Карагандинский технический университет

Кафедра: ИВС

***КУРСОВАЯ***

***РАБОТА***

По дисциплине: Методы и средства разработки программ..

*(наименование дисциплины)*

Тема: Игра в слова с компьютером.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Приняла:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Томилова Н.И.

*(оценка) (фамилия, инициалы)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (дата)*

**Члены комиссии: Выполнил:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Талғатұлы Б.

*(подпись,фамилия, и.о.) (фамилия, инициалы)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **5kirk2p**

*(подпись,фамилия, и.о) (шифр студента)*

Караганда 2021

**ЛИСТ ЗАДАНИЯ**

КАРАГАНДИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра: Информационно-вычислительные системы

Утверждаю

Зав.кафедрой\_Калинин А.А.

«6» сентября 2021г.

**ЗАДАНИЕ №\_9\_**

на курсовую работу по дисциплине:

Методы и средства технологии программированиям

Студент \_\_\_\_\_\_Талғатұлы Б.\_\_\_\_\_группа \_\_\_ИС 20-03\_\_\_\_\_\_

Тема проекта: Игра «в слова с компьютером»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исходные данные: Учебно-методический материал

|  |  |  |  |
| --- | --- | --- | --- |
| № | Содержание пояснительной записки | Сроки выполнения | Примерный объем |
| 1 | Предпроектное обследование. Разработка ТЗ | 10.09–17.09 | 6% |
| 2 | Разработка спецификации ПС | 21.09–05.10 | 6% |
| 3 | Разработка проекта программы. Разработка структуры интерфейса ПС. | 08.10–15.10 | 12% |
| 4 | Алгоритмизация | 18.10–30.10 | 12% |
| 5 | Реализация программного обеспечения | 1.11–20.11 | 18% |
| 6 | Отладка и тестирование ПС. | 20.11–05.12 | 40% |
| 7 | Оформление пояснительной записки и графического материала | 15.11–07.12 | 6% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Содержание графической части | Сроки выполнения | Количество листов | Формат |
| 1 | Диаграмма переходов состояний | 24.09-30.10 | 1 | А4 |
| 2 | Диаграмма потоков данных | 31.10-7.11 | 1 | А4 |
| 3 | Граф диалога пользовательских интерфейсов | 8.11-14.11 | 1 | А4 |
| 4 | Структурная схема ПС | 15.11-21.11 | 1 | А4 |
| 5 | Блок-схема общей функциональности программы | 22.11-30.11 | 1 | А4 |

Литература:

1\_ Томилова Н. И. Электронный учебник по дисциплине, 2021;

2\_ Прохоренок Н., Дронов В. Python 3 и PyQt 5. Разработка приложений

3\_ Сысоев М.В.  «[Программирование для "нормальных" с нуля на языке Python](file:///D:\%D0%A3%D0%BD%D0%B8%D0%B2%D0%B5%D1%80\%D0%9C%D0%98%D0%A2%D0%9F\%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B\%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%BA%D0%B0\%D0%AD%D0%A3%D0%A0_%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B8%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC_%D0%9B%D0%B0%D0%B1%D1%8B_\labs_\Literatura\Python_Москва_2018.pdf)»;

4\_ Марк Лутц - Программирование на Python, том 1, 4-е издание.

Дата выдачи задания, дата защиты проекта 06.09.2021, 10.12.2021

Руководитель проекта 06.09.2021 Томилова Н. И.

Задание принял к исполнению

Дата, подпись студента

**СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ**

В пояснительной записке к курсовому проекту приняты следующие сокращения:

|  |  |
| --- | --- |
| Обозначение | Расшифровка |
| ИВС | Информационно-вычислительные системы |
| ПК | Персональный компьютер |
| ПО | Программное обеспечение |
| ПС | Программная система |
| ЛКМ | Левая кнопка мыши |
| Рис. | Рисунок |
| см. | Смотрите (глагол) |

**ОГЛАВЛЕНИЕ**

1. [СОДЕРЖАТЕЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ 8](#_Toc89958671)
2. [ТЕХНИЧЕСКОЕ ЗАДАНИЕ 9](#_Toc89958672)

[ВВЕДЕНИЕ 9](#_Toc89958673)

[2.1. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ 9](#_Toc89958674)

[2.2. НАЗНАЧЕНИЯ ДЛЯ РАЗРАБОТКИ 9](#_Toc89958675)

[2.3. ТРЕБОВАНИЯ К ПРОГРАММЕ ПРОГРАММНОМУ ИЗДЕЛИЮ 10](#_Toc89958676)

[2.4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 11](#_Toc89958677)

[2.5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 11](#_Toc89958678)

[2.6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 11](#_Toc89958679)

[2.7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 12](#_Toc89958680)

1. [СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ 14](#_Toc89958681)

[3.1 ФУНКЦИОНАЛЬНАЯ ЧАСТЬ 14](#_Toc89958682)

[3.2 ЭКСПЛУТАЦИОННАЯ ЧАСТЬ 16](#_Toc89958683)

[3.3 МЕТОД РЕШЕНИЯ 18](#_Toc89958684)

1. [ПРОЕКТ ПРОГРАММНОЙ СИСТЕМЫ 20](#_Toc89958685)

[4.1 КОНЦЕПТУАЛЬНЫЙ АЛГОРИТМ РАБОТЫ ПС 20](#_Toc89958686)

[4.2 СТРУКТУРНАЯ СХЕМА ПС 21](#_Toc89958687)

[4.3 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС 24](#_Toc89958688)

[4.3 ГРАФ ДИАЛОГА 25](#_Toc89958689)

[4.4 ОПИСАНИЕ ИНТЕРФЕЙСА И КОМАНД МЕНЮ 26](#_Toc89958690)

1. [АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 28](#_Toc89958691)

[5.1КОНЦЕПТУАЛЬНАЯ АРХИТЕКТУРА ПО 28](#_Toc89958692)

[5.2 ДИАГРАММА КЛАССОВ 30](#_Toc89958693)

[5.3 БЛОК-СХЕМЫ ОСНОВНЫХ РЕАЛИЗОВАННЫХ ФУНКЦИЙ ПС 30](#_Toc89958694)

1. [ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 33](#_Toc89958695)

[6.1 СТРУКТУРНОЕ ТЕСТИРОВАНИЕ 33](#_Toc89958696)

[6.2 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ 34](#_Toc89958697)

[6.3 ТЕСТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА 35](#_Toc89958698)

1. [СПРАВКИ ДЛЯ ПРИЛОЖЕНИЯ ПО КП 36](#_Toc89958699)

[7.1 РУКОВОДСТВО АДМИНИСТРАТОРА 36](#_Toc89958700)

[7.2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 36](#_Toc89958701)

1. [ЗАКЛЮЧЕНИЕ 38](#_Toc89958702)
2. [ПРИЛОЖЕНИЕ 1 39](#_Toc89958703)
3. [ПРИЛОЖЕНИЕ 2 46](#_Toc89958704)

СОДЕРЖАТЕЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ

Игра в слова с компьютером как развивающая игра.

Развивающие игры — это игры на развитие памяти, мышления и т. п. Игры должны быть оригинальными, интересными и нести в себе задачу, вопрос, только в этом случае они выполняют свою цель.

В данном проекте под название «Игра в слова» необходим словарный запас, логика. Играя, можно выучить или узнать новые слова.

Принцип игры в том, что даётся выбор первого хода — человек или компьютер. В случае если первым ходит человек, должно быть загадано слово и компьютер по словарю из базы изменяет слово на новое. Это продолжается пока компьютер или человек не сможет изменять букву в словах. По заданию, максимальное количество букв в слове — 5.

Темой курсовой работы является разработка программы, которая будет обеспечивать возможность логической игры в игра в слова с компьютером на языке Python, рассчитанная на одного пользователя ПК. Целью данной курсовой работы является углубление знаний и расширение навыков по разработке алгоритмов и их реализации на персональном компьютере.

Курсовая работа выполнена в среде Python с использованием графического модуля PyQt5.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку построение компьютерной игры, предназначенной для игры в слова с компьютером. Предполагается, что использовать данный программный продукт будут люди любого возраста. В виде входных данных на него приходит длинное слово (для разгадывания), результат работы — данные об возможных существующих в русском языке вариантах комбинаций слов.

**Наименование программы**

Наименование разрабатываемой программы – «Игра в слова с компьютером»

**Назначение и область применения**

1. Настоящее техническое задание распространяется на составление программы, цель которого – развлечь и развить у пользователей способности к логическому мышлению. Игра предназначена для осмысленной непродуктивной деятельности, где мотив лежит не в ее результате, а в самом процессе. Использования данного программного продукта позволит людям провести время занимательно.
2. С помощью модуля PyQt5 в языке программирования Python можно создать удобный графический интерфейс, который будет интуитивно понятен для всех пользователей.

2.1. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Система разрабатывается на основании учебного плана кафедры ИВС по дисциплине «6B06102 Информационные системы», в качестве курсового проекта по дисциплине «Методы и средства разработки программ».

2.2. НАЗНАЧЕНИЯ ДЛЯ РАЗРАБОТКИ

Главным назначением программы представляет собой обеспечения простого и примитивного программного продукта, созданного с целью развлечь и развить у пользователей способности к логическому мышлению.

2.3. ТРЕБОВАНИЯ К ПРОГРАММЕ ПРОГРАММНОМУ ИЗДЕЛИЮ

**2.3.1. Требования к функциональным характеристикам**

**2.3.1.1. Система должна обеспечивать возможность выполнения следующих функций:**

* Выбор первого хода;
* Замены буквы;
* Проверки слова;
* Сообщение о выигрыше, в случае победы человека;

**2.3.1.2. Исходные данные**

* Словарь, база слов;
* Поиск слов;

**2.3.1.3. Результат**

Результатом работы программы является игра в слова с компьютером.

**2.3.2. Требования к надежности**

**2.3.2.1.** Предусмотреть блокировку некорректных действий пользователя при работе с приложением.

**2.3.2.2.** Обеспечить целостность приложения.

**2. 3.3. Требования к составу и параметрам технических средств**

* Система должна работать на IBM совместимых персональных компьютерах.
* Минимальная конфигурация:

а) тип процессора Intel Pentium и выше;

б) объем оперативного запоминающего устройства 1024 Мб и выше.

**2.3.4. Требования к информационной и программной совместимости**

Программа должна работать под управлением семейства операционных систем Windows 32/64x – Windows 2000, XP, Vista, 7, 8, 8.1, 10, 11.

2.4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

**2.4.1.** Разрабатываемые программные модули должны быть задокументированы, т. е. тексты программ должны содержать все необходимые комментарии.

**2.4.2.** Разрабатываемая программа должна включать справочную информацию об основных приемах работы с форматированием электронной текстовой информации.

**2.4.3. В состав сопровождающей документации должны входить:**

**2.4.3.1.** Пояснительная записка на 25–30 листах, содержащая описание разработки.

**2.4.3.2.** Руководство пользователя.

**2.4.3.3.** Графическая часть на трех листах формата А4.

**2.4.3.4.** Схема структурная программной системы.

**2.4.3.5** Диаграмма компонентов данных.

2.5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

**2.5.1.** Ориентировочная экономическая эффективность не рассчитываются.

**2.5.2.** Предполагаемая годовая потребность не рассчитывается.

**2.5.3.** Экономические преимущества: бесплатное распространение. Существующие аналоги: не существует

2.6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

**2.6.1. Стадии разработки**

Разработка должна быть проведена следующими стадиями:

* разработка технического задания;
* проектирование;
* реализация и отладка;
* внедрение.

**2.6.2. Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии проектирования должны быть выполнены перечисленные ниже этапы работ:

* Разработка спецификации программы;
* Составление проекта программы и плана отладки.

На стадии реализации и отладки должны быть выполнены перечисленные ниже этапы работ:

* Алгоритмизация. Выбор основных средств отладки;
* Программирование. Подготовка тестов;
* Отладка программы.

На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы.

**2.6.3. Сроки разработки**

1. Разработка технического задания – 1 неделя;
2. Разработка спецификации программы – 1 неделя;
3. Составление проекта программы и плана отладки – 2 недели;
4. Алгоритмизация. Выбор основных средств отладки – 2 недели;
5. Программирование. Подготовка тестов – 3 недели;
6. Отладка программы – 5 недель.

2.7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

**2.7.1. Виды испытаний**

* Испытание общей работоспособности на операционной системе Windows различном оборудовании;
* Испытания работоспособности дополнительных функций программы (открытие и т.д.).

**2.7.2. Общие требования к приемке**

* Программа должна выполнять все необходимые требования, указанные в постановке задачи;
* Программа должна иметь удобный, интуитивно-понятный пользовательский интерфейс;

СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ

3.1 ФУНКЦИОНАЛЬНАЯ ЧАСТЬ

**3.1.1 Таблица Функции программы**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Функция. Требования к составу выполняемой функции. | Входные данные | Выходные данные |
| 1 | Функция, выбора право первого хода | Данные клика мышки пользователем | Выбор хода |
| 2 | Функция, загрузки слов, словаря | Словарь в формате .txt | Отображение слов |
| 3 | Функция, проверки слова | Словарь в формате .txt | Слово, которое было предложено, не вылезает снова |
| 4 | Функция, ход игрока | Изменение буквы | Передача ходу компьютеру |

**3.1.2 Тип пользовательского интерфейса**

В приложение реализован простой интерфейс – меню.

Интерфейс-меню предполагает, что программа находится либо в состоя­нии Уровень меню, либо в состоянии Выполнение операции. В состоянии Уровень меню осуществляется вывод меню соответствующего уровня и выбор нужного пункта меню, а в состоянии Выполнение операции реализуется сценарии выбранной операции.

.

**3.1.3 Диаграммы ПС**

На рисунках продемонстрированы: диаграмма переходов состояний (рис. 3.1), диаграмма потоков данных (рис. 3.2), SADT-модель (рис. 3.3).

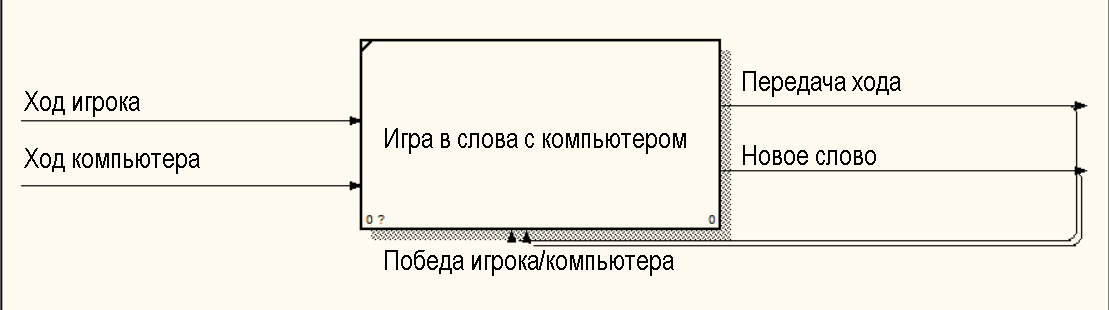


Рис 3.1 Диаграмма переходов состояний

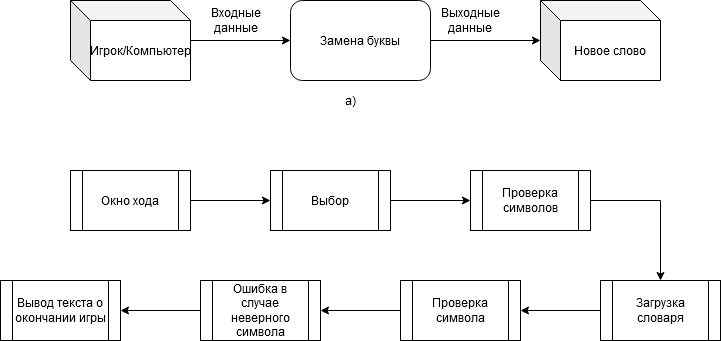
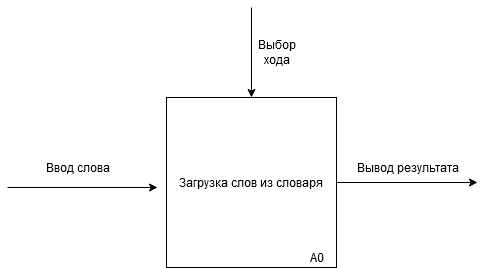
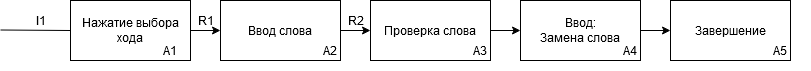


Рис 3.2 а) Диаграмма потоков данных; б) Дополненная диаграмма потоков управляющих данных;



а)



I1 – Нажатие выбора хода

R1 –Ввод

R2–Вывод результатов в игровое поле

Рис. 3.3 SADT модель: а – концептуальный уровень

3.2 ЭКСПЛУТАЦИОННАЯ ЧАСТЬ

**3.2.1 Требования к производительности**

К разработке данного ПО предоставляются следующие требования к производительности;

* обеспечить максимальную скорость вывода слова;
* обеспечить максимальную производительность программы при замене буквы;
* обеспечить минимальное использование оперативной памяти.

**3.2.2 Требования к надежности**

К разработке данного ПО предоставляются следующие требования к надежности:

* обеспечить безотказность работы программы;
* обеспечить целостность обрабатываемой информации.

**3.2.3 Обработка ошибок**

К разработке данного ПО предоставляются следующие требования к обработке ошибок:

* если пользователь заменит неверную букву, программа выведет ошибку;
* количество символов не может быть выше пяти;

**3.2.4 Интерфейсные требования**

К разработке данного ПО предоставляются следующие интерфейсные требования:

* минимальность затрат ресурсов пользователя при вводе, модификации и просмотре данных;
* максимально простое взаимодействие программы и пользователя;
* минимальный объём оперативной памяти пользователя;
* максимально простая навигация меню;
* комфортная цветовая гамма.

**3.2.5 Ограничения**

К разработке данного ПО предоставляются следующие ограничения:

* программа должна быть написана на языке программирования Python;
* программа должна иметь модульную структуру.

3.3 МЕТОД РЕШЕНИЯ

**3.3.1 Алгоритм решения основной функциональности ПО (в виде блок-схемы)**

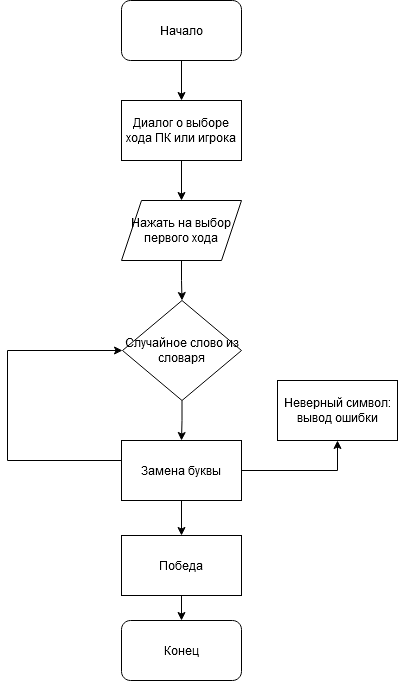
****

Рис 3.1 Блок-схема алгоритма решения основной функциональности ПО

**3.3.2 Алгоритм решения основной функциональности по (словесный)**

1. Начало: программа запустилась;
2. Выбор первого хода;
3. Предложение слова если первый ход игрока;
4. Замены буквы если первый ход компьютера;
5. Окончание игры если все слова были разыграны;
6. Если выбрать выход из игры, то произойдет завершение программы;

ПРОЕКТ ПРОГРАММНОЙ СИСТЕМЫ

4.1 КОНЦЕПТУАЛЬНЫЙ АЛГОРИТМ РАБОТЫ ПС

**Шаг 1.** Определяем структуру управляющей программы:

***Программа.***

*Инициализировать глобальные значения*

*Вывести главное окно и меню*

***Выполнять***

***Сообщение*** *Кто ходит первым*

***Если***

*Выбор Игрок*

*то предложить ввести*

***Если***

*Выбор Компьютер*

*то загрузить из словаря слово*

***Выполнять***

*Ввод буквы*

*Выбор из 5 строчек на замену буквы*

***Если***

*Нет подходящего слово*

***Вывести***

*Вы победили*

*Команда=Выход*

***Конец.***

4.2 СТРУКТУРНАЯ СХЕМА ПС

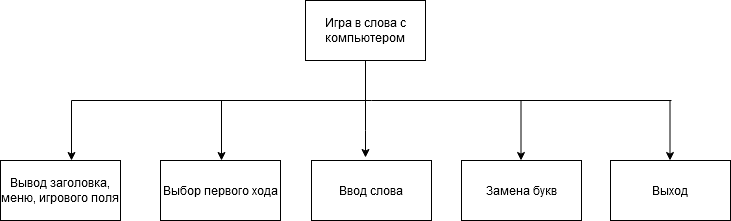
****

Рис 4.2.1 Структурная схема ПО

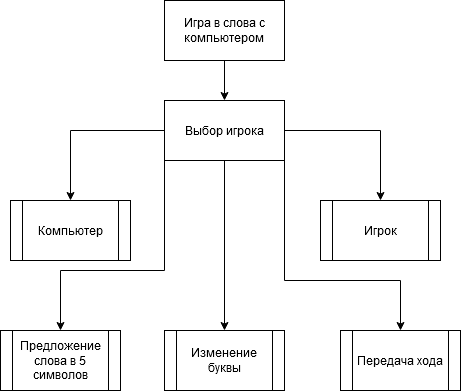
****

Рис 4.2.2 Структурная карта Константайна

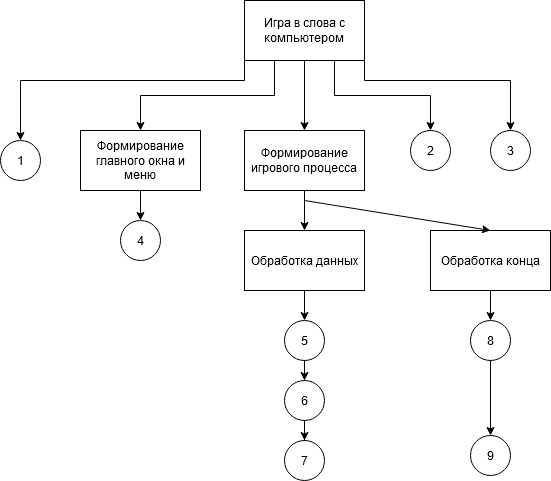
****

Рис. 4.2.3 Структурная карта Джексона

Полный список операций в структурной карте Джексона, которые должна выполнять программа:

1 - запуск программы  
2 – выбор хода   
3 - завершение программы  
4 – загрузить слова из словаря  
5 – предложить слово  
6 – заменить букву  
7 – выбор буквы на замену  
8 – закрытие программы  
9 - вывести победное окно

Псевдокод структурной карты Джексона:

***Импорт модулей***

***Запуск игры***

***Функция:***

*Сообщение = кто ходит первым*

*кнопка1 = Я*

*кнопка2 = компьютер*

*кнопка1.клик. подключение (первым ходит игрок)*

*кнопка2.клик. подключение (первым ходит компьютер)*

***Если***

*Выбор компьютера*

***то загрузить слово из словаря***

***Если***

*Выбор игрока*

***то вывести окно о вводе слов***

***Завершить игру и вывести окно о победе***

***Если***

*Невозможно подобрать слово*

***Класс для интерфейса (игра, виджеты, главное меню, интерфейс)***

***Функция:***

*Загрузка словаря;*

*Функция проверки слова*

*Функция хода игрока*

*Функция хода компьютера*

*Функция получения слова*

*Функция проверки символов*

***Окно = приложение***

***Обработка событий***

4.3 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

**4.3.1 Тип интерфейса**

В приложение реализован одноуровневый интерфейс, который включает в себя: Справка с виджетами «Об авторе», «игровое поле» и «Выход из игры».

**4.3.2 Макет интерфейса**

На рисунке представлен макет пользовательского интерфейса:

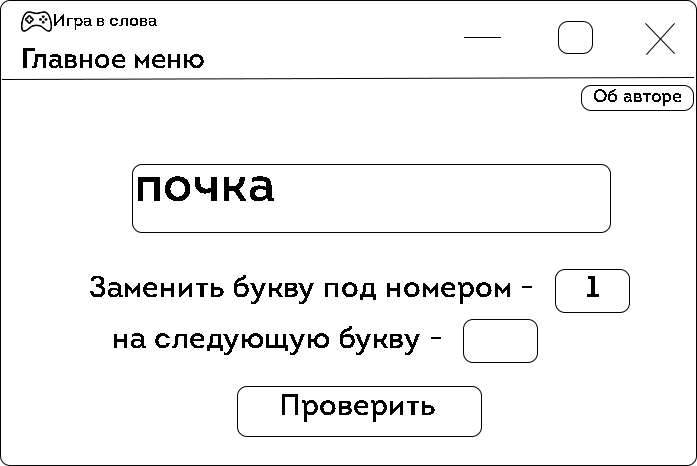
****

Рис. 4.3.2 Макет интерфейса

4.3 ГРАФ ДИАЛОГА

На рисунке представлена графа диалога:

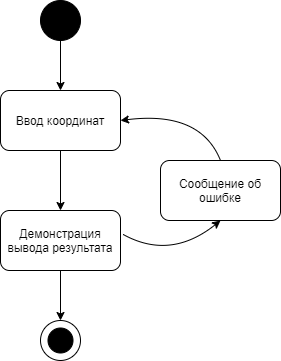


Рис. 4.3 Граф диалога

4.4 ОПИСАНИЕ ИНТЕРФЕЙСА И КОМАНД МЕНЮ

**4.4.1 Структура главного окна**

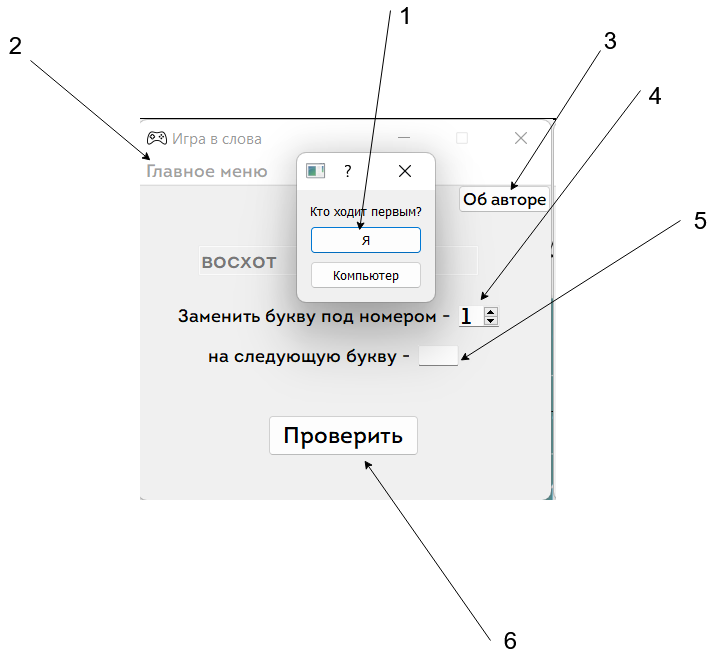


Рисунок 4.4 Структура главного окна

1 – Строка выбора хода – содержит выбор хода игрока или компьютера

2 – Строка меню – главное меню программы

3 – Кнопка об авторе – показывает создателя приложения

4 – Кнопка выбора – заменяет букву под выбранной цифрой

5 – Поле ввода буквы — ввод русской буквы

6 – Проверка на правильность замены

**4.4.2 Структура меню**

Graphical user interface, text, application, chat or text message

Description automatically generated

Рис 4.4.2 Структура меню

1 – Пункт «Главное меню» – служит для вызова главного меню

2 – Пункт «Сохранить как» - служит для сохранения

3 – Пункт «Выйти» - служит для выходи из программы

**4.4.3 Процесс изменения пользовательского интерфейса**

Пользовательский интерфейс не меняется и остается в исходном состоянии.

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1КОНЦЕПТУАЛЬНАЯ АРХИТЕКТУРА ПО

Программное обеспечение разрабатываемого курсового проекта состоит из 4 программных модулей и 1 файла справки.

Программные модули:

1. Pyqt5 – модуль, предназначенный для графического интерфейса
2. random – модуль для перемешивания элементов списка
3. sys – модуль sys обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором python.

Cхема программных модулей:

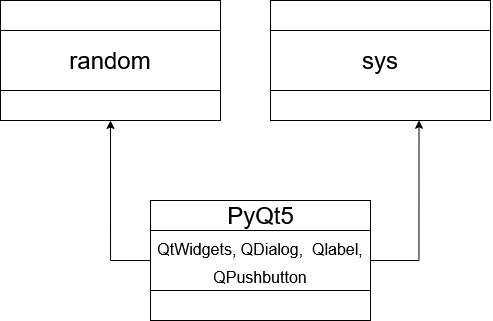


Рис. 5.1 Схема программных модулей

Таблицы описание методов модулей:

Таблица 5.1.1 PyQt5

|  |  |
| --- | --- |
| **Наименование подпрограммы** | **Описание подпрограммы** |
| QInputDialog | Функция вывода диалога |
| QPushButton | Функция кнопки |
| QLabel | Функция вывод текста |

Таблица 5.1.1 random

|  |  |
| --- | --- |
| **Наименование подпрограммы** | **Описание подпрограммы** |
| Randint() | Возвращает случайное значение |

5.2 ДИАГРАММА КЛАССОВ

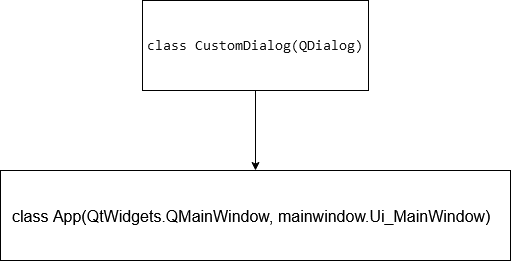


Рис 5.2 Диаграмма классов

5.3 БЛОК-СХЕМЫ ОСНОВНЫХ РЕАЛИЗОВАННЫХ ФУНКЦИЙ ПС

**5.3.1 Блок-схема функции поиска пустой ячейки.**

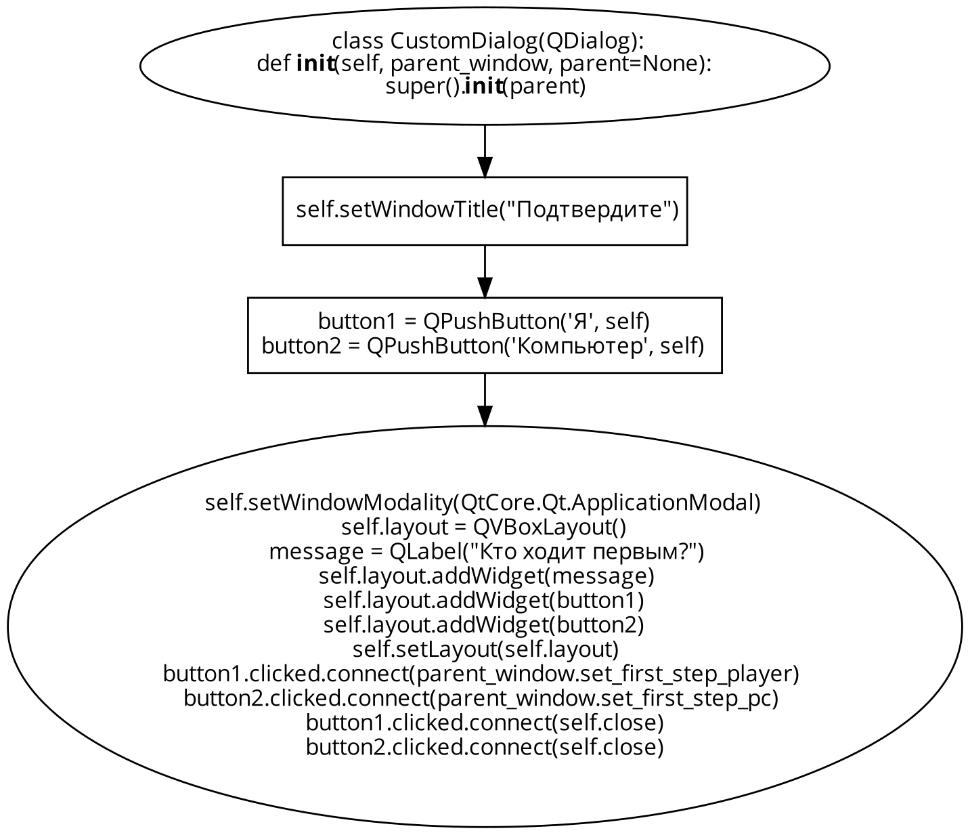


Рис. 5.3.1 Блок-схема класса выбора хода

**5.3.2 Блок-схема функции загрузки словаря и проверки слов**

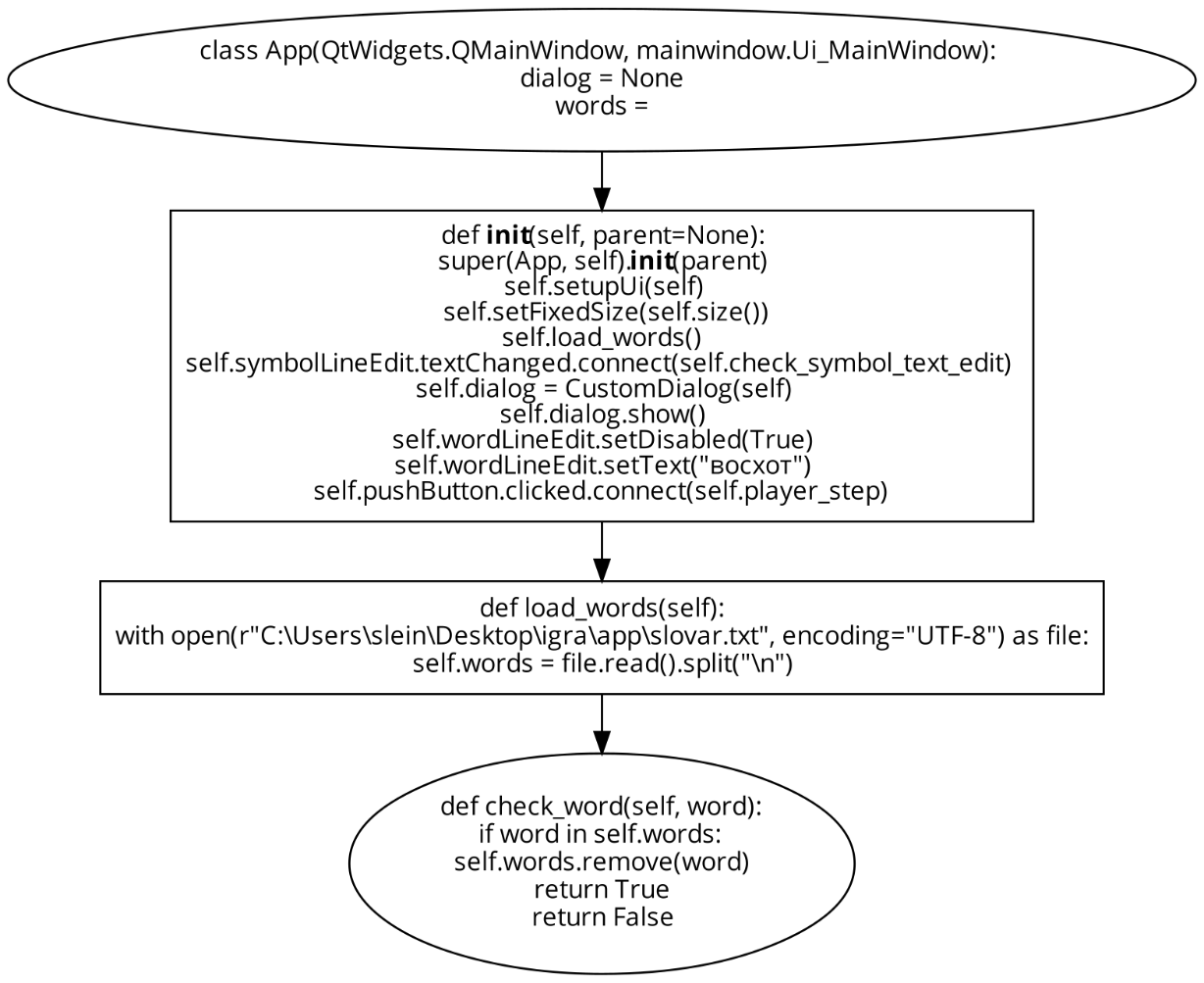


Рис. 5.3.2 Блок-схема класса загрузки словаря

**5.3.3 Блок-схема функции считывания координат**

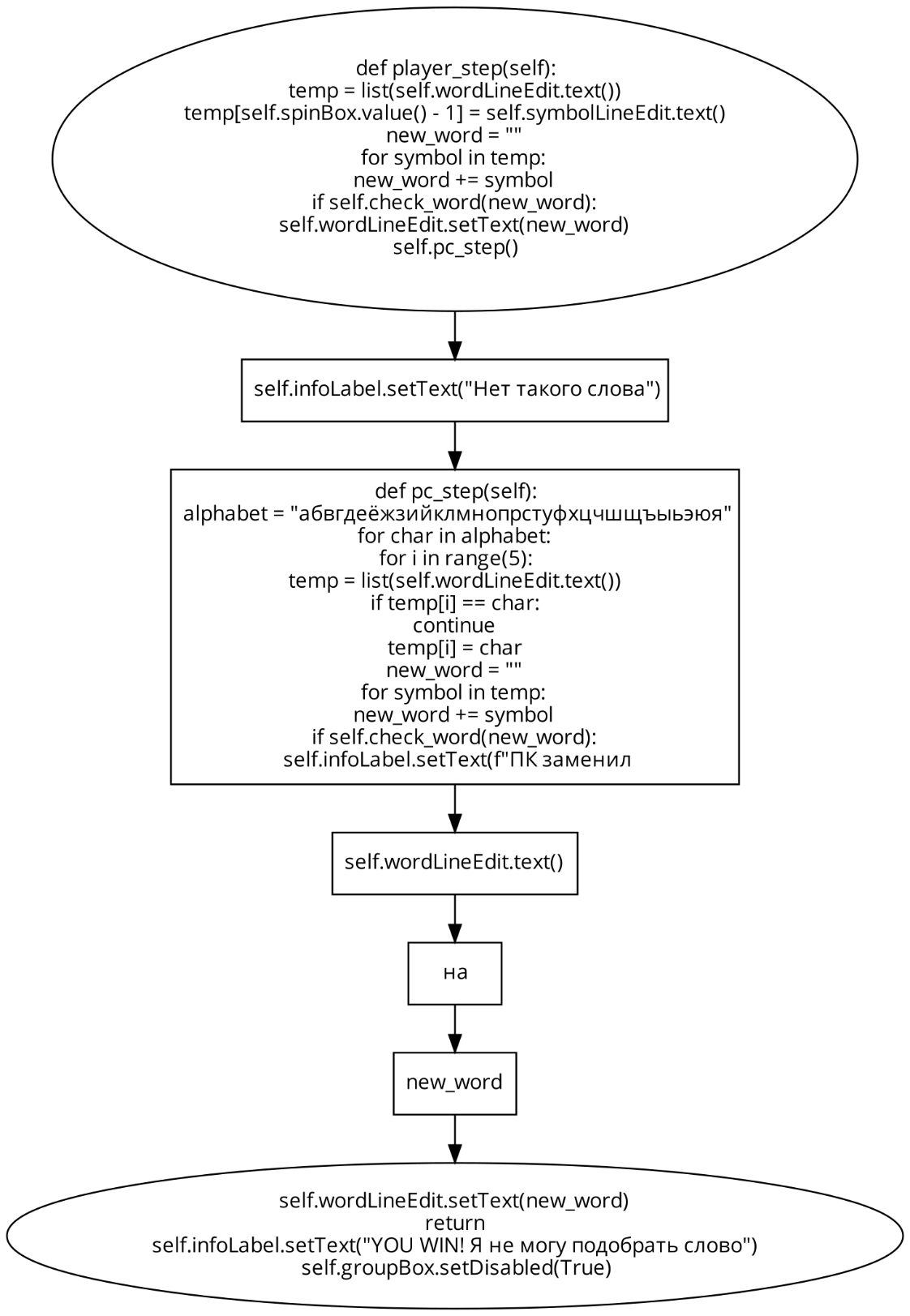


Рис. 5.3.3 Блок-схема функции проверки на правильность введенной буквы и окно о выигрыше

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

6.1 СТРУКТУРНОЕ ТЕСТИРОВАНИЕ

**6.1.1 Игровая часть**

На рисунке представлен граф передачи управления игровой части:

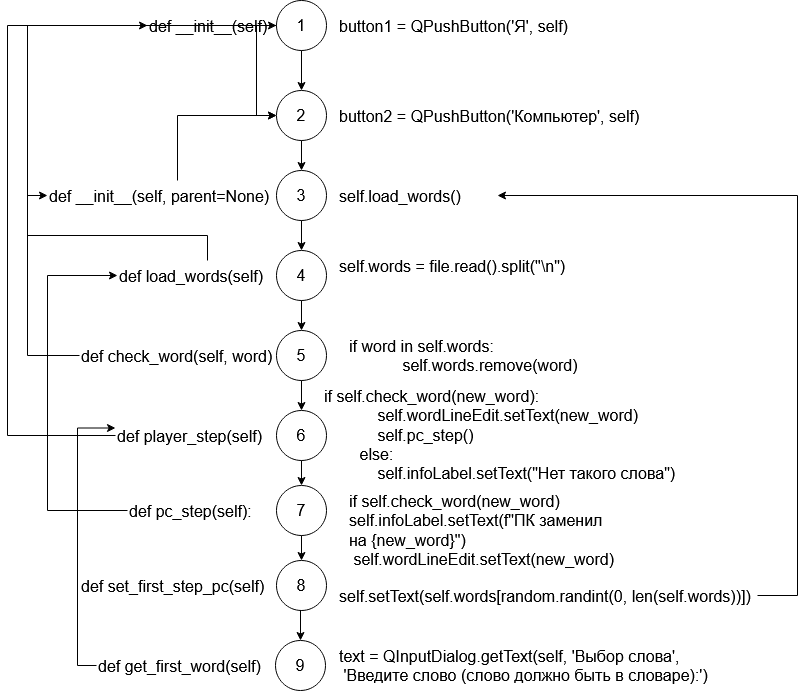
****

Рис 6.1.1 Граф передачи управления игровой части

|  |  |  |
| --- | --- | --- |
| Маршрут | Переменные | Результат |
| 1-3-4-5-7-8 | button1  self.load\_words()  check\_word | Маршрут выполняется |
| 2-3-4-5-7-8 | button2  self.load\_words()  random.randint | Маршрут выполняется |

6.2 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

**6.2.1 Тестирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Тестовые данные | Ожидаемый результат | Полученный результат | Тип ошибки |
| Словарь формата txt | Содержимое текстового файла отобразится в программе | Словарь был открыт и его содержимое отобразилось в поле строки | Нет |
| Права первого хода | Появление окна, дающая выбор первого хода | Появление окна, показывающая окно первого хода | Нет |
| Замены буквы из словаря | Замена буквы в слове | Замена буквы в слове | Нет |
| Окно для ввода слова для первого игрока | Ввод слова и проверка его из словаря | Ввод слова и проверка его из словаря | Нет |
| Окно о победе | Окно о выигрыше игрока в случае если невозможен подбор слов | Окно о выигрыше | Нет |

6.3 ТЕСТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Действие | Реакция системы | Результат |
| 1 | Щелкните по кнопке «Я» | Окно для ввода слова | Верно |
| 2 | Щелкните по кнопке «Компьютер» | Произойдет ввод слова компьютером | Верно |
| 3 | Щелкните по кнопке «Об авторе» | Произойдет открытия нового окна с информацией | Верно |
| 4 | Выберите «номер буквы» для замены | Произойдет выбор числа | Верно |
| 5 | Ввести «букву» | Произойдет ввод буквы в поле | Верно |
| 6 | Щелкните по кнопке «проверить» | Произойдет проверка на замену слова | Верно |
| 7 | Щелкните «выход» | Произойдет выход из приложения | Верно |

СПРАВКИ ДЛЯ ПРИЛОЖЕНИЯ ПО КП

7.1 РУКОВОДСТВО АДМИНИСТРАТОРА

Дополнительная установка программы не требуется, для запуска достаточно нажать двойным кликом ЛКМ по .exe файлу. Программа готова к исполнению.

7.2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

**7.2.1 Назначение программы**

Программа предназачена для осуществления игры «Игра в слова с компьютером». Программа позволяет играть в данную игру.

**7.2.2 Работа с программой**

Запуск программы производится после двойного щелчка левой кнопкой мыши по иконке приложения. После запуска появится окно программы, представленное на рис. 7.2.2

Graphical user interface, text, application

Description automatically generated

Рис. 7.2.2 Главное окно программы

Graphical user interface, text, application

Description automatically generated

Рис 7.2.3 «Окно с диалогом если кто-то одержал победу»

1. Игра начинается автоматически после запуска программы. На игровом поле расположена кнопка проверить и выбор замены буквы.
2. Для замены буквы необходимо выбрать цифру под какой нумерацией пользователь хочет заменить букву и нажать «проверить».
3. Выход из программы осуществляется по нажатию на кнопки из главного меню «Выход», либо при нажатии на крестик в правом верхнем углу. Окно исчезает, программа завершает свою работу и освобождает память компьютера

**7.2.3 Сообщения пользователю**

В результате действий пользователя, могут появляться следующие сообщения об ошибках и сообщения о подтверждениях.

7.2.3 Таблица сообщения об ошибках

|  |  |
| --- | --- |
| Окно сообщения об ошибке | Причина и метод устранения ошибки |
|  | Была введена неправильная буква для замены в слове. Стоит выбрать правильную букву или выбрать другую по нумерации |

ЗАКЛЮЧЕНИЕ

Разработка данного курсового проекта привела к следующим результатам:

* Определены требования к программе;
* Разработано техническое задание и спецификация требования к программному средству;
* Разработан проект архитектуры программной системы;
* Разработан пользовательский интерфейс;
* Произведена разработка архитектуры программного обеспечения;
* Создано рабочее программное обеспечение;
* Программа имеет все функции, которые были определены в техническом задании;
* Программа имеет максимально простой в навигации и легкий в изучении пользовательский интерфейс;
* Написано руководство пользователя к программному обеспечению;
* Было произведено тестирование и отладка программного обеспечения;
* Программа устойчива к ошибкам и имеет защитные функции от неправильных действий пользователя.

Достоинства программы:

- Программа имеет интуитивно понятный пользовательский интерфейс (GUI)

- Приложение не нагружает систему, не занимает много места на жестком диске

- Простота в использование

Недостатки:

- Приложение не имеет более обширный функционал.

Целью данной курсовой работы, являлось углубление знаний и расширение навыков по языку высокого уровня Python и их реализации на персональном компьютере, разработанная мной программа отвечает поставленным целям.

ПРИЛОЖЕНИЕ 1

Программный код основного алгоритма работы программы:

**Листинг файла index.py:**

import random

import sys

from PyQt5 import QtWidgets, QtCore #импорт модулей для работы приложение

from PyQt5.QtWidgets import QTableWidgetItem, QInputDialog, QFileDialog, QDialog, QDialogButtonBox, QVBoxLayout, QLabel, \

QPushButton

import mainwindow

class CustomDialog(QDialog): #создание диалога о выборе первого хода

def \_\_init\_\_(self, parent\_window, parent=None):

super().\_\_init\_\_(parent)

self.setWindowTitle("Подтвердите")

button1 = QPushButton('Я', self)

button2 = QPushButton('Компьютер', self)

self.setWindowModality(QtCore.Qt.ApplicationModal)

self.layout = QVBoxLayout()

message = QLabel("Кто ходит первым?")

self.layout.addWidget(message)

self.layout.addWidget(button1)

self.layout.addWidget(button2)

self.setLayout(self.layout)

button1.clicked.connect(parent\_window.set\_first\_step\_player) #выбор первый ход человек

button2.clicked.connect(parent\_window.set\_first\_step\_pc)#выбор первый ход компьютер

button1.clicked.connect(self.close) #закрыть окно при выборе

button2.clicked.connect(self.close) #закрыть окно при выборе

class App(QtWidgets.QMainWindow, mainwindow.Ui\_MainWindow): #виджеты для главного меню

dialog = None

words = []

def \_\_init\_\_(self, parent=None): #функция загрузки слов

super(App, self).\_\_init\_\_(parent)

self.setupUi(self)

self.setFixedSize(self.size())

self.load\_words()

self.symbolLineEdit.textChanged.connect(self.check\_symbol\_text\_edit)

self.dialog = CustomDialog(self)

self.dialog.show()

self.wordLineEdit.setDisabled(True)

self.wordLineEdit.setText("восхот") #начальное слово

self.pushButton.clicked.connect(self.player\_step) #окно если первый ход человек

def load\_words(self): #загрузка слов из словаря по пути

with open(r"C:\Users\slein\Desktop\igra\app\slovar.txt", encoding="UTF-8") as file:

self.words = file.read().split("\n")

def check\_word(self, word): #функция для проверки буквы

if word in self.words:

self.words.remove(word)

return True

return False

def player\_step(self): #функция хода игрока

temp = list(self.wordLineEdit.text())

new\_word = ""

for symbol in temp:

new\_word += symbol

if self.check\_word(new\_word): #если новое слово

self.wordLineEdit.setText(new\_word)

self.pc\_step() #ход компьютера

else: #если слова нет

self.infoLabel.setText("Нет такого слова") #сообщение что такого слова не существует

def pc\_step(self): #ход компьютера

alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя" #алфавит

for char in alphabet:

for i in range(5):

temp = list(self.wordLineEdit.text())

if temp[i] == char:

continue

temp[i] = char

new\_word = ""

for symbol in temp:

new\_word += symbol

if self.check\_word(new\_word):

self.infoLabel.setText(f"ПК заменил {self.wordLineEdit.text()} на {new\_word}") #замены буквы компьютером

self.wordLineEdit.setText(new\_word) #новое слово

return

self.infoLabel.setText("YOU WIN! Я не могу подобрать слово") # окно если победил игрок

self.groupBox.setDisabled(True)

def set\_first\_step\_pc(self):

self.wordLineEdit.setText(self.words[random.randint(0, len(self.words))])

def get\_first\_word(self):

text, ok = QInputDialog.getText(self, 'Выбор слова',

'Введите слово (слово должно быть в словаре):')

return text

def set\_first\_step\_player(self):

while True:

word = self.get\_first\_word()

if self.check\_word(word):

self.wordLineEdit.setText(word)

self.pc\_step()

break

def check\_symbol\_text\_edit(self): #проверка на символы

if len(self.symbolLineEdit.text()) > 5:

self.symbolLineEdit.setText(self.symbolLineEdit.text()[0])

def except\_hook(except\_type, value, t\_back):

QtWidgets.QMessageBox.critical(

app, "CRITICAL ERROR", str(value),

QtWidgets.QMessageBox.Cancel

)

sys.\_\_excepthook\_\_(except\_type, value, t\_back)

sys.excepthook = except\_hook

app = QtWidgets.QApplication([])

window = App()

window.show()

app.exec\_()

**Листинг файла mainwindow.py**

from PyQt5 import QtCore, QtGui, QtWidgets

from mainwindow2 import Ui\_OtherWindow\_2

class Ui\_MainWindow(object):

def openWindow(self):

self.window = QtWidgets.QMainWindow()

self.ui = Ui\_OtherWindow\_2()

self.ui.setupUi(self.window)

self.window.show()

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(413, 343)

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(11)

MainWindow.setFont(font)

icon = QtGui.QIcon()

icon.addPixmap(QtGui.QPixmap("../../../../../../!DOWNLOAD/console.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

MainWindow.setWindowIcon(icon)

self.centralwidget = QtWidgets.QWidget(MainWindow)

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(14)

self.centralwidget.setFont(font)

self.centralwidget.setObjectName("centralwidget")

self.infoLabel = QtWidgets.QLabel(self.centralwidget)

self.infoLabel.setGeometry(QtCore.QRect(30, 180, 271, 31))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(11)

self.infoLabel.setFont(font)

self.infoLabel.setText("")

self.infoLabel.setObjectName("infoLabel")

self.groupBox = QtWidgets.QGroupBox(self.centralwidget)

self.groupBox.setGeometry(QtCore.QRect(-20, 0, 541, 321))

self.groupBox.setTitle("")

self.groupBox.setObjectName("groupBox")

self.pushButton = QtWidgets.QPushButton(self.groupBox)

self.pushButton.setGeometry(QtCore.QRect(150, 230, 151, 41))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(14)

self.pushButton.setFont(font)

self.pushButton.setObjectName("pushButton")

self.wordLineEdit = QtWidgets.QLineEdit(self.groupBox)

self.wordLineEdit.setGeometry(QtCore.QRect(80, 60, 281, 31))

self.wordLineEdit.setObjectName("wordLineEdit")

self.spinBox = QtWidgets.QSpinBox(self.groupBox)

self.spinBox.setGeometry(QtCore.QRect(340, 120, 42, 22))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(14)

self.spinBox.setFont(font)

self.spinBox.setMinimum(1)

self.spinBox.setMaximum(5)

self.spinBox.setObjectName("spinBox")

self.label\_2 = QtWidgets.QLabel(self.groupBox)

self.label\_2.setGeometry(QtCore.QRect(90, 160, 201, 21))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(11)

self.label\_2.setFont(font)

self.label\_2.setObjectName("label\_2")

self.label = QtWidgets.QLabel(self.groupBox)

self.label.setGeometry(QtCore.QRect(60, 120, 281, 21))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(11)

self.label.setFont(font)

self.label.setObjectName("label")

self.symbolLineEdit = QtWidgets.QLineEdit(self.groupBox)

self.symbolLineEdit.setGeometry(QtCore.QRect(300, 160, 41, 21))

self.symbolLineEdit.setObjectName("symbolLineEdit")

self.btn\_open = QtWidgets.QPushButton(self.groupBox)

self.btn\_open.setGeometry(QtCore.QRect(340, 0, 93, 28))

font = QtGui.QFont()

font.setPointSize(10)

self.btn\_open.setFont(font)

self.btn\_open.setObjectName("btn\_open")

self.btn\_open.clicked.connect(self.openWindow)

MainWindow.setCentralWidget(self.centralwidget)

self.menuBar = QtWidgets.QMenuBar(MainWindow)

self.menuBar.setGeometry(QtCore.QRect(0, 0, 413, 28))

font = QtGui.QFont()

font.setFamily("Intro Regular")

font.setPointSize(11)

self.menuBar.setFont(font)

self.menuBar.setObjectName("menuBar")

self.menuFile = QtWidgets.QMenu(self.menuBar)

font = QtGui.QFont()

font.setFamily("Intro Regular")

self.menuFile.setFont(font)

self.menuFile.setObjectName("menuFile")

MainWindow.setMenuBar(self.menuBar)

self.action\_2 = QtWidgets.QAction(MainWindow)

font = QtGui.QFont()

font.setFamily("Intro Regular")

self.action\_2.setFont(font)

self.action\_2.setObjectName("action\_2")

self.action\_3 = QtWidgets.QAction(MainWindow)

font = QtGui.QFont()

font.setFamily("Intro Regular")

self.action\_3.setFont(font)

self.action\_3.setShortcut("")

self.action\_3.setObjectName("action\_3")

self.menuFile.addSeparator()

self.menuFile.addAction(self.action\_2)

self.menuFile.addAction(self.action\_3)

self.menuFile.addSeparator()

self.menuBar.addAction(self.menuFile.menuAction())

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "Игра в слова"))

self.pushButton.setText(\_translate("MainWindow", "Проверить"))

self.label\_2.setText(\_translate("MainWindow", "на следующую букву - "))

self.label.setText(\_translate("MainWindow", "Заменить букву под номером - "))

self.btn\_open.setText(\_translate("MainWindow", "Об авторе"))

self.menuFile.setTitle(\_translate("MainWindow", "Главное меню"))

self.action\_2.setText(\_translate("MainWindow", "Сохранить как"))

self.action\_2.setStatusTip(\_translate("MainWindow", "Save a file"))

self.action\_2.setShortcut(\_translate("MainWindow", "Ctrl+S"))

self.action\_3.setText(\_translate("MainWindow", "Выйти"))

self.action\_3.setStatusTip(\_translate("MainWindow", "exit"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

**Листинг файла dict generator:**

with open("C:\Users\slein\Desktop\igra\app\slovar.txt", "w", encoding="UTF-8") as file:

temp = ""

for word in words:

temp += word + "\n"

temp.rstrip("\n")

file.write(temp)

ПРИЛОЖЕНИЕ 2

Справка на плагиат, полученная на <https://antiplagiat.ru/>

Graphical user interface, text, application

Description automatically generated

Рис 9. Справка на антиплагиат

Задание по приложению:

|  |  |
| --- | --- |
| 18 | Описание: Приложение, которое переводит текст на другой язык |

Для создания приложения используется среда разработки «MIT App Inventor».

**Шаг 1.** Нужно перейти на сайт «MIT App Inventor» и создать новый проект



Рис. 10 Создание проекта

**Шаг 2.** Выбрать вкладку «User Interface» и создать «TextBox»

Graphical user interface, application

Description automatically generated

Рис 10.1 Создание Textbox

**Шаг 3.** Задать характеристики для «heigh»t и width».

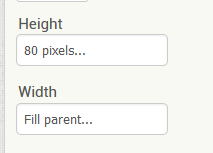


Рис 10.2 Ширина и высота

**Шаг 4.** В той же вкладке интерфейса создать кнопку

Graphical user interface

Description automatically generated

Рис 10.3 Создание кнопки

**Шаг 5.** Создать «Label» для пустой строки для переведенного слова

Рис 10.4 «Label»

**Шаг 6.** Из вкладки «Media» выбрать YandexTranslate для перевода текста

Graphical user interface, application

Description automatically generated

Рис 10.5 Добавление перевода

**Шаг 7.** На выходе получаем такой интерфейс

Graphical user interface, application, Word

Description automatically generated

Рис 10.6 Интерфейс приложения

**Шаг 8.** Переходим в раздел «Blocks» и создаем блоки для работы приложения

Graphical user interface, application

Description automatically generated

Рис 10.7 Блоки для работы приложения

**Шаг 9.** Компилируем это в готовый .apk файл для работы на телефоне:

Graphical user interface, text, application

Description automatically generated

Рис 10.8 Создание файла

На выходе получаем готовое мобильное приложение «переводчик»

A picture containing background pattern

Description automatically generated

Рис 10.9 Мобильное приложение «Переводчик»