

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет инженерно-экономический  
Кафедра экономической информатики  
Дисциплина «Программирование сетевых приложений»

«К ЗАЩИТЕ ДОПУСТИТЬ»

Руководитель курсового проекта  
ассистент кафедры экономической  
информатики

\_\_\_\_\_.\_\_\_\_\_.2021 К.А. Климович

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ БАНКОВСКИМИ  
ВКЛАДАМИ ФИЗИЧЕСКИХ ЛИЦ»**

БГУИР КП 1-40 01 02-08 015 ПЗ

Выполнил студент группы 972303  
СНОПОК Лика Александровна

\_\_\_\_\_  
(подпись студента)

Курсовой проект представлен на  
проверку \_\_\_\_\_.\_\_\_\_\_.2021

\_\_\_\_\_  
(подпись студента)

Минск 2021

# СОДЕРЖАНИЕ

Введение.....	5
1 Описание банковской сферы .....	7
2 Постановка задачи на разработку программного средства .....	9
3 Информационная модель системы .....	10
4 Информационная модель системы по учету банковских вкладов.....	15
5 Модели представления системы по учету банковских вкладов физических лиц .....	18
5.1 Спецификация вариантов использования системы .....	18
5.2 Описание диаграммы последовательности .....	19
5.3 Описание диаграммы состояний .....	20
5.4 Описание диаграммы компонентов .....	21
5.5 Описание диаграммы развертывания .....	22
5.6 Описание диаграммы классов.....	23
6 Описание алгоритмов, реализующих бизнес-логику системы по учету банковских вкладов .....	25
6.1 Схема алгоритма клиент-серверного взаимодействия.....	25
6.2 Схема алгоритма расчета суммы по вкладу .....	27
6.3 Схема алгоритма построения графика.....	28
7 Руководство пользователя.....	30
8 Результаты тестирования разработанной системы.....	46
Заключение .....	48
Список использованных источников .....	49
Приложение А .....	50
Приложение Б .....	54

## **ВВЕДЕНИЕ**

Современный период развития общества характеризуется сильным влиянием на него компьютерных технологий, которые проникают во все сферы деятельности человека, обеспечивают распространение информационных потоков в обществе, образуя глобальное информационное пространство.

В настоящее время в век высоких технологий не найдется ни одной организации в какой-либо сфере деятельности, в которой бы не использовались информационные технологии.

Информационные системы используют в различных отраслях экономики страны, начиная от сельского хозяйства и заканчивая высокотехнологичным производством. Применение технологий облегчает процессы в любой деятельности человека, что в свою очередь повышает производительность труда, снижает влияние человеческого фактора и риск возникновения ошибок в работе.

Наиболее широко информационные технологии используются в таких областях человеческой деятельности, как медицина, банковское дело, государственная и транспортная сферы. Их применение позволяют сохранять и обрабатывать огромное количество информации, которая необходима для нормальной и эффективной деятельности организации.

Помимо повышения эффективности работы компаний внедрение информационных технологий дает возможность предлагать для клиентов новые виды услуг. Благодаря непрерывному совершенствованию с каждым днем все большее количество фирм переходит на автоматизированные системы учета информации.

Исходя из всего вышесказанного была сформулирована цель данной работы: облегчить организацию учета привлеченных банковских вкладов физических лиц, а также снизить количество ошибок, возникающих во время работы.

Прежде чем приступить к выполнению поставленной цели. Необходимо обозначить объект и предмет исследования, поскольку от правильно выбранного объекта и предмета напрямую зависит весь результат работы.

Объект является более широким понятием нежели предмет, так в проекте объектом исследования будет являться банк и банковская сфера в целом.

Предмет исследования – это некий аспект объекта, который подлежит изучению. Так предметом исследования будет процесс учета и управления депозитами в банке.

Кроме определения объекта и предмета для достижения поставленной цели потребуется решение следующих задач:

- проанализировать деятельность в банковской сфере;
- исследовать бизнес-процессы, связанные с учетом вкладов;
- определить какие стороны нуждаются в автоматизации;
- разработать информационную модель в соответствии с поставленными задачами;
- спроектировать логическую и физическую модели данных;
- разработать программное обеспечение;
- протестировать и составить описание полученного ПО.

Вследствие внедрения информационных технологий организации могут успешно конкурировать на мировом рынке. И чем быстрее поступает и внедряется информация, тем большего успеха может достичь компания.

# 1 ОПИСАНИЕ БАНКОВСКОЙ СФЕРЫ

Банковская сфера – это включенная в экономическую систему взаимодействующая совокупность кредитных организаций, рынков, бизнеса [1].

Банковские операции – виды хозяйственной деятельности, к осуществлению которых допускаются исключительно организации, имеющие лицензию, выдаваемую центральными банками. Лицензирование банковской деятельности связано с необходимостью защиты денежных средств частных лиц и компаний.

К основным банковским операциям можно отнести следующие:

- привлечение денежных средств физических и юридических лиц во вклады;
- открытие и ведение банковских счетов физических и юридических лиц;
- осуществление переводов по поручению физических и юридических лиц, в том числе банков-корреспондентов, по их банковским счетам;
- инкассация денежных средств, векселей, платежных и расчетных документов и кассовое обслуживание физических и юридических лиц;
- купля-продажа иностранной валюты в наличной и безналичной формах;
- привлечение драгоценных металлов физических и юридических лиц во вклады (до востребования и на определенный срок), за исключением монет из драгоценных металлов;
- осуществление переводов денежных средств по поручению физических лиц без открытия банковских счетов, в том числе электронных денежных средств (за исключением почтовых переводов).

Вкладом (депозитом) называется способ размещения денежных средств в банках для надежного сохранения этих средств и получения дохода от суммы вложения [2].

Вклад – это результат депозитного договора между вкладчиком и банком.

По такому соглашению банк принимает депозит и может распоряжаться вложенными средствами как собственными. За это владелец вклада получает от банка определенную, заранее оговоренную плату.

Вся сумма депозита остается собственностью вкладчика и должна быть возвращена либо немедленно по желанию клиента, либо в другой срок, обозначенный в договоре.

Привлеченные во вклады средства банки используют для коммерческих операций, в том числе выдачи кредитов.

В зависимости от цели выделяют вклады:

- краткосрочные и долгосрочные;
- в отечественной и зарубежной валюте;
- с фиксированной или меняющейся процентной ставкой;
- с ограничениями по снятию и пополнению и без таких ограничений;
- для физических лиц, для организаций;
- для бизнеса, для сбережения, для расчетов и др.

В рамках данного курсового проекта более детально будут рассмотрены операции со вкладами физических лиц, к ним относятся:

- открытие и закрытие вклада;
- пополнение и снятие денежных средств;
- переводы денежных средств на другие счета вкладчика и иных лиц;
- оплата товаров и услуг;
- погашение кредитов;
- обмен валют;
- получение информации о состоянии счёта вклада и операций по нему.

Данные операции могут осуществляться как при личном посещении отделения банка вкладчиком или его представителем, так и дистанционно через систему интернет-банкинга или мобильного банкинга.

Операции из приведённого выше списка являются доступными лишь для расчётных счетов или вкладов до востребования, по которым банки не устанавливают практически никаких ограничений. Максимально ограничен вкладчик в возможности распоряжаться средствами сберегательного вклада, который можно только открыть и закрыть, другие операции по нему, кроме получения информации, не допускаются.

## **2 ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ ПРОГРАММНОГО СРЕДСТВА**

Банковский сектор – высококонкурентная среда, которая требует постоянной оптимизации бизнес-процессов. На современном этапе развития банковской сферы возникает множество новинок и революционных технологий. Меняются требования рынка и предпочтения потребителей. В условиях финансовой нестабильности банки сталкиваются с беспрецедентным операционным давлением, которое проверяет эффективность, действенность и маневренность бизнес-процессов.

В этом случае необходимы эффективные методы организационного развития, реинжиниринга и управления бизнесом, которые позволят осуществлять мониторинг и координацию в масштабах всего предприятия. Кредитные организации стремятся создать лучший банковский бизнес-процесс – достаточно интеллектуальный, чтобы успешно сочетать бизнес-цели с желаниями потребителей, и достаточно гибкий, чтобы идти в ногу с динамичной операционной средой.

Исходя из этого можно определить функциональные возможности разрабатываемой системы по управлению банковскими вкладами.

В первую очередь система должна предусматривать тривиальные операции со вкладами, такие как изучение текущих, возможность некоторой сортировки и поиска, внесение новых данных в базу или наоборот удаление ненужных, а также изменение некоторых сведений. Затем необходимо реализовать более специфические операции, которые упростят учет вкладов и операции по ним.

Во-первых, стоит рассмотреть возможность добавления такого функционала как расчет итоговой суммы и процентов по вкладу через определенный срок, это позволит снизить время, затрачиваемое на выполнение данной работы вручную или даже с использованием других программных решений.

Во-вторых, система должна уметь предоставлять некоторые аналитические сведения и строить графики по исходным данным, что даст более наглядное представление о состоянии банка.

И наконец, реализация автоматической генерации отчетов по заданным параметрам. Как итог – меньше бумажной волокиты и ошибок, связанных с этим.

### 3 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ

Формализованность и регламентированность бизнес-процессов на сегодняшний день является одним из ключевых факторов успеха любого коммерческого банка. Описание и оптимизация бизнес-процессов прямым образом влияет на операционную и стратегическую эффективность банка, что в итоге влияет на показатели прибыльности и рейтинги [4].

Рассмотрим детально процесс поиска и привлечения новых клиентов на открытие счетов по вкладу и построим для него диаграмму в нотации IDEF0. Диаграмма контекстного уровня представлена на рисунке 3.1.

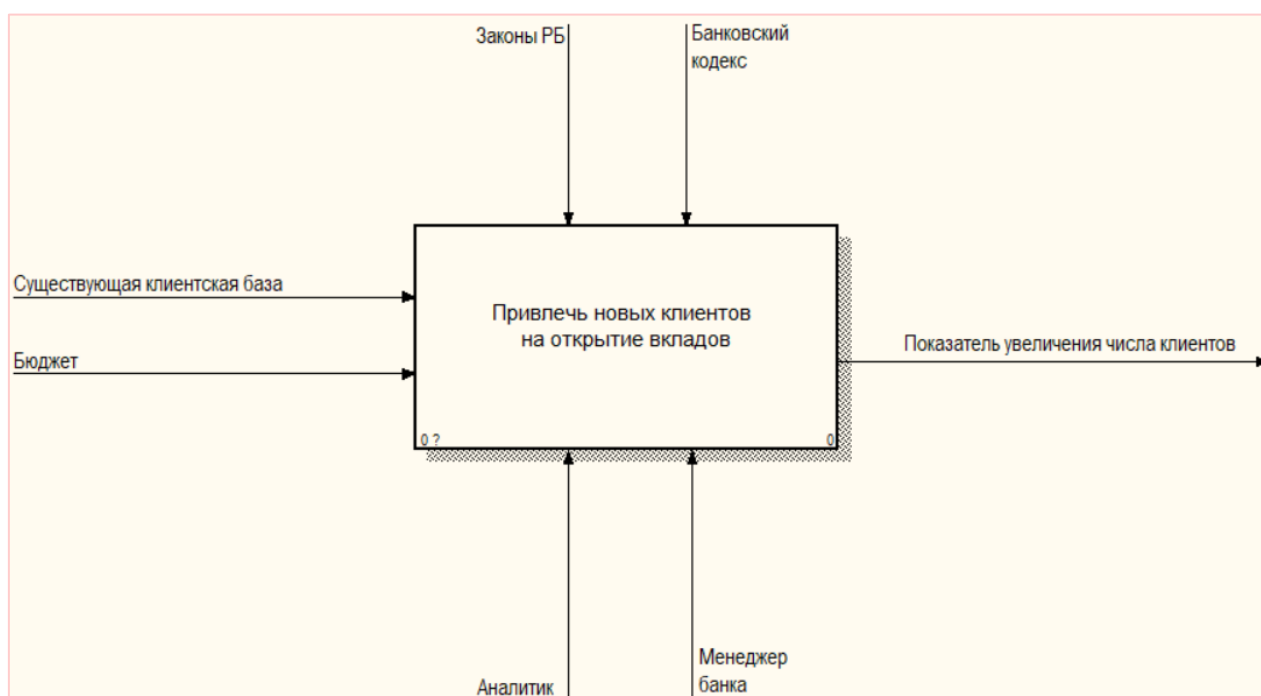


Рисунок 3.1 – Контекстная диаграмма

В качестве входных данных используются существующая клиентская база и бюджет на проведение мероприятий по привлечению клиентов. Как результат должны быть получены некие показатели того, что количество клиентов увеличилось, например, прирост клиентов; общее число открытых вкладов большее, чем до проведения мероприятий; показатель увеличения общей суммы средств, привлеченных на вклады, и другие. Работа по привлечению и поиску клиентов осуществляет аналитик и менеджер либо команды таких сотрудников, в своей работе они должны руководствоваться текущим законодательством Республики Беларусь и банковским кодексом.



Для детального рассмотрения была проведена декомпозиция контекстного уровня, в результате чего процесс был разбит ещё на четыре подпроцесса (рисунок 3.2).

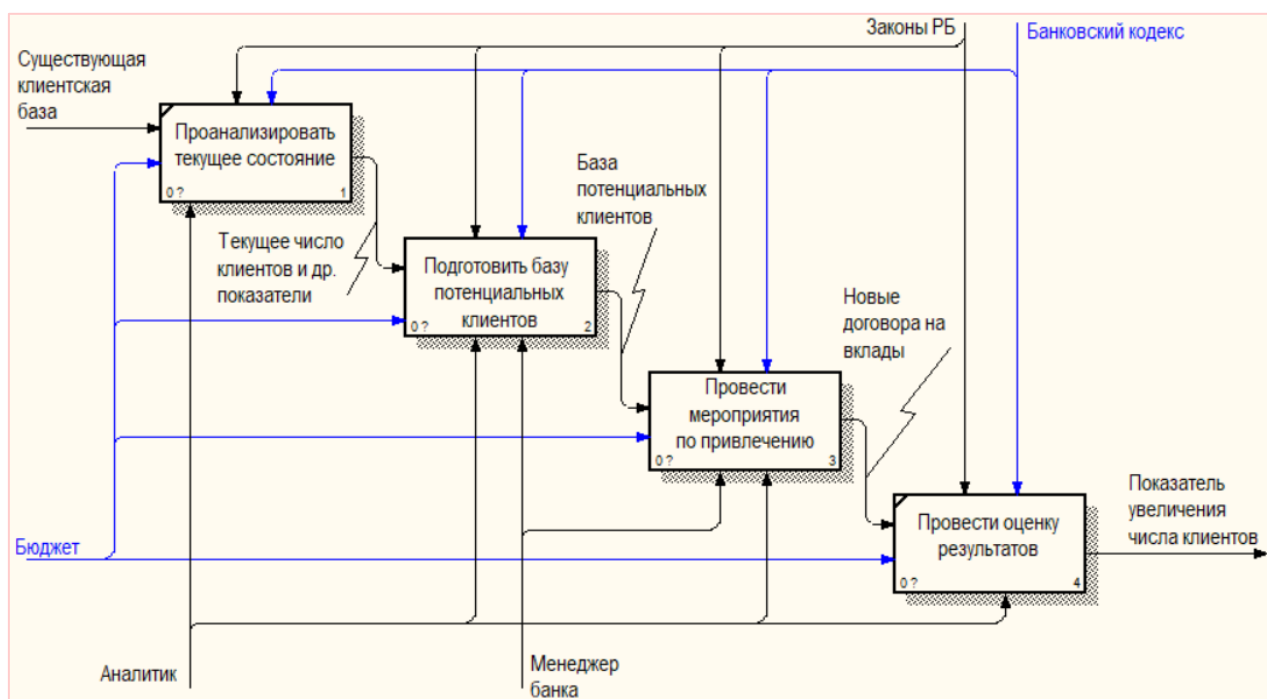


Рисунок 3.2 – Декомпозиция верхнего уровня

В качестве подпроцессов были выделены следующие:

- проанализировать текущее состояние;
- подготовить базу потенциальных клиентов;
- провести мероприятия по привлечению;
- провести оценку результатов.

Все работы по полученным подпроцессам выполняются аналитиком, на втором и третьем шаге к нему присоединяется менеджер.

Для анализа текущего состояния используются имеющаяся клиентская база и в результате работ по этому подпроцессу будут получены исходные показатели для сравнения на последнем этапе.

Во время подготовки базы потенциальных клиентов происходит изучение рынка и конкурентов, получение информации, где лучше осуществить поиск клиентов, возможно уже получение некоторых контактов. Декомпозиция подпроцесса «Подготовить базу потенциальных клиентов» приведена на рисунке 3.3.

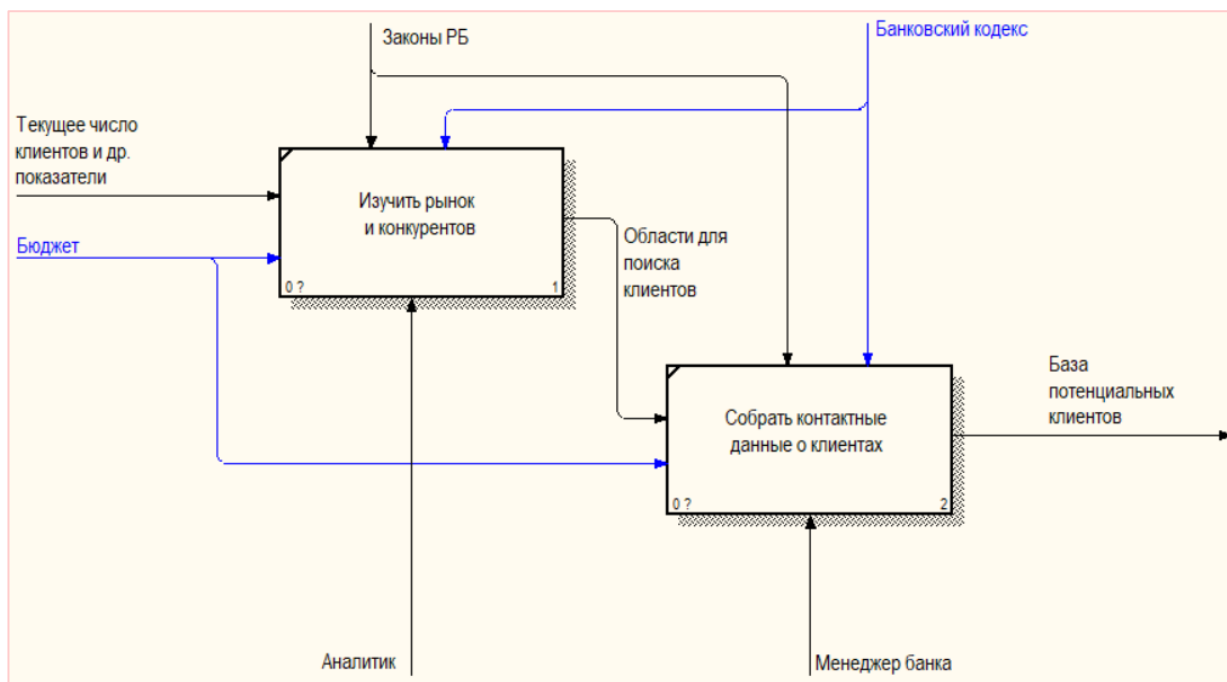


Рисунок 3.3 – Декомпозиция блока «Подготовить базу потенциальных клиентов»

После получение базы потенциальных клиентов можно проводить мероприятия по привлечению этих клиентов. Этот процесс может быть также декомпозирован на несколько этапов, что будет рассмотрено немного позже.

И заключительный этап – это проведение оценки результатов, сравнение новых показателей с исходными, которые были получены в результате анализа на первом шаге.

Вернемся к декомпозиции подпроцесса «Провести мероприятия по привлечению» (рисунок 3.4).

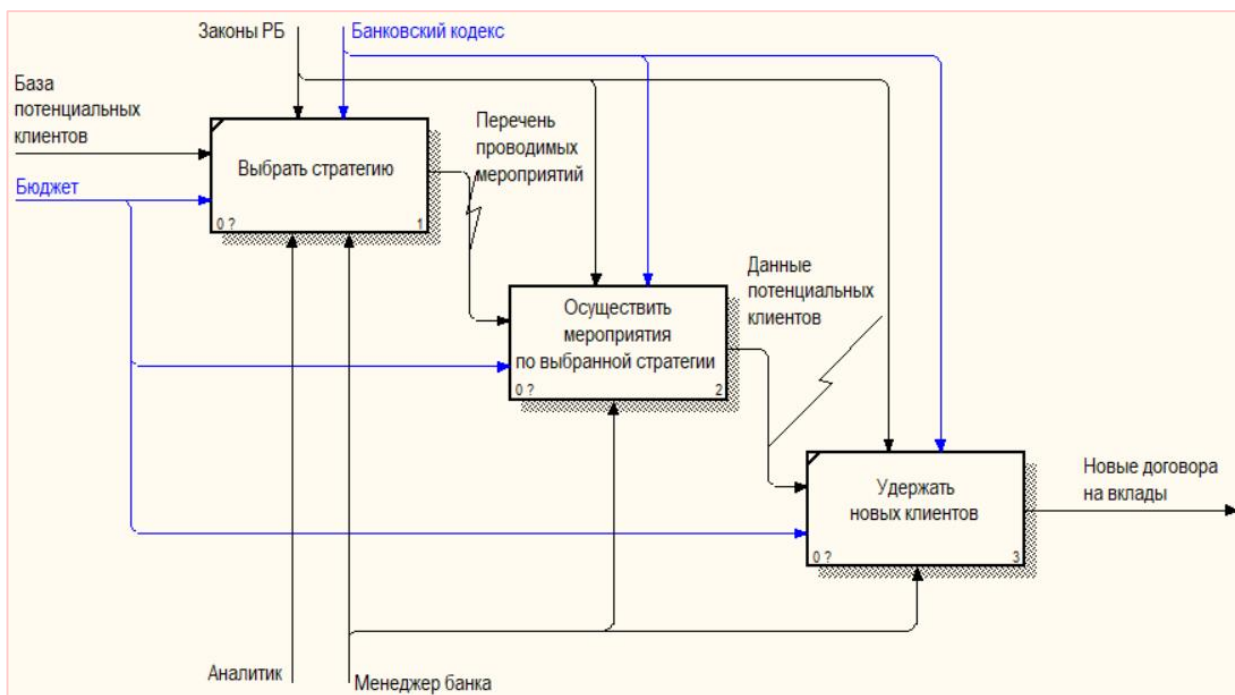


Рисунок 3.4 – Декомпозиция блока «Провести мероприятия по привлечению»

Было выделено 3 этапа для осуществления мероприятий.

Во-первых, выбрать стратегию. Имеется множество стратегий, среди которых можно выделить следующие:

- привлечение клиентов через давно и успешно обсуживающихся в данном банке;
- привлечение клиентов через определение следов успешной деятельности компаний в СМИ;
- привлечение клиентов через заказчиков и потребителей наиболее удачных, хорошо приспособленных к рынку продуктов, услуг, товаров;
- привлечение клиентов через информацию о проблемных, разваливающихся банках и перехвате их клиентов;
- привлечение клиентов путем направления потенциальным клиентам адресных писем-предложений (так называемая система direct-mail) [5].

Во-вторых, непосредственно осуществить выбранную стратегию, например, провести презентации.

И, наконец, необходимо удержать новых клиентов, чтобы они вошли базу.

В зависимости от выбранной стратегии на первом этапе второй может быть декомпозирован еще на два-три шага. В данной диаграмме были

выделены два наиболее общих для всех стратегий и представлены на диаграмме как показано на рисунке 3.5.

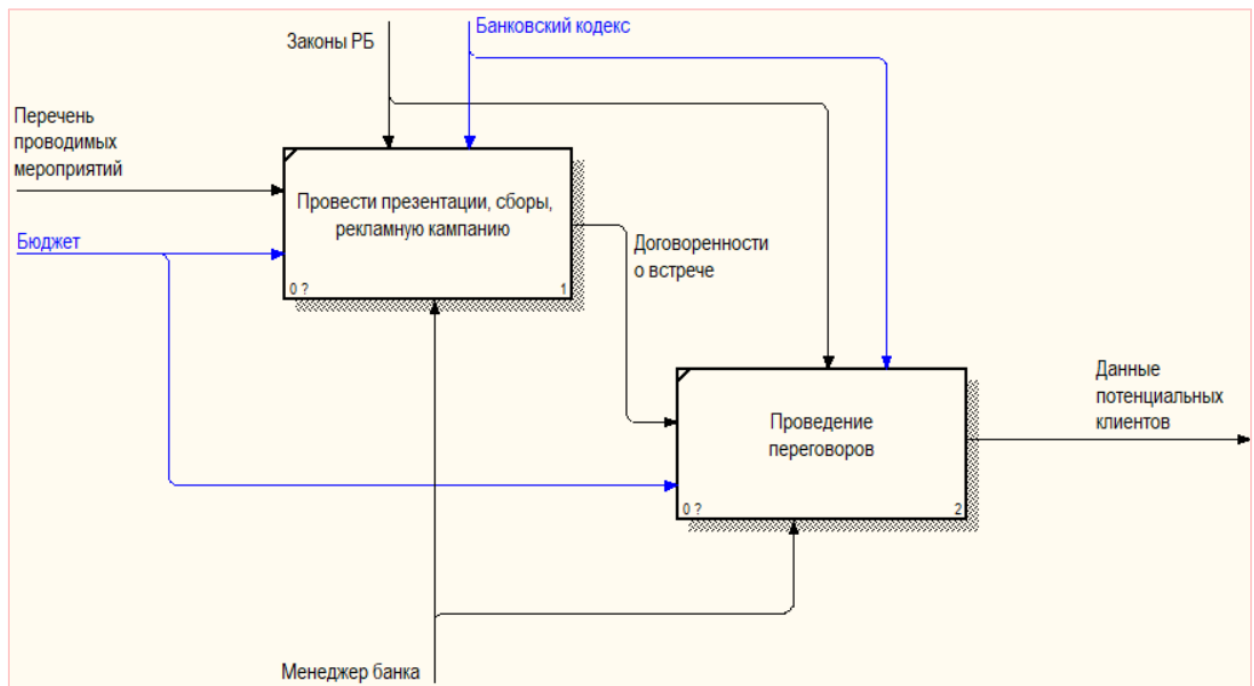


Рисунок 3.5 – Декомпозиция блока «Осуществить мероприятия по выбранной стратегии»

Первый шаг – это проведение каких-либо презентаций, рекламной кампании, по одной из стратегий рассылка предложений по e-mail.

Второй шаг – проведение переговоров с потенциальными клиентами. Как результат здесь можно получить данные клиентов, которые готовы сотрудничать с банком.

Таким образом данная диаграмма демонстрирует процесс поиска и привлечения новых клиентов, однако как правило большинство организаций, и банков в том числе, делают выбор в пользу удержания существующих клиентов, поскольку это оказывается менее затратно.

## 4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ ПО УЧЕТУ БАНКОВСКИХ ВКЛАДОВ

Схема базы данных, используемой разрабатываемым приложением, приведена на рисунке 4.1. Используемая база данных приведена к третьей нормальной форме, доказательство чего будет приведено далее.

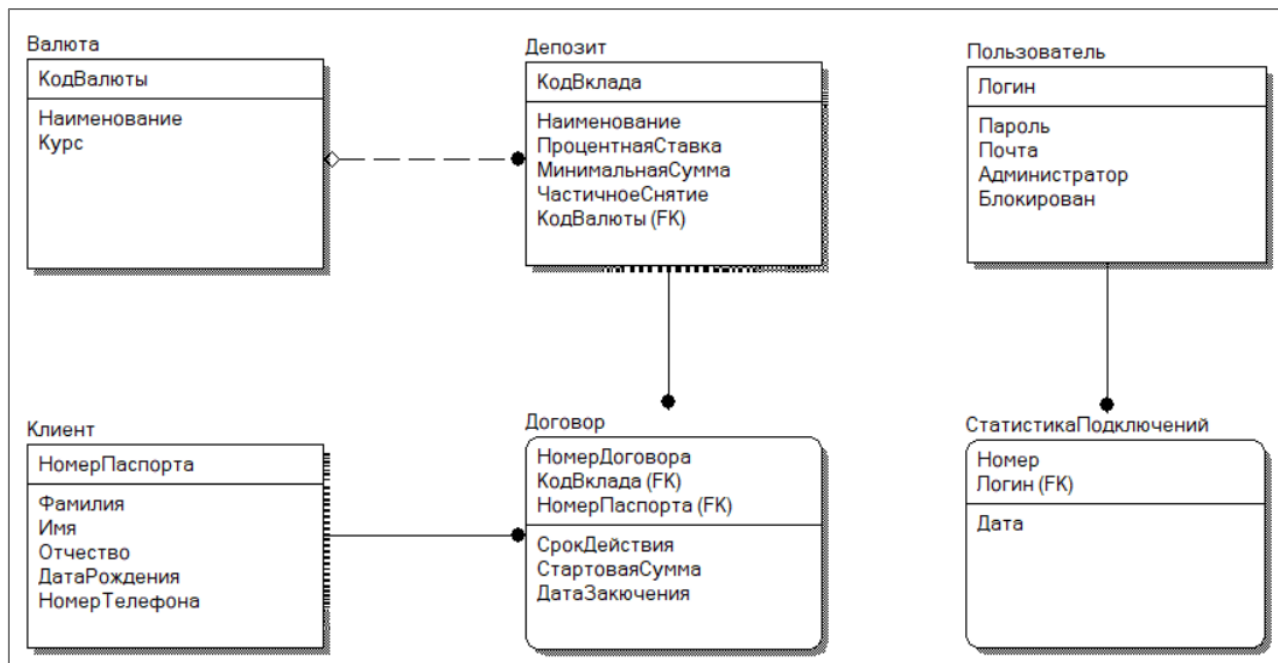


Рисунок 4.1 – Схема базы данных

В базе данных имеется шесть сущностей: «Валюта», «Депозит», «Клиент», «Договор», «Пользователь» и «СтатистикаПодключений».

Сущность «Пользователь» имеет такие атрибуты, как «Логин», «Пароль», «Адрес электронной почты», «Администратор» и «Блокирован». Значение атрибута «Администратор» может принимать два значения – «Да» или «Нет» – говорит о том, является ли пользователь с данной учетной записью администратором в системе, «Блокирован» – заблокирована данная учетная запись либо нет, в случае если аккаунт заблокирован, вход в систему будет невозможен, до тех пор, пока не будет изменено значение данного атрибута.

Сущность «СтатистикаПодключений» является вспомогательной, в ней содержатся записи о попытках входа в систему: логин пользователя и дата авторизации.

Сущность «Клиент» хранит все сведения обо всех клиентах банка и имеет следующие атрибуты:

- Номер паспорта;
- Фамилия;
- Имя;
- Отчество;
- Дата рождения;
- Номер телефона.

Каждая запись в сущности «Клиент» уникально идентифицируется значением атрибута «Номер паспорта».

Для сохранения данных о имеющихся депозитах в банке создана сущность «Депозит», она имеет такие атрибуты как:

- Код вклада;
- Наименование;
- Процентная ставка;
- Минимальная сумма;
- Частичное снятие – существует ли возможность снятия средств до окончания срока по данному вкладу;
- Код валюты – в какой валюте может быть открыт данный вклад.

По атрибуту «КодВалюты» к сущности «Депозит» присоединяется сущность «Валюта», с атрибутами, описывающими существующие валюты: мнемокод, наименование, курс.

Сущность «Договор» содержит записи о заключенных договорах на открытие вклада. Кроме атрибутов «НомерПаспорта» и «КодВклада», по которым идентифицируются клиент и вклад, есть также номер договора, срок действия, дата оформления и стартовая сумма вклада.

Далее перейдем к доказательству того, что представленная база данных приведена к третьей нормальной форме.

Нормализация – это процесс (процедура) приведения таблиц базы данных к ряду нормальных форм с целью избежания избыточности в базе данных, аномалий вставки, редактирования и удаления данных. Таблицы могут иметь неэффективную или не подходящую структуру, которую нужно нормализовать. Нормализация предусматривает разбивку исходной таблицы (отношения) на несколько новых таблиц (отношений) [3].

База данных считается правильно спроектированной (оптимальной или приближенной к оптимальной), если она отвечает требованиям нормальных

форм. Для небольших баз данных достаточно применения первых трех нормальных форм.

Чтобы база данных находилась в первой нормальной форме, необходимо чтобы выполнялись следующие условия:

- не должно быть дублирующих строк;
- в каждой ячейке хранится атомарное значение;
- в столбце хранятся данные одного типа;
- отсутствуют массивы и списки в любом виде.

Для нахождения базы данных во второй нормальной форме, ее таблицы должны удовлетворять следующим требованиям:

- таблица должна находиться в первой нормальной форме;
- таблица должна иметь ключ.

Чтобы нормализовать базу данных до третьей нормальной формы, необходимо, во-первых, привести ее ко второй нормальной и соответственно первой нормальной форме, во-вторых, сделать так, чтобы в таблицах отсутствовали неключевые столбцы, которые зависят от других неключевых столбцов [6].

Если проанализировать разработанную базу данных, то можно сделать вывод, что она находится в третьей нормальной форме: отсутствуют составные столбцы, есть первичный ключ и ни один неключевой столбец не зависит от другого.

## 5 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ ПО УЧЕТУ БАНКОВСКИХ ВКЛАДОВ ФИЗИЧЕСКИХ ЛИЦ

### 5.1 Спецификация вариантов использования системы

Диаграмма вариантов использования отражает все возможности в системе. Для разрабатываемого приложения Диаграмма вариантов использования приведена на рисунке 5.1.

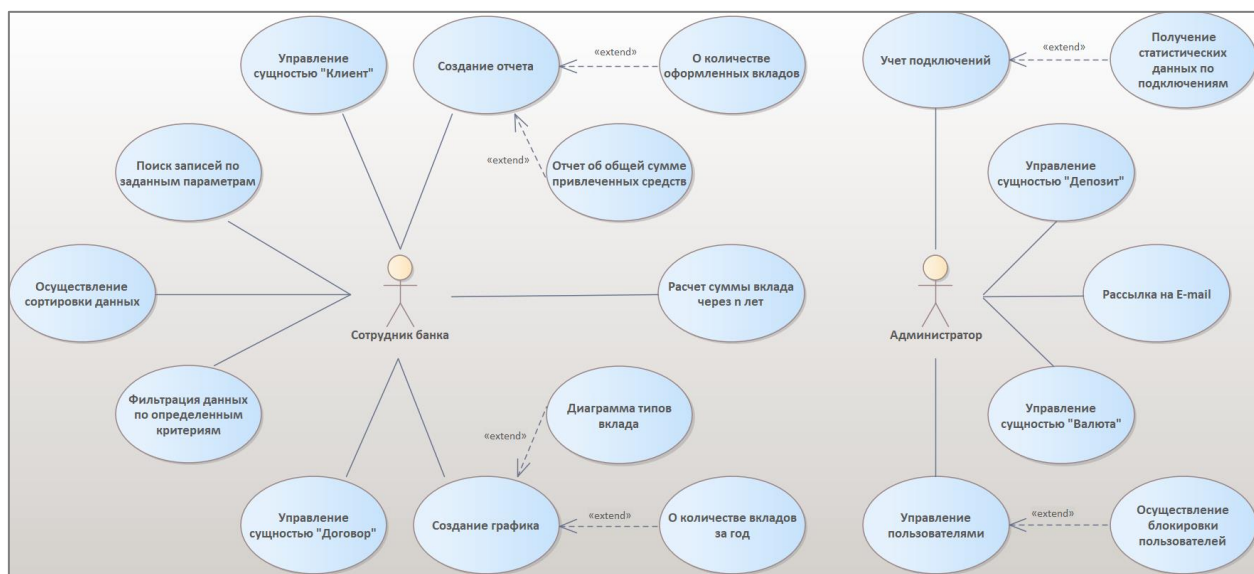


Рисунок 5.1 – Диаграмма вариантов использования в системе по управлению банковскими вкладами

В приложении имеются две роли: сотрудник банка и администратор. Спектр возможностей сотрудника банка намного шире, нежели у администратора – это достигается за счёт специфичных операций для сотрудника.

В первую очередь рассмотрим возможности администратора в системе. Среди них можно выделить такие как учет подключений, управление сущностью депозит и сущностью валюта, управление пользователями, рассылка на e-mail. Учет подключений может быть расширен за счет возможности получения статистических данных о подключениях. А управление пользователями предполагает осуществление блокировки учетных записей.

Далее перейдем к рассмотрению функционала для сотрудника.



Сотрудник банка имеет возможность управлять сущностями «Клиент» и «Договор», здесь под управлением понимается осуществление добавления, удаления, редактирования записей о клиентах и договорах. По имеющимся данным сотрудник может осуществлять сортировку, поиск и фильтрацию в соответствии с задаваемыми параметрами, например, отсортировать всех клиентов в алфавитном порядке либо просмотреть все договора, заключённые в определённый период.

На основе полученных данных в результате фильтрации, сортировки либо поиска сотрудник имеет возможность сгенерировать отчет, а также построить график.

При необходимости сотрудник также может рассчитать сумму вклада через определенное количество лет, по какой формуле производить расчёт – по формуле сложных процентов либо простых – выбирает сам сотрудник.

## 5.2 Описание диаграммы последовательности

Диаграмма последовательности, приведенная на рисунке 5.2, отражает течение процесса «Открыть вклад» во времени.

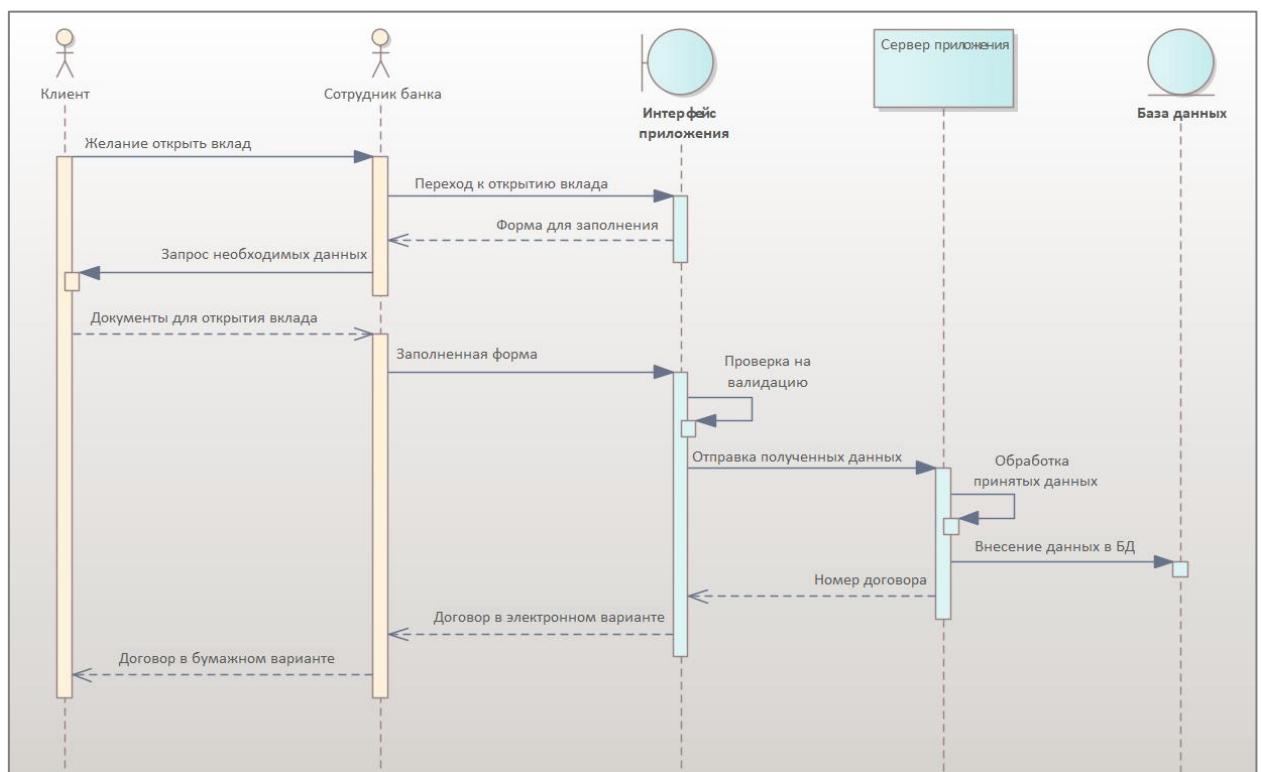


Рисунок 5.2 – Диаграмма последовательности

Процесса иницируются на выполнение, когда клиент выражает желание открыть вклад. Сразу после этого сотрудник банка выбирает в интерфейсе приложения соответствующий раздел. Далее ему предоставляется форма для заполнения, данные для которой он получает от клиента. Затем данные обрабатываются, происходит их проверка на валидацию. В случае успешного прохождения проверки данные отправляются на сервер, где происходит их дальнейшая обработка. После этого данные вносятся в базу данных. Результат обработки возвращается в интерфейс приложения, там формируется договор в электронном варианте. И на заключительном этапе договор в печатном варианте возвращается клиенту как результат процесса.

### 5.3 Описание диаграммы состояний

Диаграмма состояний, изображенная на рисунке 5.3, демонстрирует последовательность состояний, в которых будет находиться приложение в процессе открытия вклада.



Рисунок 5.3 – Диаграмма состояний

После открытия приложения программа находится в ожидании действий пользователя, при выборе необходимого действия, в данном случае – Открыть счет по вкладу, приложение переходит в следующее состояние – Предоставление формы. Пока пользователь осуществляет ввод данных программа находится в ожидании окончания этого. На следующем шаге осуществляется проверка на валидность. В зависимости от результатов данной проверки будет осуществлен либо запрос на повторный ввод, либо

произведена отправка данных на сервер. Если события будут развиваться по первому варианту, то после запроса на повторный ввод, программа вновь перейдет в состояние «Предоставление формы» и далее состояния повторяются. Если же – по второму, то программа после отправки перейдет в состояние ожидания ответа с сервера. Как только ответ будет получен последует состояние «Отображение результата». Заключительным, как и начальным, состоянием является «Ожидание запроса пользователя».

## 5.4 Описание диаграммы компонентов

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы. На рисунке 5.4 приведена диаграмма компонентов разрабатываемой системы.

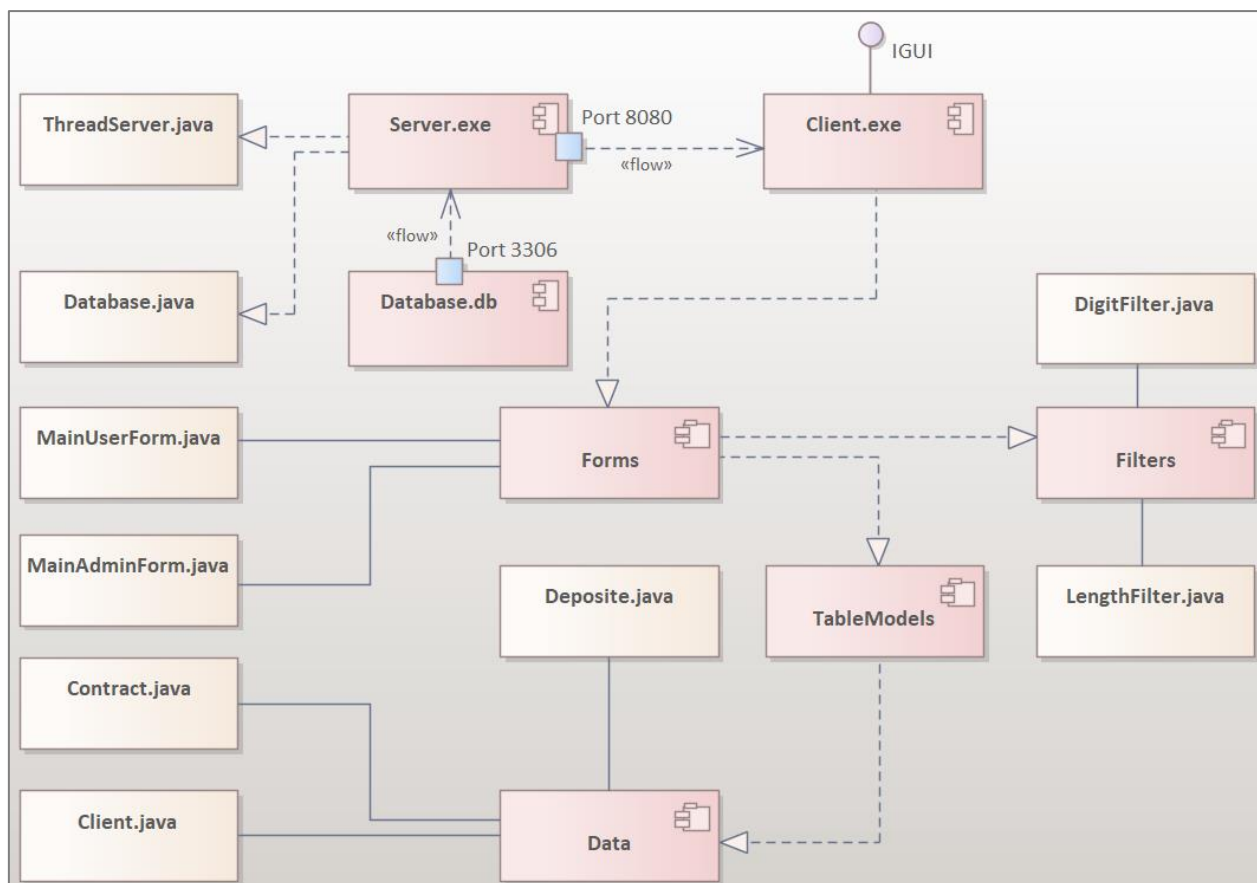


Рисунок 5.4 – Диаграмма компонентов

Система реализуется на трех компонентах: база данных, программа сервера и программ клиента. Клиентская программа реализуется на пакете классов Forms – MainAdminForm.java, MainUserForm.java.

В формах используются фильтры для ввода допустимых значений в поля, применение таких фильтров значительно снижает возможность появления некорректных данных в базе. Используемые классы фильтров – DigitFilter.java, LengthFilter.java. Реализуется пакет классов TableModels, содержащий классы для управления таблицами.

Клиентская программа взаимодействует с серверной через открытый порт 8080 на ней. Сервер реализуется на двух классах – ThreadServer.java и Database.java.

Для взаимодействия с базой данных используется локальный порт MySQL 3306.

## 5.5 Описание диаграммы развертывания

Диаграмма развертывания предназначена для описания аппаратной части системы. Она показывает на каком физическом оборудовании запускается та или иная составляющая программного обеспечения. Рисунок 5.5 демонстрирует диаграмму развертывания для разрабатываемого приложения.

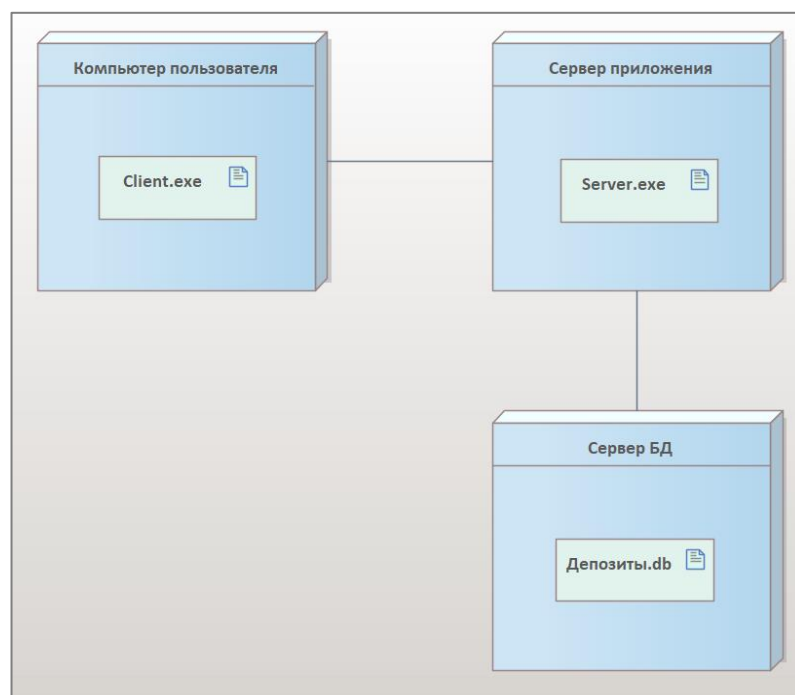


Рисунок 5.5 – Диаграмма развертывания

Приложение реализуется на трех компонентах, один из которых – это компьютер сотрудника банка. Для работы необходимо, чтобы на этот компьютер была установлена клиентская часть системы.

Узел «Сервер приложения» осуществляет обработку данных поступающих с компьютеров пользователей. На нем установлена серверная часть разрабатываемой системы.

Компонент «Сервер БД» осуществляет хранение данных пользователей и предоставляет эти данные по запросу сервера приложения.

## 5.6 Описание диаграммы классов

На рисунке 5.6 приведена диаграмма классов, реализованных в приложении с клиентской стороны.

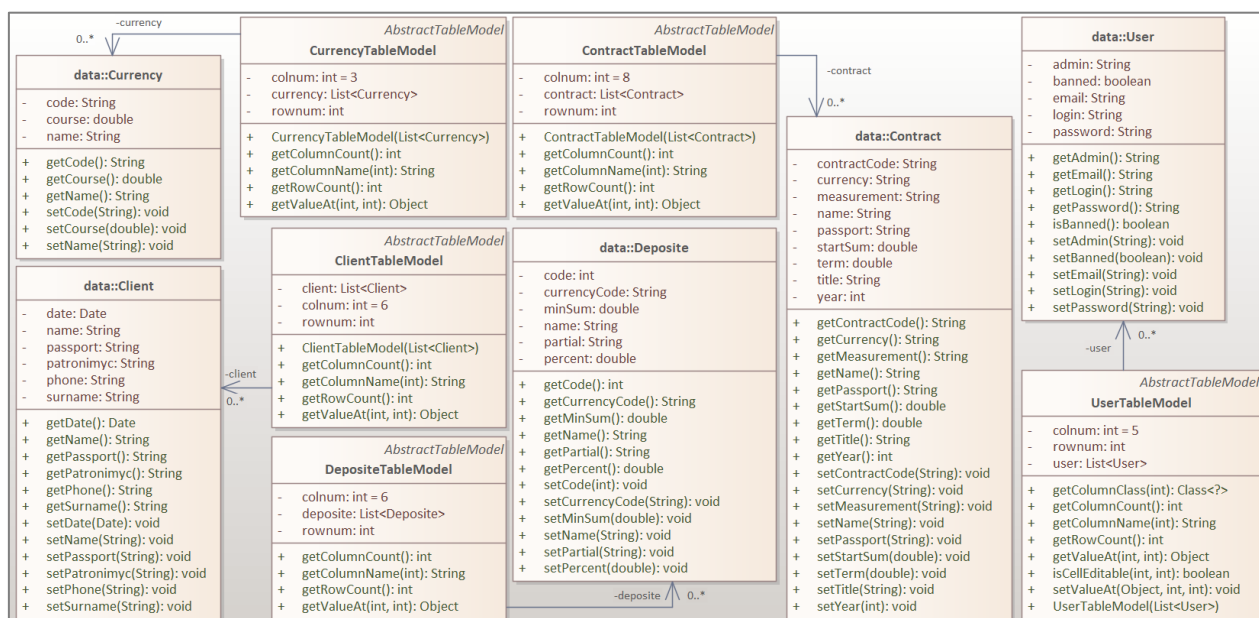


Рисунок 5.6 – Диаграмма классов, реализованных на клиенте

Представленная диаграмма отражает общую структуру иерархии классов приложения, которые реализуют основную логику программы, их атрибуты, методы и взаимосвязи.

Классы типа TableModel такие как ClientTableModel, DepositTableModel, ContractTableModel, CurrencyTableModel и UserTableModel расширяют абстрактный класс AbstractTableModel и созданы для осуществления представления данных из БД, их изменения. В свою

очередь они используют классы, которые представляют непосредственно сами данные: Client, Deposit, Contract, Currency и User.

Все классы содержат в себе конструкторы для начальной инициализации, а также необходимый набор геттеров и сеттеров. Классы представляющие модели таблиц переопределяют методы базового класса AbstractTableModel, например, метод получения данных из столбцов таблицы getValueAt(int row, int column).

Диаграмма классов, реализованных на серверной стороне приложения приведена на рисунке 5.7.

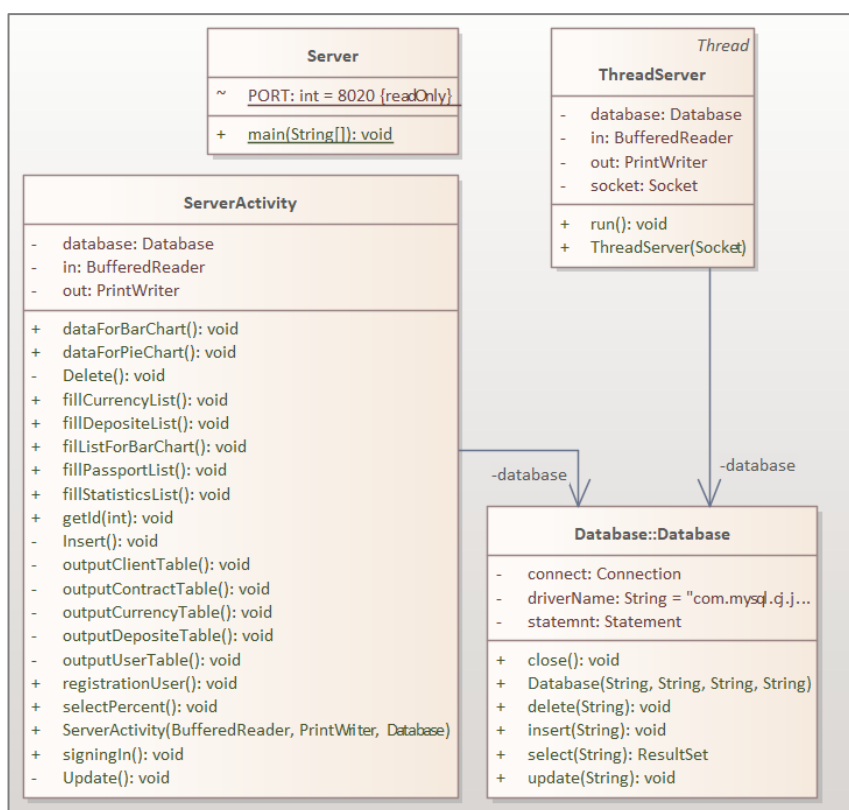


Рисунок 5.7 – Диаграмма классов, реализованных на сервере

Главный класс Server содержит в себе метод main – точку входа в программу. Класс Database отвечает за установление соединения с базой данных и инкапсулирует методы для взаимодействия с ней. ThreadServer расширяет класс Thread и реализует установку соединения с клиентом. Внутри метода run() создается объект класса ServerActivity, в котором реализован весь необходимый для работы с клиентом функционал.

## **6 ОПИСАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ БИЗНЕС-ЛОГИКУ СИСТЕМЫ ПО УЧЕТУ БАНКОВСКИХ ВКЛАДОВ**

В главе шесть приведены схемы и описания некоторых алгоритмов, которые реализуют функциональные возможности системы.

### **6.1 Схема алгоритма клиент-серверного взаимодействия**

На рисунке 6.1 приведена схема алгоритма, который описывает взаимодействие между клиентом и сервером. Прежде чем начнется обмен данными, необходимо установить соединение. В случае если соединение установить не удалось будет получено сообщение об ошибке. В случае успешной установки соединения клиент имеет возможность выбрать операцию. В зависимости от его выбора ему предоставляется форма. На следующем шаге приложение собирает данные из формы и проверяет их валидность. В случае неуспешного прохождения проверки будет получено сообщение об ошибке. Если же данные прошли проверку, происходит их пересылка на сервер. Когда данные будут обработаны сервер вернет результат на клиента. В зависимости от того вернет сервер корректные либо некорректные данные пользователю будет предоставлено либо сообщение об ошибке, либо результаты выполнения операции.

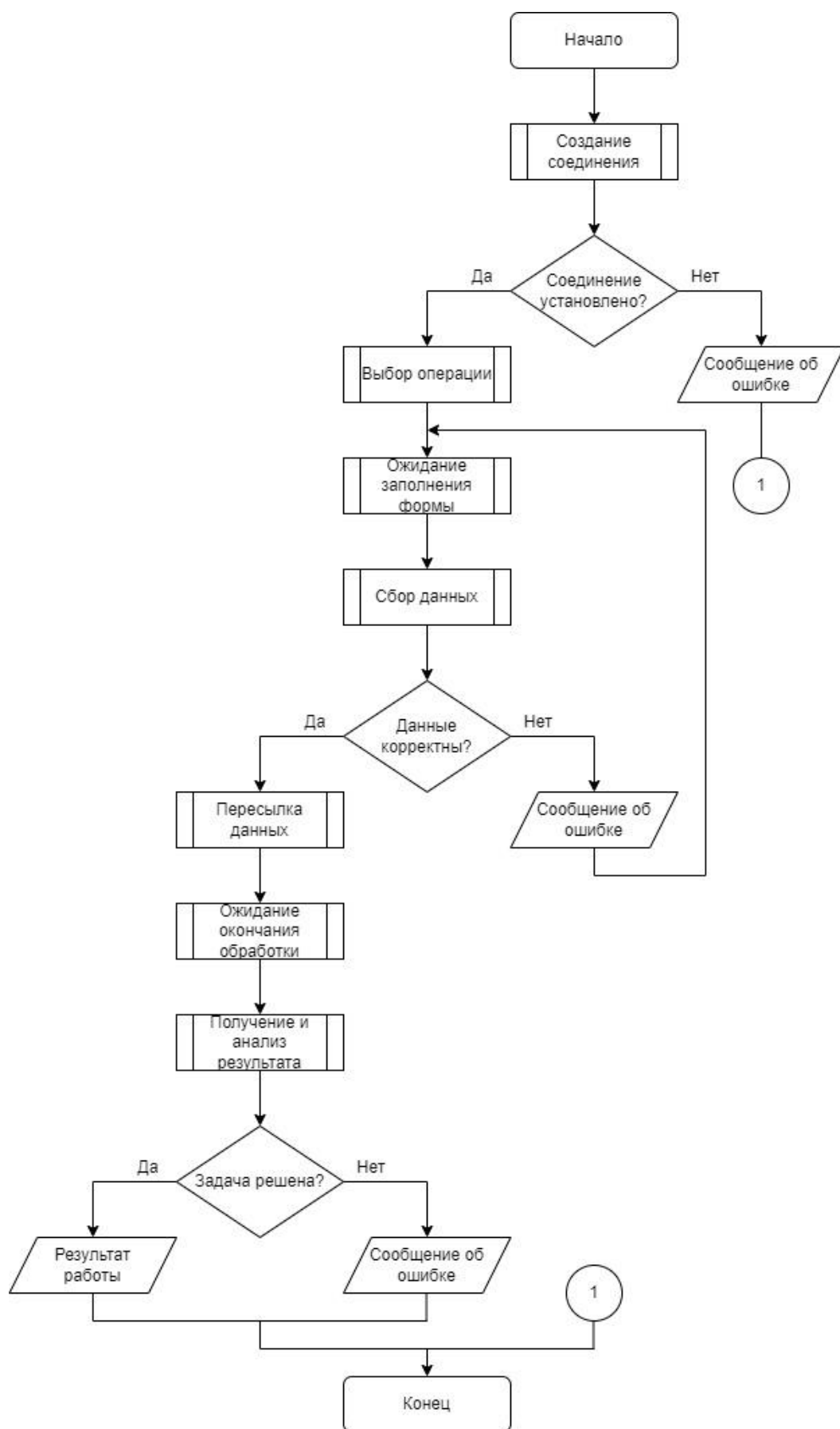


Рисунок 6.1 – Схема клиент-серверного взаимодействия



## 6.2 Схема алгоритма расчета суммы по вкладу

Расчет суммы по вкладу через определенный срок происходит в соответствии с алгоритмом, схема которого приведена на рисунке 6.2.

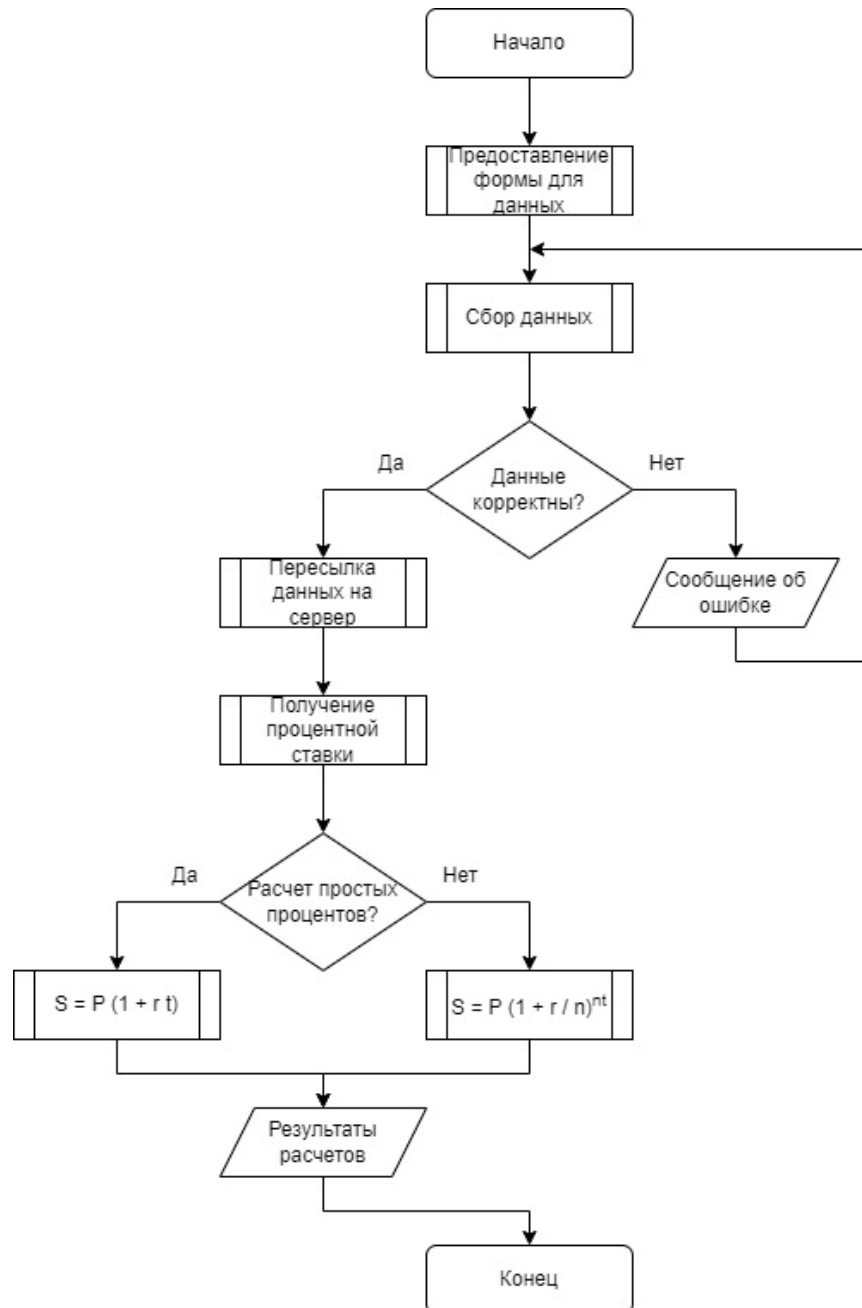


Рисунок 6.2 – Схема алгоритма расчета суммы вклада через определенный срок

В первую очередь происходит сбор данных необходимых для расчетов и их проверка на валидность. Если данные корректны, на сервер отправляется

запрос на получение дополнительных сведений, в частности процентной ставки. После получения этих данных в соответствии с выбором пользователя производится расчет суммы по одной из формул – по формуле простых процентов или по формуле сложных процентов. Результаты расчетов выводятся пользователю.

### 6.3 Схема алгоритма построения графика

На рисунке 6.3 приведена схема алгоритма построения графика в системе по исходным параметрам.

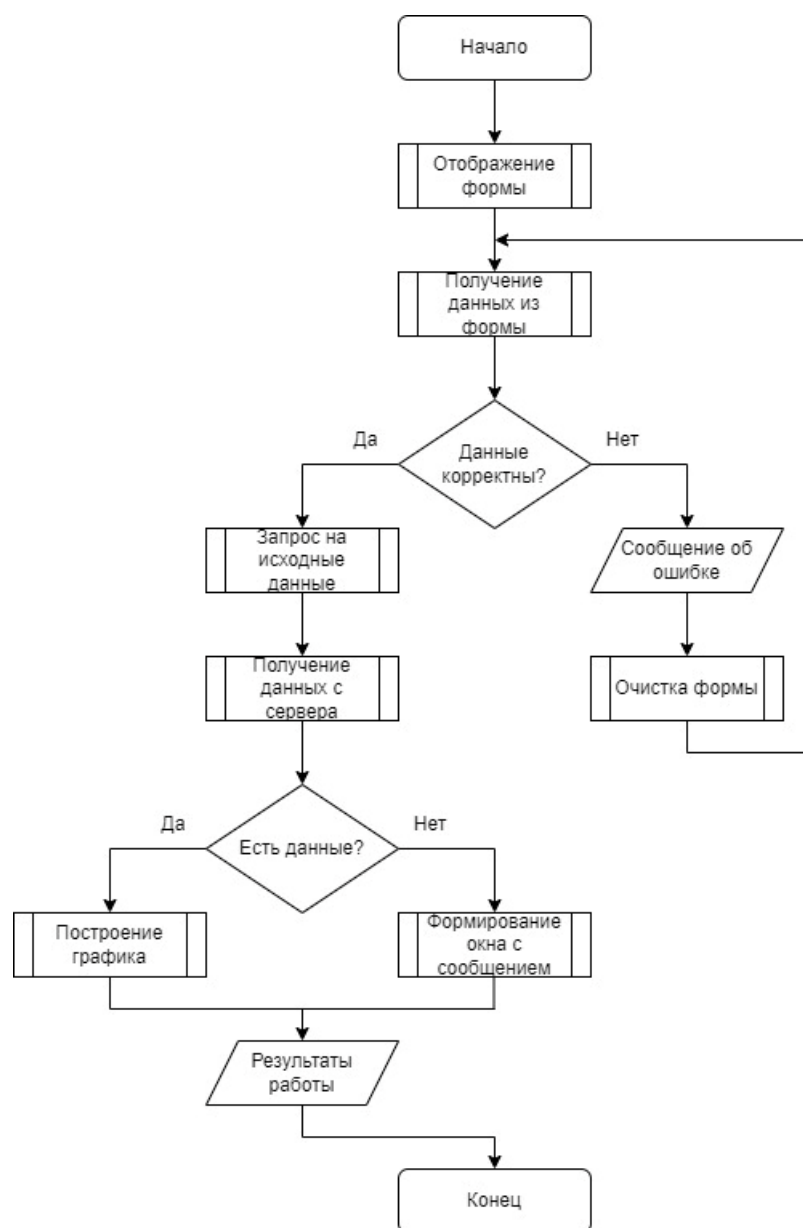


Рисунок 6.3 – Схема алгоритма построения графика

Как и в алгоритме расчета суммы по вкладу здесь в первую очередь происходит сборка параметров для выполнения операции, например, получение периода, за который должны учитываться данные, и проверка данных на валидность. Далее, в случае успешной проверки, с сервера запрашивается набор данных подходящий под указанные параметры. Если такой набор данных существует, происходит построение графика, если нет отображается соответствующее сообщение.

Таким образом, рассмотренные выше схемы алгоритмов демонстрируют некоторую логику работы программы. Аналогично описанным в данной главе алгоритмам, могут быть рассмотрены и алгоритмы работы других функций, которые выполняет разработанное приложение.

## 7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данный раздел представляет собой руководство по эксплуатации разработанной системы.

Для корректной работы приложения в первую очередь должен быть запущен сервер, где происходит подключение к базе данных и настройка корректной работы в целом. Как только сервер начал работу при подключении/отключении клиентов, при подключении к базе данных или при возникающих ошибках соответствующая этому информация отображается в консоли (пример на рисунке 7.1).

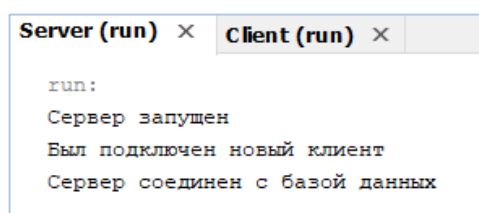


Рисунок 7.1 – Состояния сервера

После этого могут быть запущены приложения-клиенты. Запуск приложения-клиента может осуществляться как двойным кликом по Client.jar. Так и через интегрированную среду разработки.

После запуска приложения пользователю отображается окно с авторизацией, как показано на рисунке 7.2.

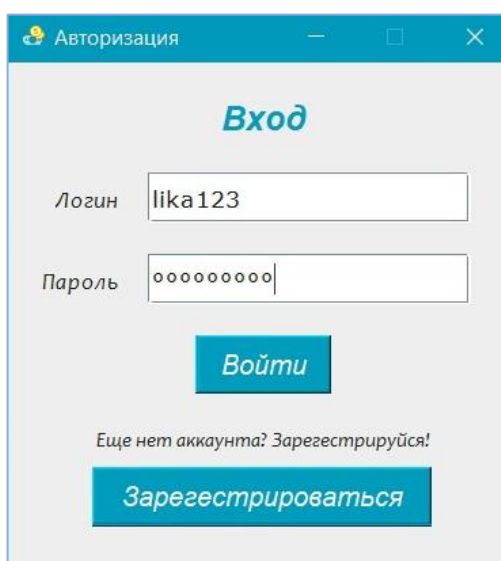
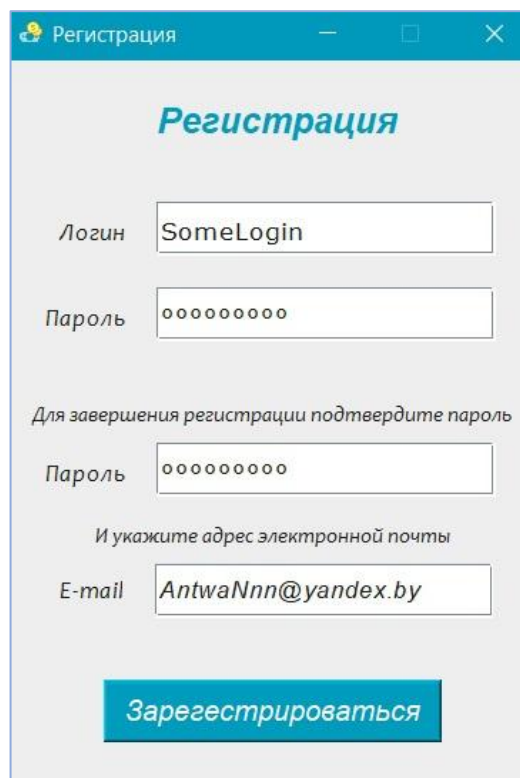


Рисунок 7.2 – Окно авторизации

Если данный пользователь уже зарегистрирован в системе, он просто осуществляет стандартный вход, если данный пользователь использует систему впервые и еще не имеет учетной записи, ему необходимо пройти регистрацию. Окно регистрации представлено на рисунке 7.3.



The image shows a web browser window titled "Регистрация" (Registration). The page has a light gray background with a blue header bar. The title "Регистрация" is displayed in a large, bold, blue font. Below the title, there are several input fields and labels. The first field is labeled "Логин" (Login) and contains the text "SomeLogin". The second field is labeled "Пароль" (Password) and contains ten dots. Below this, a text label reads "Для завершения регистрации подтвердите пароль" (To complete registration, confirm your password). This is followed by another "Пароль" (Password) field, also containing ten dots. Below that, a text label reads "И укажите адрес электронной почты" (And specify your email address). The next field is labeled "E-mail" and contains the text "AntwaNnn@yandex.by". At the bottom of the form is a large blue button with the white text "Зарегистрироваться" (Register).

Рисунок 7.3 – Окно регистрации

После заполнения формы необходимыми данными пользователь может авторизоваться в системе. Если логин и пароль, введенные пользователями, не заняты, будет выведено сообщения об успешном прохождении регистрации как показано на рисунке 7.4. В противном случае программа сообщит, что данный логин или пароль уже заняты.

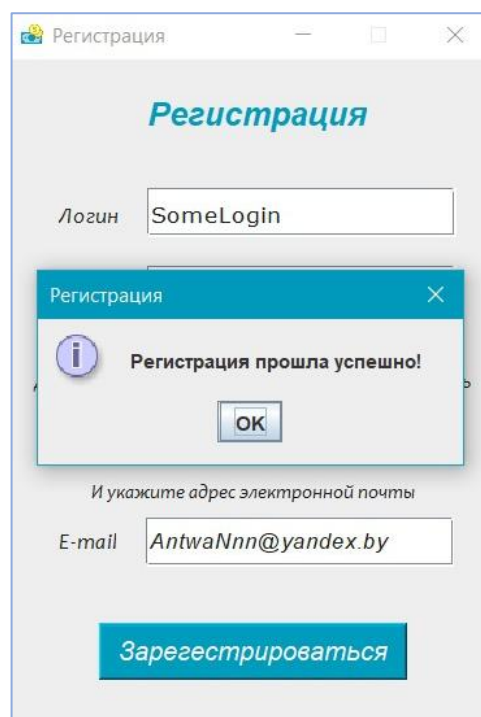


Рисунок 7.4 – Подтверждение о прохождении регистрации

Главный экран для пользователей имеет вид, приведенный на рисунке 7.5.

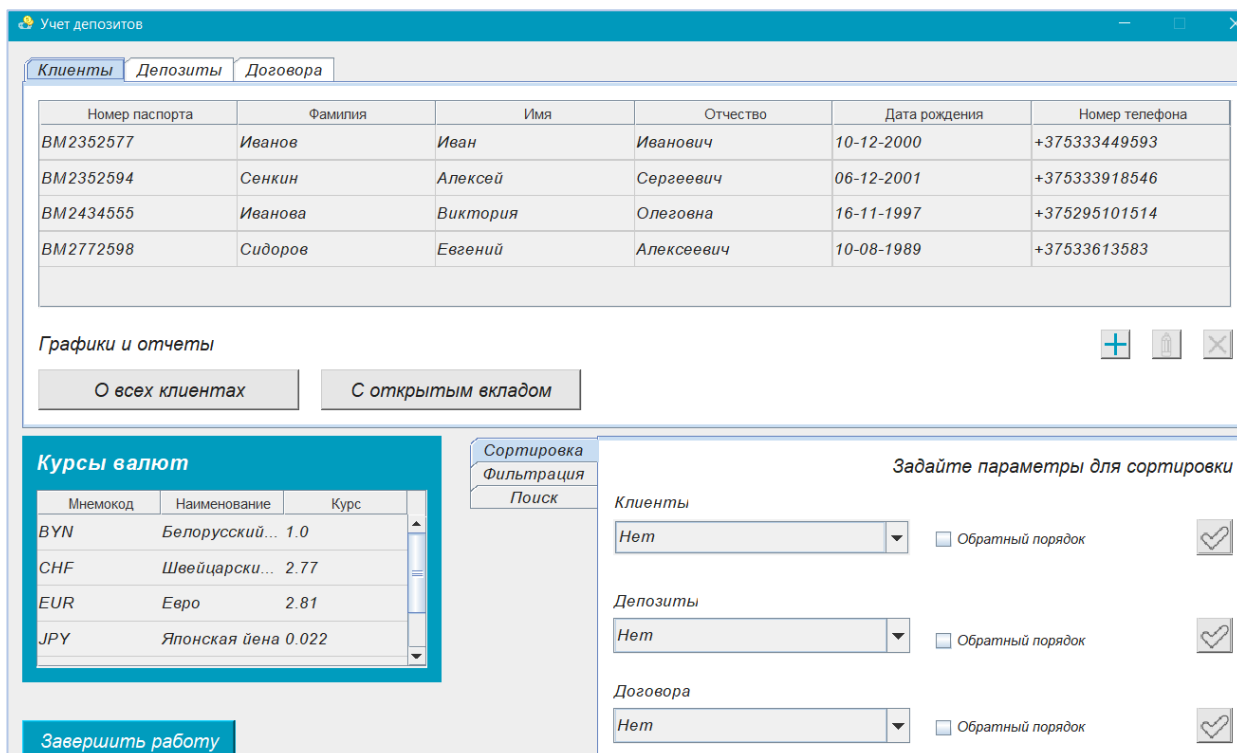


Рисунок 7.4 – Главное окно для пользователя

В верхней части окна расположена панель с таблицами, содержащими сведения о клиентах, депозитах и договорах на вклады, имеющиеся в базе данных. Кроме непосредственно таблиц на панели также расположены элементы управления для работы с данными, например, кнопки «О всех клиентах» и «С открытым вкладом» – это кнопки для генерации отчетов об общем числе клиентов и числе клиентов на которых открыт вклад соответственно, кнопки добавления, удаления и редактирования данных. Кнопки удаления и редактирования по умолчанию недоступны, чтобы активировать их, необходимо выбрать запись в таблице.

Для депозитов договоров имеются аналогичные панели (рисунок 7.5).

**Клиенты** | **Депозиты** | **Договора**

Код	Наименование	Процентная ставка	Минимальная сумма	Частичное снятие	Валюта
1	Традиционный	7.3	300.0	Нет	RUB
2	Традиционный	7.3	300.0	Нет	BYN
3	Традиционный	7.3	300.0	Нет	USD
4	Традиционный	7.3	300.0	Нет	EUR
8	Вкусный	14.3	350.0	Да	BYN

Графики и отчеты

Общее число вкладов | Отчет по вкладу | Кол-во по валюте

**Клиенты** | **Депозиты** | **Договора**

Номер договора	ФИО	Номер паспорта	Наименование	Валюта	Стартовая сумма	Срок действия	Год заключения
2151301157	Иванов Иван Ие...	BM2352577	Традиционный	RUB	480.0	5.0 г.	2020
2151721154	Иванов Иван Ие...	BM2352577	Традиционный	BYN	475.0	5.0 г.	2020
9999999999	Сенкин Алексей	BM2352594	Традиционный	RUB	320.0	24.0 мес.	2021
1000000002	Иванова Викто...	BM2434555	Вкусный	EUR	400.0	4.0 г.	2021
5555555555	Иванова Викто...	BM2434555	Традиционный	RUB	510.0	18.0 мес.	2021

Графики и отчеты

Отчет | Кол-во в год | Доля вклада | Сумма по вкладу

Рисунок 7.5 – Таблицы «Депозит» и «Договор»

Для таблицы «Депозит» существует возможность создания двух типов отчетов – об общем числе вкладов в банке и подробный отчет по определенному вкладу. Кроме этого, можно построить график, на котором отражается количество депозитов доступных каждой валюте. Окна с отчетами представлены на рисунке 7.6, а с графиком – на рисунке 7.7.

### Отчет

О предоставляемых депозитах банком

На текущую дату 10.12.2021 общее число депозитов составляет 8  
В их числе

- 1 Традиционный 7.3 300.0
- 2 Традиционный 7.3 300.0
- 3 Традиционный 7.3 300.0
- 4 Традиционный 7.3 300.0
- 8 Вкусный 14.3 350.0
- 9 Вкусный 14.3 350.0
- 10 Вкусный 14.3 350.0
- 11 Вкусный 14.3 350.0

Сохранить

### Отчет

О предоставляемых депозитах банком

На текущую дату 10.12.2021 общее число депозитов составляет 8  
С возможностью частичного снятия 4  
В их числе

- 8 Вкусный 14.3 350.0
- 9 Вкусный 14.3 350.0
- 10 Вкусный 14.3 350.0
- 11 Вкусный 14.3 350.0

А также без возможности частичного снятия 4  
В их числе

- 1 Традиционный 7.3 300.0
- 2 Традиционный 7.3 300.0
- 3 Традиционный 7.3 300.0
- 4 Традиционный 7.3 300.0

Сохранить

Рисунок 7.6 – Отчеты по депозитам

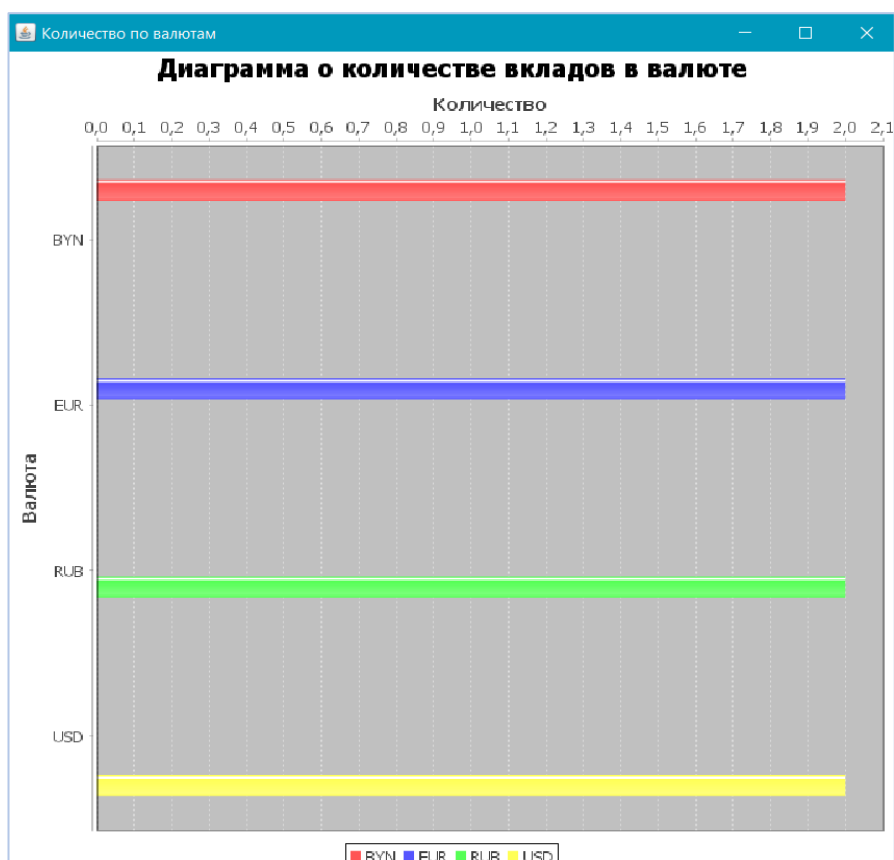


Рисунок 7.7 – График по депозитам



Чтобы сохранить отчет в файл и иметь возможность вывести его на печать, необходимо кликнуть по кнопке «Сохранить», затем система предложит ввести имя целевого файла – файла, куда будет осуществлен вывод (окно приведено на рисунке 7.8).

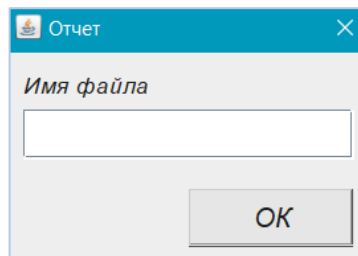


Рисунок 7.8 – Окно ввода имени файла

Далее рассмотрим возможности работы с таблицей «Договор». Как и для депозитов для договоров предусмотрена функция генерации отчета и создания графиков. Результат вывода отчета можно видеть на рисунке 7.9.

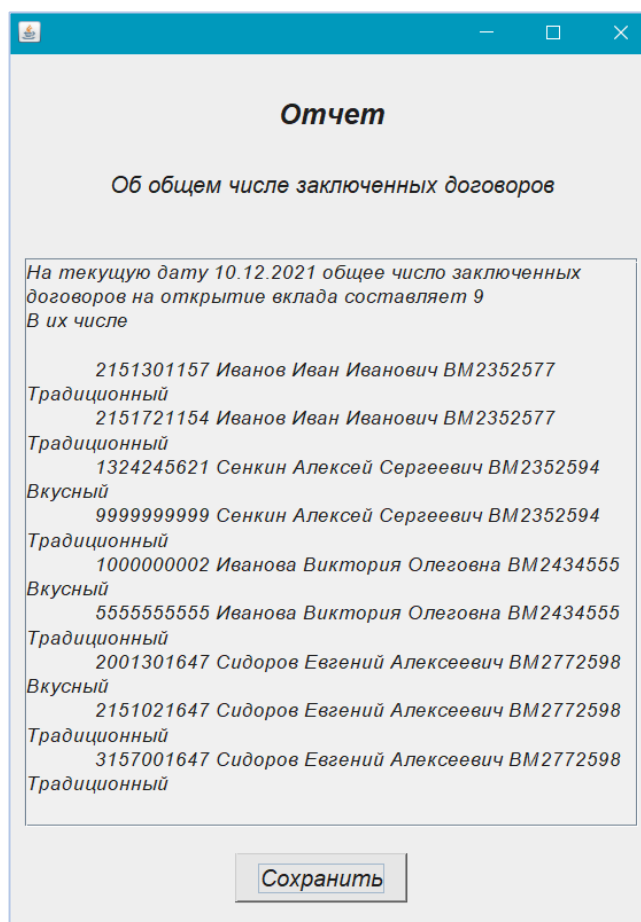


Рисунок 7.9 – Отчет по договорам

Графики для договоров предусмотрены в двух типах – столбчатая диаграмма (гистограмма) и круговая диаграмма.

Гистограмма отражает общее число договоров, заключенных за определенный год. Система также позволяет задать какие-либо определенные типы депозита и период для анализа. Пример гистограммы по всем данным за все года и по всем договорам, заключенным на вклад «Традиционный» в период с 2018 года по 2021 год, приведен на рисунке 7.10.

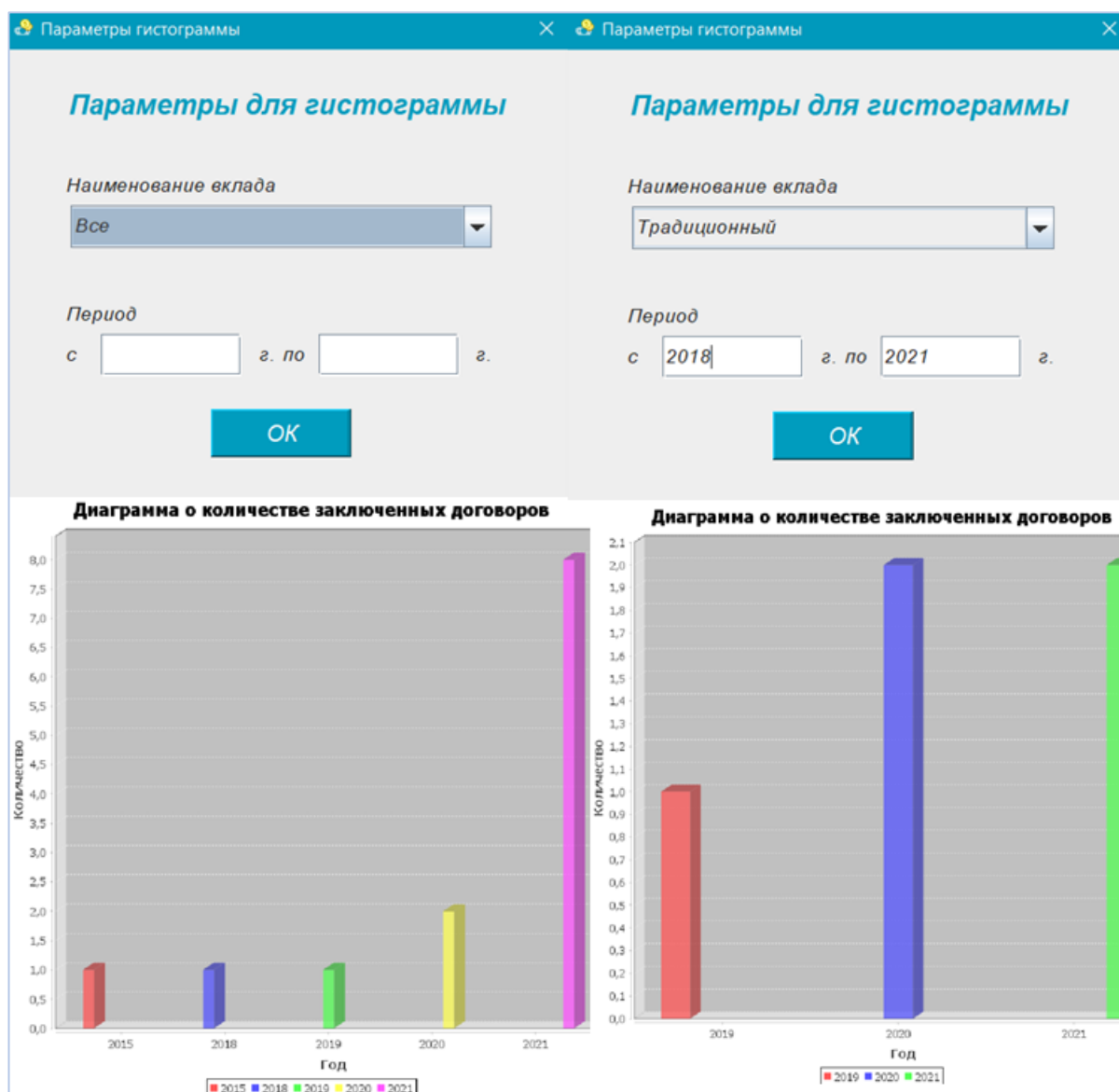


Рисунок 7.10 – Гистограммы

Круговая диаграмма показывает, сколько договоров на открытие вклада определённого вида было заключено в отношении к общему числу, расчёт может производиться как в абсолютных, так и в относительных величинах.

Пример построения круговой диаграммы в абсолютных единицах приведен на рисунке 7.11.



Рисунок 7.11 – Круговая диаграмма

Кроме этого, в соответствии с заданием были реализованы тривиальные операции для работы с данными, такие как добавления, удаление и редактирование. Как говорилось ранее, кнопки удалить и редактировать доступны только после выбора строки в таблице.

Добавление и редактирование осуществляется в соответствующей для этого форме, для обеих этих операций используются одна и та же форма, за исключением того, что для редактирования данные переносятся на форму из выбранной строки таблицы.

Пример добавления записи о клиенте и договоре приведен на рисунке 7.12.

The image shows two side-by-side windows from a software application. Both windows have a title bar with a close button (X) and a maximize button. The left window is titled 'Клиент' (Client) and the right window is titled 'Договор' (Contract). Both windows have a subtitle 'Добавить новые данные' (Add new data). The left window contains the following fields: 'Номер паспорта' (Passport number) with value 'BM1458769', 'Фамилия' (Surname) with value 'Снопок', 'Имя' (Name) with value 'Лика', 'Отчество' (Patronymic) with value 'Александровна', 'Дата рождения' (Date of birth) with value '10.02.2002', and 'Номер телефона' (Phone number) with value '+375 333619593'. The right window contains the following fields: 'Номер договора' (Contract number) with value '1324245621', 'Паспорт клиента' (Client passport) with value 'BM2352594:Сенкин Алексей Сергеевич', 'Депозит' (Deposit) with value '10. Вкусный EUR', 'Стартовая сумма' (Starting sum) with value '1600', 'Срок действия' (Term of validity) with value '3.5', and 'Год заключения' (Year of conclusion) with value '2021'. Both windows have an 'OK' button at the bottom.

Рисунок 7.12 – Добавление Клиента и Договора

В случае успешного завершения операции система сообщает об этом пользователю.

Удаления записи осуществляется кликом по кнопке «Удалить», после чего будет выведено диалоговое окно вида, представленного на рисунке 7.13.

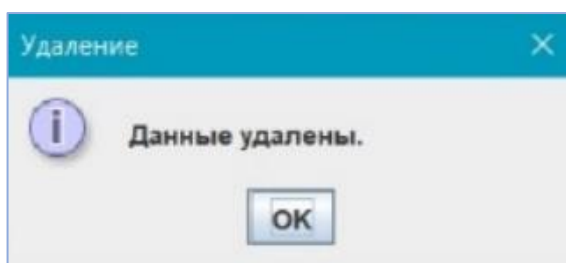


Рисунок 7.13 – Удаление записи

В Нижнем левом углу расположена таблица с данными по валютам (рисунок 7.14), в ней отражены такие сведения, как мнемокод, полное наименование и курс на текущий момент.

Курсы валют		
Мнемокод	Наименование	Курс
BYN	Белорусский...	1.0
CHF	Швейцарски...	2.77
EUR	Евро	2.81
JPY	Японская йена	0.022

Рисунок 7.14 – Курсы валют

Правее таблицей валют, размещается панель для задания параметров сортировки, фильтрации и поиска (смотреть рисунок 7.15).

Задайте параметры для сортировки

Клиенты

Нет

Обратный порядок

✓

Депозиты

Нет

Обратный порядок

✓

Договора

Нет

Обратный порядок

✓

Задайте параметры для фильтрации

Клиенты

Нет

Введите значение

✓

Депозиты

Нет

Введите значение

✓

Договора

Нет

Введите значение

✓

Задайте параметры для поиска

Клиенты

Нет

Введите значение

✓

Депозиты

Нет

Введите значение

✓

Договора

Нет

Введите значение

✓

Рисунок 7.15 – Панели сортировки, фильтрации и поиска

Сортировка записей таблицы осуществляется по следующим полям:

- клиенты – номер паспорта, фамилия, имя, отчество, дата рождения;
- депозиты – код, наименование, процентная ставка, минимальная сумма, валюта;
- договора – номер договора, ФИО клиента, номер паспорта, наименование вклада, валюта, срок действия, стартовая сумма.

Фильтрация производится по полям:

- клиенты – фамилия, имя, отчество;
- депозиты – наименование, валюта;
- договора – ФИО клиента, наименование вклада, валюта.

Для осуществления поиска необходимо выбрать одно из полей для таблицы в соответствии со следующим списком:

- клиенты – номер паспорта, фамилия, имя, отчество;
- депозиты – код, наименование;
- договора – номер договора, наименование вклада, номер паспорта, ФИО клиента.

Выбрав определённую запись в таблице «Договор», можно произвести расчёт суммы по вкладу через определенный период. Параметры для расчета, такие как срок начисления процентов, способ расчета (формула сложных процентов или формула простых процентов) устанавливаются в форме, как изображено на рисунке 7.16.

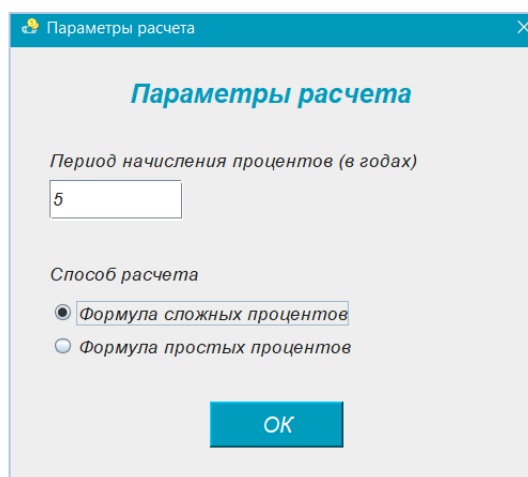


Рисунок 7.16 – Параметры расчета суммы по вкладу

По результатам расчета выводится диалоговое окно подобное тому, что представлено на рисунке 7.17.

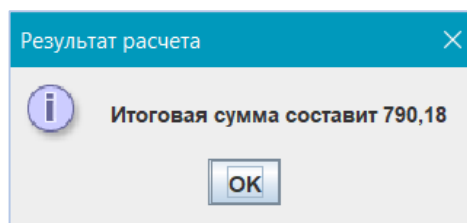


Рисунок 7.17 – Результаты расчета

На этом функциональные возможности для обычного пользователя закончены. Далее перейдем к рассмотрению системы со стороны администратора, чтобы произвести вход от имени администратора, необходимо ввести логин Administrator и пароль Administrator. После этого выводится главный экран вида как на рисунке 7.18.

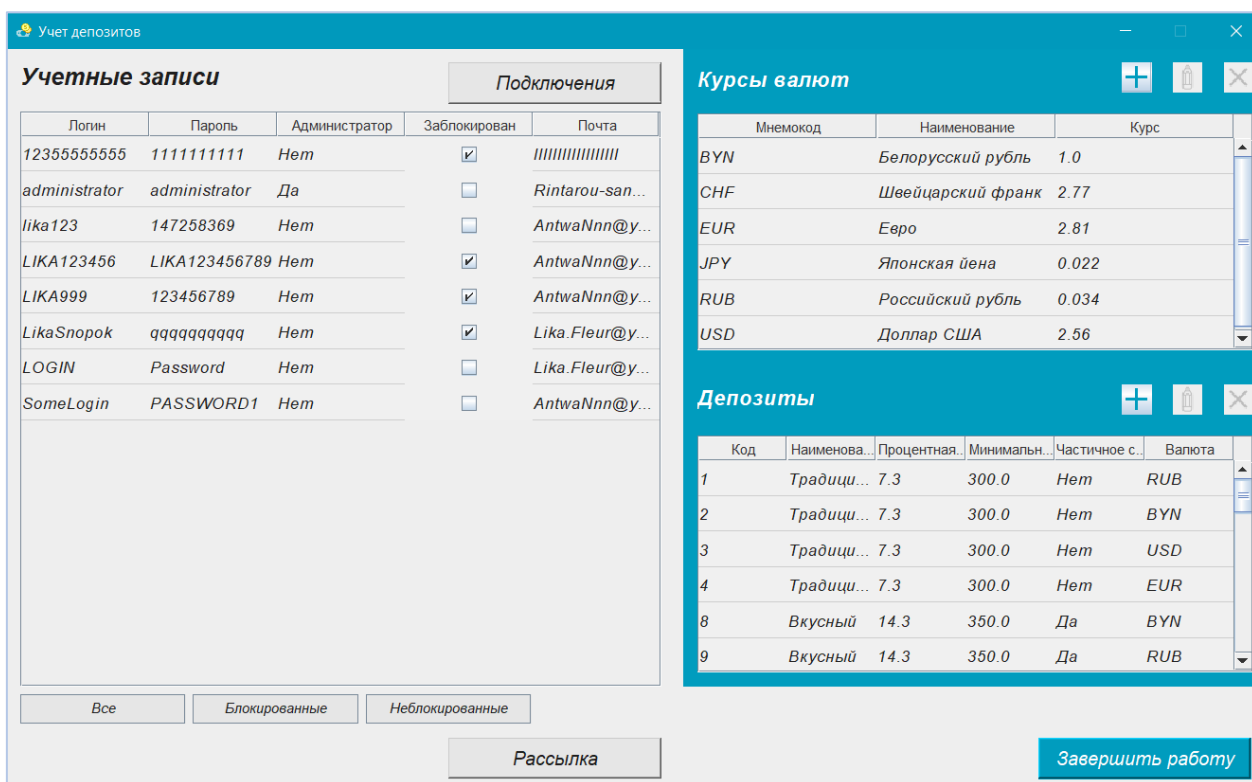


Рисунок 7.18 – Главное окно администратора

Справа на форме расположены таблицы валют и депозитов. В отличие от обычного пользователя, у администратора есть возможность вносить изменения в эти таблицы, которые будут отображены и у пользователя соответственно. Пример редактирования записей в этих таблицах приведен на рисунке 7.19.

Валюта

Депозит

Изменить данные

Мнемокод

CHF

Наименование

Швейцарский франк

Курс

2.77

OK

Редактировать

Код

2

Наименование

Традиционный

Процентная ставка

7

Минимальная сумма

300

Частичное снятие

☐ Да ☒ Нет

Валюта вклада

BYN Белорусский рубль

OK

Редактировать

Код

2

Наименование

Традиционный

Процентная ставка

7

Минимальная сумма

300

Частичное снятие

☐ Да ☒ Нет

Валюта вклада

ВУН Белорусский рубль

OK

Рисунок 7.19 – Редактирование данных

Слева размещается таблица с учетными записями всех пользователей (смотреть рисунок 7.21).

Учетные записи			Подключения	
Логин	Пароль	Администратор	Заблокирован	Почта
12355555555	1111111111	Hem	<input checked="" type="checkbox"/>	IIIIIIIIIIIIIIIIIIII
administrator	administrator	Да	<input type="checkbox"/>	Rintarou-san...
Iika123	147258369	Hem	<input type="checkbox"/>	AntwaNnn@y...
LIKA123456	LIKA123456789	Hem	<input checked="" type="checkbox"/>	AntwaNnn@y...
LIKA999	123456789	Hem	<input checked="" type="checkbox"/>	AntwaNnn@y...
LikaSnopok	qqqqqqqqqq	Hem	<input checked="" type="checkbox"/>	Lika.Fleur@y...
LOGIN	Password	Hem	<input type="checkbox"/>	Lika.Fleur@y...
SomeLogin	PASSWORD1	Hem	<input type="checkbox"/>	AntwaNnn@y...

Все
Блокированные
Неблокированные

Рисунок 7.20 – Учетные записи пользователей

В колонке «Заблокирован» можно изменять значение, если флажок установлен, пользователь заблокирован и вход в систему с данной учетной записи не может быть осуществлен. При попытке входа с такой учетной записи будет получено предупреждение (рисунок 7.21).



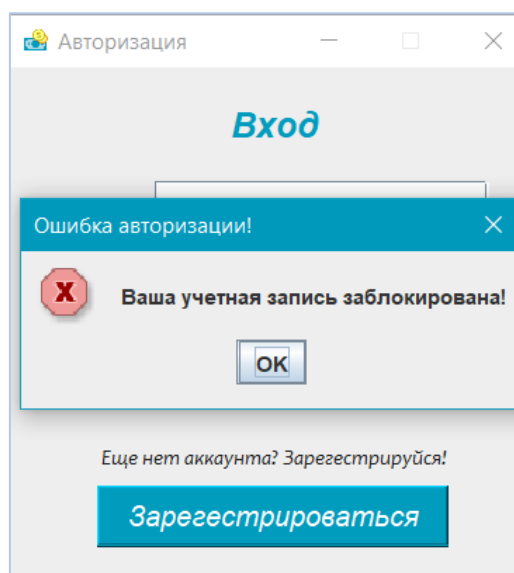


Рисунок 7.21 – Сообщение о блокировке

По клику на кнопке «Подключения» появляется окно со списком авторизаций пользователей (вид окна на рисунке 7.22). Есть возможность очистить историю входов по одной записи либо все сразу.

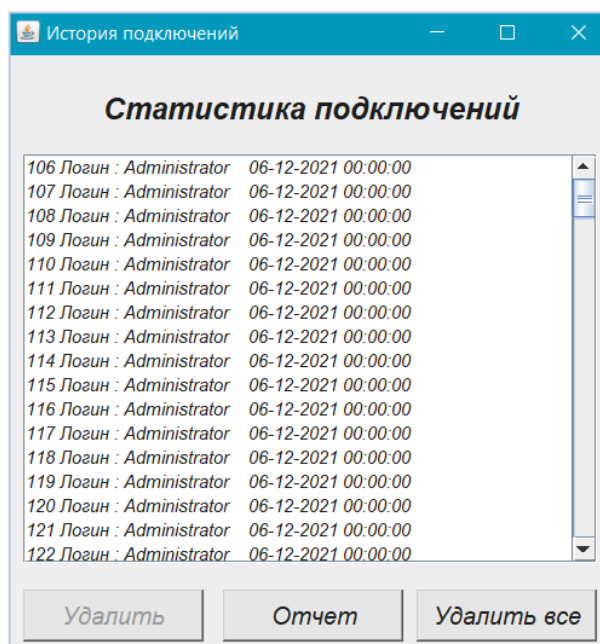


Рисунок 7.22 – Статистика подключений

Кроме этого, на основе выведенной статистике может быть составлен отчет как представлено на рисунке 7.23.

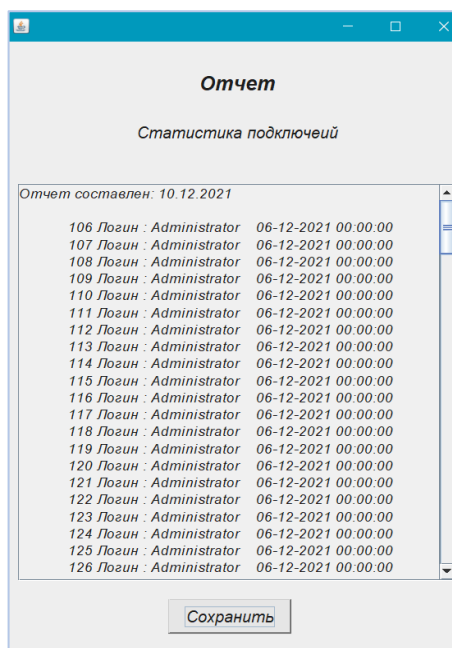


Рисунок 7.23 – Отчет по статистике подключений

Помимо всего вышесказанного, администратор также может управлять рассылкой писем на e mail, для этого после клика по кнопке «Рассылка» необходимо заполнить форму как на рисунке 7.24.

**Написать письмо**

Тема письма

Тема письма

Текст письма

Выберите получателей

Логин	Почта	Получатель
administrator	Rintarou-san@yandex.by	<input checked="" type="checkbox"/>
lika123	AntwaNnn@yandex.by	<input type="checkbox"/>
LIKA123456	AntwaNnn@yandex.by	<input type="checkbox"/>
LIKA999	AntwaNnn@yandex.by	<input type="checkbox"/>
LikaSnopok	Lika.Fleur@yandex.by	<input checked="" type="checkbox"/>
LOGIN	Lika.Fleur@yandex.by	<input type="checkbox"/>
SomeLogin	AntwaNnn@yandex.by	<input checked="" type="checkbox"/>

Отправить

Рисунок 7.24 – Рассылка по e-mail

В нижней части окна в таблице пользователи, отмеченные флажком в колонке «Получатель» будут включены в рассылку, пример такого письма, полученного на почту, приведен на рисунке 7.25.

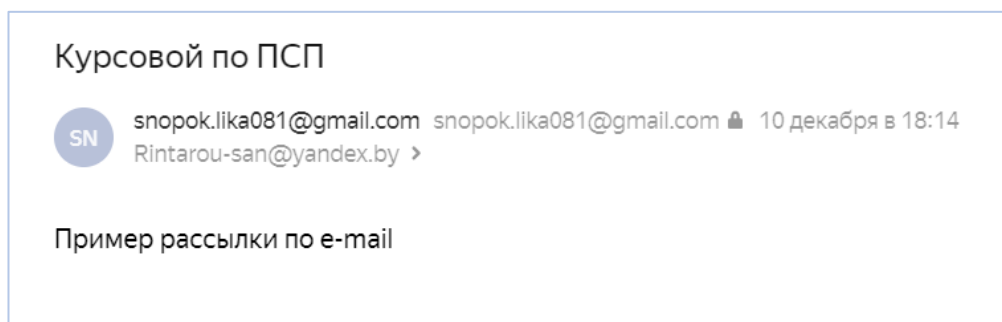


Рисунок 7.25 – Пример письма

Таким образом в данной главе были рассмотрены возможности разработанного приложения со стороны администратора и со стороны обычных пользователей.

## 8 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ

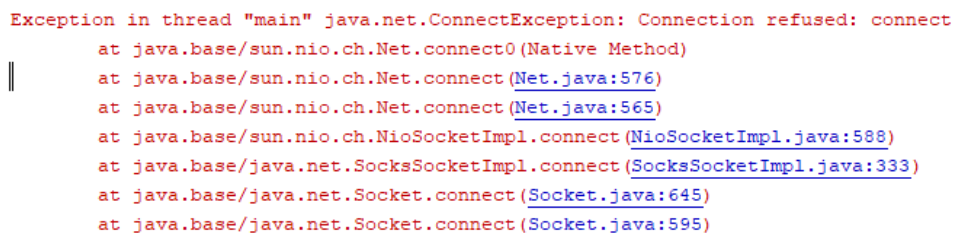
Тестирование – это очень важный и необходимый инструмент, который можно использовать, чтобы минимизировать количество ошибок в приложении.

К видам тестирования можно отнести:

- модульное тестирование – проверка работы программы на уровне отдельных модулей (классов, методов);
- интеграционное тестирование – проверка совместной работы нескольких модулей;
- системное тестирование – проверка работы системы в целом;
- функциональное тестирование;
- тестирование производительности;
- тестирование удобства использования;
- тестирование безопасности.

Для разработанной системы было проведено системное тестирование, т. е. проверка корректности работы всего приложения в целом.

Первая ошибка, с которой можно столкнуться – запустить клиентское приложение раньше серверного (рисунок 8.1)



```
Exception in thread "main" java.net.ConnectException: Connection refused: connect
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:576)
    at java.base/sun.nio.ch.Net.connect(Net.java:565)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:588)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:333)
    at java.base/java.net.Socket.connect(Socket.java:645)
    at java.base/java.net.Socket.connect(Socket.java:595)
```

Рисунок 8.1 – Не установлено соединение

Чтобы избежать ошибок с получением некорректных данных из форм при добавлении записей, проводимых расчетах и т. д., на поля были наложены фильтры. Таким образом, если поле предназначено для ввода числовых значений, то приложение не позволит ввести любой другой символ, кроме числового. Для некоторых полей важно количество символов в целом, например, номер телефона в формате xx xxx xx xx не может иметь более девяти цифр, поэтому в таких полях ограничен ввод на количество символов, однако там, где требуется ввести фамилию, имя, отчество количество символов

ограничивается размером соответствующего поля в базе данных. Пример подобной ошибки приведен на рисунке 8.2.

```
дек. 11, 2021 12:04:08 AM Database.Database insert  
SEVERE: null  
com.mysql.cj.jdbc.exceptions.MySQLDataTruncation: Data truncation: Data too long for column 'фамилия' at row 1  
    at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:104)  
    at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)  
    at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
```

Рисунок 8.2 – Ошибка «Слишком длинное значение»

В случае пустых полей, которые не предусмотрены настройками базы данных также вызовут ошибки (рисунок 8.3).

```
дек. 11, 2021 12:07:03 AM Database.Database insert  
SEVERE: null  
com.mysql.cj.jdbc.exceptions.MySQLDataTruncation: Data truncation: Incorrect date value: '' for column 'ДатаРождения' at row 1  
    at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:104)  
    at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)  
    at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
```

Рисунок 8.3 – Ошибка «Пустое значение»

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной курсовой работы было разработано приложение для учета банковских вкладов физических лиц. Для раскрытия темы, были решены следующие задачи:

- изучена предметная область, выявлены теоретические аспекты необходимые для разработки системы;
- выстроен алгоритм функционирования программного продукта;
- разработано приложение;
- на основе полученных результатов составлена пояснительная записка.

В процессе разработки были рассмотрены актуальные вопросы разработки ПО, исследована архитектура клиент-сервер, рассмотрен порядок взаимодействия клиентской стороны с серверной, рассмотрены преимущества двухуровневой и трехуровневой архитектуры, а также взаимодействие сервера с базой данных.

И в заключении составлена пояснительная записка, в которой приведены все диаграммы и алгоритмы, описывающие структуру приложения, а также руководство пользователя и руководство по разворачиванию данной системы.

Подводя итог, можно сказать, что использование новых технологий значительно повышает эффективность работы банка за счет использования преимуществ информационных систем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Институциональная структура банковской сферы: понятие и элементы [Электронный ресурс] [https://bstudy.net/711839/ekonomika/institutsionalnaya\\_struktura\\_bankovskoy\\_sfer\\_ponyatie\\_elementy](https://bstudy.net/711839/ekonomika/institutsionalnaya_struktura_bankovskoy_sfer_ponyatie_elementy)
- [2] Словарь банковских терминов [Электронный ресурс] <https://myfin.by/wiki/term/vklad-depozit>
- [3] Нормализация. Понятие и необходимость применения [Электронный ресурс] <https://www.bestprog.net/ru/2020/10/17/databases-normalization-concept-and-necessity-of-application-modification-anomalies-examples-ru/>
- [4] Методика описания бизнес-процессов банка [Электронный ресурс] [https://www.businessstudio.ru/articles/article/metodika\\_opisaniya\\_biznes\\_protsessov\\_bank\\_oversiya](https://www.businessstudio.ru/articles/article/metodika_opisaniya_biznes_protsessov_bank_oversiya)
- [5] Как банк может привлечь новых клиентов [Электронный ресурс] <https://www.klerk.ru/bank/articles/440033>
- [6] Нормализация базы данных [Электронный ресурс] <https://info-comp.ru/>

## ПРИЛОЖЕНИЕ А

### (обязательное)

### Листинг кода

#### Организация соединения с клиентом:

```
ServerSocket servSocket = new ServerSocket(PORT);
    System.out.println("Сервер запущен");

    try {
        while (true) {
            Socket socket = servSocket.accept();
            System.out.println("Был подключен новый клиент");
            try {
                new ThreadServer(socket);
            }
            catch (IOException e) {
                socket.close();
            }
        }
    } finally {
        servSocket.close();
    }
}
```

#### Класс соединения с базой данных:

```
public class Database {

    private String driverName = "com.mysql.cj.jdbc.Driver";
    private Connection connect;
    private Statement statemnt;

    public Database(String drivName, String url, String name, String pass)
throws ClassNotFoundException, SQLException {
        this.driverName = drivName;
        Class.forName(this.driverName);
        this.connect = DriverManager.getConnection(url, name, pass);
        this.statemnt = this.connect.createStatement();
        this.statemnt.execute("set character set utf8");
        this.statemnt.execute("set names utf8");
    }

    public void insert(String sqlString) {
        try {
            statemnt.executeUpdate(sqlString);
        } catch (SQLException ex) {
            Logger.getLogger(Database.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    public void delete(String sqlString) {
        try {
            statemnt.executeUpdate(sqlString);
        }
    }
}
```



## Продолжение приложения А

```
        } catch (SQLException ex) {
            Logger.getLogger(Database.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    public void update(String sqlString) {
        try {
            statemnt.executeUpdate(sqlString);
        } catch (SQLException ex) {
            Logger.getLogger(Database.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    public ResultSet select(String sqlString) {
        ResultSet rs = null;
        try {
            rs = statemnt.executeQuery(sqlString);
        } catch (SQLException ex) {
            Logger.getLogger(Database.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return rs;
    }

    public void close() {
        try {
            connect.close();
            statemnt.close();
        } catch (SQLException ex) {
            Logger.getLogger(Database.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}
```

### Метод для построения графика:

```
private void ButtonGraphActionPerformed(java.awt.event.ActionEvent evt) {
    BarChartDialog dlg = new BarChartDialog(this, true);
    dlg.setLocationRelativeTo(null);
    dlg.setVisible(true);

    Stream server = new Stream();
    String executeStr = "SELECT `ГодЗаклучения`,
`Депозит`.`Наименование`, COUNT(`ГодЗаклучения`) AS Количество FROM `Договор`
"
        + "INNER JOIN `Депозит` ON
`Договор`.`КодВклада`=`Депозит`.`Код`";
    if (!dlg.getDeposite().equals("")) {
        executeStr += " WHERE Наименование='" + dlg.getDeposite() +
"";
    }
}
```

## Продолжение приложения А

```
    }
    if (!dlg.getSince().equals("")) {
        executeStr += " AND ГодЗаклучения >= " + dlg.getSince();
    }
    if (!dlg.getFor().equals("")) {
        executeStr += " AND ГодЗаклучения <= " + dlg.getFor();
    }
    executeStr += " GROUP BY `ГодЗаклучения` ORDER BY
`ГодЗаклучения`";
    server.sendInt(14);
    server.sendString(executeStr);
    try {
        int count = server.getInt();
        DefaultCategoryDataset dataSet = new
DefaultCategoryDataset();
        for (int i = 0; i < count; i++) {
            String[] data = server.getString().split(":");
            dataSet.setValue(Integer.parseInt(data[1]), data[0],
data[0]);
        }
        JFreeChart chart = ChartFactory.createBarChart3D("Диаграмма
о количестве заключенных договоров", "Год", "Количество",
dataSet, PlotOrientation.VERTICAL, true, true, true);
        CategoryPlot P = (CategoryPlot) chart.getPlot();
        ChartFrame frame = new ChartFrame("Количество по годам",
chart);

        frame.setVisible(true);
        frame.setSize(700, 700);
    } catch (IOException ex) {
        Logger.getLogger(MainAdminForm.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```

### Метод рассылки писем на E-mail:

```
private void SendToEmailButtonActionPerformed(java.awt.event.ActionEvent
evt){

    EmailSendDialog dlg = new EmailSendDialog(this, true);
    dlg.setLocationRelativeTo(null);
    dlg.setVisible(true);

    Sender tlsSender = new Sender("snopok.lika081@gmail.com",
"Vjhb_Jufq17.02");
    tlsSender.send(dlg.getSubject(),dlg.getText(),
"snopok.lika081@gmail.com", dlg.getRecipients());
}
```

### Класс фильтра для строк:

```
public class DigitFilter extends DocumentFilter {

    private static final String DIGITS = "\\d+";
```

## Продолжение приложения А

```
@Override
    public void insertString(FilterBypass fb, int offset, String string,
        AttributeSet attr) throws BadLocationException {

        if (string.matches(DIGITS)) {
            super.insertString(fb, offset, string, attr);
        }
    }

    @Override
    public void replace(FilterBypass fb, int offset, int length, String
        string, AttributeSet attrs) throws BadLocationException {
        if (string.matches(DIGITS)) {
            super.replace(fb, offset, length, string, attrs);
        }
    }
}
```

### Класс паттерна Фабрика:

```
public class TableModelFactory {

    public MyTableModels createModel(TypeOfModel type, List<Object> list)
    {

        MyTableModels model = null;
        switch (type) {
            case CLIENT_TABLE:
                model = new ClientTableModel(list);
                break;
            case DEPOSITE_TABLE:
                model = new DepositTableModel(list);
                break;
            case CONTRACT_TABLE:
                model = new ContractTableModel(list);
                break;
            case CURRENCY_TABLE:
                model = new CurrencyTableModel(list);
                break;
            case USER_TABLE:
                model = new UserTableModel(list);
                break;
        }
        return model;
    }
}
```

## ПРИЛОЖЕНИЕ Б

### Скрипт базы данных

```
CREATE DATABASE Депозиты
Use депозиты;

CREATE TABLE `Пользователь` (
  `Логин` VARCHAR(20) NOT NULL,
  `Пароль` VARCHAR(20) NOT NULL,
  `Админ` VARCHAR(3) NOT NULL,
  `Заблокирован` BOOLEAN,
  PRIMARY KEY(`Логин`),
  CONSTRAINT Статус_check CHECK (`Админ` IN ('Да', 'Нет'))
);

CREATE TABLE `Валюта` (
  `Мнемокод` VARCHAR(3) NOT NULL,
  `Наименование` VARCHAR(40) NOT NULL,
  `Курс` NUMERIC(10, 4) NOT NULL,
  PRIMARY KEY (Мнемокод),
  CONSTRAINT Курс_check CHECK(`Курс` > 0)
);

CREATE TABLE `Депозит` (
  `Код` INT NOT NULL AUTO_INCREMENT,
  `Наименование` VARCHAR(30) NOT NULL,
  `ВалютаВклада` VARCHAR(3) NOT NULL,
  `ПроцентнаяСтавка` NUMERIC(5, 2) NOT NULL,
  `МинимальнаяСумма` NUMERIC(10, 2) NOT NULL,
  `ЧастичноеСнятие` VARCHAR(3) NOT NULL,
  PRIMARY KEY(`Код`),
  CONSTRAINT ЧастичноеСнятие_check CHECK (`ЧастичноеСнятие` IN ('Да',
'Нет')),
  CONSTRAINT КодВалюты_FK FOREIGN KEY (`ВалютаВклада`) REFERENCES
`Валюта`(`Мнемокод`) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE `Клиент` (
  `НомерПаспорта` VARCHAR(9) NOT NULL,
  `Фамилия` VARCHAR(30) NOT NULL,
  `Имя` VARCHAR(30) NOT NULL,
  `Отчество` VARCHAR(30) NOT NULL,
  `ДатаРождения` DATE NOT NULL,
  `НомерТелефона` VARCHAR(13),
  PRIMARY KEY (НомерПаспорта),
  CONSTRAINT ДатаРождения_check CHECK (ДатаРождения <= '2002-12-31')
);

CREATE TABLE `Договор` (
  `НомерДоговора` VARCHAR(10) NOT NULL,
  `НомерПаспорта` VARCHAR(9) NOT NULL,
  `КодВклада` INT NOT NULL,
```

## Продолжение приложения Б

```
`СрокДействия` NUMERIC(10, 2) NOT NULL,  
`Единица` VARCHAR(10) NOT NULL,  
`ГодЗаклучения` INT NOT NULL,  
`НачальнаяСумма` NUMERIC(10,2) NOT NULL,  
PRIMARY KEY (`НомерДоговора`),  
CONSTRAINT ГодЗаклучения_check CHECK (`ГодЗаклучения` >=1945 AND  
`ГодЗаклучения` <=2021),  
CONSTRAINT НомерПаспорта_FK FOREIGN KEY (`НомерПаспорта`) REFERENCES  
`Клиент`(`НомерПаспорта`) ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT Вклад_FK FOREIGN KEY (`КодВклада`) REFERENCES `Депозит`(`Код`)  
ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE `Статистика`(  
`Код` INT AUTO_INCREMENT,  
`Логин` VARCHAR(20) NOT NULL,  
`Дата` DATE NOT NULL,  
PRIMARY KEY(`Код`)  
);  
  
--ДОБАВЛЕНИЕ ЗАПИСЕЙ  
INSERT INTO `Валюта` VALUES ('USD', 'Доллар США', 2.49);  
INSERT INTO `Валюта` VALUES ('EUR', 'Евро', 2.81);  
INSERT INTO `Валюта` VALUES ('RUB', 'Российский рубль', 0.034);  
INSERT INTO `Валюта` VALUES ('JPY', 'Японская йена', 0.022);  
  
INSERT INTO `Клиент` VALUES('BM2352448', 'Петров', 'Петр', 'Петрович',  
'1991-03-12', '+375333617793');  
INSERT INTO `Клиент` VALUES('BM2772598', 'Сидоров', 'Сергей',  
'Алексеевич', '1989-08-10', '+375333613583');  
  
INSERT INTO `Депозит`(`Наименование`, `ВалютаВклада`, `ПроцентнаяСтавка`,  
`МинимальнаяСумма`, `ЧастичноеСнятие`) VALUES ('Традиционный', 'RUB', 7.3, 300,  
'Нет');  
INSERT INTO `Депозит`(`Наименование`, `ВалютаВклада`, `ПроцентнаяСтавка`,  
`МинимальнаяСумма`, `ЧастичноеСнятие`) VALUES ('Традиционный', 'BYN', 7.3, 300,  
'Нет');  
INSERT INTO `Депозит`(`Наименование`, `ВалютаВклада`, `ПроцентнаяСтавка`,  
`МинимальнаяСумма`, `ЧастичноеСнятие`) VALUES ('Традиционный', 'USD', 7.3, 300,  
'Нет');  
INSERT INTO `Депозит`(`Наименование`, `ВалютаВклада`, `ПроцентнаяСтавка`,  
`МинимальнаяСумма`, `ЧастичноеСнятие`) VALUES ('Традиционный', 'EUR', 7.3, 300,  
'Нет');  
  
INSERT INTO `Договор` VALUES ('2151001644', 'BM2352448', 2, 3, 'г.');
```

```
INSERT INTO `Договор` VALUES ('2151021647', 'BM2772598', 4, 3, 'мес.');
```

```
INSERT INTO `Договор` VALUES ('2151301647', 'BM2352448', 1, 5, 'г.');
```

```
INSERT INTO `Договор` VALUES ('3157001647', 'BM2772598', 3, 2.4, 'г.');
```