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

Яруллин Я.Р.

(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: Яруллину Я.Р группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Домофон»

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, 14 рисунков, 8 используемых источников.

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

Объектом исследования является программное обеспечение и симулятор домофона, который способен открывать дверь при поднесении ключа, автоматически закрывать её по истечению определённого времени, а также оповещать жильцов строения о желании незнакомца войти внутрь.

Цель работы состоит в разработке проекта программного обеспечения «Домофон» 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 домофоны служат одним из основных средств защиты частной собственности от нежелательных лиц, несущих угрозу для общества. К числу достоинств, послуживших причиной такого глобального распространения технологии, можно отнести простоту обращения, надёжность, возможность обеспечить высокую степень защиты. Применение домофонов в жилых зданиях стала неотъемлемой частью человеческой жизни.

Однако, несмотря на повсеместное использование данной технологии, она никогда не смогла бы обеспечить надлежавший уровень защиты без тщательного исследования предметной области и проведения различных тестов, учитывающих всевозможные взаимодействия с компонентами системы. Для уменьшения денежных затрат стали использовать симуляторы - «имитаторы (обычно механические или компьютерные), задача которых состоит в имитации управления каким-либо процессом, аппаратом или транспортным средством».

Таким образом, симулятор домофона – имитатор домофонной системы, который позволяет управлять вызывным блоком и переговорным устройством.

**1 Формулировка задачи**

Задачей данного курсового проекта является разработка модели программного обеспечения встроенного микропроцессора для домофона, регулирующего доступ в помещение. Домофон должен состоять из следующих компонентов:

– панели с кнопками для ввода номера квартиры и двумя параметризованными кнопками «Вызов» и «Стереть» для осуществления дополнительных операций;

– цифрового дисплея для визуального отображения, вводимой информации;

– электромеханическая дверь, способная полностью заблокировать вход в строение;

– звонка для оповещения жильцов о желании незнакомого человека проникнуть в здание.

Панель с кнопками устанавливается с наружной стороны двери, домофон устанавливается с внутренней стороны двери, звонок устанавливается внутри охраняемого помещения. В обычном состоянии дверь домофона закрыта. Попытка доступ в помещение осуществляется после набора номера, состоящего из трех цифр. Во время набора номера квартиры введенные цифры отображаются на дисплее. Если номер набран и нажата соответствующая кнопка, то осуществляется вызов в данную квартиру, после чего человек внутри квартиры должен решить – открывать дверь или нет. Содержимое дисплея очищается. Кнопка «Вызов» используется для подачи звукового сигнала внутри помещения. Кнопка «Стереть» используется для смены номера квартиры. Внутри квартиры есть переговорное устройство, оснащенное тремя кнопками: «Связь», «Открыть дверь», «Блокировка». Кнопка «Связь» отвечает за активацию функции переговора между людьми. Кнопка «Открыть дверь» отвечает за открытие двери для человека снаружи. Кнопка « Блокировка» блокирует звуковой сигнал и доступ к данной квартире.

# 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 для проекта «Домофон» 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 бизнес-процессов относительно конкурентов. 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 «Домофон».

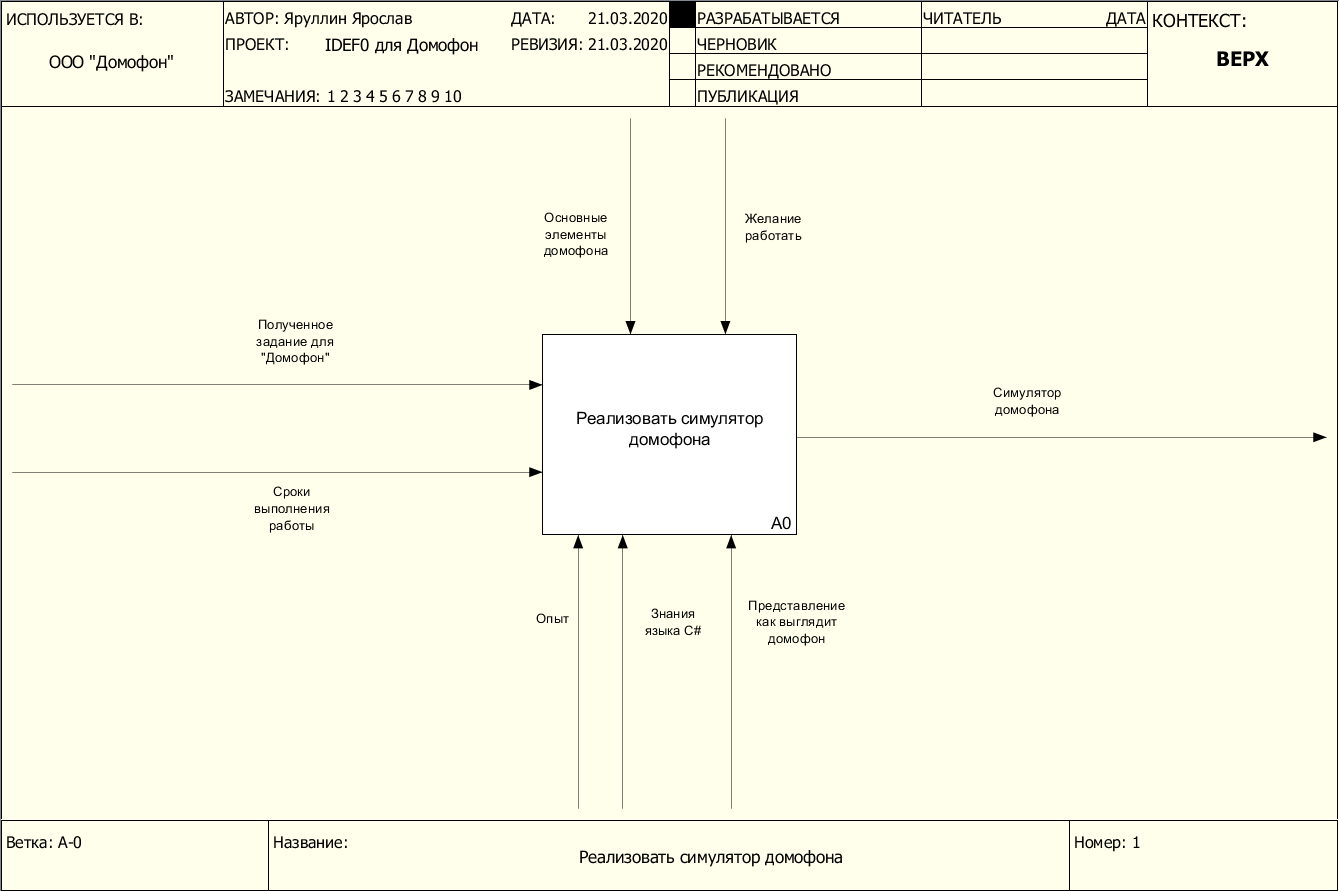


Рисунок 1 – Модель As-Is проекта «Домофон»

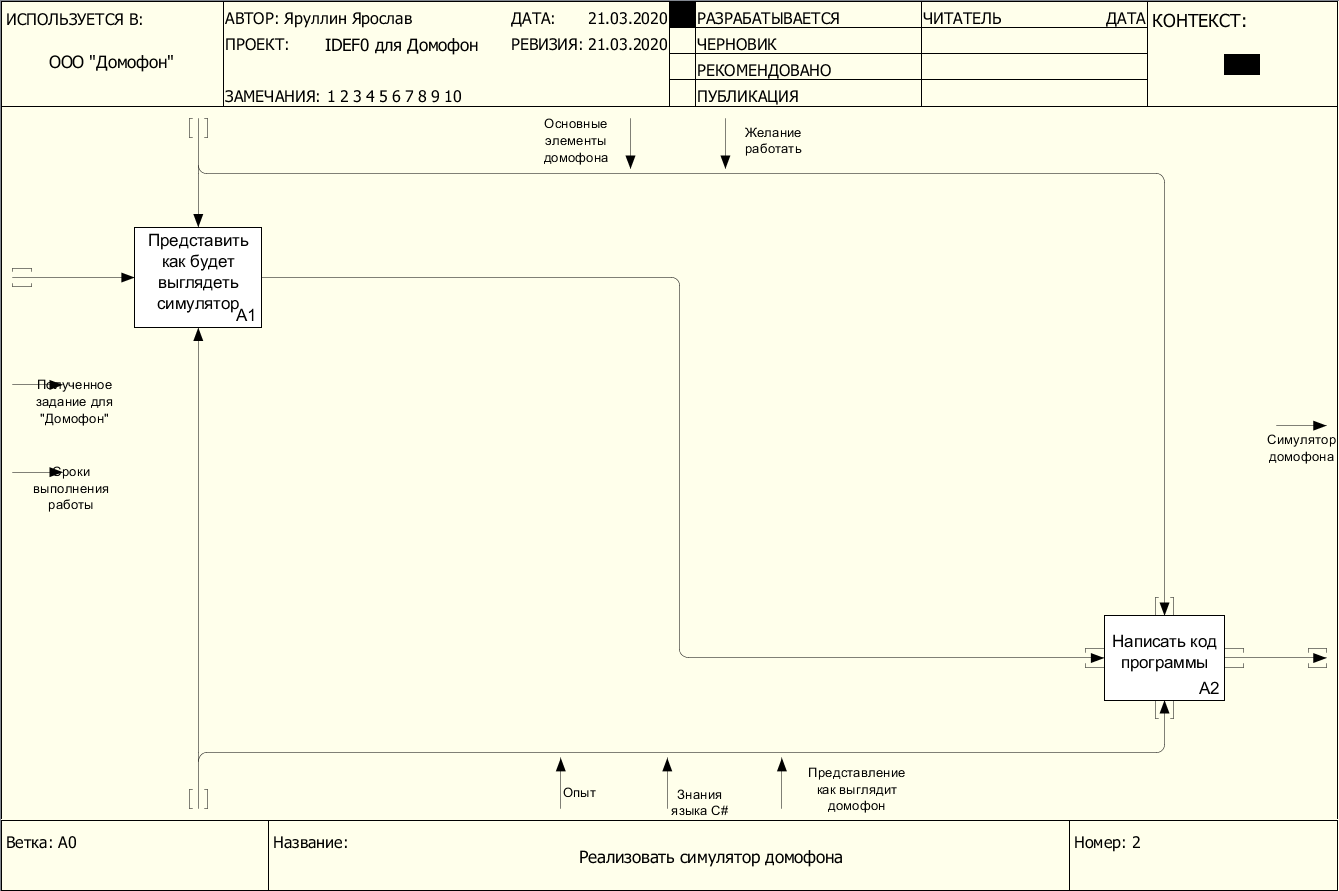


Рисунок 2 – Декомпозиция проекта «Домофон»

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

На рисунке 3 можно видеть модель кода «Домофон» после декомпозиции.

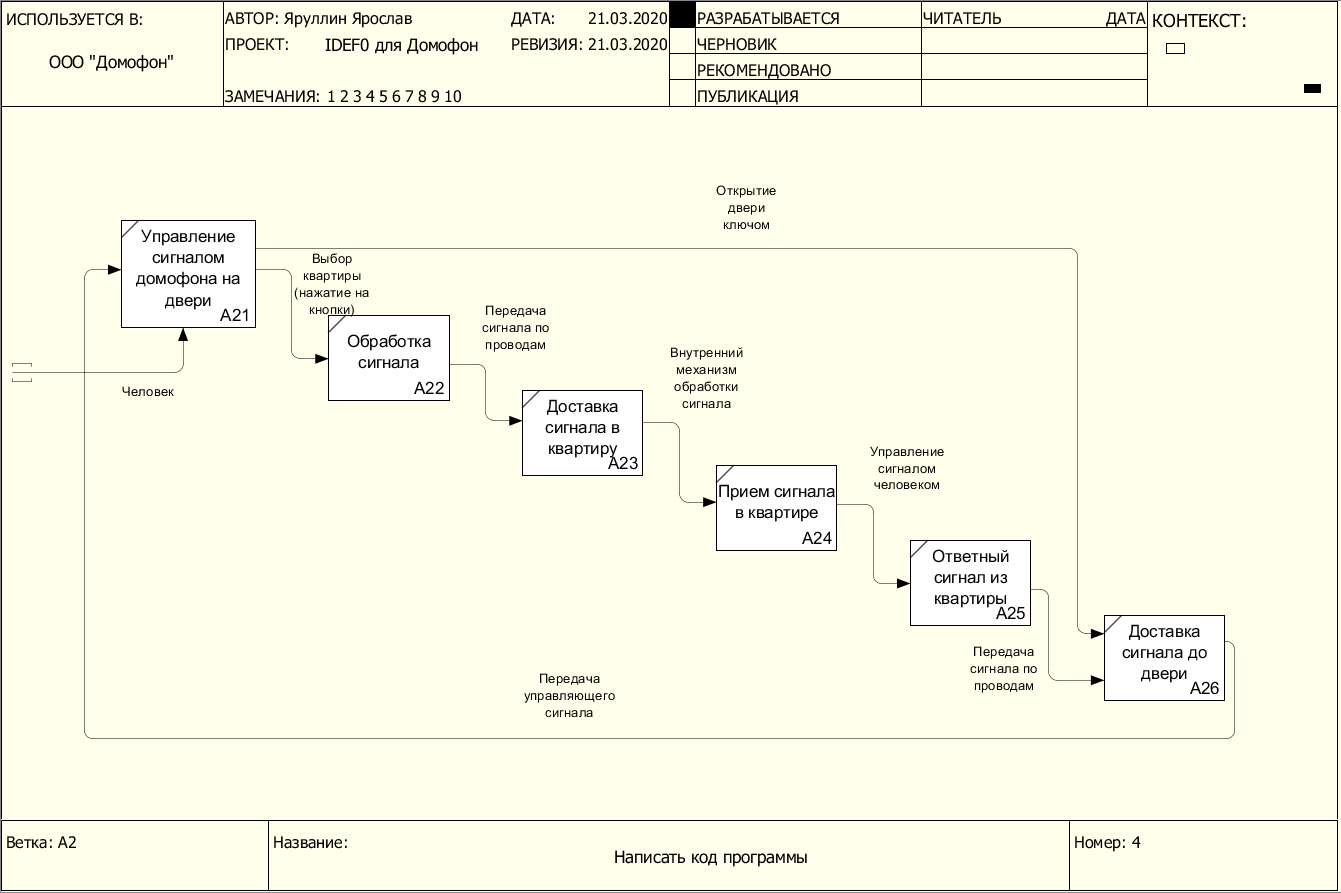


Рисунок 3 – Декомпозиция кода программы

Входными данными для данной системы являются действия человека, необходимые для того, чтобы открыть дверь, а также нажатие на кнопку звонка, предназначенного для оповещения жильцов дома.

Управление происходит благодаря кнопочному интерфейсу, контроллеру устройства домофона, а также правилам, описанным в информационной безопасности.

Механизмом реализации работы системы являются жильцы дома, ключ от домофона и программное обеспечение домофона.

Результатом деятельности системы является подача звукового сигнала для оповещения жильцов дома.

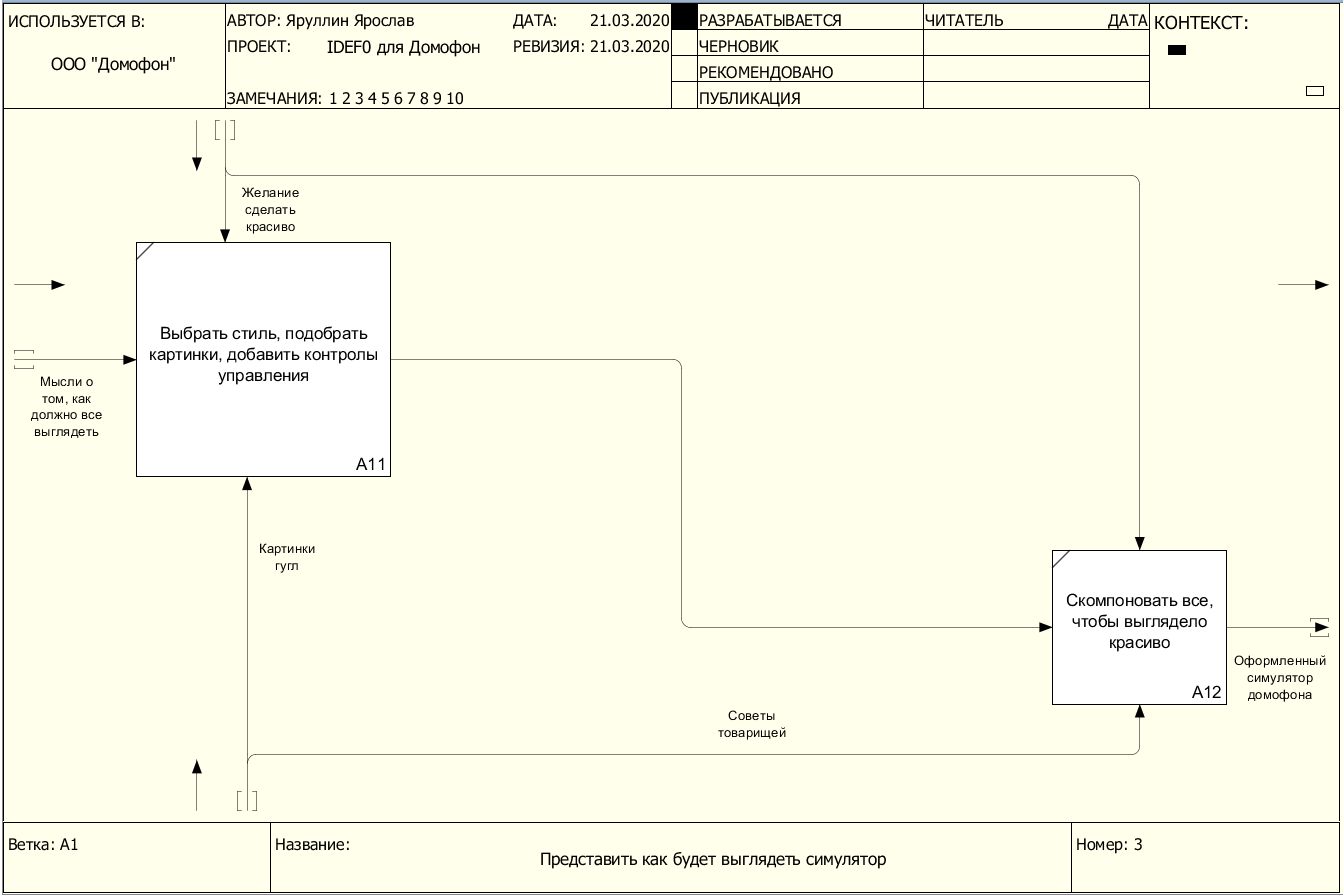


Рисунок 4 – Декомпозиция представления как должен выглядеть симулятор

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

– приём входных данных от пользователя;

– анализ полученного номера квартиры;

– воздействие на дверь человек из квартиры.

**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 «Домофон» была получена следующая диаграмма DFD (рис. 5).

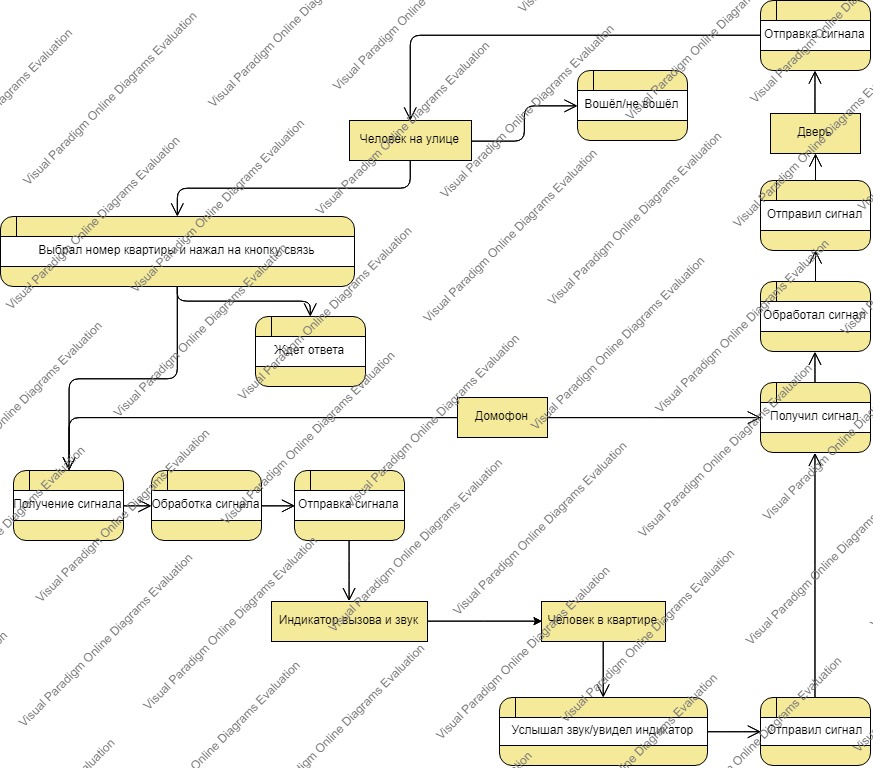


Рисунок 5 – Диаграмма DFD-системы «Домофон»

Внешними сущностями данной системы являются человек на улице, человек в квартире, а также сам домофон. 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.»

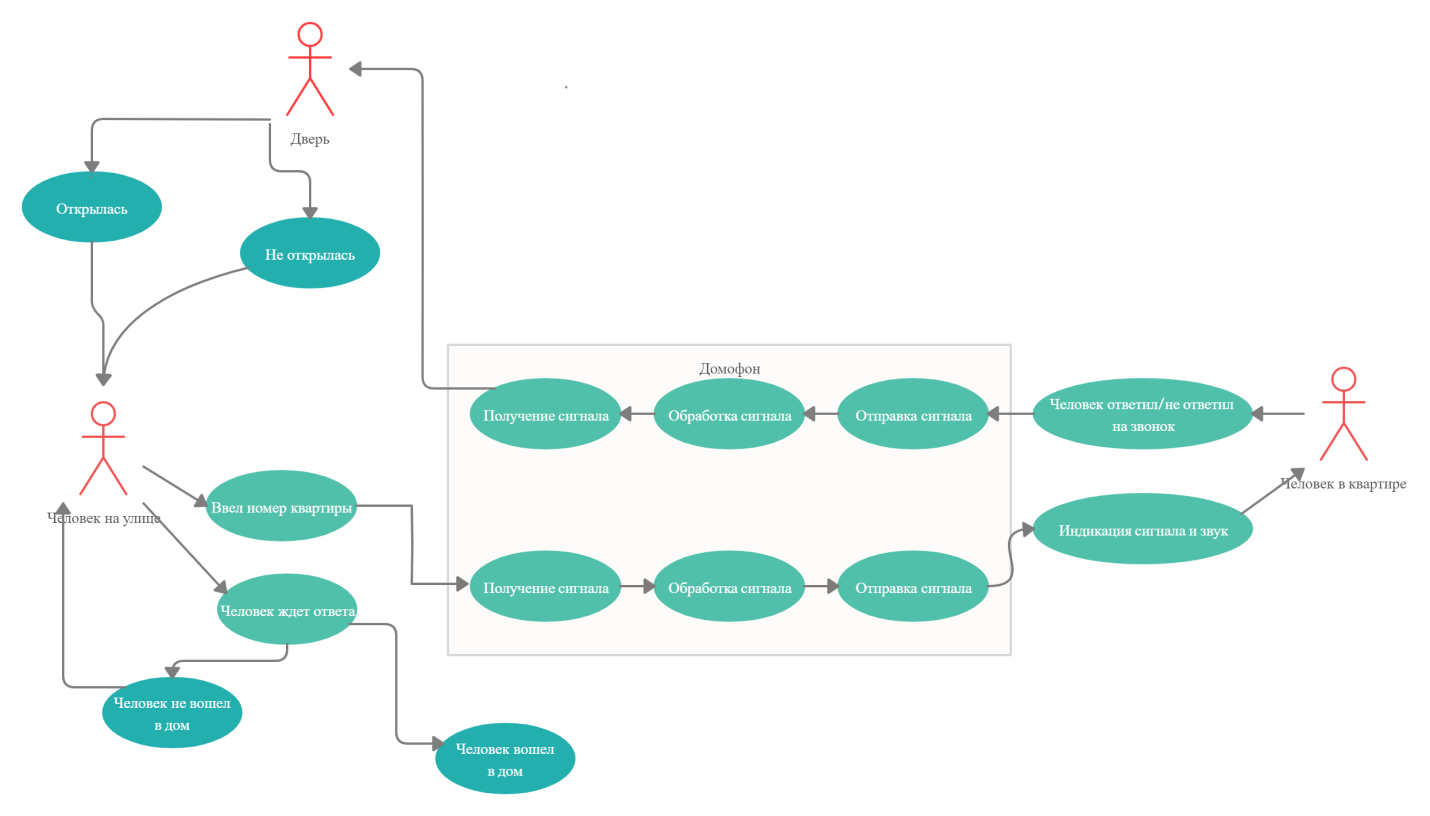


Рисунок 6 – UML-диаграмма системы «Домофон»

# 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.»

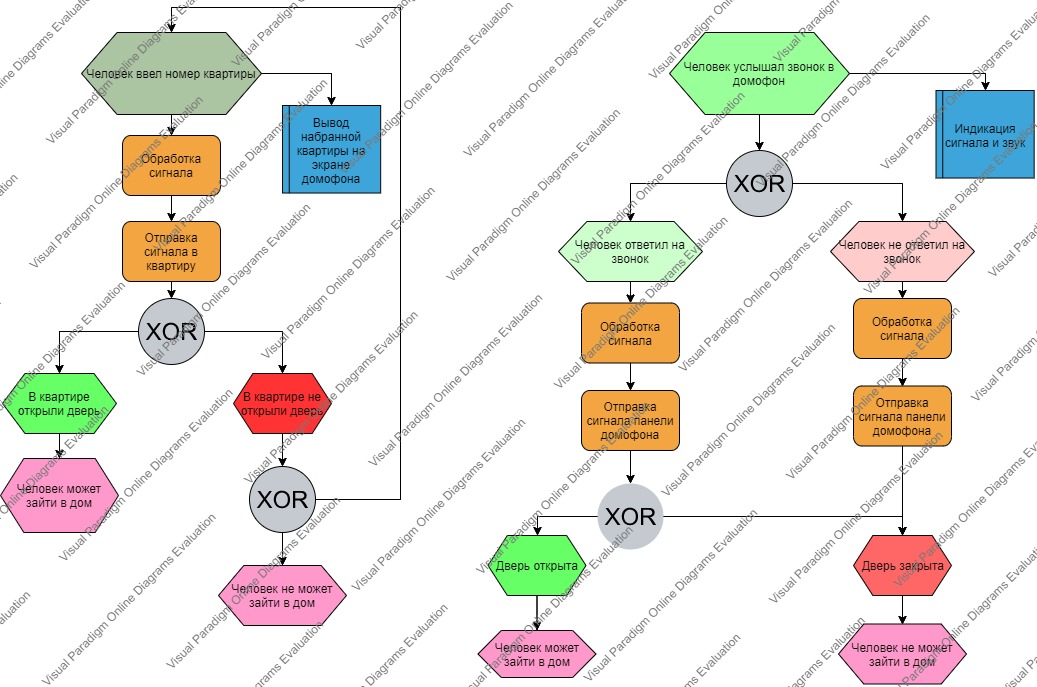


Рисунок 7 – EPC-диаграмма системы «Домофон»

# 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.»

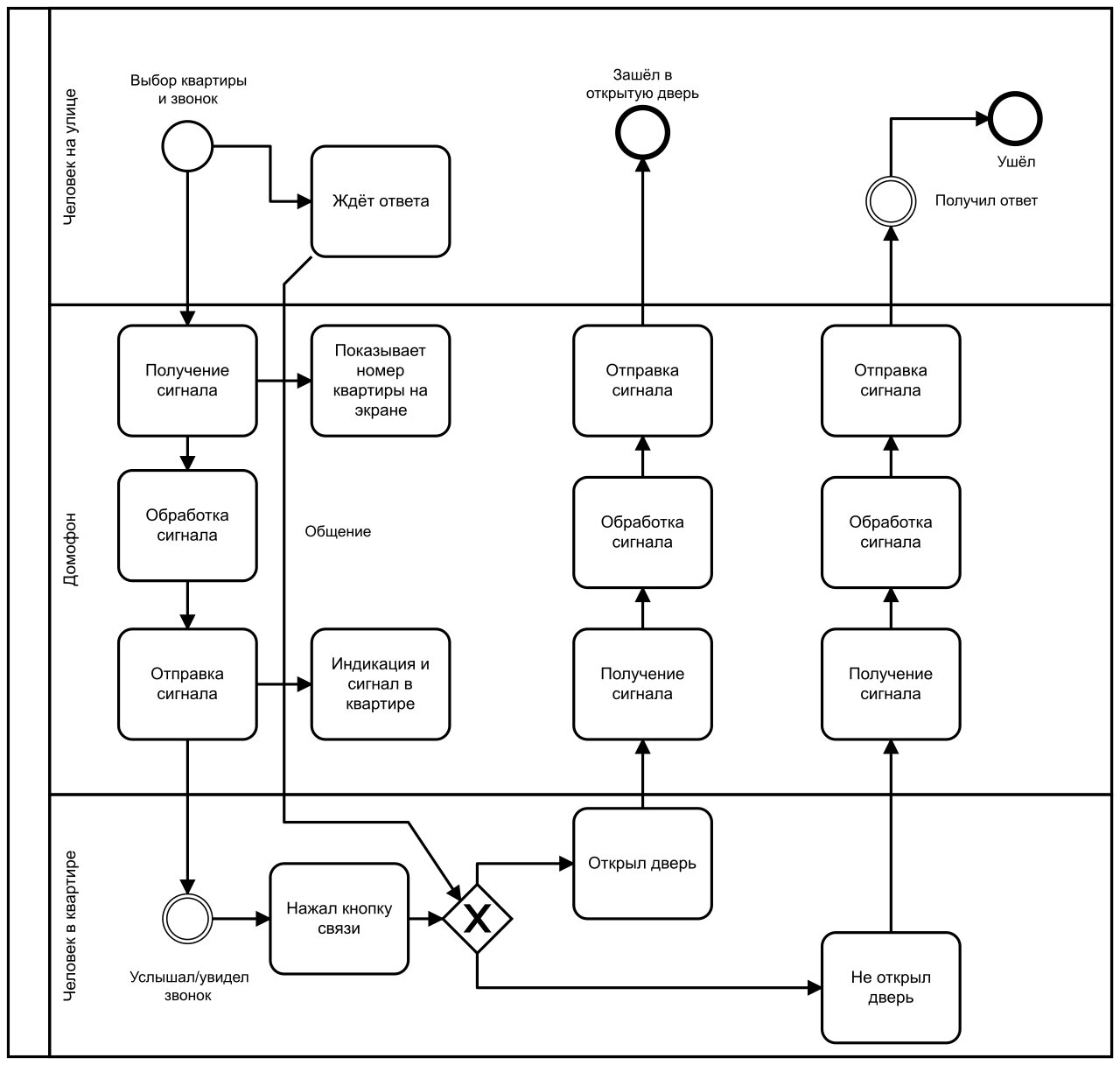


Рисунок 8 – Диаграмма BPMN «As-Is» и «To be»

# 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+.

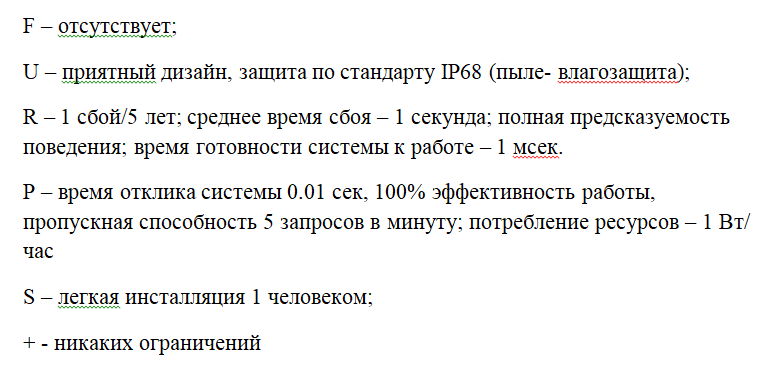


Рисунок 9 – FURPS+ для системы «Домофон»

# 9 Результаты машинного тестирования программы



Рисунок 10 – Открытие двери домофона с ключа



Рисунок 11 – Вызов квартиры 111



Рисунок 12 – Открытие двери, если был вызов



Рисунок 13 – Блокировка вызова

**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 квартиры, ключ от домофона, блок управления домофона в квартире.



Рисунок 14 – Начальный интерфейс программы

Для начала работы пользователь должен ввести номер квартиры, на который есть ограничения: он не должен начинаться с 0. Далее надо нажать на кнопку «Вызов» или «С», если введена не та квартира. После нажатия кнопки «Вызов» кнопка «Вызов» больше недоступна для нажатия. Программа издаст характерный звук – сигнал вызова, который длится, пока таймер программы не достигнет определенного значения. Во время работы таймера можно нажать лишь 1 кнопку управления – «Связь», которая в интерфейсе обозначена кнопкой с телефонной трубкой. После нажатия на «Связь» играет характерный звук и запускается новый таймер – таймер разговора. Во время работы таймера можно нажать на кнопку с изображением двери – «Открыть дверь». После чего издается звук открытия двери, картинка закрытой двери меняется на картинку открытой двери. После окончания таймера дверь закрывается и картинка меняется на закрытую дверь.

До нажатия кнопки «Вызов» работает кнопка с замком – «Блокировка» - она не позволяет вызывать квартиру, т.е. в данной квартире отключен звук.

В любой момент времени можно открыть дверь при помощи домофонного ключа: надо потянуть за домофонный ключ и навести его на соответствующее место.

**Заключение**

В результате выполнения данного курсового проекта была спроектирована система «Домофон» 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 «Домофон» 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 – 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



# WINPROJ_vkHjHnGy2wПриложение Б – Диаграмма Ганта

# Приложение В – Листинг программы

Таблица 2 – Характеристика переменных к заданию

|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Смысл переменной | Назначение переменной | Ограничения |
| Bip | Звук вызова | Исходная | Нет |
| Svyaz | Звук связи | Исходная | Нет |
| Open | Звук открытой двери | Исходная | Нет |
| vizov | Флаг вызова | Исходная | Булевая |
| otvet | Флаг ответа | Исходная | Булевая |
| open | Флаг открытой двери | Исходная | Булевая |
| green | Флаг зеленого индикатора | Исходная | Булевая |
| flagsvyaz | Флаг первого нажатия кнопки «связь» | Вспомогательная | Булевая |
| flag | Флаг нажатия кнопки «блокировка» | Исходная | Булевая |

*Продолжение таблицы 2*

|  |  |  |  |
| --- | --- | --- | --- |
| isDown | Флаг нажатия левой кнопки мыши | Исходная | Булевая |
| textBox1.Text | Значение на панели вызова (номер квартиры) | Исходная | Строковая,  количество символов<=4 |
| timer1.Interval | Интервал 1 таймера | Исходная | Целочисленная |
| timer2.Interval | Интервал 2 таймера | Исходная | Целочисленная |
| timer3.Interval | Интервал 3 таймера | Исходная | Целочисленная |
| timer4.Interval | Интервал 4 таймера | Исходная | Целочисленная |
| timer5.Interval | Интервал 5 таймера | Исходная | Целочисленная |
| pictureBox1.BackgroundImage | Картинка индикатора блокировки | Исходная | Нет |
| pictureBox7.BackgroundImage | Картинка состояния двери | Исходная | Нет |

*Продолжение таблицы 2*

|  |  |  |  |
| --- | --- | --- | --- |
| pictureBox8.BackgroundImage | Картинка ключа от домофона | Исходная | Нет |

Код программы:

namespace Domofon

{

public partial class Form1 : Form

{

SoundPlayer Bip = new SoundPlayer(Domofon.Properties.Resources.pisk);

SoundPlayer Svyaz = new SoundPlayer(Domofon.Properties.Resources.svyaz);

SoundPlayer Open = new SoundPlayer(Domofon.Properties.Resources.open);

public Form1()

{

AllowTransparency = true;

InitializeComponent();

textBox1.Text = ".";

button1.BackColor = System.Drawing.Color.Transparent;

}

bool vizov = false;

bool otvet=false;

bool open = false;

bool green= true;

int flagsvyaz = 1;

long flag = 0;

bool isDown;

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "1";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "2";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "3";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "4";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "5";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "6";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "7";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "8";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "9";

}

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

{

if (textBox1.Text.Length <= 3)

textBox1.Text += "#";

}

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

{

if (textBox1.Text.Length <= 3)

if (textBox1.Text.StartsWith(".") && textBox1.Text.EndsWith(".")) { }

else textBox1.Text += "0";

}

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

{

textBox1.Text = ".";

}

private void timer1\_Tick(object sender, EventArgs e)

{

timer3.Stop(); // прекращаем моргание, значит не ответили

pictureBox1.Visible = true;

timer1.Stop(); // прекращаем ожидание ответа

Bip.Stop(); // остановка звука вызова

textBox1.Text = "."; // обнуляем табло

if(otvet==false && open==false) // если ответа не было, то разблокируем кнопку

button16.Enabled = true;

vizov = false; // вызова больше нет

}

private void timer2\_Tick(object sender, EventArgs e)

{

pictureBox7.BackgroundImage = (Domofon.Properties.Resources.doorclose);

label1.Text = "Дверь закрыта";

timer2.Stop(); // окончание времени открытой двери

button16.Enabled = true; // разблокируем кнопку вызова

open = false; // дверь закрыта

Open.Stop(); // прекращаем звук открытой двери

}

private void timer3\_Tick(object sender, EventArgs e) // моргание зеленого индикатора

{

if (pictureBox1.Image != Domofon.Properties.Resources.КруК)

{

if (pictureBox1.Visible == false)

pictureBox1.Visible = true;

else pictureBox1.Visible = false;

}

}

private void timer4\_Tick(object sender, EventArgs e) // ответили но проигнорили

{

timer4.Stop();

if (otvet == true && open == false)

button16.Enabled = true;

}

private void timer5\_Tick(object sender, EventArgs e) // таймер на открытие с ключа

{

timer5.Stop();

Open.Stop();

vizov = false;

otvet = false;

open = false;

pictureBox4.Enabled = true;

pictureBox5.Enabled = true;

pictureBox7.BackgroundImage = (Domofon.Properties.Resources.doorclose);

label1.Text = "Дверь закрыта";

textBox1.Text = ".";

button16.Enabled = true;

}

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

{

{

if (textBox1.Text.StartsWith(".") && textBox1.Text.EndsWith(".") || green == false) { }

else

{

timer3.Interval = 250;

timer3.Start(); //моргание

timer1.Interval = 10000; // таймер на вызов

timer1.Start();

Bip.PlayLooping(); // играет звук вызова

vizov = true;

button16.Enabled = false;

}

}

}

private void pictureBox4\_Click(object sender, EventArgs e) // ответ

{

flagsvyaz = 1;

timer3.Stop(); // прекращается моргание

Bip.Stop(); // остановка звука

pictureBox1.Visible = true;

if (vizov == true && flagsvyaz == 1)

{

Svyaz.Play();

otvet = true;

flagsvyaz++;

}

if (otvet == false)

{

timer4.Interval = 10000; // таймер на ответ, но неоткрытие

timer4.Start();

}

}

private void pictureBox5\_Click(object sender, EventArgs e) // открыть

{

if (textBox1.Text.StartsWith(".") && textBox1.Text.EndsWith(".") || otvet == false) { }

else

{

pictureBox7.BackgroundImage = (Domofon.Properties.Resources.dooropen);

label1.Text = "Дверь открыта";

open = true;

textBox1.Text = ".";

vizov = false;

otvet = false;

timer2.Interval = 10000; // таймер на открытую дверь

timer2.Start();

Open.PlayLooping(); // играет звук открытой двери

}

}

private void pictureBox8\_MouseUp(object sender, MouseEventArgs e) // открыть с ключа

{

isDown = false;

if (pictureBox8.Location.X > 696 && pictureBox8.Location.X < 756 && pictureBox8.Location.Y > 282 && pictureBox8.Location.Y < 344 && open == false)

{

pictureBox4.Enabled = false;

pictureBox5.Enabled = false;

Bip.Stop();

timer3.Stop();

button16.Enabled = false;

pictureBox7.BackgroundImage = (Domofon.Properties.Resources.dooropen);

label1.Text = "Дверь открыта";

open = true;

textBox1.Text = ".";

pictureBox1.Visible = true;

timer5.Interval = 10000;

timer5.Start();

Open.PlayLooping();

}

pictureBox8.Location = new Point(706, 388);

}

private void pictureBox6\_Click(object sender, EventArgs e) // блокировка

{

if (vizov == false)

{

flag++;

if (flag % 2 == 1)

{

pictureBox1.BackgroundImage = Domofon.Properties.Resources.КруК;

green = false;

}

else { pictureBox1.BackgroundImage = Domofon.Properties.Resources.Кру; green = true; }

}

}

private void pictureBox8\_MouseDown(object sender, MouseEventArgs e)

{

isDown = true;

}

private void pictureBox8\_MouseMove(object sender, MouseEventArgs e)

{

Control c = sender as Control;

if (isDown)

{

c.Location = this.PointToClient(Control.MousePosition);

}

}

}

}