tipi;
CREATE TABLE tipi (
  id_ti INTEGER NOT NULL AUTO_INCREMENT,
  tip VARCHAR(255),
  PRIMARY KEY (id_ti)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

DROP TABLE IF EXISTS kach_param;
CREATE TABLE kach_param (
  id_ka INTEGER NOT NULL AUTO_INCREMENT,
  opisanie VARCHAR(255),
  vagnost INTEGER NOT NULL DEFAULT 1,
  PRIMARY KEY (id_ka)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

DROP TABLE IF EXISTS kol_param;
CREATE TABLE kol_param (
  id_ko INTEGER NOT NULL AUTO_INCREMENT,
  opisanie VARCHAR(255),
  napravl INTEGER NOT NULL DEFAULT 1,
  vagnost INTEGER NOT NULL DEFAULT 1,
  PRIMARY KEY (id_ko)
) ENGINE=innodb DEFAULT CHARSET=utf8;
SET autocommit=1;

DROP TABLE IF EXISTS zn_kach_param;
CREATE TABLE zn_kach_param(
  id_zn INTEGER NOT NULL AUTO_INCREMENT,
```

```
id_ka INTEGER NOT NULL,  
znachenie TEXT,  
rang INTEGER NOT NULL DEFAULT 0,  
PRIMARY KEY (id_zn)  
) ENGINE=innodb DEFAULT CHARSET=utf8;  
SET autocommit=1;
```

```
DROP TABLE IF EXISTS zaem_kol;  
CREATE TABLE zaem_kol (  
id_za INTEGER NOT NULL,  
id_ko INTEGER NOT NULL,  
znach FLOAT DEFAULT NULL,  
PRIMARY KEY (id_za, id_ko)  
) ENGINE=innodb DEFAULT CHARSET=utf8;  
SET autocommit=1;
```

```
DROP TABLE IF EXISTS zaem_kach;  
CREATE TABLE zaem_kach (  
id_za INTEGER NOT NULL,  
id_ka INTEGER NOT NULL,  
id_zn INTEGER DEFAULT NULL,  
PRIMARY KEY (id_za, id_ka)  
) ENGINE=innodb DEFAULT CHARSET=utf8;  
SET autocommit=1;
```

```
DROP TABLE IF EXISTS users;  
CREATE TABLE users (  
id_use INTEGER NOT NULL AUTO_INCREMENT,  
fio VARCHAR(255),  
login VARCHAR(255),  
psswd VARCHAR(255),  
rights VARCHAR(15) DEFAULT '11111',  
PRIMARY KEY (id_use)  
) ENGINE=innodb DEFAULT CHARSET=utf8;  
SET autocommit=1;
```

Приложение Б

Листинги программных модулей

Модуль редактирования кредитных параметров

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

namespace arm
{
    public partial class EditCreditParams : Form
    {
        public main mF;
        int currRow=-1, currCol=-1;
        string id_za="-1";
        int currentMouseOverCol;
        int currentMouseOverRow;
        Dictionary<int, int> di = new Dictionary<int, int>();
        Dictionary<int, string> changed = new Dictionary<int, string>();
        bool allowChange = false;

        public EditCreditParams(main fM, int r, int c)
        {
            InitializeComponent();
            currRow = r;
            currCol = c;

            if (currRow < 0 && currCol < 0)
                this.Close();

            mF = fM;

            id_za = mF.mainGrd.Rows[r].Cells[0].Value.ToString();
            string sql;
            textBox1.Text = "\"" +
mF.mainGrd.Rows[r].Cells[2].Value.ToString() + " " +
mF.mainGrd.Rows[r].Cells[4].Value.ToString() + "\" | " +
mF.mainGrd.Rows[r].Cells[3].Value.ToString() + " | т: " +
mF.mainGrd.Rows[r].Cells[9].Value.ToString() + " | адр: " +
mF.mainGrd.Rows[r].Cells[5].Value.ToString() + ", " +
mF.mainGrd.Rows[r].Cells[6].Value.ToString() + ", " +
mF.mainGrd.Rows[r].Cells[7].Value.ToString() + ", " +
mF.mainGrd.Rows[r].Cells[8].Value.ToString();

            List<List<String>> sqlresult = new List<List<string>>();

            // Проверка наличия количественных параметров. Если нет, то
            их создание для заемщика

```

```

        bool res = mF.sqlSelect("select count(id_ko) from zaem_kol
where id_za="+id_za, ref sqlresult);
        if (!res || sqlresult.Count == 0 || sqlresult.Count > 1)
        {
            MessageBox.Show("Ошибка проверки наличия количественных
параметров");
            return;
        }
        if (sqlresult[0][0] == "0")
        {
            sql = "insert into zaem_kol(id_za, id_ko) select " +
id_za + ", id_ko from kol_param";
            mF.sqlInsert(sql);
        }

        // Проверка наличия качественных параметров. Если нет, то их
создание для заемщика
        res = mF.sqlSelect("select count(id_ka) from zaem_kach where
id_za=" + id_za, ref sqlresult);
        if (!res || sqlresult.Count == 0 || sqlresult.Count > 1)
        {
            MessageBox.Show("Ошибка проверки наличия качественных
параметров");
            return;
        }
        if (sqlresult[0][0] == "0")
        {
            sql = "insert into zaem_kach(id_za, id_ka) select " +
id_za + ", id_ka from kach_param";
            mF.sqlInsert(sql);
        }

        loadKolParams();
        loadKachParams();
    }

    public void loadKolParams()
    {
        // Загрузка количественных параметров
        btnKol1.Enabled = false;
        allowChange = false;
        mF.showSQLinGridView(ref di, 5, @"select
            a.id_za, a.id_ko, b.opisanie, b.napravl, a.znach
            from zaem_kol as a
            inner join kol_param as b on a.id_ko=b.id_ko
            where a.id_za=" + id_za,
            this.gridKol,
            "id_za,id_ko,Описание параметра,Направление
ранжирования, Значение параметра",
            "0,1",
            "0,0,590,100,100");
        allowChange = true;
    }

    public void loadKachParams()
    {
        // Загрузка качественных параметров
        arm.main.links myLinksTmp2 = new arm.main.links("zaem_kach");
    }

```

```

        mF.OpenCatalog(@"
            select
            a.id_ka, b.opisanie, b.vagnost, a.id_zn, a.id_za,
concat(c.znachenie, '; Ранг: ', c.rang) as znach
            from zaem_kach as a
            inner join kach_param as b on a.id_ka=b.id_ka
            left outer join zn_kach_param as c on a.id_zn=c.id_zn
            where a.id_za=" + id_za,

" id_ka#int#1,opisanie#str#0,vagnost#int#0,id_zn#int#1,id_za#int#1,znach#
str#0",

            "Качественные параметры",
            "id_ka,Описание параметра,Важность,id_zn,id_za,Значение
параметра",

            "0,290,100,0,0,300",
            "0,3,4",
            myLinksTmp2,
            this.gridKach,
            this.btnKach1,
            this.btnKach2,
            this.btnKach3,
            false,
            "1,2");
    }

    private void EditCreditParams_Load(object sender, EventArgs e)
    {
        this.Location = new Point((Screen.PrimaryScreen.Bounds.Width
- this.Width) / 2,
            (Screen.PrimaryScreen.Bounds.Height - this.Height) / 2);
    }

    private void gridKach_MouseClick(object sender, MouseEventArgs e)
    {
        currentMouseOverCol = gridKach.HitTest(e.X, e.Y).ColumnIndex;
        currentMouseOverRow = gridKach.HitTest(e.X, e.Y).RowIndex;

        if (e.Button == MouseButton.Right)
        {
            ContextMenu m = new ContextMenu();
            m.MenuItems.Clear();

            if (currentMouseOverRow >= 0)
            {
                if
                (gridKach.Rows[currentMouseOverRow].Cells[0].Value.ToString() != "")
                {
                    System.Windows.Forms.MenuItem itm1 = new
                    System.Windows.Forms.MenuItem("Выбрать новое значение параметра");
                    itm1.Tag = 1;
                    m.MenuItems.Add(itm1);
                    itm1.Click += new
                    System.EventHandler(contextMenuItem_Click1);
                }
            }

            m.Show(gridKach, new System.Drawing.Point(e.X, e.Y));
        }
    }

```

```

    }

    public void contexMenuItem_Click1(object sender, System.EventArgs
e) // клик по контекстному меню таблицы
    {
        System.Windows.Forms.MenuItem itm =
(System.Windows.Forms.MenuItem) sender;
        int op = Convert.ToInt32(itm.Tag);
        switch (op)
        {
            case 1: editKachParam();
                break;
        }
    }

    public void editKachParam()
    {
        EditZnachKachParam ed = new EditZnachKachParam(mF,
gridKach.Rows[currentMouseOverRow].Cells[4].Value.ToString(),
gridKach.Rows[currentMouseOverRow].Cells[3].Value.ToString(),
gridKach.Rows[currentMouseOverRow].Cells[0].Value.ToString(),
gridKach.Rows[currentMouseOverRow].Cells[1].Value.ToString(),
gridKach.Rows[currentMouseOverRow].Cells[5].Value.ToString());
        ed.ShowDialog();
        loadKachParams();
    }

    private void gridKol_CellValueChanged(object sender,
DataGridViewCellEventArgs e)
    {
        if (allowChange)
        {
            changed.Add(e.RowIndex,
this.gridKol.Rows[e.RowIndex].Cells[4].Value.ToString());
            btnKol1.Enabled = true;
        }
    }

    private void btnKol3_Click(object sender, EventArgs e)
    {
        loadKolParams();
    }

    private void btnKol2_Click(object sender, EventArgs e)
    {
        SaveFileDialog saveDlg = new SaveFileDialog();
        saveDlg.Filter = "Excel Files|*.xlsx";
        saveDlg.DefaultExt = ".xlsx";
        saveDlg.Title = "Выгрузка количественных параметров в файл
Excel";

        saveDlg.FileName = "Количественные параметры.xlsx";
        if (saveDlg.ShowDialog() == DialogResult.OK)
        {
            try
            {
                string path =
arm.main.GetFullPathWithoutExtension(saveDlg.FileName);

```

```

        mF.printGrid(gridKol, "Количественные параметры",
path);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка. Не удалось сохранить файл
Excel\r\n" + ex.Message);
    }
}

private void btnKoll_Click(object sender, EventArgs e)
{
    foreach (int key in changed.Keys)
    {
        string val = changed[key];
        string id_za =
gridKol.Rows[key].Cells[0].Value.ToString();
        string id_ko =
gridKol.Rows[key].Cells[1].Value.ToString();
        string sql = "update zaem_kol set znach=" + val + " where
id_za=" + id_za + " and id_ko=" + id_ko;
        //MessageBox.Show(sql);
        mF.sqlOther(sql);
    }
    loadKolParams();
}
}
}

```

Модуль редактирования значения качественного параметра

```

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

namespace arm
{
    public partial class EditZnachKachParam : Form
    {
        const char DELIM = ',';
        public main mF;
        string id_za, id_zn, id_ka, param, value;
        Dictionary<int, string> currComboDict1 = new Dictionary<int,
string>();

        public EditZnachKachParam(main fM, string idZa, string idZn,
string idKa, string par, string v)
        {
            InitializeComponent();

```

```

        mF = fM;
        id_za = idZa;
        id_zn = idZn;
        id_ka = idKa;
        param = par;
        value = v;

        textBox1.Text = param;
        button1.Enabled = false;

        loadParams();
        comboBox1.Text = v;
    }

    public void loadParams()
    {
        string sql = "select id_zn, znachenie, rang from
zn_kach_param where id_ka=" + id_ka;
        showSQLinComboCellAndDict1(sql, value, ref comboBox1, ref
currComboDict1, "#,#, Пар: #");
    }

    public void showSQLinComboCellAndDict1(string sql, string
valsInCell, ref ComboBox combo, ref Dictionary<int, string> dict, string
prefPost)
    {
        List<List<String>> sqlresult = new List<List<string>>();
        string str;

        bool res = mF.sqlSelect(sql, ref sqlresult);
        if (res && sqlresult.Count > 0)
        {
            combo.Items.Clear();
            dict.Clear();

            //string[] hideC = hiddenCols.Split(DELMIM).ToArray();

            List<List<string>> lst = new List<List<string>>();
            List<List<string>> lst1 = new List<List<string>>();

            string[] valsCell = valsInCell.Split(DELMIM).ToArray();

            List<string> tmp = prefPost.Split(DELMIM).ToList();
            for (int i = 0; i < tmp.Count; i++)
                lst.Add(tmp[i].Split('#').ToList());

            for (int i = 0; i < sqlresult.Count; i++)
            {
                str = "";
                for (int j = 1; j < sqlresult[i].Count; j++)
                    str += lst[j][0] + sqlresult[i][j] + lst[j][1];
                combo.Items.Add(str);
                dict.Add(i, sqlresult[i][0]);
            }
        }
    }

    private void EditZnachKachParam_Load(object sender, EventArgs e)

```

```

        {
            this.Location = new Point((Screen.PrimaryScreen.Bounds.Width
- this.Width) / 2,
                (Screen.PrimaryScreen.Bounds.Height - this.Height) / 2);
        }

        private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            button1.Enabled = true;
        }

        private void EditZnachKachParam_FormClosing(object sender,
FormClosingEventArgs e)
        {
            if (button1.Enabled)
                if (MessageBox.Show("У вас есть несохраненные
данные.\r\nВсе равно продолжить?", "Внимание", MessageBoxButtons.YesNo)
== DialogResult.No)
                    e.Cancel = true;
                else
                    e.Cancel = false;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string sql = "UPDATE zaem_kach SET id_zn=" +
currComboDict1[comboBox1.SelectedIndex] + " WHERE id_za=" + id_za + " AND
id_ka=" + id_ka;
            mF.sqlOther(sql);

            button1.Enabled = false;
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }

    }
}

```

Приложение В
Сертификат научной статьи



СЕРТИФИКАТ участника

XLII студенческая международная
научно-практическая конференция
Молодежный научный форум

Погодин Даниил Витальевич

На конференцию представлена научная работа:

«Проектирование системы принятия решения на основе оценки
рейтинга заемщиков кредитной организации»

Научный руководитель:

Киргизова Елена Викторовна

д-р философии в области культурологии,
профессор философии Международной
кадровой академии, член Евразийской Академии
Телевидения и Радио





А. Лебедева

Москва
03.04.2019