**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель, доцент департамента «Программной инженерии», кандидат технических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.С. Белова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия», кандидат технических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.05.15-01 81 01-1** | | **ИНТЕРАКТИВНАЯ АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ НАНООБЪЕКТОВ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.05.15-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ219  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Фирсов Ф.А. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |
|  | | |
| **Москва 2023** | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕНО  RU.17701729.05.15-01 81 01-1-ЛУ | |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.05.15-01 81 01-1** | | **ИНТЕРАКТИВНАЯ АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ НАНООБЪЕКТОВ**  **Пояснительная записка**  **RU.17701729.05.15-01 81 01-1**  **Листов 31** | | | |
|  |  | | |
|  | | |
| **Москва 2023** | | | |

|  |  |
| --- | --- |
|  |  |
|  |  |

СОДЕРЖАНИЕ

[**АННОТАЦИЯ 4**](#_Toc132054687)

[**ВВЕДЕНИЕ 6**](#_Toc132054688)

[**1.1. Наименование программы 6**](#_Toc132054689)

[**1.2. Документ, на основании которого ведется разработка 6**](#_Toc132054690)

[**2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 7**](#_Toc132054691)

[**2.1. Назначение программы 7**](#_Toc132054692)

[**2.1.1. Функциональное назначение 7**](#_Toc132054693)

[**2.1.2. Эксплуатационное назначение 7**](#_Toc132054694)

[**2.2. Краткая характеристика области применения 7**](#_Toc132054695)

[**3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 8**](#_Toc132054696)

[**3.1. Постановка задачи на разработку программы 8**](#_Toc132054697)

[**3.2. Описание и обоснование архитектуры программы 9**](#_Toc132054698)

[**3.2.1. Описание архитектуры программы 9**](#_Toc132054699)

[**3.2.2. Обоснование архитектуры программы 1**](#_Toc132054700)**2**

[**3.3. Описание алгоритма работы программы 13**](#_Toc132054704)

[**3.4. Описание выбора метода организации входных и выходных данных 14**](#_Toc132054706)

[**3.5. Описание и обоснование выбора состава технических и программных средств 14**](#_Toc132054707)

[**3.5.1. Состав технических и программных средств 14**](#_Toc132054708)

[**3.5.2. Обоснование выбора технических и программных средств 15**](#_Toc132054709)

[**4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 16**](#_Toc132054710)

[**4.1. Ориентировочная экономическая эффективность 16**](#_Toc132054711)

[**4.2. Предполагаемая потребность 16**](#_Toc132054712)

[**4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 16**](#_Toc132054713)

[**5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 18**](#_Toc132054714)

[**ПРИЛОЖЕНИЕ 1 19**](#_Toc132054715)

[**ПРИЛОЖЕНИЕ 2 20**](#_Toc132054716)

[**ПРИЛОЖЕНИЕ 3 21**](#_Toc132054717)

[**ПРИЛОЖЕНИЕ 4 22**](#_Toc132054718)

[**ПРИЛОЖЕНИЕ 5**](#_Toc132054719) **30**

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 3**](#_Toc132054720)**2**

# **АННОТАЦИЯ**

Настоящий программный документ представляет собой пояснительную записку к программному проекту «Интерактивная автоматизированная система обработки экспериментальных данных нанообъектов», используемого для обработки результатов физико-химических обследований нанообъектов. Работа над проектом велась в сотрудничестве с доктором химических наук, профессором кафедры цифровых и аддитивных технологий (ЦАТ) Института перспективных технологий и индустриального программирования (ИПТИП) МИРЭА - Российского технологического университета (РТУ МИРЭА). Соответственно система может быть использована сотрудниками кафедры в будущих исследованиях свойств нанообъектов.

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием организации, утвердившей данный документ.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

В разделе «Технические характеристики» присутствуют следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание и обоснование алгоритма работы программы, описание и обоснование математического алгоритма работы программы, описание выбора метода организации входных и выходных данных, описание работы с файловой системой, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов[1];
2. ГОСТ 19.102-77 Стадии разработки[2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов[3];
4. ГОСТ 19.104-78 Основные надписи[4];
5. ГОСТ 19.105-78 Общие требования к программным документам[5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом[6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению[7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78[8], ГОСТ 19.604-78[9].

Перед прочтением данного документа следует ознакомиться с терминологией, приведенной в Приложении 1.

**ВВЕДЕНИЕ**

* 1. **Наименование программы**

**Наименование темы разработки:** «Мобильное приложение для онлайн соревнования в решении кроссвордов».

**Наименование темы разработки на английском языке:** «Mobile Application for Online Competitions in Solving Crosswords».

* 1. **Документ, на основании которого ведется разработка**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

Документ утвержден организацией: Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Программная инженерия».

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Назначение программы**
      1. **Функциональное назначение**

Программа позволяет пользователям участвовать в соревнованиях по решению кроссворда с другими пользователями, а также узнавать свой рейтинг исходя из проведенных рейтинговых игр. Имеет развлекательное назначение.

* + 1. **Эксплуатационное назначение**

Предположительными конечными пользователями могут быть как дети школьного возраста, так и взрослые люди, увлеченные решением кроссвордов. Соревновательный эффект в игре должен способствовать расширению аудитории и поддержанию ее интереса к игре.

* 1. **Краткая характеристика области применения**

«Мобильное приложение для онлайн соревнования в решении кроссвордов» — это игровое приложение, которое позволит не только решать кроссворды в одиночку, но и соревноваться с другими игроками. Пользователи смогут сразиться как с друзьями, так и с равными по силе игроками в рейтинговой игре в борьбе за рейтинговые ранги.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

«Мобильное приложение для онлайн соревнования в решении кроссвордов» должно предоставлять возможность выполнения следующих задач:

1. Предоставление пользователю интерфейса для регистрации в приложении.
2. Предоставление пользователю интерфейса для входа в аккаунт в приложении.
3. Смену пароля или игрового имени в интерфейсе страницы настройки.
4. Предоставление интерфейса с данными о текущем ранге и очках повышения.
5. Доступ начать рейтинговую, обычную или тренировочную игру меню.
6. Свободное перемещение между страницами настройки, главным меню и статистикой.
7. Предоставления интерфейса, отображающего кроссворд и описания для загаданных слов.
8. Предоставления интерфейса с информацией о количестве отгаданных слов игроком и его соперником.
9. Возможность проверки слова, введенного пользователем в кроссворд.
10. После окончание игры предоставление информации о победе/поражении и опционально о изменении очков повышения и ранга игрока.
11. Отражения информации на странице статистика о:
    1. Отображение общего количества сыгранных игр.
    2. Отображение количества сыгранных рейтинговых игр.
    3. Отображение количества сыгранных обычных игр.
    4. Отображение общего количества побед.
    5. Отображение количества побед в рейтинговых играх.
    6. Отображение количества побед в обычных играх.
    7. Отображение общего процента побед.
    8. Отображение процента побед в рейтинговых играх.
    9. Отображение процента побед в обычных играх.
    10. Отображение общего количества отгаданных слов.
    11. **Описание и обоснование архитектуры программы**
        1. **Описание архитектуры программы**

«Мобильное приложение для онлайн соревнования в решении кроссвордов» является клиент-серверным мобильным приложением приложением. Мобильное приложение (клиент) написано с использованием среды разработки Android Studio. Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, является официальным инструментом для создания приложений на платформе Android с использованием встренных библиотек этой платформы. Для написания кода клиента был выюран язык Java, так как он является одним из “классических” языков платформы Android, поддерживается Android Studio и является универсальным и комфортным инструментом. Клиентская часть отвечает за предоставления пользователю интерфейса, описанного выше.

Для создания серверной части был выбран язык Python и его встроенные библиотеки socket и thread. Серверная часть обеспечивает генерацию кроссвордов, ответы на запросы проверки слов, синхронизации игроков и прочую логику взаимодействия клиентов друг с другом и базой данных. Python был выбран из-за его удобства в использовании механизмов связывающих серверную часть с остальными модулями и многопоточной работы, а также лаконичности кода.

Для создания базы данных была выбрана реляционная система управления базами данных MySQL. Эта СУБД была выбрана, так как является одной из самых распространённых баз данных, поддерживает работу с Python, на котором написана серверная часть, и показывает хорошую производительность и эффективность.

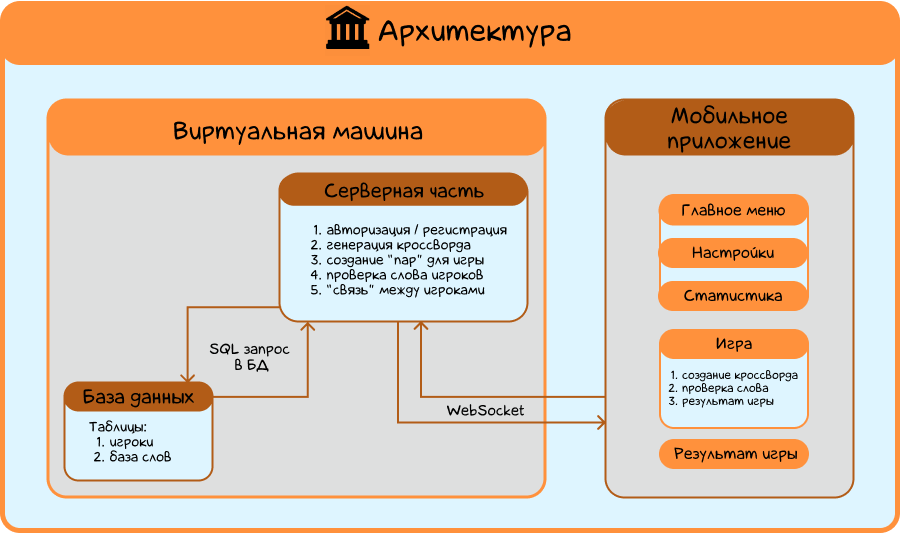


Рисунок 1. Общая архитектура.

Вначале рассмотрим схему взаимодействия элементов системы. Взаимодействие между компонентами можно разбить на следующие фазы:

1. Пользователь совершает действие в мобильном приложении, требующее запроса на сервер.
2. Мобильное приложение обрабатывает действие пользователя и создает объект запроса на сервер.
3. Мобильное записывает в WebSocket запрос и отправляет его серверу.
4. Сервер распознает тип запроса.
5. (Опционально в зависимости от типа запроса) Сервер подключается и создает запрос в базу данных.
6. Сервер генерирует ответ на запрос клиента и записывает результат в WebSocket. Отправляет назад клиенту.
7. Мобильное приложение обрабатывает результат и выводит результат действия пользователю.

Теперь давайте более подробно рассмотрим внутреннее устройство описанных компонентов.

1. **Пользователь совершает действие в мобильном приложении, требующее запроса на сервер.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Запросы на сервер мобильного приложения** | | | | |
| **событие** | **действие** | **страница** | **тип запроса** | **название метода** |
| вход в систему | Нажатии кнопки “войти” | страница входа | login | setOnClickListener |
| регистрация нового пользователя | Нажатии кнопки “регистрация” | страница регистрации | register | setOnClickListener |
| отмена поиска игры | Нажатии кнопки “отмена” | ожидание игры | cancel queue | setOnClickListener |
| проверка нахождения противника | регулярно | ожидание игры | check queue | check\_queue |
| начать новую игру | Нажатии кнопки “Поехали!” | главная страница | add to queue | setOnClickListener |
| отгадал ли противник новые слова | регулярно | игра | game status | gameStatus |
| проверка слова | нажатие кнопки с галочкой | игра | check word | setOnClickListener |

Рисунок 2. Таблица “Запросы на сервер мобильного приложения”

При желании совершить действие указанное в первом столбце таблицы на рисунке 2 пользователь совершает действие указанное в одноименном столбце. После чего запрос обрабатывается на этапе 2.

1. **Мобильное приложение обрабатывает действие пользователя и создает объект запроса на сервер.**

Действие пользователя для каждого события обрабатывается в методе, указанном в столбце “название метода” таблицы на рисунке Для запроса создается JSON объект содержащий тип из таблицы и конкретные для каждого типа запроса данные.

1. **Мобильное записывает в WebSocket запрос и отправляет его серверу.**

Создание WebSocket и запи и *public static JSONObject executePost(JSONObject json)* класса *LoginMenu.* В качестве единственного параметра принимается JSON объект, сгенерированный на стадии 2.

1. **Сервер распознает тип запроса.**

Обработка происходит в методе *request\_to\_client(data: str)* файла *Requester.py*. В качестве входного параметра принимается записанный в строку JSON объект. По полю *type* определяется тип входящего запроса

1. **Сервер подключается и создает запрос в базу данных.**

Если для получения результата запроса нужно обратиться в базу данных вызывается один из методов *select\_from\_db, insert\_to\_db* из файла *DbWork.py* для получания или вставки данных.

1. **Сервер генерирует ответ на запрос клиента и записывает результат в WebSocket. Отправляет назад клиенту.**

В методе request\_to\_client(data: str) из файла *Requester.py* генерируется ответ в виде JSON объекта с полем *status* соответствующим значению коду состояния HTTP и конкретной для каждого запроса информацией ответа. Метод *client\_thread* записывает его в WebSocket и отправляет клиенту.

1. **Мобильное приложение обрабатывает результат и выводит результат действия пользователю.**

Обработка происходит в том же методе, где инициализирован запрос.

* + 1. **Обоснование архитектуры программы**

При разработке «Мобильное приложение для онлайн соревнования в решении кроссвордов» было принято несколько важных архитектурных решений. Объясним каждое из них:

1. Разделение на клиент (мобильное приложение) и сервер. В первую очередь «Мобильное приложение для онлайн соревнования в решении кроссвордов» представляет собой мобильное приложение для платформы Android, поэтому было необходимо выбрать платформу, которая предоставляет возможности по разработке графического приложения. Для этих целей было выбрано Android Studio как рекомендованное Google, создателем платформы Android. На ее основе довольно просто написать необходимые интерфейсы, а также прописать логику и работу со сторонними системами. Общение клиентов друг с другом должен кто-то “регулировать”, поэтому появилась необходимость в серверном приложении. Кроме того, генерация кроссворда обязывает к хранению базы данных содержащие все слова, а хранить ее на телефоне не желательно, как в целях безопасности, так и не эффективна. Так как сервер не реализует приложение и минимально контактирует с операционной системой, то жесткой привязки к JAVA и Android Studio уже нет. Наиболее подходящим для задач сервера был выбран язык Python, поскольку вместе с ним можно использовать удобные библиотеки для работы с базой данных и WebSocket и его код лаконичен.
2. В процессе написания серверной обработки запросов я использовал принципы "слоевой" архитектуры. Это способствовало правильной декомпозиции кода, что в свою очередь улучшило читаемость, облегчило тестирование и обнаружение ошибок. Кроме того, такая архитектура обеспечивает возможность добавления новой логики в будущем (если это будет необходимо), делая код более гибким и адаптируемым к изменениям.
   1. **Описание алгоритма работы программы**

Разберем алгоритм работы «Мобильного приложения для онлайн соревнования в решении кроссвордов» по шагам:

1. На первом этапе пользователь запускает приложение системы, в результате чего загружается страница входа;
2. Пользователь переходит на страницу регистрации с помощью соответствующей кнопки или переходит к пункту 4;
3. Пользователь создает новый аккаунт указал логин, пароль и игровое имя.
4. Пользователь входит в приложение используя указанный пре регистрации логин и пароль.
5. Пользователь может перемещаться между страницами статистика, главное меню и настройки.
6. С помощь кнопок влево и вправо в главном меню можно изменить режим игры.
7. Пользователь нажимает кнопку “поехали!” в главном меню и перемещается на страницу ожидания игры.
8. Ожидание подбора соперника и генерации кроссворда.
9. Пользователь способен переключаться между клетками кроссворда нажатием на клетку. Повторное нажатие меняет ориентацию выбранного участка.
10. С помощью клавиш пользователь вводит слово отвечающие загадке в выбранные клетки.
11. Пользователь проверяет верность слова при нажатии кнопки с галочкой. Если слово отгадано верно оно блокируется, показывается сопернику, а игрок получает победное очко.
12. Когда все слова отгаданы пользователь перемещается на страницу результатов.
13. Если у игрока очков больше, чем у соперника, то игра выиграна. В противном случае - проигрыш.
14. В случае выигрыша в рейтинговой игре пользователь повышает свой ранг.
    1. **Описание выбора метода организации входных и выходных данных**

Мобильное и серверное приложение не имеет входных и выходных данных. Все данные хранятся в базе данных. Для корректной работы программы необходима таблица words в датасете WordDuel на текущем sql сервере. Таблица состоит из трех столбцов: word\_id – уникальное целое число, word – слово(строка), definition – загадка(строка)

* 1. **Описание и обоснование выбора состава технических и программных средств**
     1. **Состав технических и программных средств**

Для нормального функционирования мобильного приложения требуется смартфон, оснащенный следующими техническими компонентами:

1. 100 мб свободного дискового пространства;
2. 256 мб ОЗУ или более;
3. Доступ в локальную или глобальную сеть

Для работы мобильного приложения необходим следующий состав программных средств:

1. OC Android 3.8 или больше;

Для нормального серверного приложения требуется компьютер/сервер оснащенный пропорционально нагрузке на систему.

Для работы серверного приложения необходим следующий состав программных средств:

1. OC Windows 7+ или Linux
2. Python 3 c библиотеками threading, mysql, socket
   * 1. **Обоснование выбора технических и программных средств**

Программе необходимо как минимум 256 мб оперативной памяти, поскольку в иных случаях ресурсов компьютера может не быть недостаточно для корректного отображения приложения.

Для работы серверного приложения необходим компьютер c мощностями пропорциональными одновременному количеству активных пользователей.

1. **ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**
   1. **Ориентировочная экономическая эффективность**

В рамках данной работы расчет экономической эффективности не предусмотрен.

* 1. **Предполагаемая потребность**

«Мобильного приложения для онлайн соревнования в решении кроссвордов» исключительно в развлекательных целях.

* 1. **Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами**

Сравнение «Мобильного приложения для онлайн соревнования в решении кроссвордов» с аналогами представлено в следующей таблице:



Основным преимуществом по сравнению с конкурентами является смешение идей и механик классических приложений с разгадыванием кроссвордов с идеями соревновательных онлайн игр. Из классических кроссвордов будут взяты основы геймплея – угадывание слов, размещенных на поле, при помощи данных описаний, а из соревновательных онлайн игр взяты идеи о противостоянии 2-х игроков в решении одного кроссворда, механики рейтинга и рангов, ограничение по времени, нетривиальный подбор игроков и другое.

1. **Кроссворды на русском**

Классический представитель мобильного приложения для решения кроссвордов с большим набором заготовленных заранее заданий.

1. **Эрудит онлайн**

Мобильная версия словесной игры Эрудит построенной на составлении слов и выкладывании их на поле по принципу кроссворда. В отличие от классических кроссвордов позволяет соревноваться нескольким игрокам, однако уходит от отгадывания слов к составлению их из данных букв.

1. **Words of Wonders**

Во многом похожа на Эрудит, но все слова состоят из одних и тех же букв данных изначально и отличаются только длинной и порядком букв.

1. **Clash Royale**

Один из самых популярных примеров соревновательных онлайн игр на мобильные устройства. Геймплей построен на противостоянии примерно равных по силе игроков в дуэлях по результатам, которые меняется рейтинг, арена (аналог рангу) и открываются новые возможности. Является прекрасным примеров реализации классических механик соревновательных онлайн игр таких как: соревновательные игры, рейтинг, система друзей, система кланов.

1. **ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**
2. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
12. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. John Torjo «Boost.Asio C++ Network Programming»
14. The Art of Computer Programming, Donald Ervin Knuth
    1. Volume 1: Fundamental Algorithms. Third Edition (Reading, Massachusetts: Addison-Wesley, 1997)
    2. Volume 2: Seminumerical Algorithms. Third Edition (Reading, Massachusetts: Addison-Wesley, 1997)
    3. Volume 4B: Combinatorial Algorithms, Part 2 (Upper Saddle River, New Jersey: Addison-Wesley, 2023)
15. Документация по NumPy [Электронный ресурс] / NumPy community. Режим доступа: <https://numpy.org/> , свободный.

# **ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

|  |
| --- |
| Таблица 1 |

|  |  |
| --- | --- |
| Термин | Определение |
|  |
| Соревнование, игра | Ограниченный по времени этап игры, когда два участника соревнования решают общий кроссворд на время. |
| Ранг игрока | Показатель мастерства игрока определяемый исходя из проведенных игроком рейтинговых игр. Состоит из ключевого слова и буквы, например «Золото A». |
| СУБД | Система управления базы данных |

# **ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КОМПОНЕНТОВ ИНФРАСТУКТУРЫ СИСТЕМЫ**

Описание и функциональное назначение компонентов инфраструктуры системы.

Таблица 2

|  |  |  |
| --- | --- | --- |
| Название компонента | Тип | Назначение |
| Мобильное приложение WordDuel | Мобильное приложение | Основная часть мобильного приложения для онлайн соревнования в решении кроссвордов |
| Серверное приложение | Скрипты | Запускает серверную часть программы |

**ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

**МОБИЛЬНОГО ПРИЛОЖЕНИЯ**

Описание и функциональное назначение классов.

Таблица 3

|  |  |
| --- | --- |
| Класс | Назначение |
| Word | Класс, описывающий слово, как часть кроссворда. |
| Crossword | Класс, описывающий кроссворд. Хранит информацию о позициях всех слов, о текущем статусе (отгадана/не отгадана/пустая) всех клеток. |
| LoginMenu | Класс, обрабатывающий логику поведения и данные, которые использует окна регистрации. |
| RegisterMenu | Класс, обрабатывающий логику поведения и данные, которые использует окно регистрации. |
| StatisticMenu | Класс, обрабатывающий логику поведения и данные, которые использует окно статистики. |
| WaitGameMenu | Класс, обрабатывающий логику поведения и данные, которые использует окно ожидание игры. |
| ResultActivityMenu | Класс, обрабатывающий логику поведения и данные, которые использует окно результатов игры. |
| MainActivity | Класс, обрабатывающий логику поведения и данные, которые использует главного меню. |
| SettingsMenu | Класс, обрабатывающий логику поведения и данные, которые использует окно настройки. |

**ПРИЛОЖЕНИЕ 4**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ WPF-ПРИЛОЖЕНИЯ**

Таблица 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей, методов и свойств класса Word | | | | |
| Поля | | | | |
| Наименование | Модификатор  доступа | Тип аргумента | Назначение | |
| map | private | int[][][] | Информация о клетках поля кроссворда. | |
| words | private | Word[] | Информация о словах загаданных в кроссворде. | |
| crossword\_id | public | int | Уникальный идентификатор кроссворда. | |
| wordid\_to\_id | public | JSONObject | Словарь преобразования идентификатора слова в базе данных в локальный идентификатор. | |
| Методы | | | | |
| Наименование | Модификатор  доступа | Тип | Аргументы | Назначение |
| getStatus | public | int | int i, int j, boolean d | Получения статуса клетки по координатам в зависимости от ориентации. |
| getStatus | public | int | int i, int j | Получения статуса клетки по координатам. |
| getHeight | public | int | - | Получения ширины кроссворда. |
| getWidth | public | int | - | Получения высоты кроссворда. |
| getOrientation | public | boolean | int id | Получения ориентации слова по его локальному идентификатору. |
| getSize | public | int | int id | Получения длины слова по его локальному идентификатору. |
| getDefinition | public | String | int id | Получения загадки слова по его локальному идентификатору. |
| getWordId | public | String | int id | Получения глобального идентификатора слова по его локальному идентификатору. |
| getI | public | int | int id | Получения первой координаты слова по его локальному идентификатору. |
| getJ | public | int | int id | Получения второй координаты слова по его локальному идентификатору. |
| setGood | public | int | int id | Устанавливает клутку в статус отгаданной. |

Таблица 4.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей, методов и свойств класса GameActivity | | | | |
| Свойства | | | | |
| Наименование | Модификатор  доступа | Тип | Назначение | |
| selectedI | private | int | Первая координата выбранной клетки | |
| selectedJ | private | int | Вторая координата выбранной клетки | |
| selectedId | private | int | Идентификатор слова, в который входит выбранная клетка | |
| selectedOrientation | private | boolean | Ориентация слова, в который входит выбранная клетка | |
| crossword | private | Crossword | Информация о кроссворде | |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. | |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. | |
| background | private | VideoView | Видео на заднем фоне. | |
| your\_score | public | int | Количество отгаданных противником слов. | |
| enemy\_score | public | int | Количество отгаданных игроком слов. | |
| crossword\_id | private | int | Идентификатора текущего кроссворда. | |
| width | private | int | Ширина клетки кроссворда. | |
| used | private | Set<String> | Множество отгаданных слов. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| makeGood | private | void | int word\_id | Установка слова в статус, отгаданного с блокировкой и перекраской клеток. |
| setWord | private | void | int word\_id | Запись слова в кроссворд, если оно отгадано игроком. |
| createCrossword | public | void | - | Отрисовка кроссворда. |
| unmarkWord | private | void | - | Отмена выделения текущего слова. |
| markWord | private | void | int i, int j, boolean orient | Выделить соответствующее слово. |
| createKeyboard | private | void | - | Отрисовка клавиатуры. |
| addSpace | private | void | LinearLayout keyboard | Добавление пустого места в клавиатуру. |
| addLine | private | void | LinearLayout linearLayout, List<String> letters, int height, int width, int k | Добавление строки в клавиатуру. |
| selectLetter | private | void | int i, int j, boolean orient | Выбор буквы в кроссворде. |
| findFirstWord | private | void | - | Поиск не отгаданного слова. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |
| gameStatus | private | void | - | Синхронизация с соперником отгаданных слов. |

Таблица 4.3

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса LoginMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |
| executePost | public | JSONObject | JSONObject json | Создание запроса на сервер. |
| showRegistor | private | void | - | Открыть меню регистрации |

Таблица 4.4

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса RegisterMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |

Таблица 4.5

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса ResultActivityMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |

Таблица 4.6

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса SettingsMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |

Таблица 4.7

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса StatisticMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |
| count\_games | public | int | Количество сыгранных игр. |
| games\_wins | public | int | Количество выигранных игр. |
| words\_wins | public | int | Количество отгаданных слов. |
| rangs\_games | public | int | Количество сыгранных игр в рейтинговых матчах. |
| rangs\_wins | public | int | Количество выигранных игр в рейтинговых матчах. |
| series\_rang | public | int | Лучшая серия побед в рейтинговых матчах. |
| series\_rang\_cur | public | int | Текущая серия побед в рейтинговых матчах. |
| series\_comm | public | int | Лучшая серия побед в обычных матчах. |
| series\_comm\_cur | public | int | Текущая серия побед в обычных матчах. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |

Таблица 4.8

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса WaitGameMenu | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| mCurrentVideoPosition | private | int | Текущий момент видео, отображаемого на фоне. |
| background | private | VideoView | Видео на заднем фоне. |
| mMediaPlayer | private | MediaPlayer | Плеер видео на заднем фоне. |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| onCreate | protected | void | Bundle savedInstanceState | Создание окна. |
| onPause | protected | void | - | Остановка видео при закрытии окна. |
| onResume | protected | void | - | Продолжение видео при открытии окна. |
| onDestroy | protected | void | - | Завершение видео при закрытии окна. |

Таблица 4.9

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей, методов и свойств класса Word | | | |
| Свойства | | | |
| Наименование | Модификатор  доступа | Тип | Назначение |
| i | public | int | Первая координата первой буквы. |
| j | public | int | Вторая координата первой буквы. |
| size | public | int | Длина слова |
| orientation | public | boolean | Ориентация слова |
| definition | public | String | Загадка для кроссворда. |
| word\_id | public | String | Глобальный идентификатор слова. |

# **ПРИЛОЖЕНИЕ 5**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ СЕРВЕРНОГО ПРИЛОЖЕНИЯ**

Таблица 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей и методов класса CrosswordGenerator | | | |
| Поля | | | |
| Наименование | | | Назначение |
| word\_grid | | | Поле кроссворда |
| words | | | Список слов |
| character\_comparison\_grid | | | Таблица сравнения символов |
| word\_location\_info | | | Информация о положении слов в кроссворде |
| Методы | | | |
| Наименование | Статический | Аргументы | Назначение |
| find\_comparisons | - |  | Поиск колличества пересечений в кроссворде |
| compare\_words | + | word1: str, word2: str | Возвращает строку из всех символов, которые есть в обоих словах. |
| create\_grid | - | - | Создает поле кроссворда |
| place\_words | - | selected: str, orientation: bool, starting\_position: tuple | Размещает выбранное слово в выбранной позиции |
| test\_placement | - | intersection: list, orientation: bool, possible: list, word: str | Тестовое размещение слов в кроссворде для оценки количества пересечений. |
| test\_collision | - | possible\_intersections: list, intersection: list, word: str, orientation:bool | Тестовое размещение слова в кроссворде для оценки количества пересечений. |
| find\_intersections | - | key: int, count: list | Поиск пересечений. |
| empty\_column\_left | - | - | Проверка на возможность удалить левый столбец. |
| empty\_column\_right | - | - | Проверка на возможность удалить правый столбец. |
| clear\_left\_column | - | - | Удалить левый столбец. |
| clear\_right\_column | - | - | Удалить правый столбец. |
| get\_crossword | - | - | Возвращает информацию о кроссворде. |

Таблица 5.2

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей и методов скрипта Requester | | | |
| Поля | | | |
| Наименование | | | Назначение |
| games | | | Текущие игры |
| crossword\_id | | | Номер следующей создаваемой игры |
| queue | | | Очереди игроков |
| users\_status | | | Текущий статус игроков |
| queue\_semaphore | | | Семафор для синхронизации работы потоков с очередью |
| Методы | | | |
| Наименование | Статический | Аргументы | Назначение |
| generate\_crossword | - | - | Генерация нового кроссворда. |
| login | - | request: dict | Запрос входа в аккаунт. |
| register | - | request: dict | Запрос регистрации аккаунта. |
| check\_word | - | request: dict | Проверка верности слова. |
| add\_to\_queue | - | request: dict | Добавление игрока в очередь ожидание игры. |
| check\_queue | - | request: dict | Проверка очереди ожидания игры на нахождение соперника. |
| game\_status | - | request: dict | Проверка на отгадывание противником новых слов. (Синхронизация с противником) |
| cansel\_queue | - | request: dict | Отмена поиска игры. |
| request\_to\_client | - | data: str | Ответ на запрос клиента: получение запроса и перенаправление запроса в соответсвующий метод. |
| update\_results | - | request: dict | Обновление дынных после завершения игры. |

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |