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: «Мини-АТС»

(тема курсовой работы)

Выполнил студент 2 курса группы 18-К-АС1

Ефимов А.С.

(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: Ефимов А.С. группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Мини-АТС»

План работы:

1. Изучение предметной области

2. Проектирование

3.  Описание реализованных диаграмм

Объем работы:

а) пояснительная записка 32 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 Ефимов А.С Ф.И.О.

**Реферат**

Курсовая работа: 32 страница, 13 рисунков, 8 используемых источников.

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

Объектом исследования является программное обеспечение и симулятор мини-АТС, который способен устанавливать соединение с набранным номером, а в случае удачного соединения появляется возможность сбросить вызов, также внесение номеров в список.

Цель работы состоит в разработке проекта программного обеспечения «Мини АТС» 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 мини-АТС. 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](#_Toc41129621)

[1 Формулировка задачи 6](#_Toc41129622)

[2 Диаграмма Ганта 6](#_Toc41129623)

[3 Создание модели As-Is в стандарте IDEF0 8](#_Toc41129624)

[4 Диаграмма потоков данных (DFD) 12](#_Toc41129625)

[5 UML 13](#_Toc41129626)

[6 EPC 14](#_Toc41129627)

[7 BPMN 15](#_Toc41129628)

[8 FURPS+ 17](#_Toc41129629)

[9 Результаты машинного тестирования программы 18](#_Toc41129630)

[10 Системные требования 22](#_Toc41129631)

[11 Руководство пользователя 23](#_Toc41129632)

[Заключение 25](#_Toc41129633)

[Приложение А – Проверка на антиплагиат 27](#_Toc41129634)

[Приложение Б – Диаграмма Ганта 28](#_Toc41129635)

[Приложение В – Листинг программы 29](#_Toc41129636)

**Введение**

Мини-АТС используется в повседневной жизни организаций. Мини-АТС служит для упорядочивания хаотичных сигналов звонков в оптимальный, чтобы каждый звонок дошел туда куда нужно. Преимуществом АТС является то, что все внутренние звонки являются бесплатными, а платить остается только за внешние звонки.

Однако, несмотря на использование данной установкой, она не может быть использована достаточно широко, чтобы удовлетворить потребности всех, поэтому её часто используют для удовлетворения требований организаций, обычному человеку это технология не требуется. Сейчас существует много аналогов функций мини-АТС, которые реализуют себя лучше и продуктивней, и они направлены конкретно на выполнение одной цели, а не как мини-АТС всего понемногу.

Таким образом, симулятор мини-АТС – имитация одной из возможности работы устройства, который позволяет управлять возможностью соединения между абонентами одной сети.

**1 Формулировка задачи**

Задачей данного курсового проекта является разработка программы, которая имитирует связь между абонентами сети и устанавливает соединение, либо уведомляет о недоступности абонента. Мини-АТС должен состоять из следующих компонентов:

– панель для набора цифр и визуального вывода;

– аналог списка, в который можно внести номер, чтобы не забыть его;

– кнопки сброса и вызова с прилегающим визуальным отображением установления соединения или недоступностью абонента.

Панель ввода номера с визуальным выводом находится в левой части экрана, которая используется пользователем для того, чтобы набрать номер другого пользователя для установки соединения между ними. Аналог контактной книги находится в правой части экрана, в которой есть возможность внесения номеров через панель ввода номера. Для того чтобы начать вызов, нужно внести номер и нажать на кнопку вызов, тогда система обработает запрос и выведет результат, если связь установлена, станет активна кнопка сброс, которая помогает завершить вызов и возвращает всё в первоначальное состояние, для повторного вызова.

# 2 Диаграмма Ганта

Диаграмма Ганта — «это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Диаграмма визуализируется как горизонтальные полосы, расположенные между двумя осями: списком задач по вертикали и датами по горизонтали.

Диаграмма отображает заданные пользователем задачи и даты, чтобы быть более точным в отображение.

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 для проекта «Мини-АТС» 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.

Модель AS-IS — это модель «как есть», т.е. модель уже существующего процесса/функции. Обследование процессов является обязательной частью любого проекта создания или развития системы. Построение функциональной модели AS-IS позволяет четко зафиксировать какие информационные объекты используются при выполнении функций различного уровня детализации. На основе анализа текущих процессов информационной обучающей системы была создана следующая AS-IS модель, которая позволяет выделить и систематизировать процессы, протекающие в данной системе при её функционировании.

IDEF0 – методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

На рисунке 1 была представлена модель As-Is проекта «Мини-АТС».

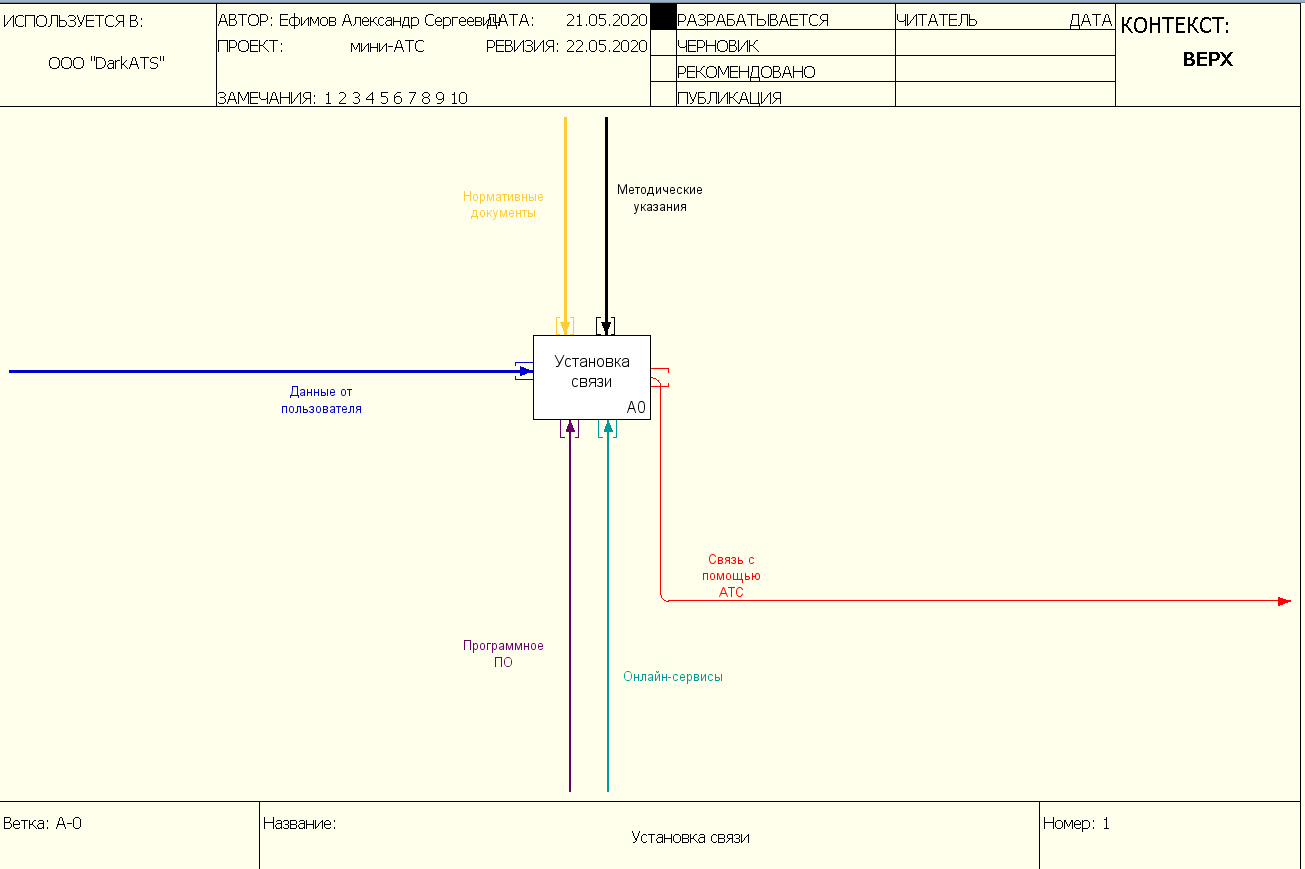


Рисунок 1 – Модель As-Is проекта «мини-АТС»

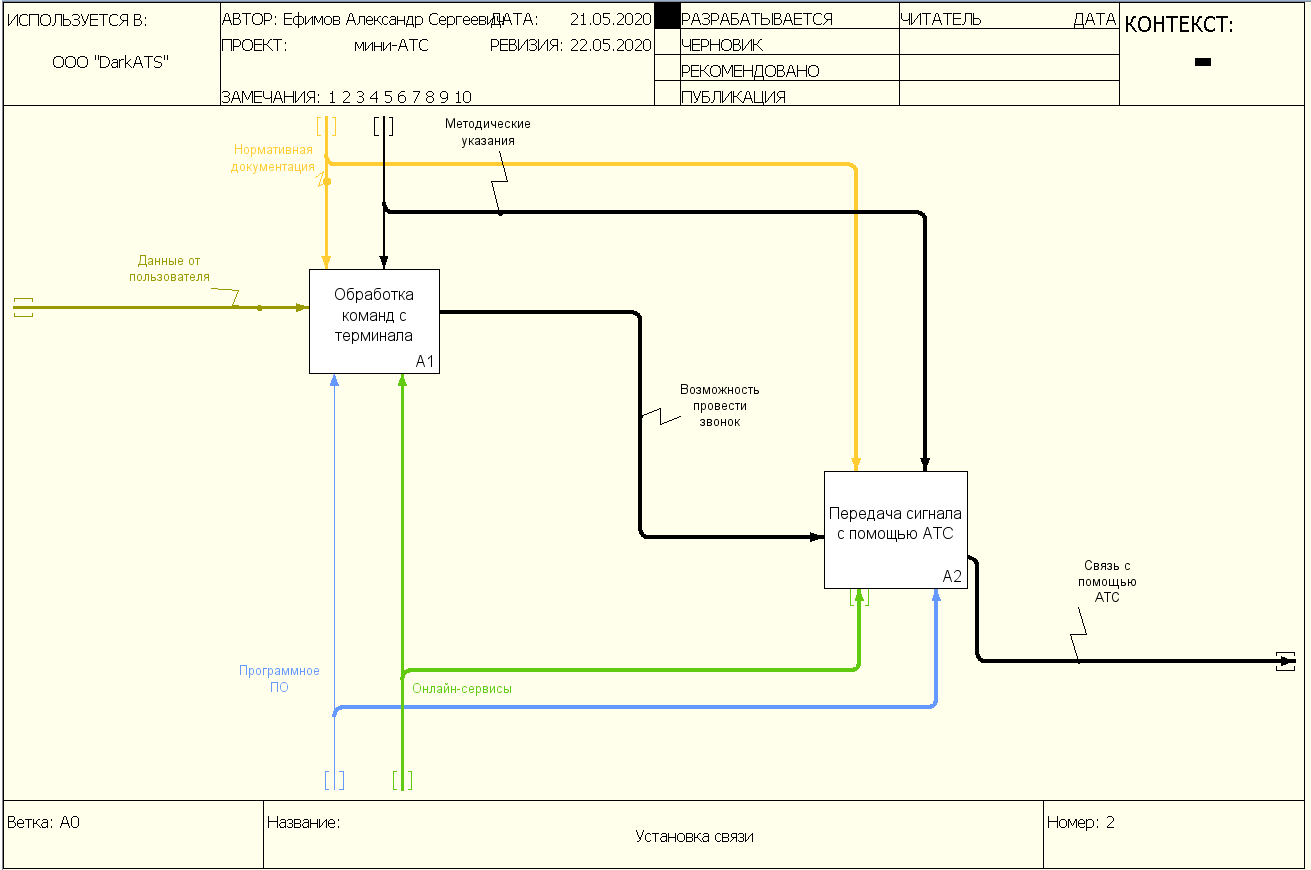


Рисунок 2 – Декомпозиция проекта «мини-АТС»

Полученная модель системы может быть представлена в более подробном виде путём разбиения на большее количество составных элементов.

На рисунке 3 можно видеть модель кода «Мини-АТС» после декомпозиции.

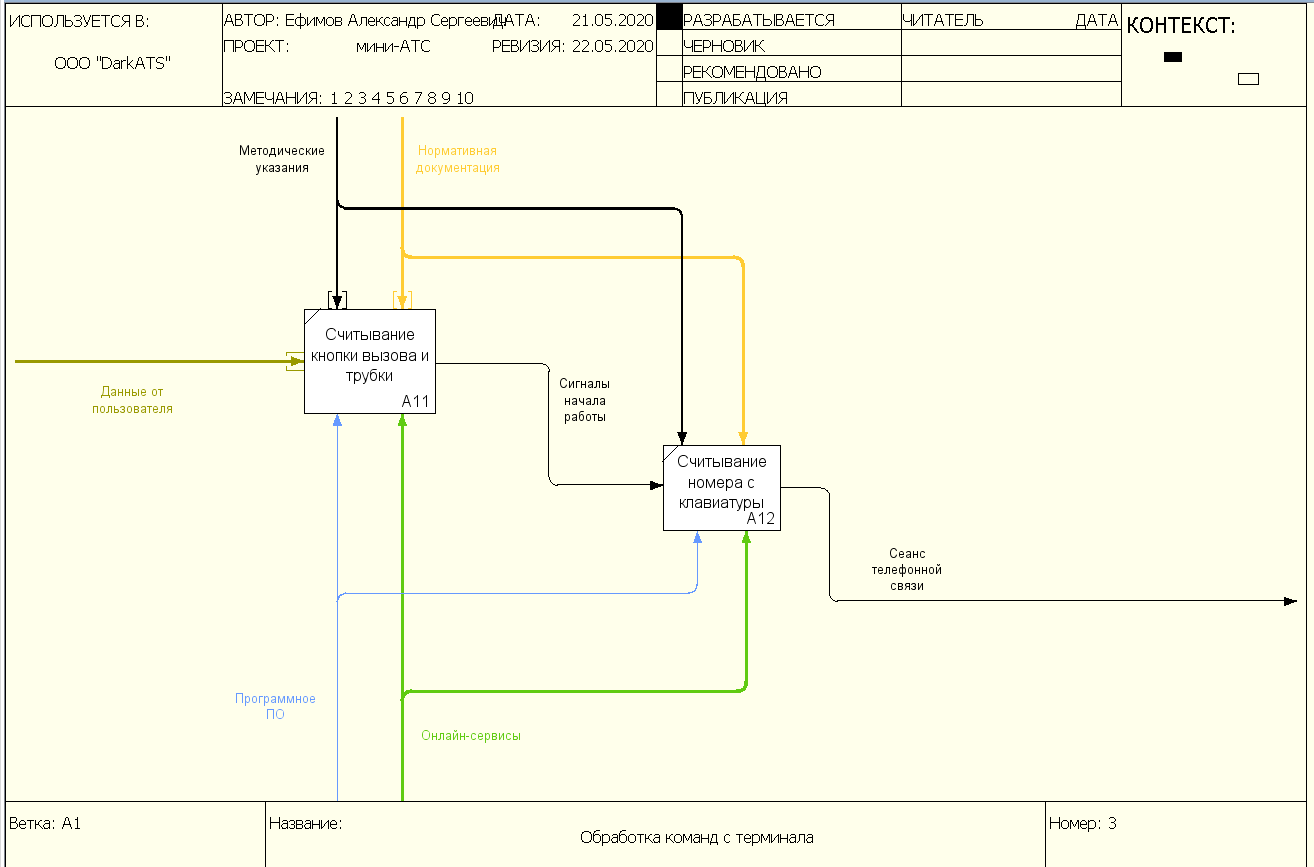


Рисунок 3 – Декомпозиция кода программы

Входными данными для данной системы являются действия человека, необходимые для того, чтобы АТС обработала сигналы.

Управление происходит благодаря кнопочному интерфейсу, контроллеру устройства телефона, который связан с мини-АТС.

Механизмом реализации работы системы являются пользователи, панель телефона и программное обеспечение мини-АТС.

Результатом деятельности системы является текстовое уведомление о передаче сигнала.

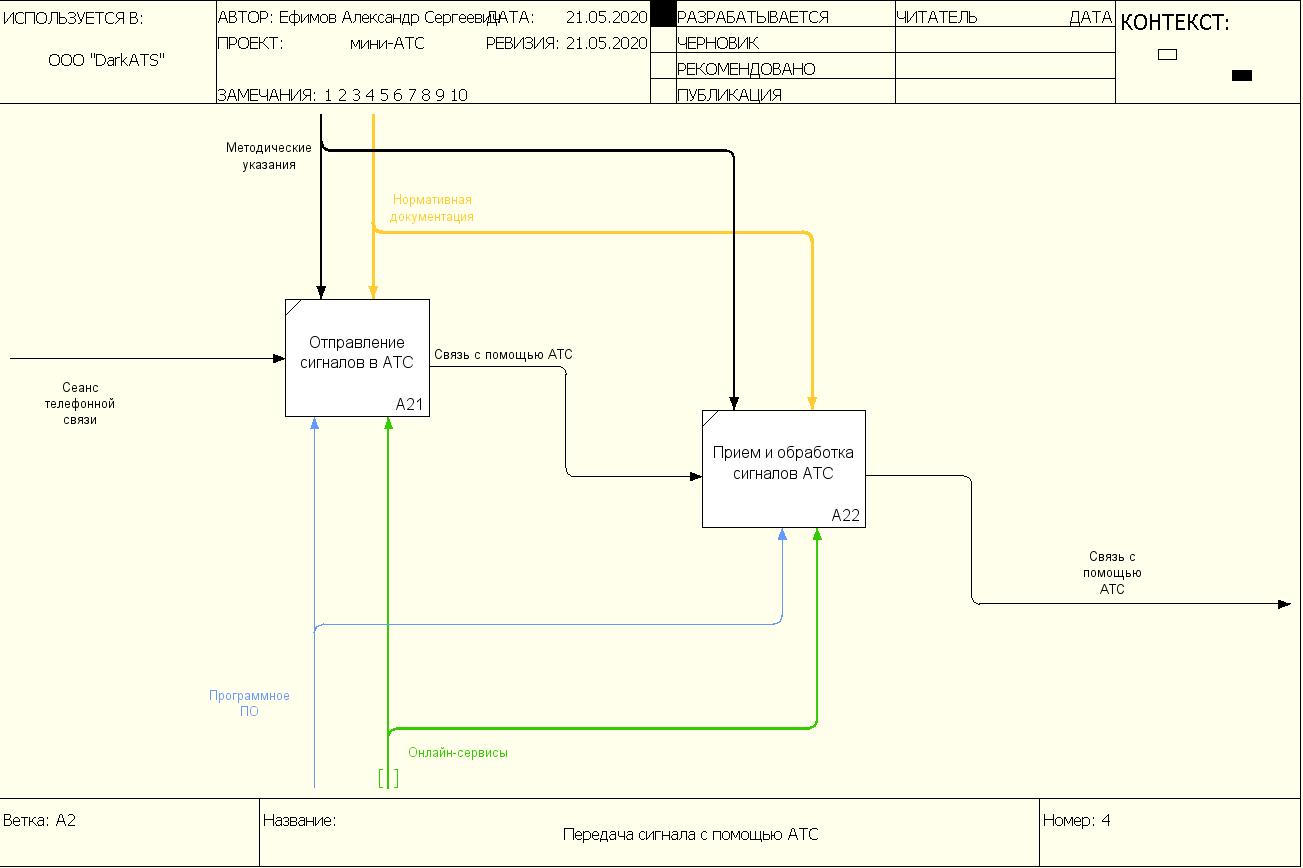


Рисунок 4 – Декомпозиция представления как должен выглядеть симулятор

Таким образом работа, рассматриваемой системы разбивается на три основанных функции:

– приём входных данных от пользователя;

– обработка сигнала мини-АТС;

– соединение между пользователями.

**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 «Мини-АТС» была получена следующая диаграмма DFD (рис. 5).

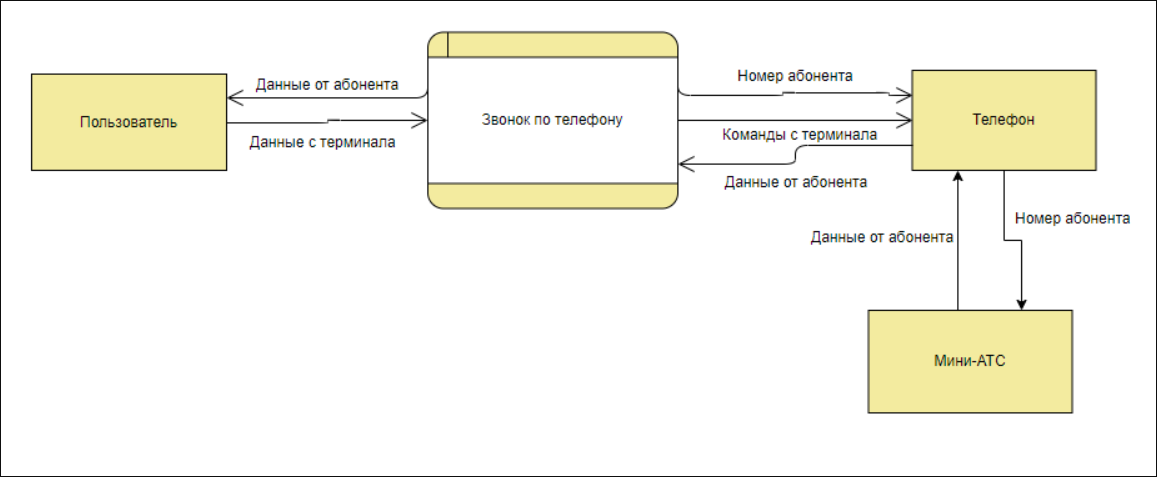


Рисунок 5 – Диаграмма DFD-системы «мини-АТС»

Внешними сущностями данной системы являются пользователь, устройство ввода данных, а также сам мини-АТС. Система не содержит баз данных.

**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.»

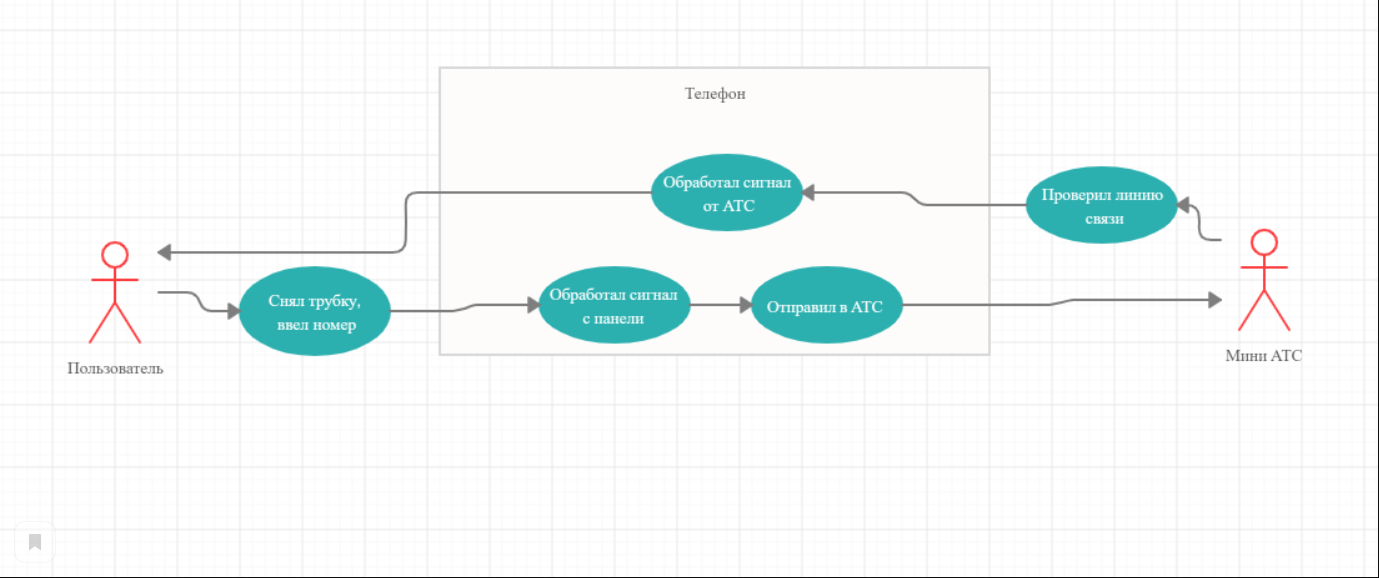


Рисунок 6 – UML-диаграмма системы «мини-АТС»

# 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 инструмент независимый 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.»

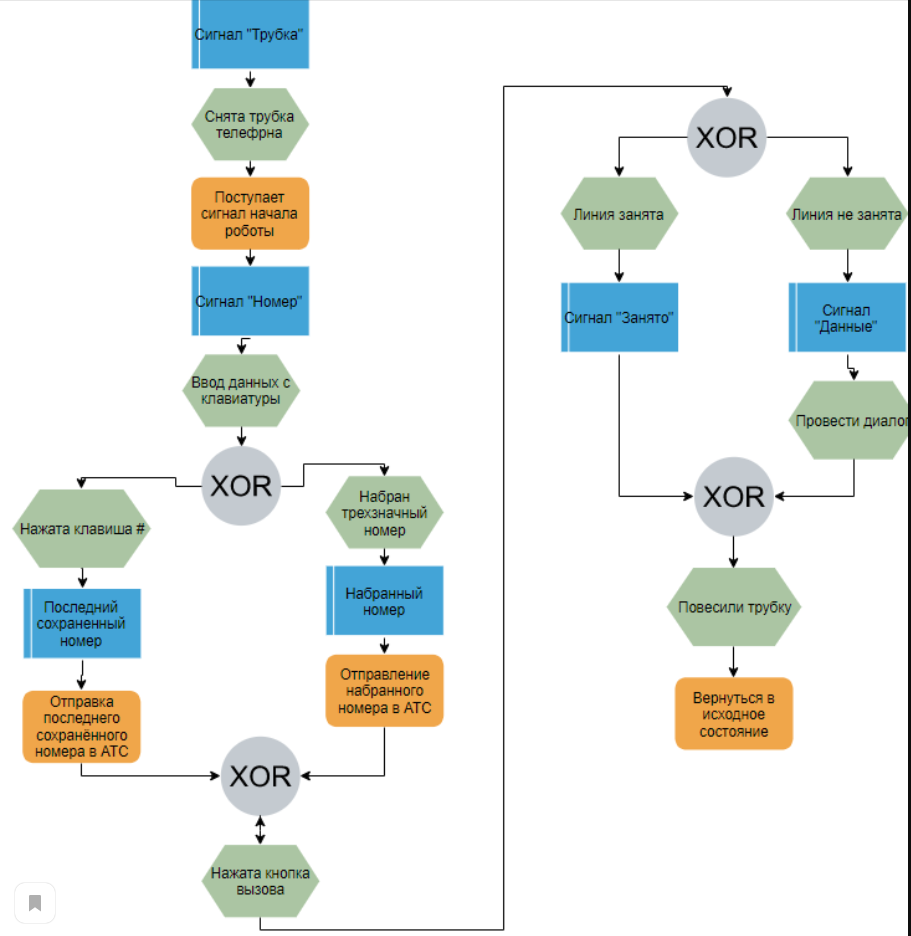


Рисунок 7 – EPC-диаграмма системы «мини»

# 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.»

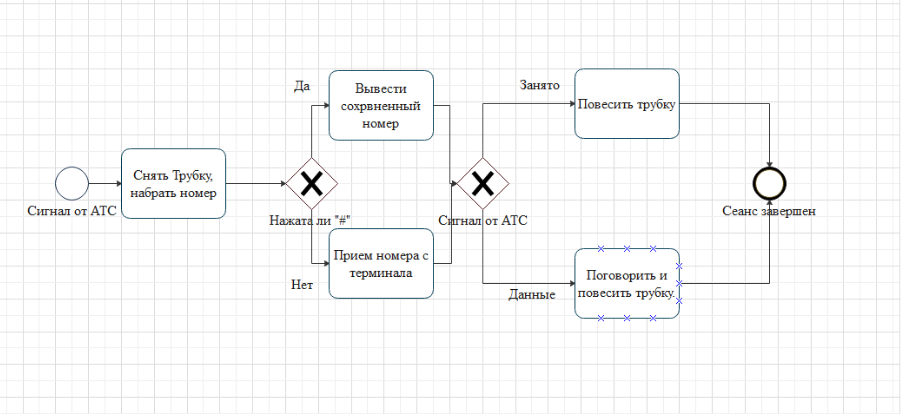


Рисунок 8 – Диаграмма BPMN «As-Is» и «To be»

# 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 – отсутствует;

U – интуитивная понятность;

R – режим доступности 24/7, защита от прослушивания, большая дальность связи;

P – долговечные аккумуляторы, моментальный отклик;

S – простая установка и настройка, не требующая профессиональных знаний;

+ - никаких ограничений

# 9 Результаты машинного тестирования программы



Рисунок 9 – Ввод номера и добавление номера в мини-АТС



Рисунок 10 – Неудачный вызов абонента 123



Рисунок 11 – Второй вариант неудачного вызова 321

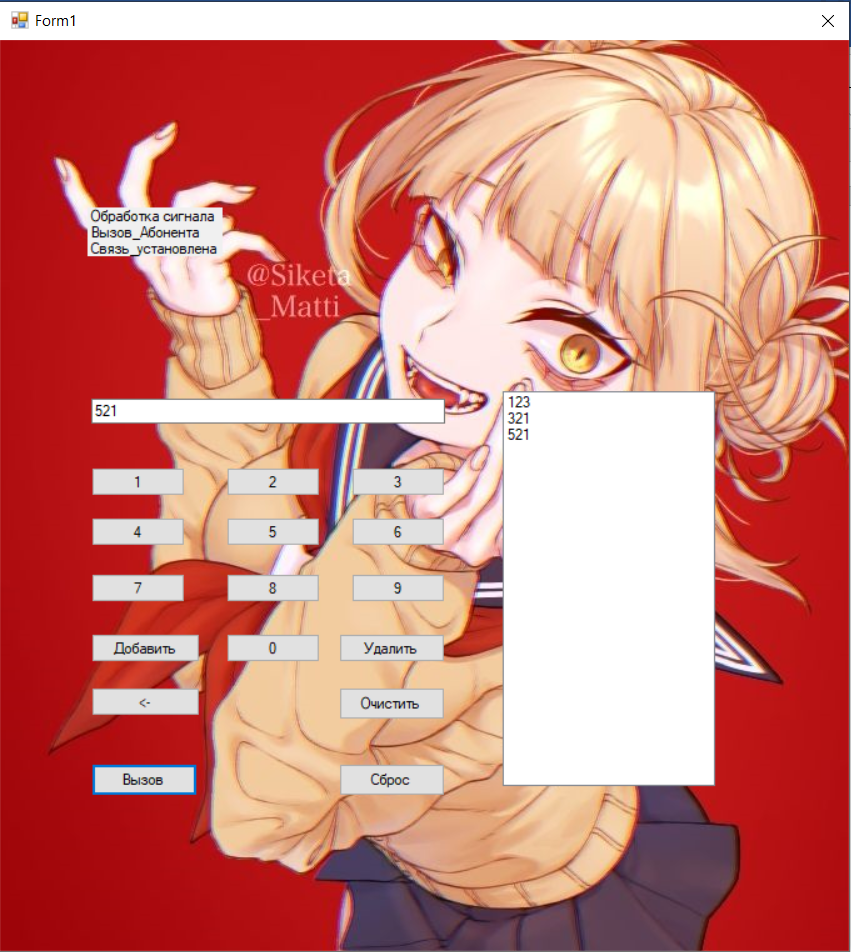


Рисунок 12 – Установка связи 521 и активная кнопка сброс

**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**

Запуск программы можно осуществить следующим способом: открыть исполняемый файл.

Запуск программы при помощи исполняемого файла:

* 1. Находим исполняемый файл.
  2. Запускаем исполняемый файл с форматом .exe.

Теперь перед нами интерфейс программы: панель ввода номера абонента, книга мини-АТС, оповещения о работе мини-АТС.

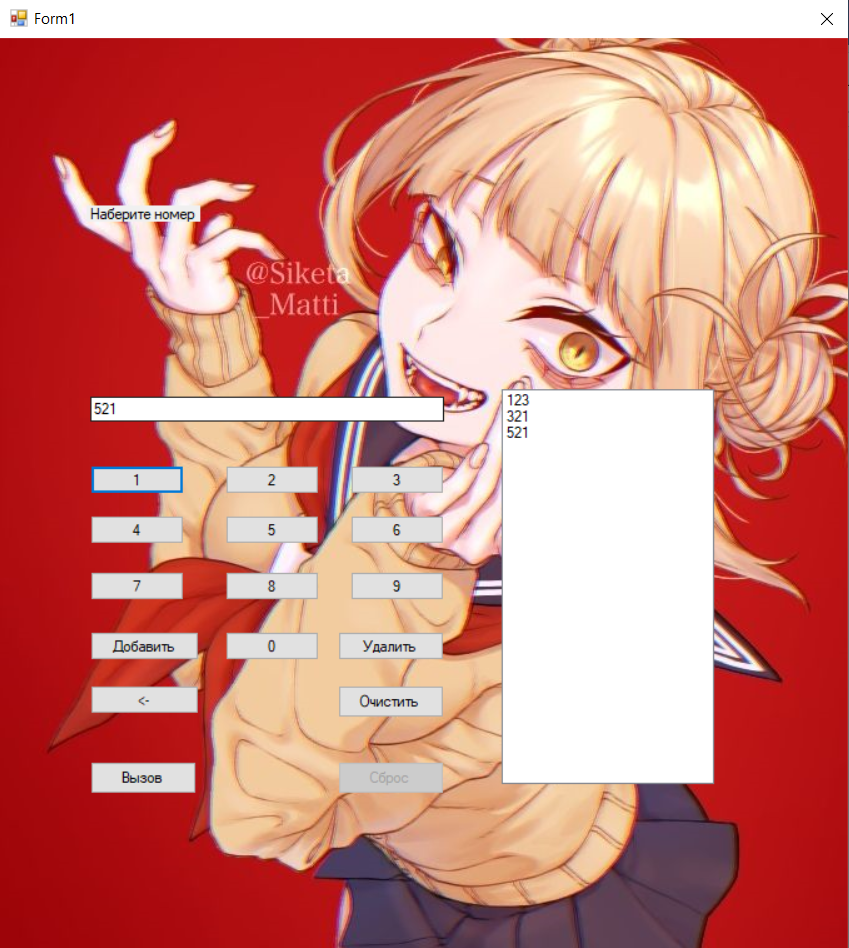


Рисунок 13 – Начальный интерфейс программы

Для начала работы пользователь должен ввести номер абонента, на который есть ограничения: он состоит из 3 цифр. Далее надо нажать на кнопку «Вызов» или «Очистить», если введен не верный номер. После нажатия кнопки «Вызов», текстовое оповещение укажет о состояние сигнала. Во время работы запроса можно нажать на кнопку «Вызов», «Очистить», «🡨», «Удалить» и «Сброс» если соединение установлено. После установки связи играет сигнал и активируется кнопка «Сброс». Чтобы сбросить до стартового экрана, нужно нажать на кнопку «Сброс».

До нажатия кнопки «Вызов» нужно ввести номер в панель управления.

**Заключение**

В результате выполнения данного курсового проекта была спроектирована система «Мини-АТС» на языке высокого уровня C#, которая была соединена с системой «Телефон». Полученные диаграммы обладают 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.

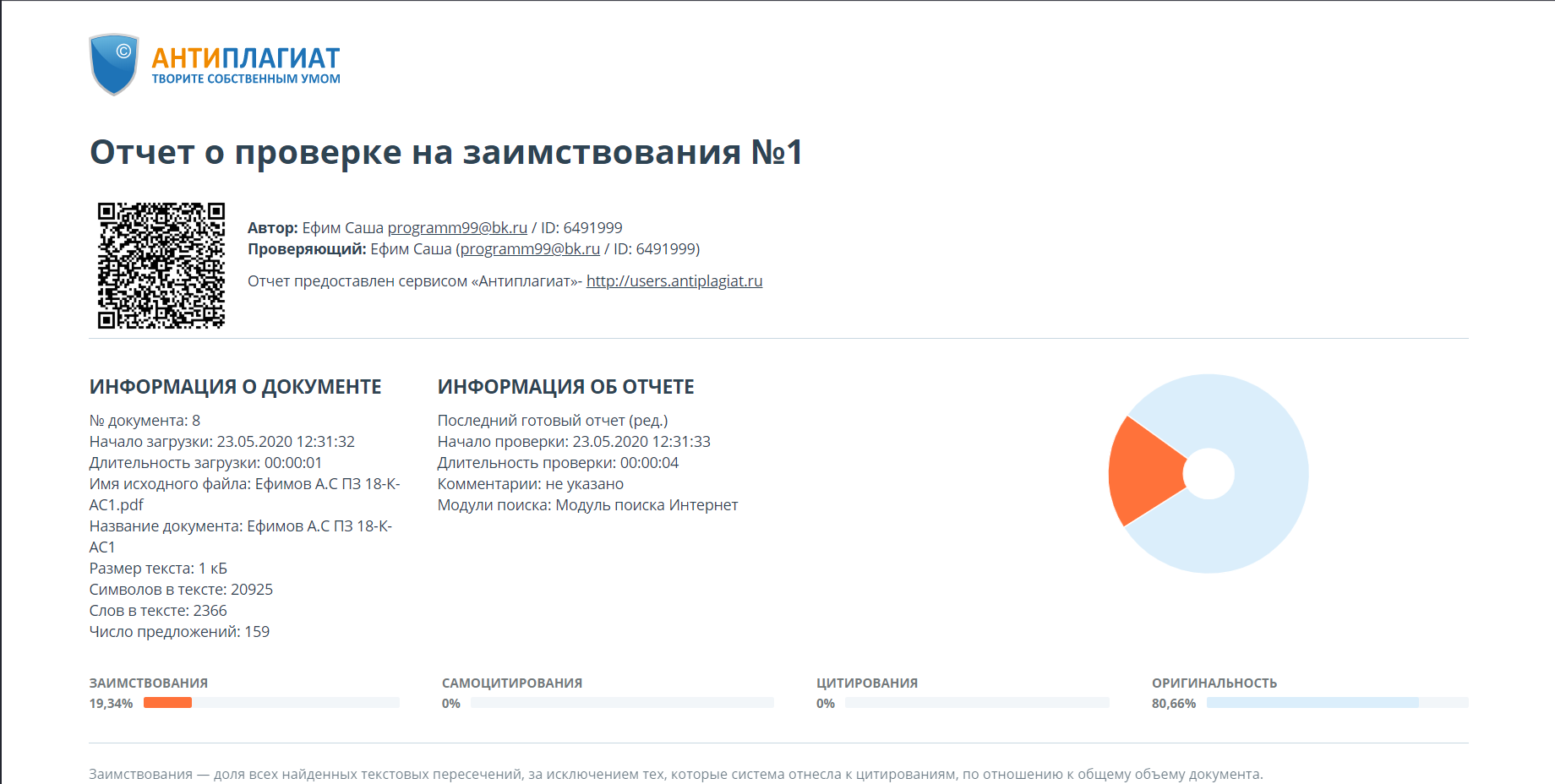
Во время выполнения работы, разработка «Мини-АТС» несла в себе новые знания о работе с языком высокого уровня С#. Новая программа объединяет в себе две разные программы, которые взаимодействуют между собой и не конфликтуют.

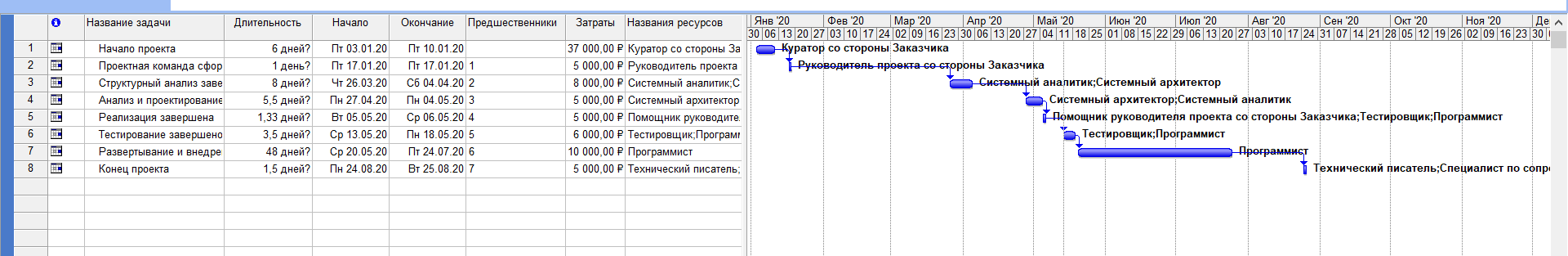
Были получены важные знания и практические навыки как в области использования объектно-ориентированных языков программирования в целом, так и в области построения диаграмм проектирования, отображающих поведение различных организационных структур.

**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 – yarajtf/intercom. [Электронный ресурс]: - Режим доступа: <https://github.com/yarajtf/intercom> (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

# Приложение В – Листинг программы

Таблица 2 – Характеристика переменных к заданию

|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Смысл переменной | Назначение переменной | Ограничения |
| ats | Список | Исходная | Целая |
| atss | Список | Исходная | Целая |
| rnd | Рандом | Исходная | Нет |
| i | Очередь | Вспомогательная | Целая |
| lenght | Длинна | Исходная | Целая |
| text | Текст | Исходная | Булевая |
| textBox1.Text | Значение на панели вызова | Исходная | Булевая |
| pictureBox1.BackgroundImage | Картинка | Исходная | Нет |

Код программы:

namespace Курсач

{

public partial class Form1 : Form

{

public Form1() : base()

{

InitializeComponent();

}

enum ats : int

{

Абонент\_недоступен, Вызов\_Абонента

}

enum atss : int

{

Абонент\_недоступен, Связь\_установлена, Перезвоните\_позже

}

Random rnd = new Random();

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

textBox1.Text += "1";

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text += "2";

}

private void button3\_Click(object sender, EventArgs e)

{

textBox1.Text += "3";

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Text += "4";

}

private void button5\_Click(object sender, EventArgs e)

{

textBox1.Text += "5";

}

private void button6\_Click(object sender, EventArgs e)

{

textBox1.Text += "6";

}

private void button7\_Click(object sender, EventArgs e)

{

textBox1.Text += "7";

}

private void button8\_Click(object sender, EventArgs e)

{

textBox1.Text += "8";

}

private void button9\_Click(object sender, EventArgs e)

{

textBox1.Text += "9";

}

private void button10\_Click(object sender, EventArgs e)

{

textBox1.Text += "0";

}

private void button11\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

listBox1.Items.Add(textBox1.Text);

}

}

private void button12\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

listBox1.Items.Remove(textBox1.Text);

}

}

private void button14\_Click(object sender, EventArgs e)

{

textBox1.Clear();

}

private void button13\_Click(object sender, EventArgs e)

{

int lenght = textBox1.Text.Length - 1;

string text = textBox1.Text;

textBox1.Clear();

for (int i = 0; i < lenght; i++)

{

textBox1.Text = textBox1.Text + text[i];

}

}

private void button15\_Click(object sender, EventArgs e)

{

button16.Enabled = false;

label1.Text = "Обработка сигнала";

label1.Visible = true;

Thread.Sleep(300);

if(label1.Text != " ")

{

ats n = (ats)rnd.Next(2);

label1.Text += "\n" + n.ToString();

if(n == ats.Абонент\_недоступен)

{

Thread.Sleep(300);

label1.Text += "\n" + atss.Перезвоните\_позже.ToString();

}

else

{

Thread.Sleep(300);

atss n1 = (atss)rnd.Next(2);

label1.Text += "\n" + n1.ToString();

if (n1 == atss.Связь\_установлена)

{

button16.Enabled = true;

}

}

}

}

private void button16\_Click(object sender, EventArgs e)

{

label1.Text = "Наберите номер";

button16.Enabled = false;

}

}

}