# Морзе ++ документация

# Исходные данные. Бизнес цели

Азбука Морзе является желанным объектом для изучения людьми разного возраста. Она привлекает своей кажущейся простотой, хотя на самом деле овладеть ей довольно сложно. Однако люди с радостью бы преодолевали все возможные препятствия на пути, если бы у них было подходящее для этого программное средство. Подобных средств в открытых источниках можно найти немало, однако найти то, которое будет совмещать в себе «приятное с полезным» , где «полезных» функций будет в достатке, не так уж и легко. В связи с этим возникла проблема – создать удобное, многофункциональное и приятное в использовании программное средство для того, чтобы его пользователи могли обучиться азбуке Морзе.

Идея проекта состоит в создании программного средства, помогающего пользователю обучиться азбуке Морзе.

Предметной областью проекта является азбука Морзе – способ [знакового кодирования](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4" \t "/home/pda/Documents\\x/_blank), представление [букв](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D0%BA%D0%B2%D0%B0" \t "/home/pda/Documents\\x/_blank) [алфавита](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82" \t "/home/pda/Documents\\x/_blank), [цифр](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%B0" \t "/home/pda/Documents\\x/_blank), [знаков препинания](https://ru.wikipedia.org/wiki/%D0%97%D0%BD%D0%B0%D0%BA_%D0%BF%D1%80%D0%B5%D0%BF%D0%B8%D0%BD%D0%B0%D0%BD%D0%B8%D1%8F" \t "/home/pda/Documents\\x/_blank) и других [символов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB" \t "/home/pda/Documents\\x/_blank) последовательностью [сигналов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB" \t "/home/pda/Documents\\x/_blank): длинных («[тире](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D1%80%D0%B5_(%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)" \t "/home/pda/Documents\\x/_blank)») и коротких («[точек](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BA%D0%B0" \t "/home/pda/Documents\\x/_blank)»).

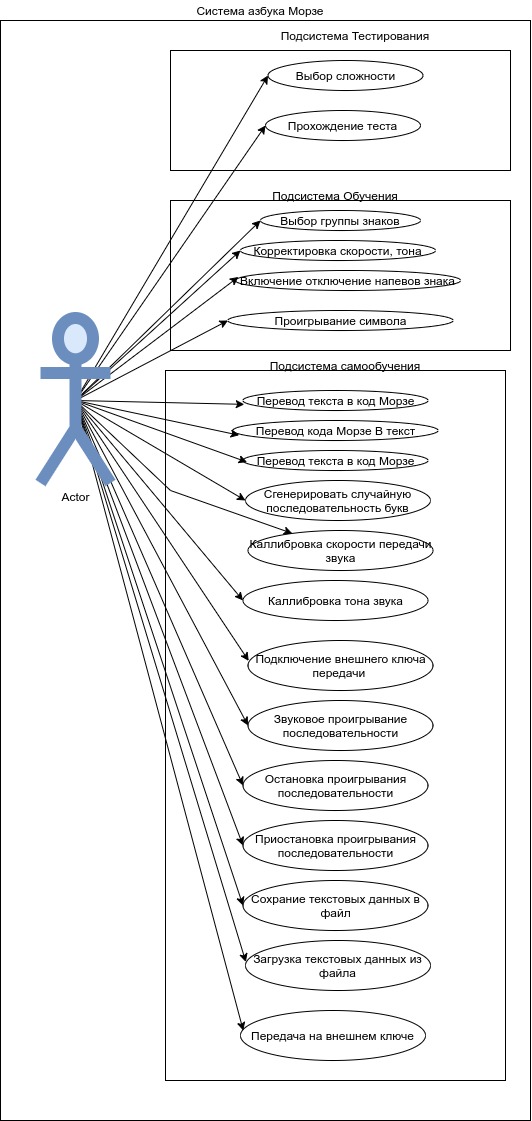
Идеальный конечный результат-программное средство должно само проводить обучение пользователей; само определять уровень подготовки пользователей, знакомых с правилами кодирования. При этом затраты на разработку самого программного средства должны быть минимальными

Программное средство должно быть многофункциональным и тщательно проработанным, чтобы иметь возможность конкурировать с уже существующими программными продуктами, но при этом возникают сложности в плане реализации.

ИКР невозможно достичь по той причине, что способность к обучению у разных людей различна. Кроме того, не всегда можно точно определить уровень подготовки пользователя: при тестировании есть вероятность непреднамеренной ошибки, а высокая скорость воспроизведения сигнала может не позволить определить нужный символ.

Необходимо создать такое программное средство, которое бы обеспечивало возможность эффективного обучения азбуки Морзе, позволяло бы совершенствовать навыки кодирования и с высокой точностью определяло бы уровень подготовки пользователя и его прогресс в освоении «Морзянки»

# Диаграмма вариантов использования



Изображение 1-"Диаграмма Use Cases"

**Подключение внешнего ключа передачи.**

Идентификатор 3-1 (первая цифра-номер подсистемы, вторая цифра-номер UC).

Автор- Проскурин Денис Александрович

Дата создания: ориентировочно 04.2017

Основной актор: обучающийся

Краткое описание: внешний ключ передачи представляет собой механическое устройство для замыкания цепи, посредством которого предаются коротки и длинные звуковые сигналы. Данное устройство необходимо для обучения передачи закодированных сообщений посредством Азбуки Морзе. Внешний ключ представляет собой плату с двумя кнопками (короткий сигнал, длинный сигнал), которая подключается к компьютеру через com-Port. Через это устройство механическими действиями подаются сигналы на компьютер в программу и компьютер воспроизводит звуки указанной скорости и тональности.

Условие тригер: Нажатие соответствовавшего переключателя.

Предварительные условия: подключения внешнего ключа, выбор нужного com-порта.

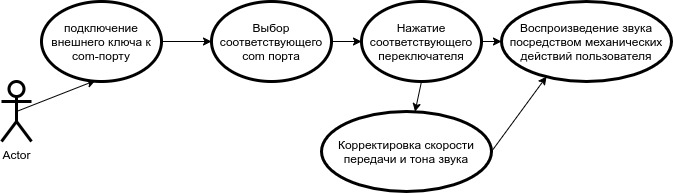
Выходные условия: Нажатие соответствующего переключателя, удаление внешнего ключа из com-порта.

Нормальное направление: воспроизведение звука, которое задаётся механическими действиями пользователя посредством внешнего ключа передачи.

Альтернативные направления: корректировка звука, тона передачи..

Исключения: не имеются.

Последовательность действий данного кейза изображена на рисунке (Изображение 2).



Изображение 2-"UC подключение внешнего ключа передачи"

# User stories

* Я как обучающийся хочу знать правила кодирования русского алфавита чтобы принимать и передавать зашифрованные сообщения.
* Я как обучающийся хочу уметь пользоваться «ключом передачи» чтобы при быстро передавать сообщения на расстояния.

***Критерий приёмки:*** Внешнее устройство должно быть подключено, флаг «Использования внешнего устройтва» должен быть активен.

* Я как обучающийся хочу протестировать свои знания чтобы знать свой уровень знания Азбуки Морзе.

***Критерий приёмки*:** Нужно выбрать соответствующий уровень сложности, ответить на все вопросы теста.

* Я как обучающийся хочу иметь доступ к справочным материалам чтобы узнать необходимую информацию касательно Азбуки Морзе.
* Я как обучающийся хочу иметь возможность корректировки настроек звука для качественного и комфортного обучения.
* Я как обучающийся хочу иметь возможность самопроизвольного кодирования, декодирования Азубуки Морзей для самопроверки.
* Я как обучающийся хочу иметь возможность сохранения и загрзуки кодированных, декодированных данных в(из) файл(а) для самоорганизации учебного процесса.
* Я как обучающийся хочу иметь возможность генерации текста из заданного буквенного диапазона для проверки своих знаний.

# Функциональные требования

В программном средстве должен быть реализован следующий набор функций:

* Перевод информации в код Морзе и из кода Морзе

***1 случай:***

**ВХОД:** Набор букв русского языка и цифры.

**ВЫХОД:** Последовательность точек и тире.

**МЕТОД:** Таблица кодирования «Азбука Морзе».

***2 случай:***

**ВХОД:** Последовательность точек и тире.

**ВЫХОД:** Набор букв русского языка и цифры.

**МЕТОД:** Таблица кодирования «Азбука Морзе».

* Генерация случайной последовательности определенной длины из символов с целью последующего перевода в код Морзе. Перечень символов и длина цепочки задаются пользователем.

**ВХОД:** Длина цепочки, множество символов русского алфавита.

**ВЫХОД**: Цепочка указанной длины.

**МЕТОД**: Каждый символ цепочки получен случайным образом из множества символов русского алфавита, который задал пользователь.

* Передача радиосигнала

**ВХОД:** последовательность нулей и единиц.

**ВЫХОД**: Длинный звуковой сигнал, короткий звуковой сигнал..

**МЕТОД**: Последовательность единиц и нулей задаётся механическими нажатиями на «внешний ключ».

* Вывод справочной таблицы соответствия текстовых и цифровых символов их закодированному представлению
* Воспроизведение кодовых сигналов, соответствующих выбранным пользователем символам, с возможностью настройки скорости и тона воспроизведения.

***1 случай:***

**ВХОД:** Набор букв русского языка и цифры, тон (в Гц), скорость (знаки/в минуту).

**ВЫХОД:** Звуковая последовательность точек и тире.

**МЕТОД:** Таблица кодирования «Азбука Морзе».

***2 случай:***

**ВХОД:** Последовательность точек и тире,тон (в Гц), скорость (знаки/в минуту).

**ВЫХОД:** Звуковая последовательность точек и тире.

**МЕТОД:** Таблица кодирования «Азбука Морзе».

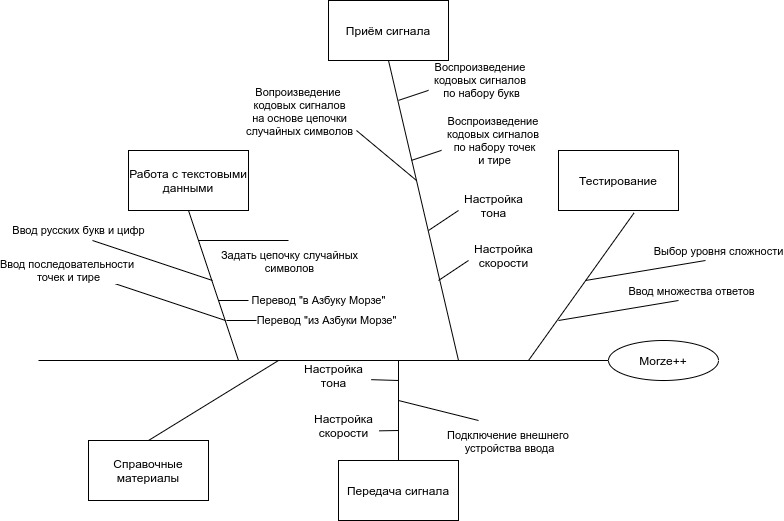
* Система тестирования на знание азбуки Морзе с выбором уровня сложности

**ВХОД:** Уровень сложности теста, множество ответов.

**ВЫХОД:** Процент правильного выполнения от общего числа тестов.

**МЕТОД:** Сравнение множество ответов пользователя с множеством правильных ответов.

# Карта функций



# Атрибуты качества

* **•Функциональность;**

Данный атрибут заявляет о том, что ПО должно обладать необходимым функционалом для быстрого и эффективного обеспечения обучению азбуке Морзе.

* **Высокая Производительность;**

Высокая производительность необходима, чтобы пользователь мог комфортно пользоваться ПО без задержек и ожидания.

* **• Надежность;**

ПО должно быть надёжным, чтобы не прерывать процесс обучения ошибками программы. Система должна сохранять состояние после перезапуска системы или после сбоя. 80 % данных должно быть сохранено до состояния отказа программы.

* **• Удобство Использования;**

разрабатываемая ПС очень проста в освоении и также удобна, интерфейс данной ПС достаточно дружелюбен, так как существует множество вспомогательных средств, которые могут указать на тот или иной объект с подробным объяснением, которое выводится на экран пользователя.

* **• Невысокая Начальная Стоимость Системы;**

Снижение расходов на создание ПО является одной из приоритетных целей заказчика.

* **• Низкие Эксплуатационные Расходы;**

Снижение расходов на эксплуатацию ПО является одной из приоритетных целей заказчика.

• **Оцениваемость;**

Пользователь должен понимать преимущества использования данного ПО по сравнению с аналогичными продуктами.

**Доступность;**

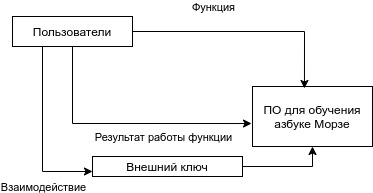
Разрабатываемая ПС имеет высокий уровень доступности, так как пользователь, скачавший её может пользоваться ей без ограничений по времени и ресурсам 98.5 % времени в году.

**Лёгкость в эксплуатации;**

Данная ПС имеет достаточно простую архитектуру, что позволяет пользователям с лёгкостью в ней разобраться. Пользователь знакомый с понятиями «Азбуки Морзе» освоит 90% функционала примерно за 10 минут, пользователям без такого опыта понадобиться 30 минут на освоение 90% функционала.

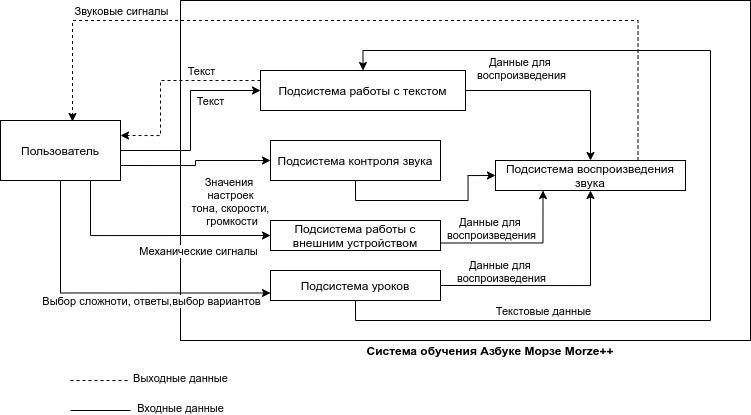
# КД/АКД

Контекстная диаграмма, содержащая нулевой процесс (Изображение 3).



Изображение 3-Контекстная диаграмма

Архитектурная контекстная диаграмма первого уровня (Изображение 4).

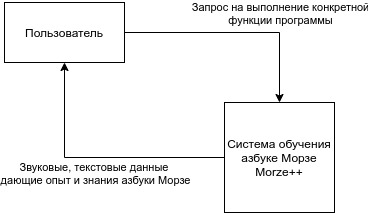


Изображение 4-Архитектурно-контекстная диаграмма

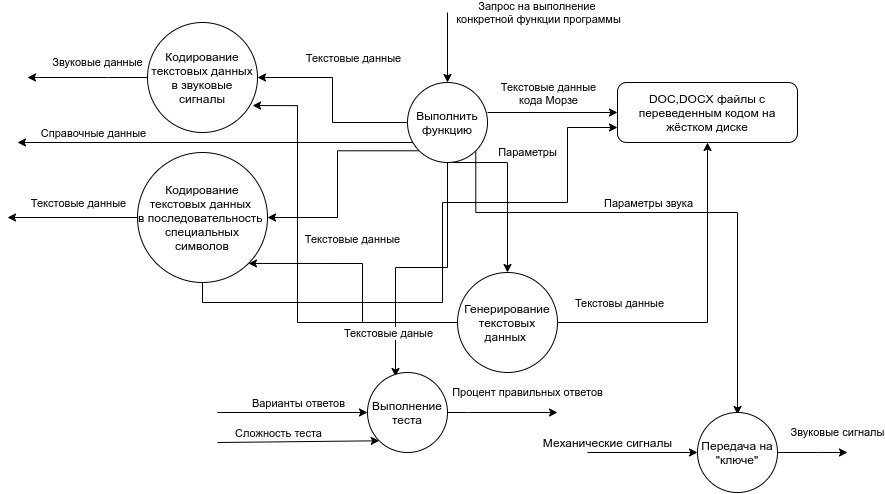
# Артефакты системы

# DFD1,DFD2

В данном разделе прилагает диаграмма потоков данных первого уровня (Изображение 5) и диаграмма потоков данных 2 уровня(декомпозиция диаграммы потоков данных первого уровня Изображение 6).



Изображение 5-ДПД 1 уровня



Изображение 6-ДПД 2 уровня

# Тестирование

**Тестирование функция check\_phone.**

**Тестирование по принципу “Чёрного ящика”.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Входные данные | Номер правильного класса эквивалентности | Правильный класс эквивалентности | Номер неправильного класса эквивалентности | Неправильный класс эквивалентности |
| Код сотрудника | 1 | Код сотрудника является натуральным числом в диапазоне от одного до ста. | 2 | Является числом больше 100 |
| 3 | Является числом меньше 1 |
| 4 | Не является числом |
| Ф.И.О. | 5 | Фамилия, имя, отчество не должны быть длинной больше 20 символов русского алфавита. Первая буква заглавная, остальные маленькие. | 6 | Длина больше 20 символов |
| 7 | Первая буква не заглавная |
| 8 | Заглавных букв больше чем 1 |
| 9 | Содержит символы не принадлежащие русскому алфавиту |
| 10 | Состоит из двух слов |
| Номер паспорта | 11 | является натуральным 6-значным числом, находящимся в диапазоне от 100000 до 999999. | 12 | Число больше 999999 |
| 13 | Число меньше 100000 |
| 14 | Не является числом |
| Категория | 15 | - является целым числом в промежутке от 1 до 3 | 16 | Число больше 3 |
| 17 | Число меньше 1 |
| 18 | Не является числом |
| Ставка за час (тыс. руб.) | 19 | является натуральным числом в диапазоне от 1 до 100. | 20 | Число больше 100 |
| 21 | Число меньше 1 |
| 22 | Не является числом |
| Код предприятия | 23 | является натуральным 5-значным число в диапазоне от 10000 до 99999. | 24 | Число больше 99999 |
| 25 | Число меньше 10000 |
| 26 | Не является числом |
| Наименование предприятия | 27 | Является набором букв русского алфавита длиной не более 20-ти символов. Причём название предприятия начинается с большой буквы, остальные буквы являются маленькими. | 28 | Длина больше 20 символов |
| 29 | Первая буква не заглавная |
| 30 | Заглавных букв больше чем 1 |
| 31 | Содержит символы не принадлежащие русскому алфавиту |
| 32 | Состоит из двух слов |
| Телефон предприятия | 33 | Является 11-значным натуральным числом, начинающимся с цифры 8. | 34 | Число больше 99999999999 |
| 35 | Число меньше 10000000000 |
| 36 | Не является числом |
| Дата выполнения работы | 37 | Набор букв русского алфавита и цифр, имеющих следующий формат: “цифра(от 1 до 31)”\_“Первый три заглавные буквы месяца“\_Год (4 цифры). | 38 | Набор букв русского алфавита и цифр, не имеющих следующий формат: “цифра(от 1 до 31)”\_“Первый три заглавные буквы месяца“\_Год (4 цифры). |
| Список кодов сотрудников | 39 | Однострочный список, элементы которого имеют формат: (“код сотрудника”:” кол-во часов”), где кол-во часов является целым неотрицательным числом в диапазоне от 0 до 1280. | 40 | Отсутствует запятая между элементами |
| 41 | Отсутствует разделитель между элементами |
| 42 | Кол-во часов больше 1280 |
| 43 | Кол-во часов меньше 0 |
| Год, введённый пользователем с клавиатуры | 44 | является натуральным числом в диапазоне от 1000 до 9999. | 45 | Число больше 9999 |
| 46 | Число меньше 1000 |
| 47 | Не является числом |
| Строка файла сотрудники | 48 | Содержит данные | 49 | Не содержит данных |
| Строка файла предприятия | 50 | Содержит данные | 51 | Не содержит данных |
| Строка файла выполненная работа | 52 | Содержит данные | 53 | Не содержит данных |
| Файл сотрудники | 54 | Содержит данные | 55 | Пустой |
| Файл предприятия | 56 | Содержит данные | 57 | Пустой |
| Файл выполненная работа | 58 | Содержит данные | 59 | Пустой |
| Строка файла сотрудники | 60 | Содержит все необходимые поля | 61 | В строке не хватает полей |
| Строка файла предприятия | 62 | Содержит нужное кол-во разделителей | 63 | Число разделителей меньше необходимого |
| 64 | Число разделителей больше необходимого |
| Строка файла выполненная работа | 65 | Содержит нужное число пробелом между полями | 66 | Число пробелом больше необходимого |
| 67 | Число пробелов меньше обходимого |
| Код сотрудника | 68 | Является уникальным идентификатором | 69 | Совпадает с кодом другого сотрудника |
| Предприятие | 70 | Имеет сотрудников | 71 | Не имеет ни одного сотрудника |
| Номер предприятия | 72 | Является уникальным идентификатором | 73 | Совпадает с номером другого предприятия |
| Название предприятия | 74 | Является уникальным | 75 | Совпадает с названием другого предприятия |
| Файл сотрудники должен существовать | 76 | Файл существует | 77 | Файл не существует |
| Файл предприятия должен существовать | 78 | Файл существует | 79 | Файл не существует |
| Файл выполненная работа должен существовать | 80 | Файл существует | 81 | Файл не существует |
| Файл сотрудники должен состоять не более чем из 100 строк | 82 | Файл содержит 100 или менее строк | 83 | Файл содержит более 100 строк |
| Файл предприятия не должен состоять более чем из 10 строк | 84 | Файл содержит 10 или менее строк | 85 | Файл содержит более 10 строк |
| Файл выполненная работа должен состоять не более чем из 10 строк | 86 | Файл содержит 10 или менее строк | 87 | Файл содержит более 10 строк |

**Построение тестов.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | Входные данные | | Номер покрытого прав. Класса эквивалентности | Номер покрытого неправ. Класса эквивалентности |
|  | Сотрудники.txt | 22 Проскурин Денис Александрович 351619 3 33  23 Ли Василий Эдуардович 355619 3 30  24 Матевасян Каринэ Лёвовна 390619 3 31 | 1,5,11,15,19,23,27,33,37,39,44,48,50,52,54,56,58,60,62,65,68,70,72,74,76,78,80,82,84,86 |  |
| Предприятия.txt | 12342| Арес |89235673082|  23532| Калисто | 89246984082|  23233| Магомед |89246784022| |
| Выполненная работа.txt | 12345 8 ЯНВ 2017 23:34,24:34,22:23  23532 8 ЯНВ 2017 22:34,23:34,24:23  23233 8 ЯНВ 2017 23:34,24:34,22:23 |
| Клавиатура | 2017 |  | 2, 24,38,45 |
|  | Сотрудники.txt | 101 Проскурин Денис Александрович 351619 3 33 |
| Предприятия.txt | 123425| Арес |89235673082| |
| Выполненная работа.txt | 12345 8 ЯНВАРЬ 2017 23:34,24:34,22:23 |
| Клавиатура | 20122 |
|  | Сотрудники.txt | 19 проскурин Денис Александрович 351619 3 33 |  | 7,41,46 |
| Предприятия.txt | 123425| Арес |89235673082| |
| Выполненная работа.txt | 12345 8 ЯНВАРЬ 2017 2334,2434,2223 |
| Клавиатура | 201 |
|  | Сотрудники.txt | 19 Проскуриняниновивосян Денис Александрович 351619 3 33 |  | 6,28,40 |
| Предприятия.txt | 123425| Аресаресаресаресаресар |89235673082| |
| Выполненная работа.txt | 12345 8 ЯНВ 2017 23:3424:3422:23 |
| Клавиатура | 2012 |
|  | Сотрудники.txt | 0 ПроскУрин Денис Александрович 351619 3 33 |  | 3,8,25,26,47 |
| Предприятия.txt | 1234| Арес |89235673082| |
| Выполненная работа.txt | 12g45 8 ЯНВ 2017 23:34,24:34,22:23 |
| Клавиатура | 201л |
|  | Сотрудники.txt | 2п Проск Yрин Денис Александрович 3512619 4 101 |  | 4,9,10,12,16,20,29,31,32,34,42,43 |
| Предприятия.txt | 12342| Аr ес |892356723082| |
| Выполненная работа.txt | 12345 8 ЯНВ 2017 23:3400,24:-1,22:23 |
| Клавиатура | 2017 |
|  | Сотрудники.txt | 22 ПроскурНн ДеНис Александрович 35619 0 0 |  | 13,17,21,30,35 |
| Предприятия.txt | 12342| Арес |892356782 | |
| Выполненная работа.txt | 12345 8 ЯНВ 2017 23:34,24:34,22:23 |
| Клавиатура | 2017 |
|  | Сотрудники.txt | 22 Проскурин Денис Александрович 35f619 3f 3s3 |  | 14,18,22,36 |
| Предприятия.txt | 12342| Арес |892356730f82| |
| Выполненная работа.txt | 12345 8 ЯНВ 2017 23:34,24:34,22:23 |
| Клавиатура | 2017 |
|  | Сотрудники.txt |  |  | 49,51,53,55,57,59 |
| Предприятия.txt |  |
| Выполненная работа.txt |  |
| Клавиатура | 2017 |
|  | Сотрудники.txt | 22 Проскурин Денис Александрович 351619  22 Ли Василий Эдуардович 345321 3 33 |  | 61,63,66,69 |
|  | Предприятия.txt | 12342| Арес 9235673082| |
|  | Выполненная работа.txt | 12345\_\_8 ЯНВ 2017 23:34,24:34,22:23 |
|  | Клавиатура | 2017 |
|  | Сотрудники.txt | 11 Проскурин Денис Александрович 351619 3 33 |  | 64,67,71,73,75 |
|  | Предприятия.txt | |12342| Арес |89235673082|  12342| Калисто |89235673082|  12242| Калисто |89235673082| |
|  | Выполненная работа.txt | 12342 ЯНВ 2017 |
|  | Клавиатура | 2017 |

**Тестирование по принципу “Белого ящика”.**

**function check\_phone**(substr: **string**): **boolean**;

**const**

l\_phone = 11;//длина номера телефона 11 цифр

range = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'];//множество целых чисел

**var**

i: **byte**;//счётчик

flag: **boolean**;//булевая переменная значение которой присваивается функции

**begin**

flag := true;(1)

**if** (length(substr) = l\_phone)(2) **and** (substr[1] = '8')(3) **then**

**begin**

(4)**for i** := 2 **to** l\_phone **do** (5)**if not** (substr[i] in range) (6) **then** flag := false;

**end else** flag := false; (7)

**result** := flag;(8)

**end;**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Substr | i | Flag | Result |
| 1,2,3,4,5,8 | 89965554466 | 0 | True | True |
| 2 | True |
| 3 | True |
| 4 | True |
| 5 | True |
| 6 | True |
| 7 | True |
| 8 | True |
| 9 | True |
| 10 | True |
| 11 | True |
| № | Substr | i | Flag | Result |
| 1,3,7,8 | 899625236142 | 0 | True | False |
| 0 | False |
| № | Substr | i | Flag | Result |
| 1,2,7,8 | 79965554466 | 0 | True | False |
| 0 | False |
| № | Substr | i | Flag | Result |
| 1,2,3,4,5,6,8 | 89296В54ВЫ9 | 0 | True | False |
| 2 | True |
| 3 | True |
| 4 | True |
| 5 | True |
| 6 | False |
| 7 | False |
| 8 | False |
| 9 | False |
| 10 | False |
| 11 | False |

**Тестирование функция check\_names.**

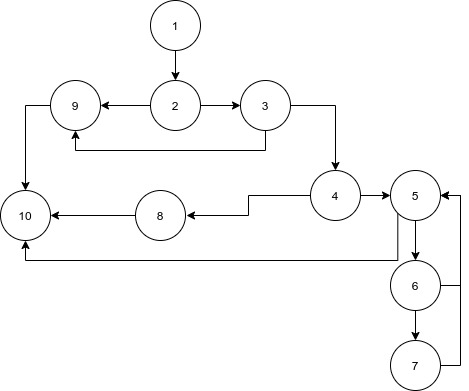
**Тестирование по принципу “Чёрного ящика”.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Входные данные | Номер правильного класса эквивалентности | Правильный класс эквивалентности | Номер неправильного класса эквивалентности | Неправильный класс эквивалентности |
| Фамилия, имя, отчество, название предприятия. | 1 | Набор букв русского алфавита длиной не более 20-ти символов, начинающийся с заглавной буквы | 2 | Не начинается с заглавной буквы |
| 3 | Длина больше 20 символов |
| 4 | Нет строки |
| 5 | Не является набором букв русского алфавита |

**Построение тестов.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | Входные данные | Номер покрытого прав. Класса эквивалентности | Номер покрытого неправ. Класса эквивалентности | Выходные данные |
|  | Калисто | 1 |  | True |
|  | денис |  | 2 | False |
|  | Калистокалистокалисто |  | 3 | False |
|  | ‘’ |  | 4 | False |
|  | Калisto |  | 5 | False |
|  | калистокалистокалисто |  | 2,3 | False |
|  | калisto |  | 2,5 | False |
|  | калистокалistoкалисто |  | 2,3,5 | False |

**Тестирование по принципу “Белого ящика”.**

****

Изображение 7-Flow Graph

M=E-N+2P (цикломатическая сложность графа).

M=13-10+2\*3=-3

**Independent paths**

1. 1 2 9 10
2. 1 2 3 9 10
3. 1 2 3 4 8 10
4. 1 2 3 4 5 10
5. 1 2 3 4 5 6 5 10
6. 1 2 3 4 5 6 7 5 10

**function check\_names**(substr: **string**): **boolean**;

**const**

B\_alf = ['А'..'Я'];//множество заглавных букв русского языка

L\_alf = ['а'..'я'];//множество строчных букв русского языка

max\_l = 20;//максимальная длина ФИО или названия предприятия

**var**

i: **byte**;//счётчик

flag: **boolean**;//булевая переменная значение которой присваивается функции

**begin**

flag := true; (1)

**if** (length(substr) <= max\_l)(2) **and** (length(substr) <> 0)(3) **then**

**begin**

**if** substr[1] **in** B\_alf **then**(4)

**begin**

**for** i := 2 **to** length(substr) **do**(5)

(6)**if not** (substr[i] **in** L\_alf) t**hen** (7)flag := false;

**end else** flag := false; (8)

**end else** flag := false; (9)

**result** := flag;(10)

**end;**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Substr | Flag | I | Substr[i] | Result |
| 1,2,3,4,5,10 | Калисто | True | 0 | ‘’ | True |
| True | 2 | а |
| True | 3 | л |
| True | 4 | и |
| True | 5 | с |
| True | 6 | т |
| True | 7 | о |
| № | Substr | Flag | I | Substr[i] | Result |
| 1,2,3,8,10 | денис | True | 0 | ‘’ | False |
| False | 0 | ‘’ |
| № | Substr | Flag | I | Substr[i] | Result |
| 1,2,9,10 | ‘’ | True | 0 | ‘’ | False |
| False | 0 | ‘’ |
| № | Substr | Flag | I | Substr[i] | Result |
| 1,2,3,4,5,6,7,10 | Калisto | True | 0 | ‘’ | False |
| True | 2 | а |
| True | 3 | л |
| False | 4 | i |
| False | 5 | s |
| False | 6 | t |
| False | 7 | o |

# Стратегия и тактика тестирования

Поскольку данный проект (прототип) был написан с использованием языка Delphi7(он же Pascal) с использованием процедурного подхода, и модульной архитектуры проекта. Из этого вытекает , что стратегией к тестированию является выделение модулей системы (Модуль работы с текстом, модуль работы со звуком, Модуль для работы с тестами, модуль обучения, модуль работы с внешним устройствои). Архитектурное решение было таково что позволило разделить на такие модули стратегия состоит в нисходящем подходе к тестированию, по отдельности берем модули, декомпозируем до функций. Далее функции в зависимости от объёма будем тестировать по белому или чёрному ящику (по белому если flow graph функции будет не таким большим) и по чёрному в противном случае. Далее тестирование функций, которые используют атомарные функции по вышеперечисленному принципу, и так до тех пор пока не дойдём до уровня модуля. Для тестирования модуля работы с текстом можно сделать простеньку программку которая будет работать по принцепу assert в с++ или unit тестов, поскольку правила кодирования азбуки морзей известны и примеров и литературы с примерам в сети огромное множество. Тестирование модуля работы со звуком наиболее трудоёмкое, т.к. программу для тестирования звука не так просто написать, нужно слушать звуковое закодированное сообщение, а можно схитрить и выводить вместе звуком соответствующие знаковые сигналы( точка, тире) для длинных и коротких звуков и тогда можно также использовать вспомогательные программки для тестирования. Модуль работы с тестами использует часть кода из модуля работы со звуком и текстом, так что часть тестов будет покрыта предыдущими, а тактика тестирования этого модуля будет примерно такой же. Модуль обучения, является нечно похожим на справочный интерактивный материал, который также будет использовать код из модуля звука и текста, тактика тестирования несильно отличается от предыдущих. Модуль работы с внешним устройством немного обособлен от других. С внешнего устройства должны в риал-тайме приходить сигналы булевского характера 0 и 1. 0-воспроизвезти короткий звук, 1 -длинный. Данный модуль хоть и использует часть кода работы со звуком, автоматизировать тестирование данного модуля на мой взгляд никак невозможно и тестирование проводится в ручную.

После «такого» подхода к тестированию функций модуля, мы частично протестировали взаимосвязи между модулями. Осталось проверить выполняются ли правильно функциональные требования программной системы. Далее это всё дело можно «добить» monkey тестами.