Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности

Кафедра Информационных систем и программирования

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Профиль Защищенные автоматизированные системы управления

**КУРСОВАЯ РАБОТА**

по дисциплине Технологии и методы программирования

(наименование дисциплины)

на тему: «Домофон»

(тема курсовой работы)

Выполнил студент 2 курса группы 18-К-АС1

Корниенко Н,М

(Ф.И.О.)

Допущен к защите\_\_\_\_\_\_\_

Руководитель (нормоконтролер) работы \_\_\_\_\_\_\_\_\_О.Б. Попова

Защищен \_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии Н.В. Кушнир\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

К.Е. Тотухов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Краснодар

2020

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности

Кафедра Информационных систем и программирования

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Профиль Защищенные автоматизированные системы управления

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_\_\_\_\_М.В. Янаева

«12» февраля 2020 г.

**ЗАДАНИЕ**

на курсовую работу

Студенту: Корниенко Н.М группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Банкомат»

План работы:

1. Изучение предметной области

2. Проектирование

3.  Описание реализованных диаграмм

Объем работы:

а) пояснительная записка 33 с.

Рекомендуемая литература

1.  Йордон. «Объектно-ориентированный анализ и проектирование систем»

2.  Роберт А. Максимчук. «UML для простых смертных»

3.  «Автоматизация проектирования вычислительных систем.» ред. М.Брейер

Срок выполнения: с «15» февраля по «11» мая 2020г.

Срок защиты: с «11» мая по «14» июня 2020 г.

Дата выдачи задания «15» февраля 2020г.

Дата сдачи работы на кафедру «01» июня 2020 г.

Руководитель работы \_\_\_\_\_\_\_\_\_ к.т.н., доцент Попова О.Б.

(должность, подпись,)

Задание принял студент Яруллин Я.Р Ф.И.О.

**Реферат**

Курсовая работа: 33 страница, 24 рисунков, 8 используемых источников.

Ключевые слова: UML, BPMN, ГАНТ, EPC, FURPS+ , IDEF0, DFD, ДИАГРАММЫ. ПРОЕКТИРОВАНИЕ, ПОСЛЕДОВАТЕЛЬНОСТИ, МОДЕЛЬ, КЛАСС, БАНКОМАТ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ВИРТУАЛЬНЫЙ.

Объектом исследования является программное обеспечение банкомата, в функционал которого входят такие возможности как: прием пластиковых карт, распознавание их принадлежности и проверки введенного пин кода, блокировка карт в случае трех попыток не верного ввода пин кода, вывод информации о счете клиента, выдача наличных по запросу в случае если их достаточно как на счету клиента так и в самом терминале, загрузка средств оператором терминала. Цель работы состоит в разработке проекта программного обеспечения «Банкомат» с использованием диаграмм разного вида, в полной мере описывающих как внутреннее устройство исследуемой системы, так и всевозможные взаимодействия между её компонентами.

В результате были получены диаграммы, обладающие исчерпывающей информацией о программном обеспечение домофона. К ним относятся: диаграмма Ганта, UML-диаграмма, IDEF0-диаграмма, DFD-диаграмма, EPC-диаграмма, BPMN «As-Is» и BPMN «To be», документ FURPS+.

**Содержание**

[Введение 5](#_Toc39780977)

[1 Формулировка задачи 6](#_Toc39780978)

[2 Диаграмма Ганта 7](#_Toc39780979)

[3 Создание модели As-Is в стандарте IDEF0 8](#_Toc39780980)

[4 Диаграмма потоков данных (DFD) 12](#_Toc39780981)

[5 UML 14](#_Toc39780982)

[6 EPC 15](#_Toc39780983)

[7 BPMN 16](#_Toc39780984)

[8 FURPS+ 17](#_Toc39780985)

[9 Результаты машинного тестирования программы 18](#_Toc39780986)

[10 Системные требования 20](#_Toc39780987)

[11 Руководство пользователя 21](#_Toc39780988)

[Заключение 23](#_Toc39780989)

[Список использованных источников 23](#_Toc39780989)4

[Приложение А – Проверка на антиплагиат 25](#_Toc39780990)

[Приложение Б – Диаграмма Ганта 26](#_Toc39780991)

[Приложение В – Листинг программы 27](#_Toc39780992)

**Введение**

В настоящее время банкоматы служат одним из основных средств выдачи наличных денежных средств. К числу достоинств, послуживших причиной такого глобального распространения технологии, можно отнести простоту обращения, надёжность, разгрузку банковских филиалов от больших очередей из желающих списать со своего счета наличные денежные средства. Применение банковских терминалов повсеместно стало неотъемлемой частью человеческой жизни.

Для разработки и отладки программного обеспечения банковского терминала в данном проекте используется симулятор банкомата, позволяющий в полной мере изучить взаимодействие всех системообразующих компонентов и провести полноценное их тестирование.

**1 Формулировка задачи**

Задачей данного курсового проекта является разработка модели программного обеспечения встроенного микропроцессора для банкомата. Банкомат должен состоять из следующих компонентов:

– цифрового дисплея для визуального отображения, вводимой информации;

– панель управления;

– приемник кредитных карт;

– хранилище денежных средств;

– лотка для выдачи денежных средств;

– хранилище конфискованных кредитных карт;

– встроенный принтер для выдачи справок о проведенных операциях;

Панель управления представляет собой обычную клавиатуру с урезанным функционалом (доступен ввод исключительно представленной цифрами информации). Ввиду невозможности реализации физического приемника кредитных карт реализован виртуальный (прием кредитной карты производится посредством считывания введенных пользователем данных карты по запросу системы). Хранилище денежных средств по той же причине представлено в виде виртуального и хранится вне памяти системы и считывается при каждом ее запуске. Лоток для выдачи денежных средств так же представлен в виртуальном виде и реализован через списание средств с счета терминала и клиента. Хранилище конфискованных кредитных карт реализовано в виде реестра, куда заносятся данные карт, число неудачных попыток авторизации которых превысило 3. Встроенный принтер реализован через вывод справки о произведенной операции.

# 

# 2 Диаграмма Ганта

Диаграмма Ганта — «это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Придумал американский инженер Генри Гант (Henry Gantt). Выглядит это как горизонтальные полосы, расположенные между двумя осями: списком задач по вертикали и датами по горизонтали.

На диаграмме видны не только сами задачи, но и их последовательность. Это позволяет ни о чём не забыть и делать всё своевременно.

Ключевым понятием диаграммы Ганта является «веха» — метка значимого момента в ходе выполнения работ, общая граница двух или более задач. Вехи позволяют наглядно отобразить необходимость синхронизации, последовательности в выполнении различных работ. Вехи, как и другие границы на диаграмме, не являются календарными датами. Сдвиг вехи приводит к сдвигу всего проекта. Поэтому диаграмма Ганта не является, строго говоря, графиком работ. Кроме того, диаграмма Ганта не отображает значимости или ресурсоемкости работ, не отображает сущности работ (области действия). Для крупных проектов диаграмма Ганта становится чрезмерно тяжеловесной и теряет всякую наглядность.»

Диаграмма Ганта проекта «Банкомат» находится в «Приложении Б».

**3 Создание модели As-Is в стандарте IDEF0**

Чтобы оценить возможности, разрабатываемой системы необходимо построить её базовую модель, которую можно представить в виде диаграммы As-Is.

Диаграмма As-Is – это функциональная модель системы «как есть», позволяющая узнать где находятся слабые места, в чём будут состоять преимущества и недостатки, протекающих в ней. Применение данной модели позволит чётко зафиксировать какие информационные объекты принимают участие в жизненном цикле системы, какая информация будет поступать на вход и что будет получаться на выходе. Модель As-Is, строится с использованием нотации IDEF0.

IDEF0 – это графическая нотация, предназначенная для описания бизнес-процессов. Система, описываемая в данной нотации, проходит через декомпозицию или, иными словами, разбиение на взаимосвязанные функции. Для каждой функции существует правило сторон:

– стрелкой слева обозначаются входные данные;

– стрелкой сверху – управление;

– стрелкой справа – выходные данные;

– стрелкой снизу – механизм.

Учитывая всё вышеперечисленное на рисунке 1 была составлена модель As-Is проекта «Банкомат». Полученная модель может быть представлена в более подробном виде путём разбиения на большее количество составных элементов.

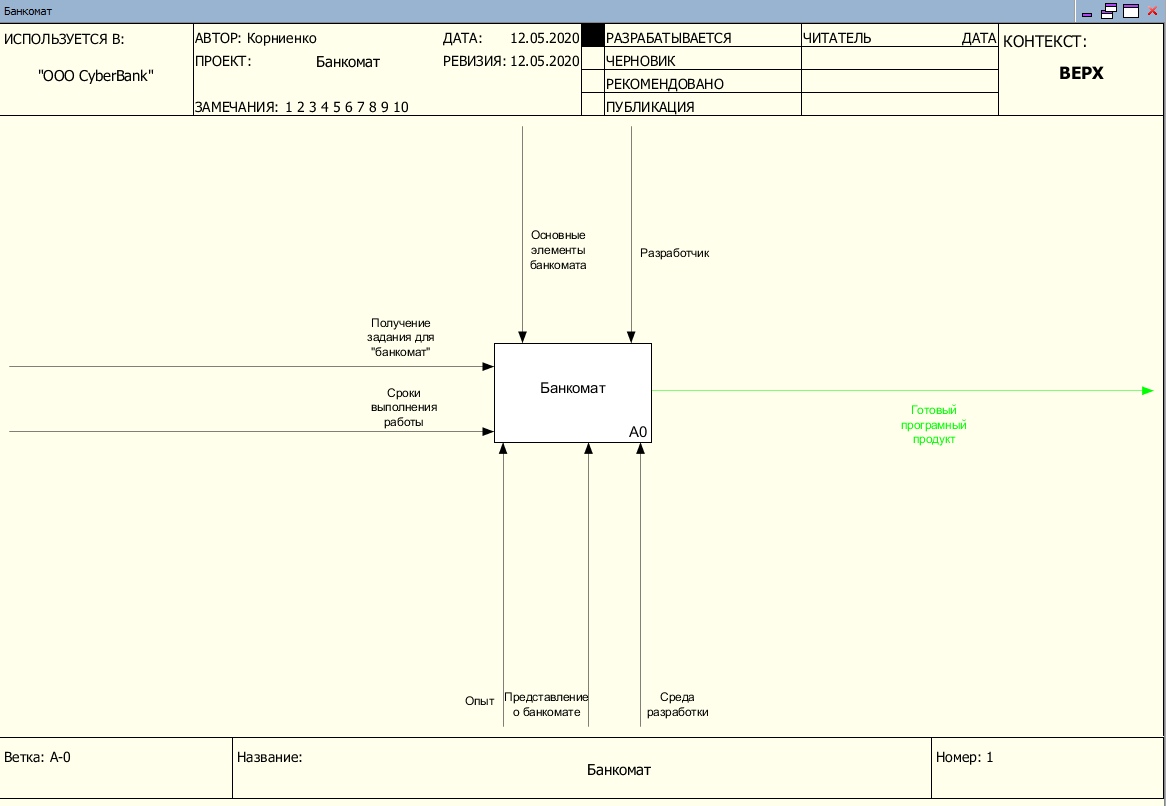


Рисунок 1 – Модель As-Is проекта «Банкомат»

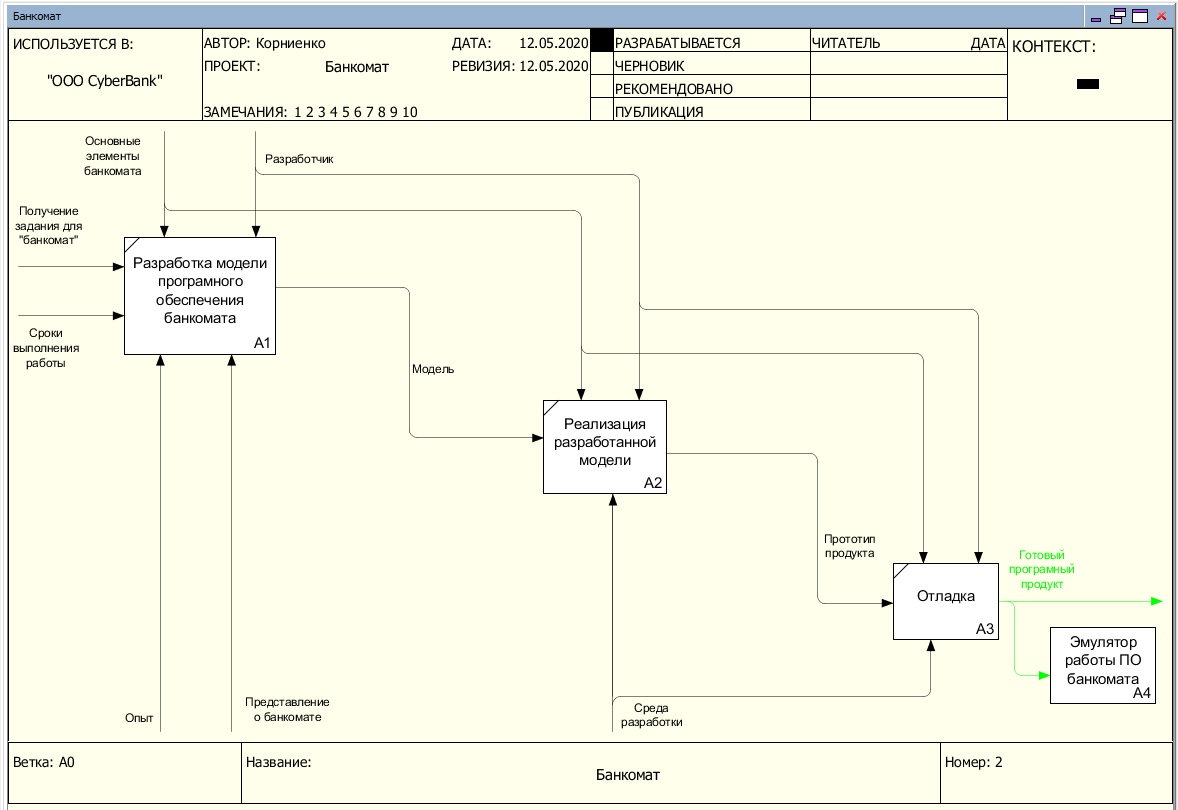


Рисунок 2 – Декомпозиция проекта «Банкомат»

Ниже представлены декомпозиции составляющих элементов разработки конечного продукта

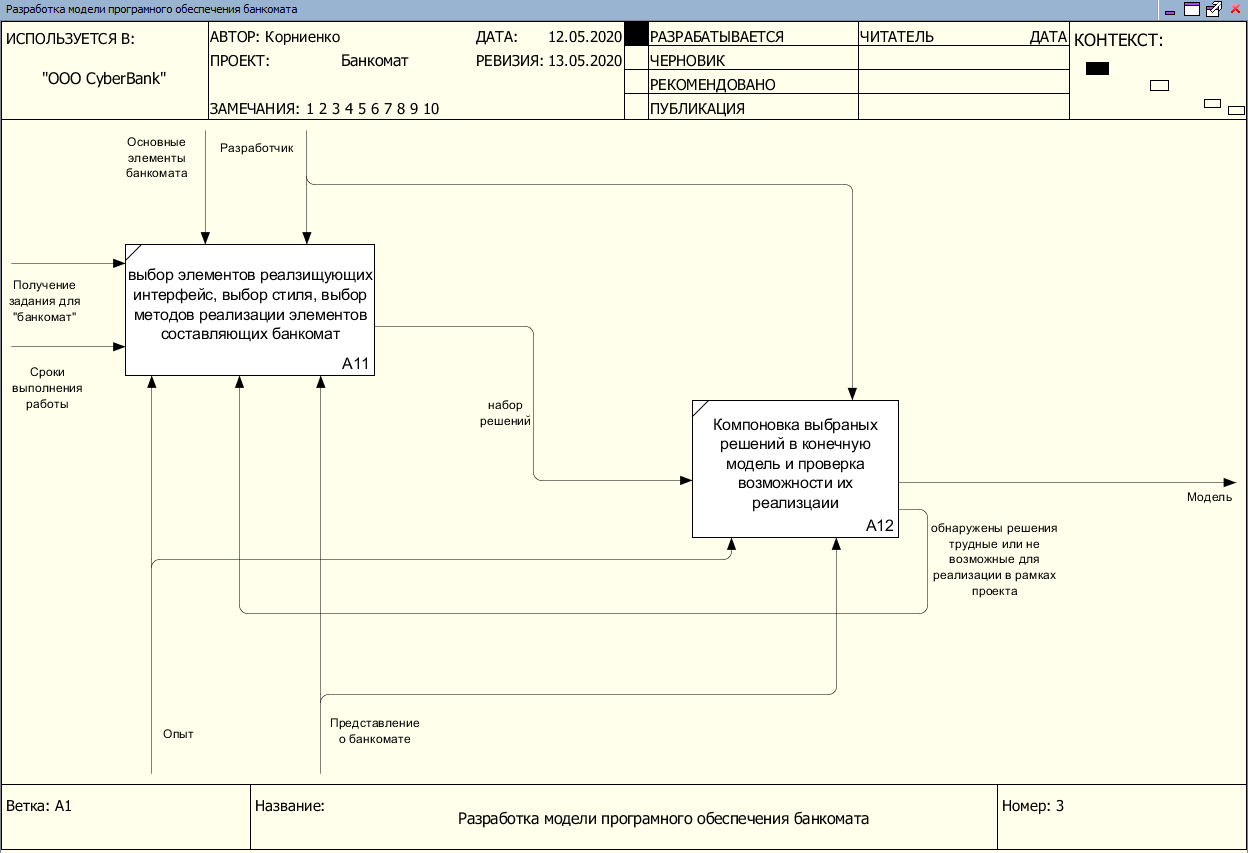


Рисунок 3 – Декомпозиция разработки модели програмы

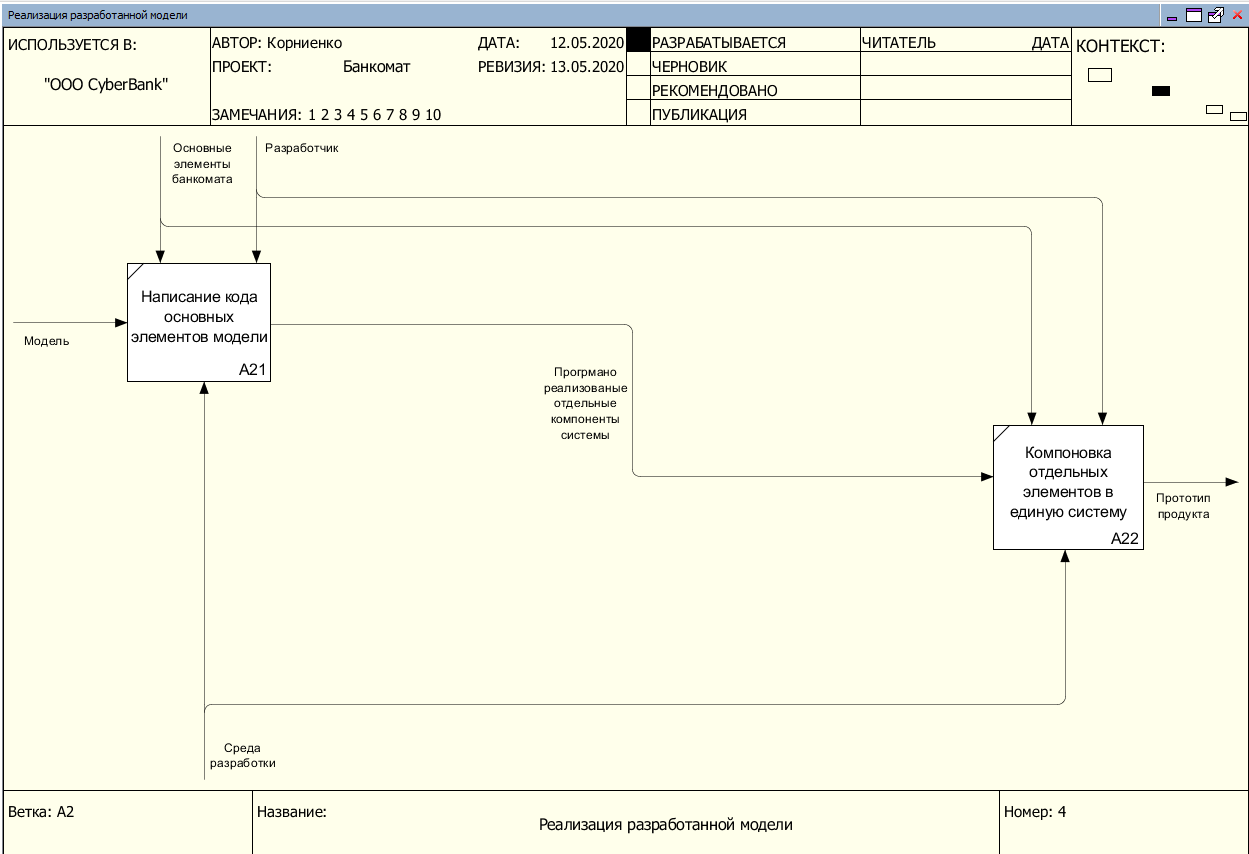


Рисунок 4 – Декомпозиция реализации модели

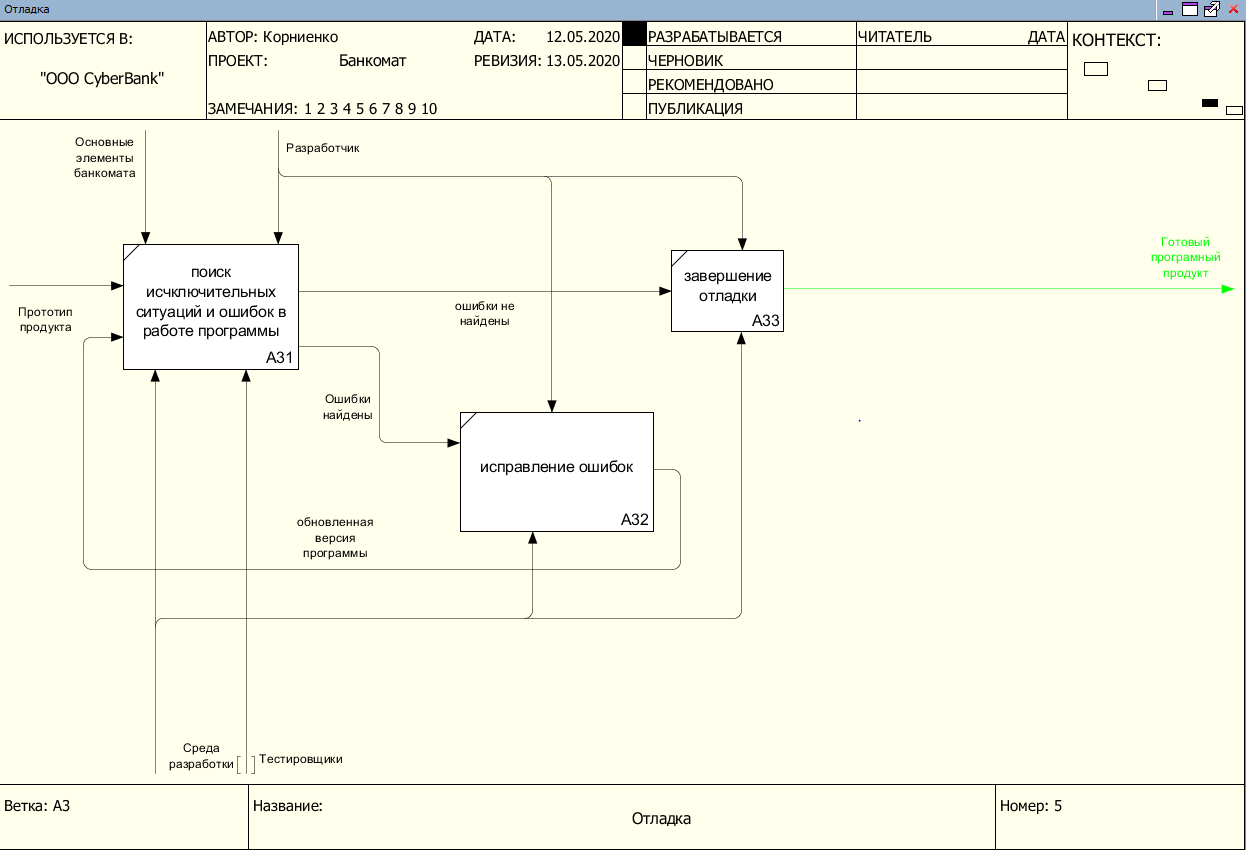


Рисунок 5 – Декомпозиция отладки

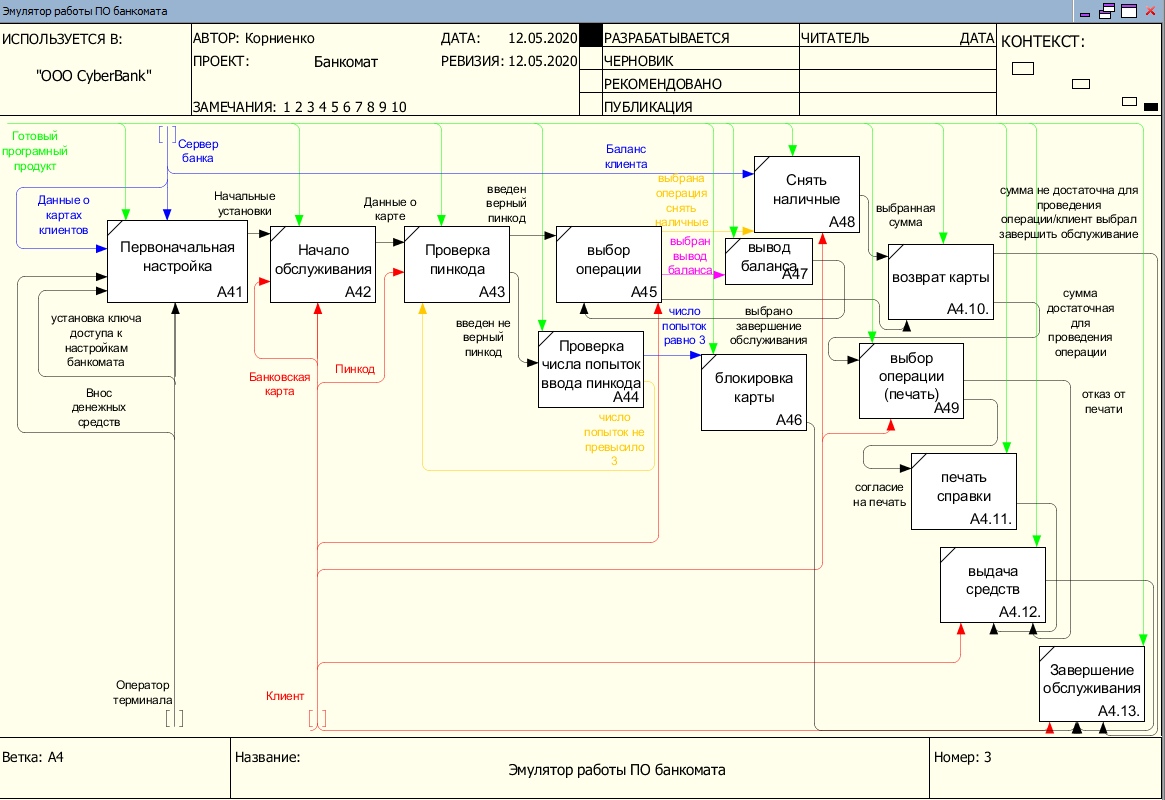


Рисунок 6 – Декомпозиция кода программы

Входными данными для данной системы в случае работы оператора с ней являются данные о картах клиентов, внос денежных средств, установка ключа доступа для дальнейшего обслуживания терминала.

Входными данными для данной системы в случае работы клиента с ней являются карта, пинкод, действия в меню обслуживания, действия в меню печати справки.

Механизмом реализации работы системы являются оператор терминала и клиент.

Результатом деятельности системы является выполнение запрошенных клиентом операций.

**4 Диаграмма потоков данных (DFD)**

Диаграмма потоков данных DFD (DataFlowDiagrams) – «это методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Диаграмма DFD – это один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.»

В результате декомпозиции системы «Банкомат» была получена следующая диаграмма DFD (рис. 5).

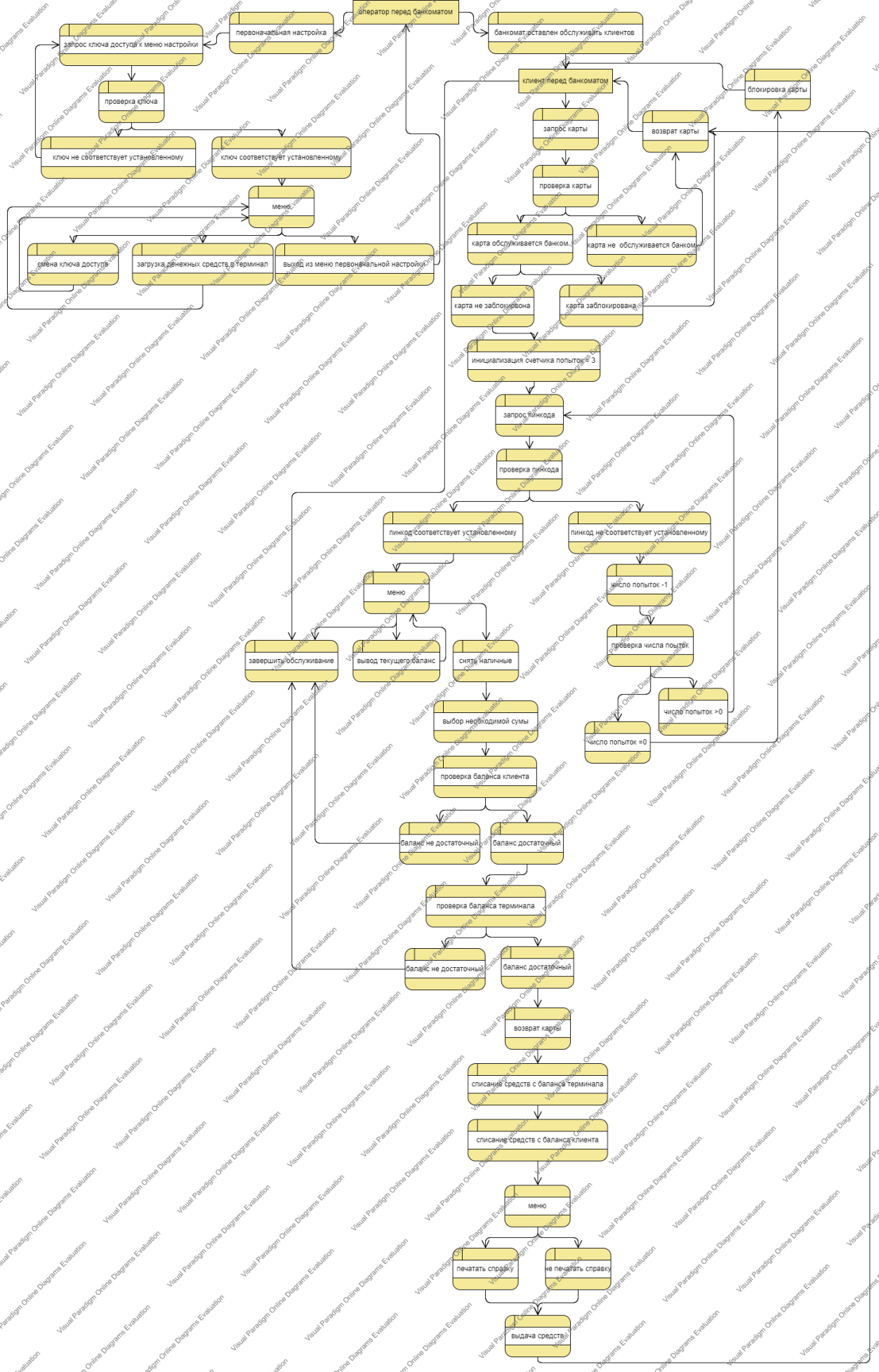


Рисунок 7 – Диаграмма DFD-системы «Банкомат»

**5 UML**

UML (англ. Unified Modeling Language — «унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.»

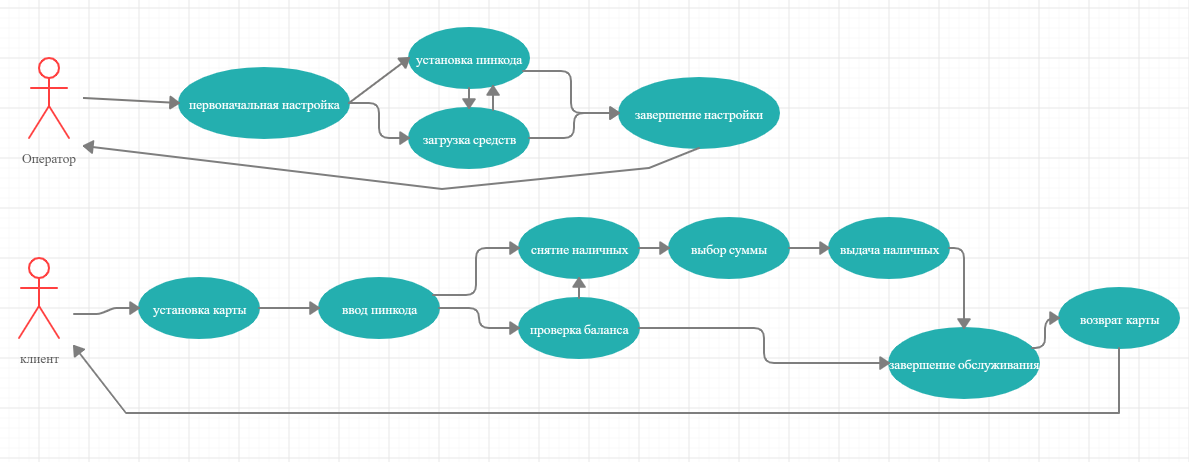


Рисунок 8 – UML-диаграмма системы «Банкомат»

# 6 EPC

Событийная цепочка процессов (EPC-диаграмма, англ. event-driven process chain) — «тип блок-схемы, используемой для бизнес-моделирования. EPC может быть использована для настройки системы планирования ресурсов предприятия (ERP), и для улучшений бизнес-процессов.

Организации используют EPC-диаграммы для планирования потоков работ бизнес-процессов. Существует ряд инструментов для создания EPC-диаграмм, некоторые из этих средств поддерживают инструментонезависимый формат обмена данными EPC — язык разметки EPML. EPC-диаграммы используют символы нескольких видов, чтобы показать структуру потока управления (последовательность решений, функции, события и другие элементы) бизнес-процесса.

EPC-метод был разработан Августом-Вильгельмом Шеером в рамках работ над созданием ARIS в начале 1990-х годов. Используется многими организациями для моделирования, анализа и реорганизации бизнес-процессов.»

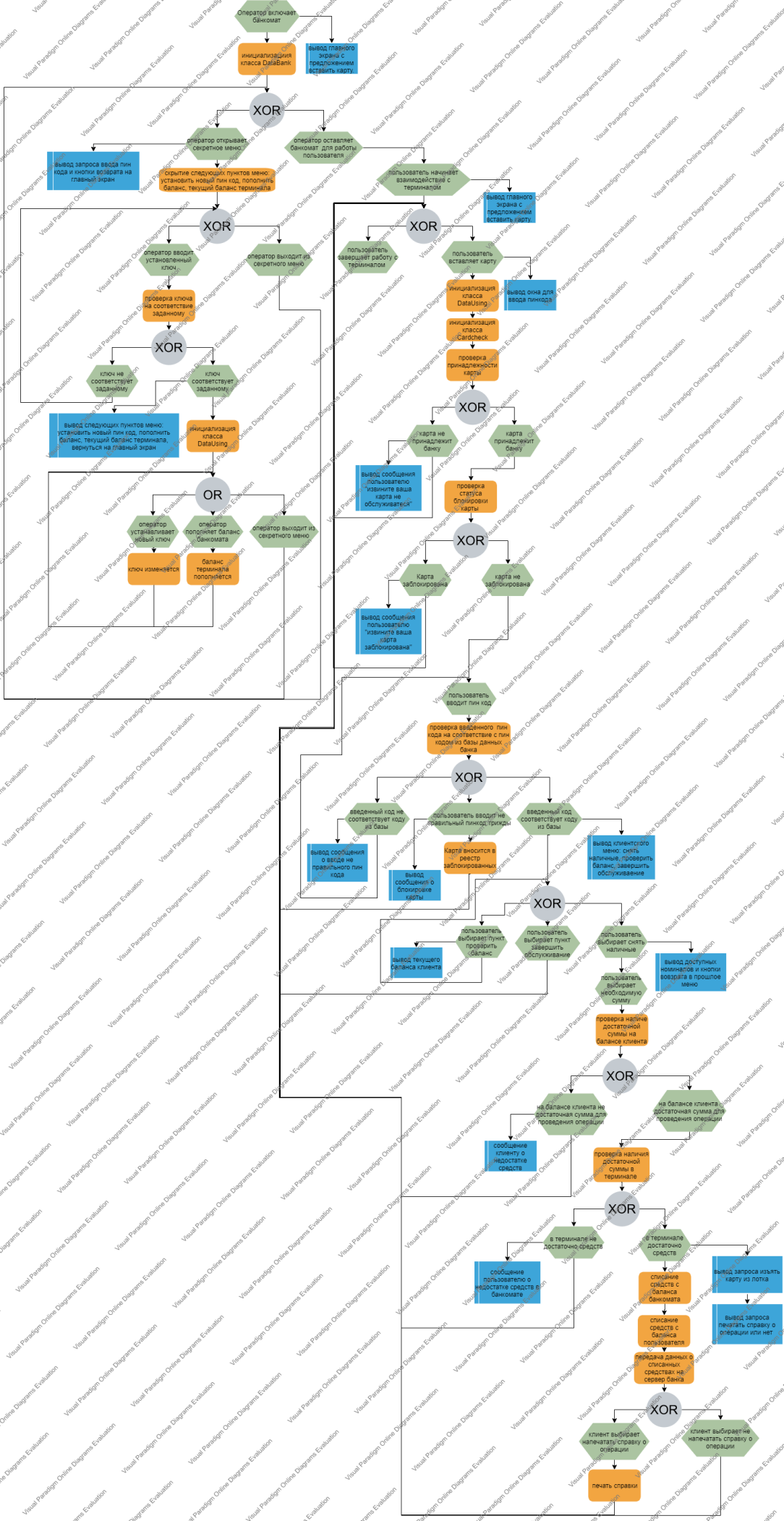


Рисунок 9 – EPC-диаграмма системы «Банкомат»

# 7 BPMN

BPMN (Business Process Management Notation) – «это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса.

Говоря проще, такая нотация представляет собой описание графических элементов, используемых для построения схемы протекания бизнес-процесса.

Как минимум, такая схема нужна, чтобы выстроить в соответствии с ней бизнес процесс и понятно регламентировать его для всех участников.

Как максимум, моделирование BPMN позволяет впоследствии провести автоматизацию бизнес-процессов в соответствии с имеющейся схемой.»

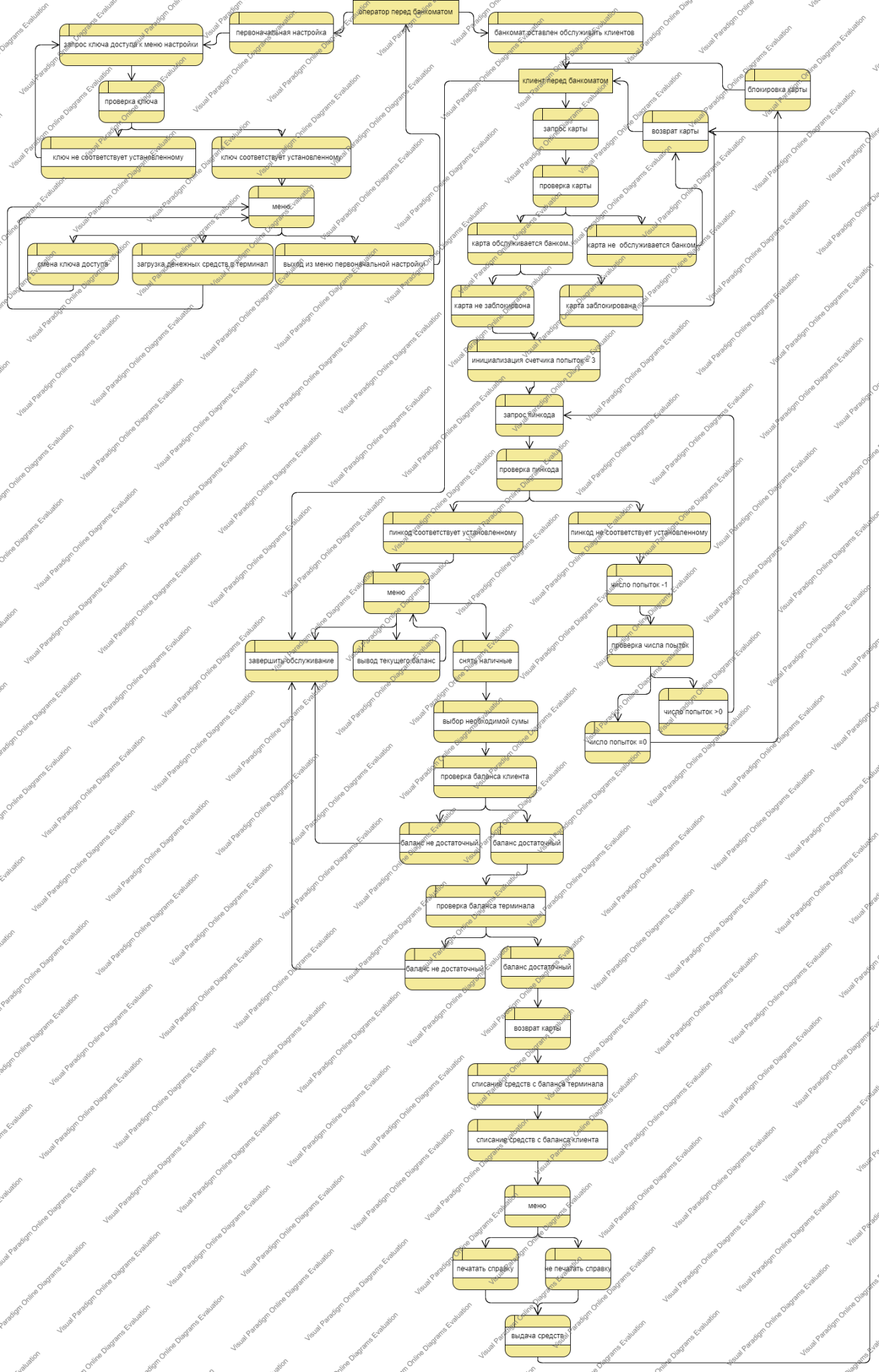


Рисунок 10 – Диаграмма BPMN «As-Is» и «To be»

# 8 FURPS+

Классификация требований к системе FURPS+ была разработана Робертом Грэйди (Robert Grady) из Hewlett-Packard и предложена в 1992 году. Сокращение FURPS расшифровывается так:

* Functionality, функциональность
* Usability, удобство использования
* Reliability, надежность
* Performance, производительность
* Supportability, поддерживаемость

+ необходимо помнить о таких возможных ограничениях, как:

* ограничения проектирования, design
* ограничения разработки, implementation
* ограничения на интерфейсы, interface
* физические ограничения, physical

Если применить к этой классификации популярное разделение требований на функциональные и нефункциональные, то к последним следует отнести все перечисленные выше группы кроме первой, т.е. URPS+.

F – исчерпывающий набор функций, повышенная безопасность.

U – интуитивный интерфейс.

R – сбои исключены, предусмотрены и исключены все возможные ошибки; время готовности системы к работе – 1 сек.

P – время отклика системы 0.1 сек, 100% эффективность работы.

S – требует предварительной настройки, поддержку наличия средств для выдачи клиентам

+ - возможность обработки карт сторонних банков

FURPS+ для системы «Банкомат»

# 9 Результаты машинного тестирования программы

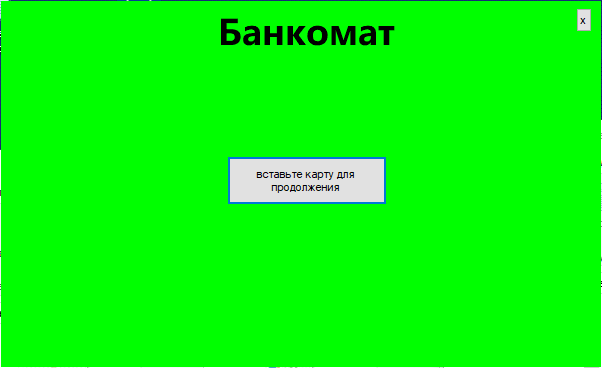


Рисунок 11 – Основное окно

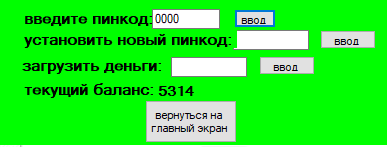


Рисунок 12 – Меню обслуживания терминала

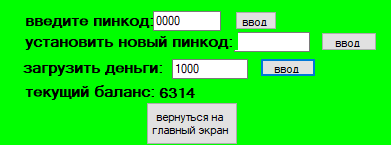


Рисунок 13 – Внос средств

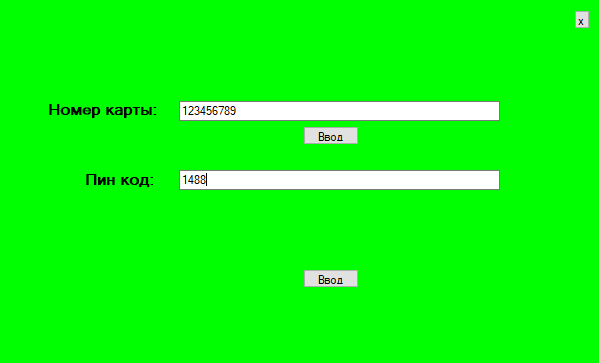


Рисунок 14 – Ввод карты

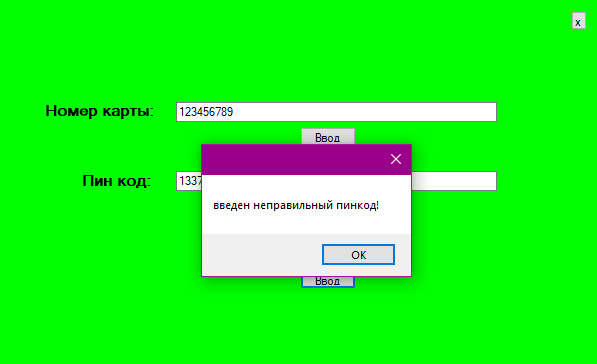


Рисунок 15 – Попытка ввода неверного пинкода

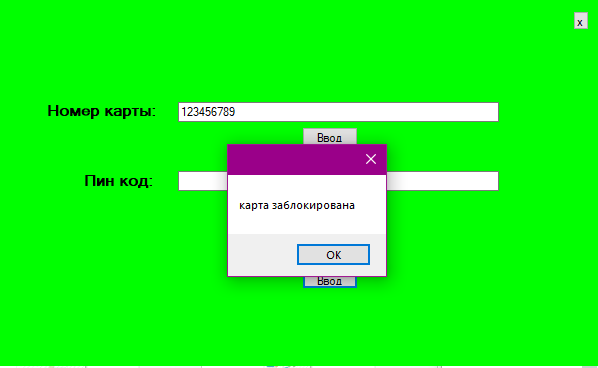


Рисунок 16 – Блокировка карты при совершении трех попыток ввести неверный пинкод

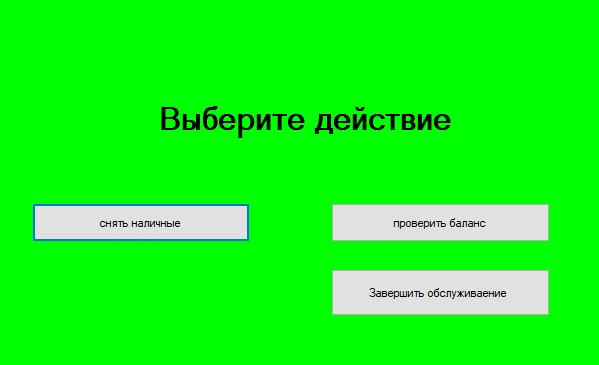


Рисунок 17 – Меню обслуживания клиента

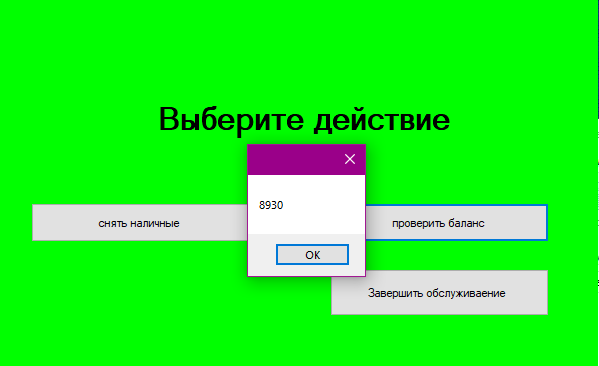


Рисунок 18 – Отображения баланса



Рисунок 19 – Снятие наличных

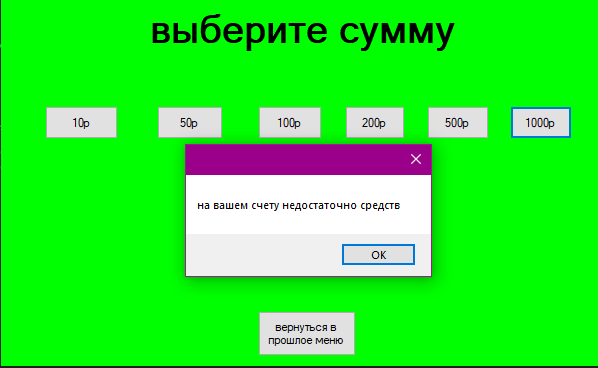


Рисунок 20 – Недостаток средств на балансе клиента

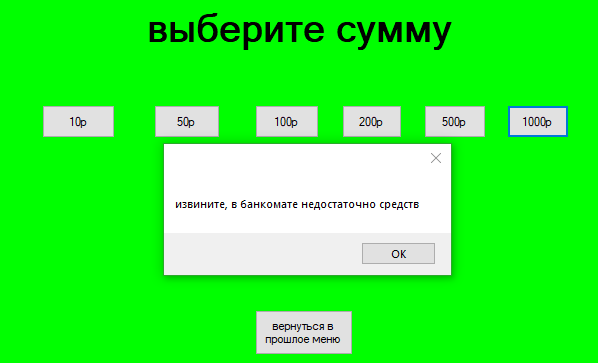


Рисунок 21 – Недостаток средств на балансе терминала

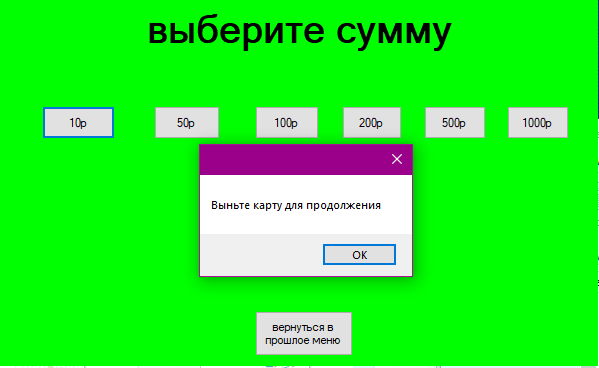


Рисунок 21 – Извлечение карты

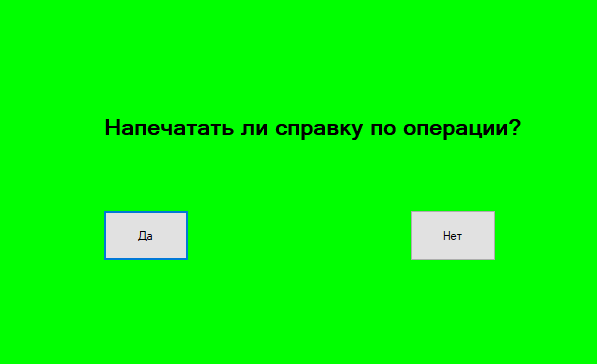


Рисунок 22 – Запрос на печать справки

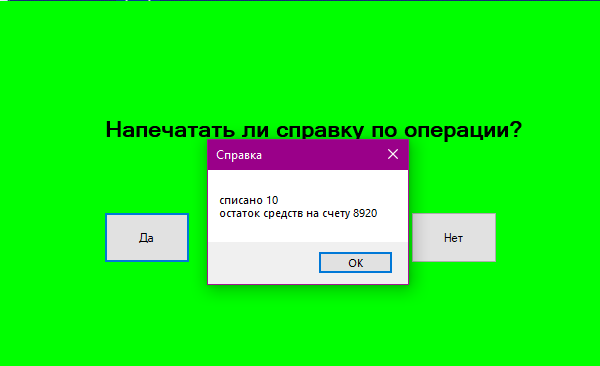


Рисунок 23 – Справка

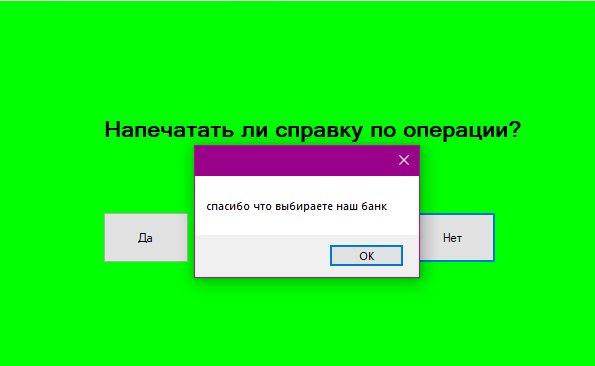


Рисунок 24 – Завершение обслуживания

**10 Системные требования**

Таблица 1 – Системные требования программы

|  |  |
| --- | --- |
| Процессор | 2.5 ГГц |
| Оперативная память | 150 Мб |
| Монитор | 1920 x 1080 |
| Свободное место на носителе | 15 Мб |
| Устройства взаимодействия с пользователем | Клавиатура и мышь |
| Программное обеспечение | Visual Studio 2019 года последней версии |

**11 Руководство пользователя**

Перед первым запуском необходимо скачать папку bank из хранилища проекта и разместить ее по адресу C:\Bank.

atmbalance - хранит баланс банкомата. (баланс не может быть больше чем 2^31)

blockedcards - хранит номера заблокированных карт.

cardinfo хранит информацию о картах клиентов в формате: строка номер карты строка пинкод строка баланс.

Для доступа в меню настроек следует в программе нажать на "банкомат" и ввести стандартный код "0000".

Настройка включает в себя смену стандартного кода и внос средств в терминал.

После завершения настройки система переходит в режим обслуживания реализуя основные функции банкомата.

Для начала обслуживания клиенту необходимо ввести данные карты и ввести пинкод. В случае трех попыток ввода неверного пинкода карта блокируется системой и считается невозможной к использованию. Для восстановления карты следует удалить ее данные из «blockedcards». В случае ввода верного пинкода клиента встречает меню обслуживания включающее в себя: Снятие наличных, отображение баланса, завершение обслуживания.

При выборе снятия наличных клиент попадает в меню выбора необходимой суммы. После выбора он получает уведомление необходимости изъять карту из банкомата без чего дальнее обслуживание не возможно. За изъятием карты следует проверка наличия достаточного колличества средств на счету клиента и в хранилище банкомата для проведения операции. В случае если на счету терминала или клиента не достаточно средств клиент получает уведомление об этом и его обслуживание завершается. В случае достатка средств на обоих счетах клиенту предлагается печать справки после чего происходит выдача средств, их списание с обоих счетов, выдача справки(опционально) и завершение обслуживания.

**Заключение**

В результате выполнения данного курсового проекта была спроектирована система «Банкомат» на языке высокого уровня C#, позволяющая наглядно продемонстрировать работу всех её компонентов. Полученные диаграммы обладают простой и понятной, даже для человека незнакомого с данной областью, структурой, описывающей каждый аспект системы с различных сторон.

При построении диаграмм использовались основные правила и принципы моделирования, включающие графическое представление объектов и связей между ними, иерархическое построение, а также названия, отражающие назначение той или иной сущности, или взаимодействия.

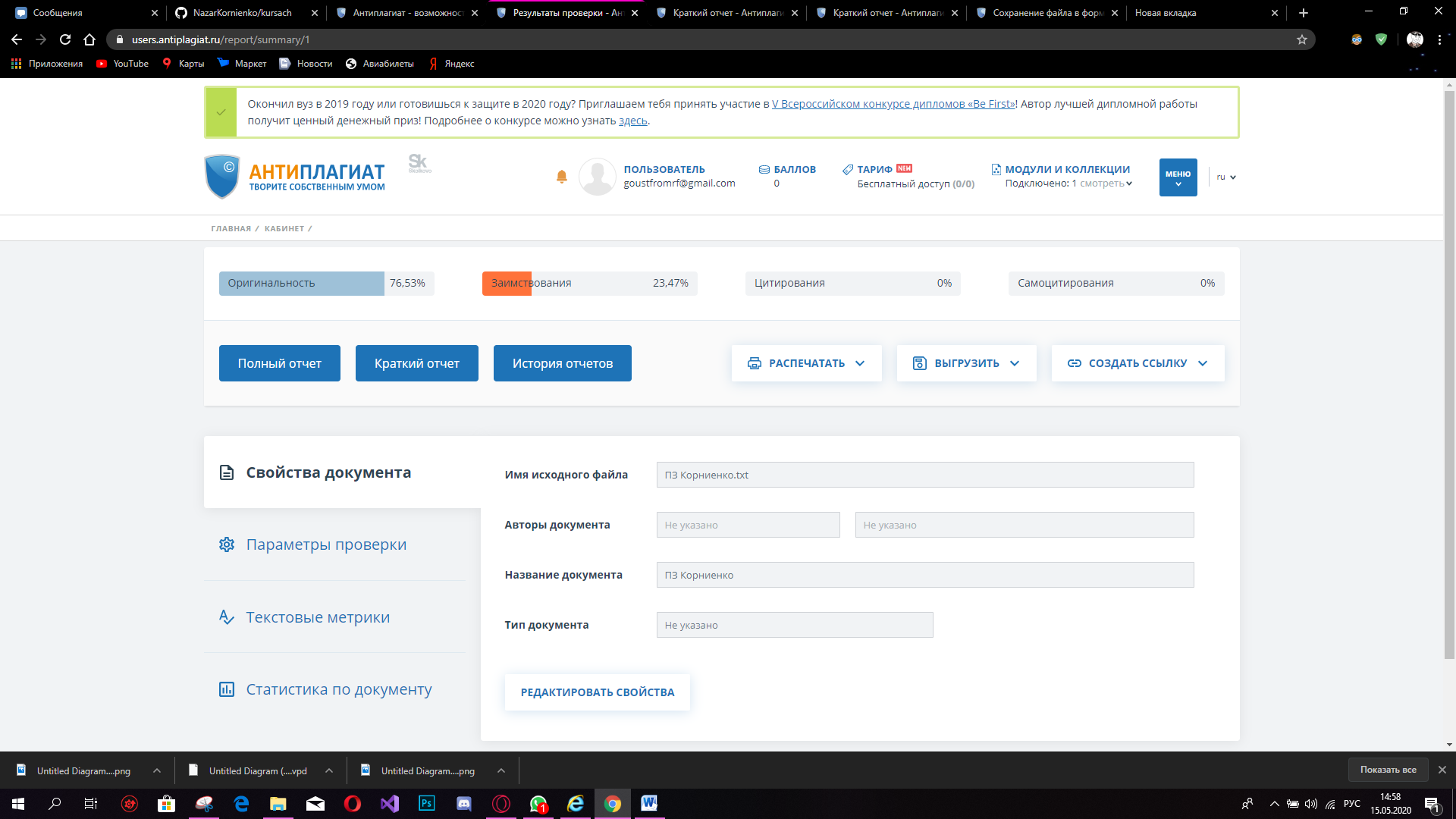
Благодаря детальному разбору проекта при помощи диаграмм проектирования, полученных в процессе разработки, можно с уверенностью сказать, что полученная система «Банкомат» полностью соответствует современным стандартам безопасности и способна выполнять различные взаимодействия, как для коммуникации с человеком, так и для повышения уровня защищённости.

Были получены важные знания и практические навыки как в области использования объектно-ориентированных языков программирования в целом, так и в области построения диаграмм проектирования, отображающих поведение различных организационных структур.

**Список использованных источников**

1. Ларман, Крэг. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / КрэгЛарман. - Москва: Гостехиздат, 2017. - 736 c.
2. Роберт А. Максимчук. UML для простых смертных / Роберт А. Максимчук, Эрик Дж. Нейбург. - Москва: СИНТЕГ, 2014. - 272 c.
3. Йордон, Эдвард. Объектно-ориентированный анализ и проектирование систем / Эдвард Йордон , Карл Аргила. - М.: ЛОРИ, 2014. - 264 c.
4. SoloLearn – C# Tutorial. [Электронный ресурс]: - Режим доступа: <https://www.sololearn.com/Course/CSharp/> (Дата обращения 13.03.2020).
5. Википедия. [Электронный ресурс]: - Режим доступа: [https://ru.wikipedia.org](https://ru.wikipedia.org/wiki/Тестовые_функции_для_оптимизации) (Дата обращения 17.09.2019).
6. GitHub – NazarKornienko/kursach. [Электронный ресурс]: - Режим доступа: <https://github.com/NazarKornienko/kursach> (Дата обращения 06.05.2020).
7. Comindware – Нотация BPMN 2.0 [Электронный ресурс]: - Режим доступа: <https://comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/> (Дата обращения 28.02.2020)
8. SysAna– Требования к системе: классификация FURPS+ [Электронный ресурс]: - Режим доступа: <https://sysana.wordpress.com/2010/09/16/furps/> (Дата обращения 03.03.2020)

# Приложение А – Проверка на антиплагиат



# Приложение Б – Диаграмма Ганта

