**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  доцент департамента  программной инженерии факультета компьютерных наук,  канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дударев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **МАТЕМАТИЧЕСКАЯ ВИКТОРИНА ДЛЯ ДВУХ ИГРОКОВ «МАТЕМАТИЧЕСКАЯ ДУЭЛЬ»**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ186  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /М.А. Семенов /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
|  | |  |

**Москва 2019**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 81 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **МАТЕМАТИЧЕСКАЯ ВИКТОРИНА ДЛЯ ДВУХ ИГРОКОВ «МАТЕМАТИЧЕСКАЯ ДУЭЛЬ»**  **Пояснительная записка**  **RU.17701729.04.01-01 81 01-1**  **Листов 28** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2019**

ОГЛАВЛЕНИЕ

[1. ВВЕДЕНИЕ 4](#_Toc8825987)

[1.1. Наименование программы 4](#_Toc8825988)

[1.2. Документы, на основании которых ведется разработка 4](#_Toc8825989)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc8825990)

[2.1. Назначение программы 5](#_Toc8825991)

[2.1.1. Функциональное назначение 5](#_Toc8825992)

[2.1.2. Эксплуатационное назначение 5](#_Toc8825993)

[2.2. Краткая характеристика области применения 5](#_Toc8825994)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc8825995)

[3.1. Постановка задачи на разработку программы 6](#_Toc8825996)

[3.2. Описание алгоритма и функционирования программы 6](#_Toc8825997)

[3.2.1. Описания работы программы при обмене данными на сервере 6](#_Toc8825998)

[3.2.2. Описания взаимодействия экранов программы 6](#_Toc8825999)

[3.2.3. Описание серверной части приложения 6](#_Toc8826000)

[3.2.4. Описание алгоритма получения неповторяющихся задач в одной игре 7](#_Toc8826001)

[3.2.5. Описания алгоритма входа в приложение 7](#_Toc8826002)

[3.2.6. Описание принципа размещения данных в БД 7](#_Toc8826003)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 8](#_Toc8826004)

[3.3.1. Описание метода организации входных и выходных данных 8](#_Toc8826005)

[3.3.2. Обоснования выбора метода организации входных и выходных данных 8](#_Toc8826006)

[3.4. Описание и обоснование выбора состава технических и программных средств 8](#_Toc8826007)

[3.4.1. Состав технических и программных средств 8](#_Toc8826008)

[3.4.2. Обоснование выбора технических и программных средств 8](#_Toc8826009)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 10](#_Toc8826010)

[4.1. Предполагаемая потребность 10](#_Toc8826011)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 10](#_Toc8826012)

[ПРИЛОЖЕНИЕ 1 11](#_Toc8826013)

[ТЕРМИНОЛОГИЯ 11](#_Toc8826014)

[ПРИЛОЖЕНИЕ 2 12](#_Toc8826015)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 12](#_Toc8826016)

[ПРИЛОЖЕНИЕ 3 13](#_Toc8826017)

[ПРАВИЛА ИГРЫ 13](#_Toc8826018)

[1. Один из пользователей приглашает другого пользователя в игру 13](#_Toc8826019)

[7. ПРИЛОЖЕНИЕ 4 14](#_Toc8826020)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 14](#_Toc8826021)

[ПРИЛОЖЕНИЕ 5 15](#_Toc8826022)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ 15](#_Toc8826023)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Математическая викторина для двух игроков «Математическая дуэль»»

Наименование программы для пользователя – «MathDuel».

## Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1012-0 2 от 10.12.18 «Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение программы

### Функциональное назначение

Функциональным назначением данного приложения является приобретение и закрепление знаний в области математики. Подготовка к решению математических задач по алгебре, геометрии, комбинаторике уровня от 7 до 11 класса. Подготовка к успешной сдаче ОГЭ и ЕГЭ. Развитие логического мышления.

### Эксплуатационное назначение

Данная игра может быть использована людьми, которые желают порешать математические задачи в соревновательном режиме.

## Краткая характеристика области применения

«Математическая викторина для двух игроков «Математическая дуэль» – программа, позволяющая проводить досуг и развивать логическое мышление, а также развивать навыки решения математических задач различного уровня сложности по различным темам. Область применения игры «Математическая викторина для двух игроков «Математическая дуэль» – развлекательная и образовательная сферы.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

Разрабатываемая программа должна соответствовать:

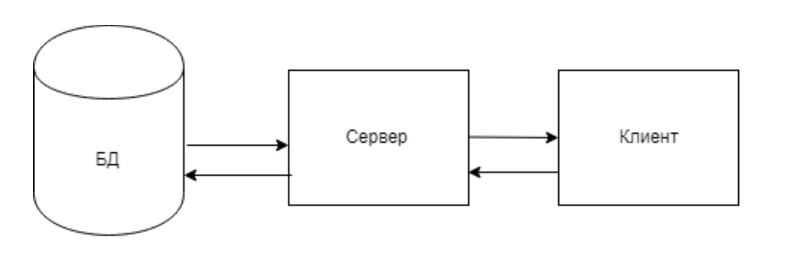
- описанным в техническом задании функциональным характеристикам (Математическая викторина для двух игроков «Математическая дуэль». Техническое задание)

- описанным в техническом задании требованиям к интерфейсу (Математическая викторина для двух игроков «Математическая дуэль». Техническое задание)

- описанным в техническом задании требованиям к надежности (Математическая викторина для двух игроков «Математическая дуэль». Техническое задание)

## Описание алгоритма и функционирования программы

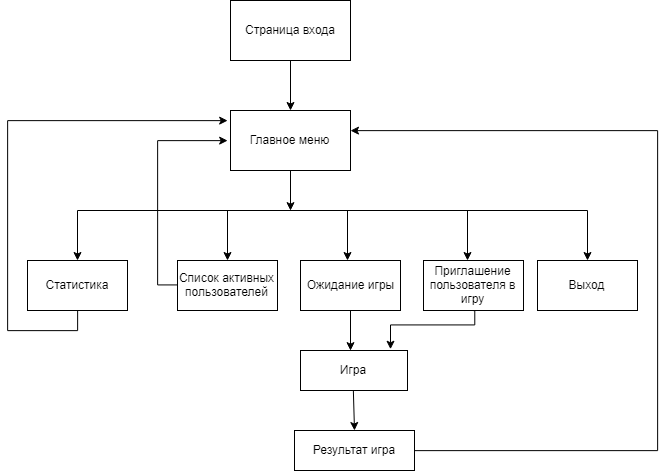
### Описания работы программы при обмене данными на сервере

Для удобства изменения данных в приложении (добавления, изменения или удалении информации о пользователях, задачах и статистике) в программе используется СУБД Microsoft SQL Server (Приложение 2 [10]). Принцип работы программы с приведен на рис. 1:

*Рисунок 1.* Обмен данными между клиентом и сервером

### Описания взаимодействия экранов программы

Принцип взаимодействия экранов приложения приведен на рис. 2, для создания страниц и управления ими использовался Xamarin (Приложение 2 [11]):



*Рисунок 2.* Взаимодействие экранов программы

### Описание серверной части приложения

Сервер работает по протоколу TCP, каждый запрос клиента обрабатывается в отдельном потоке, поэтому сервер может обрабатывать несколько запросов одновременно (Приложение 2[12]). Сервер использует TCP-порт 8888 так как он гарантирует доставку сообщений в прядке их отправления. Сервер возможно переместить на другое устройство, для этого необходимо изменить IP адрес хоста и клиента в классах ClientObject и ServerObject решения ChatServer. Так же так как сервер обращается к БД то необходимо изменить строку подключения к БД в классе ServerObject решения ChatServer. Все данные, которыми обмениваются сервер и клиент, передаются в байтовом представлении. При получении данных они переводятся в строковый вид и обрабатываются. Все сообщения имеют соответствующие знаки различия, специальный набор букв в начале сообщения, чтобы корректно его обработать.

### Описание алгоритма получения неповторяющихся задач в одной игре

Получаем все задачи по заданной теме, затем генерируем пять неповторяющихся случайных чисел. Каждое число лежит в пределе от 0 до количества полученных задач, после этого по номеру задачи получаем заданный список.

### Описания алгоритма входа в приложение

Пользователь вводит имя и пароль, затем данные отправляются на сервер. После чего с сервер обращается к базе данных для получения ответа существует ли такой игрок, или же возможно зарегистрировать нового игрока с таким именем и паролем. Сервер отправляет ответ клиенту, после чего пользователь либо успешно входит в приложение, либо видит сообщение о необходимости изменения данных. Пароль и логин не должны содержать пробелов и символов «/». Логин и пароль также имеют ограничения по длине.

### Описание принципа размещения данных в БД

На рис. 3 показана диаграмма базы данных, она демонстрирует как связаны данные в БД.

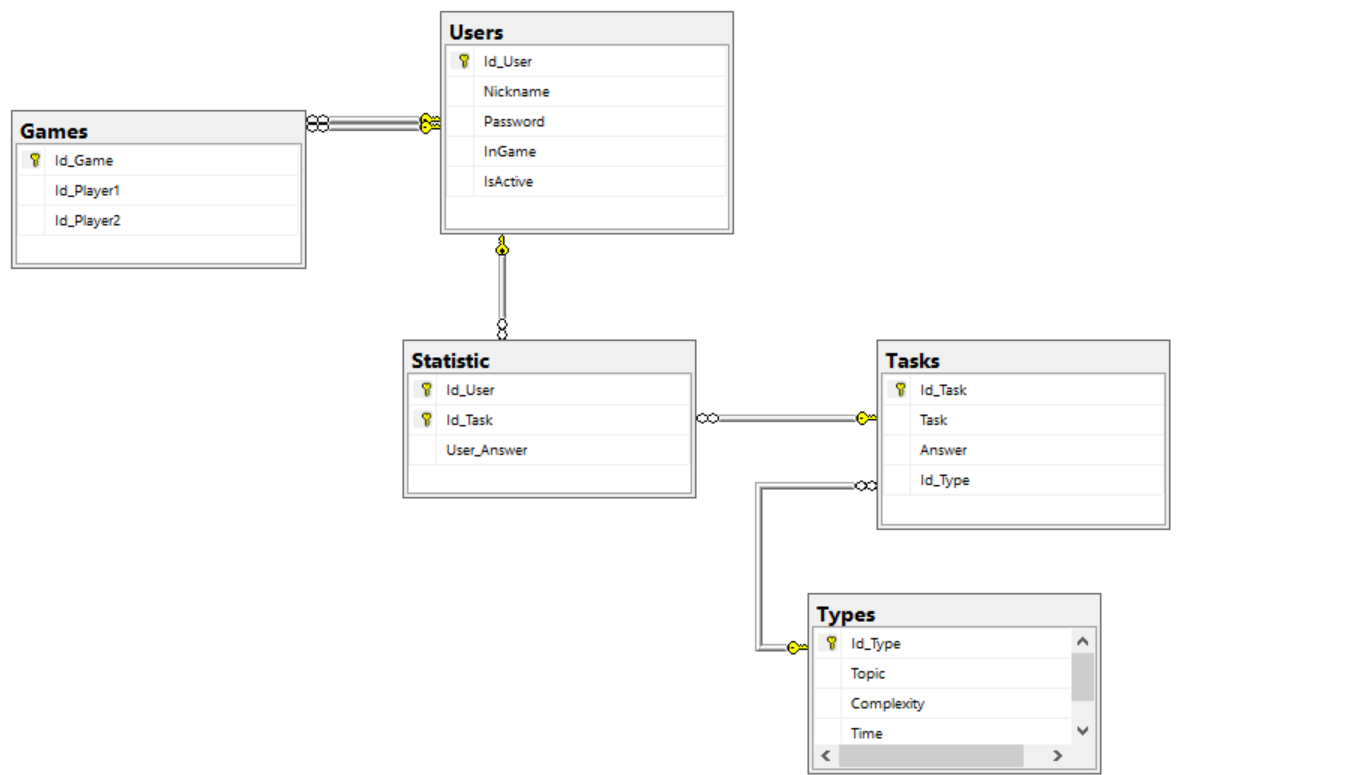
Таблица Users хранит информацию о пользователе, находится ли он в приложении, играет ли сейчас с другими пользователями, его никнейм и пароль, каждый пользователь имеет индивидуальное ID по которому задается первичный ключ.

Таблица Games хранит информацию об игре двух пользователей, их ID и ID игры, после завершения игры данные об игре удаляются.

Таблица Statistic хранит ответы пользователей на задачи, если пользователь второй раз решает задачу, то данные изменяются, с помощью этой таблицы можно определить, сколько задач решил пользователь и сколько из них он решил правильно.

Таблица Tasks хранит тип задач, задачи и правильные ответы на них, каждая задача имеет свои индивидуальный ID по которому задается первичный ключ.

Таблица Types хранит информацию о различных типах задач, тип формируется из темы задачи и сложности, каждый тип имеет индивидуальный ID по которому задается первичный ключ.



*Рисунок 3.* Диаграмма базы данных

## Описание и обоснование выбора метода организации входных и выходных данных

### Описание метода организации входных и выходных данных

Входные данные в программе – имя и пароль пользователя, действия пользователя в приложении (нажатие на кнопку). Приложение предоставляет возможность открыть ранее сохранённую и установленную версию в виде apk файла (для смартфона на платформе Android).

Выходные данные в программе – отклики приложения на действия пользователя в игре, сохранение измененной информации в базе данных.

### Обоснования выбора метода организации входных и выходных данных

Использование БД и СУБД позволяет сохранять данные, оптимально расходуя память устройства, на котором расположен сервер.

## Описание и обоснование выбора состава технических и программных средств

### Состав технических и программных средств

Для работы программы необходим следующий состав технических средств:

Клиентская часть:

- сертифицированное устройство на платформе Android 8.1 и выше

- объем свободной памяти не менее 100 Mb.

Серверная часть.

- 1 Gb оперативной памяти.

- объем свободной памяти не менее 500 Mb.

Сетевое подключение:

- необходимо наличие сетевого подключения со скоростью не ниже 1 МБ/сек.

### Обоснование выбора технических и программных средств

Данный состав технических и программных средств позволяет программе быстро и корректно обрабатывать запрос пользователя.

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Предполагаемая потребность

«Математическая викторина для двух игроков «Математическая дуэль»» – программа, направленная на развитие навыков решения математических задач. Основная аудитория — это ученики средней и старшей школы, которые увлекаются математикой, готовятся к ОГЭ, ЕГЭ и олимпиадам.

## Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Поиск в сети Интернет на момент создания приложения выявил аналоги такие как: «Клевер» и «Математическая игра для двоих», «Test your mathematics skills».

Данное приложение:

1. Специализировано на подготовку к олимпиадам;
2. Не требует финансовых инвестиций на этапе разработки;
3. Бесплатное распространение;
4. Не имеет ограничений в сроке службы.

# ПРИЛОЖЕНИЕ 1

## ТЕРМИНОЛОГИЯ

Ниже приведен список необходимых терминов для ознакомления.

**Главное меню** – экран, который видит пользователь после входа в игру.

**Вход** – прохождение проверки корректности имени и пароля игрока.

**Статистика** – количество решенных задач и количество правильно решенных задач.

**Активные пользователи** – пользователи, которые прошли подсистему входа в игру и находятся либо в игре, либо в главном меню.

**Победитель игры** – игрок, который имеет больше правильных ответов на предложенные задачи на задачи.

**Данные об игре** – таблица, в которой показаны ответы игроков и правильные ответы на задачи.

**Сервер** – компьютер, на котором происходит работа с базой данных и обработка запросов клиента.

**Клиент** – пользователь, который обменивается данными с сервером.

**Ожидание игры** – процесс ожидания приглашения в игру.

**Игра** – процесс решения 5 задач двумя различными игроками.

**Правильно решенная задача** – задача, ответ на которую в точности совпадает с ответом игрока.

**БД** – (база данных), совокупность данных, хранимых в соответствии со схемой данных, действия над которыми выполняют по правилам средств моделирования данных.

**СУБД** – (Система управления базами данных), совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление, создание и использование баз данных.

**Сетевой порт** — параметр протоколов TCP и UDP, определяющий назначение пакетов данных в формате IP, передаваемых на хост по сети.

# ПРИЛОЖЕНИЕ 2

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способ

10. Язык запросов SQL [Электронный ресурс]// URL: <https://sql-language.ru/>(Дата обращения: 11.05.2019, режим доступа: свободный).

11. Программирование Xamarin [Электронный ресурс]// URL: <https://university.xamarin.com/>(Дата обращения: 11.05.2019, режим доступа: свободный).

12. Создание сервера [Электронный ресурс] ]// URL: <https://metanit.com/sharp/net/4.3.php> (Дата обращения: 11.05.2019, режим доступа: свободный).

# ПРИЛОЖЕНИЕ 3

## ПРАВИЛА ИГРЫ

# Один из пользователей приглашает другого пользователя в игру

1. Второй пользователь либо принимает приглашение, находясь в разделе «PlayGame» главного меню, либо отклоняет его, пользователь автоматически отклоняет приглашение, если он не ожидает игры.
2. Пригласивший игрок выбирает тему и сложность пяти задач, из которых будет состоять игра.
3. Пользователи решают задачи, без ограничения по времени, до тех пор, пока оба игрока не решат по 5 задач.
4. Пользователи видят итог игры, то есть победителя игры и ответы противника, свои ответы, правильные ответы на игру.
5. Данные о решенных задачах сохраняются и учитываются в статистике.

# ПРИЛОЖЕНИЕ 4

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 4.1

Описание и функциональное назначение классов

|  |  |
| --- | --- |
| Класс | Назначение |
| ServerObject | Контролирует работу сервера. |
| ClientObject | Контролирует сообщения с клиентом на сервере. |
| Game | Хранит информацию об игре между двумя игроками. |
| Program | Запускает работу сервера. |
| User | Контролирует отправку и обработку сообщении полученных с сервера |
| CurrentUser | Хранит информацию о текущем пользователе. |
| MainPage | Контролирует процесс регистрации и входа пользователя в игру. |
| MainMenu | Предоставляет пользователю возможность воспользоваться функционалом программы. |
| MainMenuMaster | Инициализирует поля главного меню. |
| MainMenuDetail | Предоставляет пользователю возможность пригласить другого игрока в игру. |
| MainMenuMenuItem | Хранит информацию о полях главного меню. |
| AcceptPage | Предоставляет пользователю принять приглашение в игру или отклонить его. |
| App | Инициализирует приложение, отвечает за работу приложения. |
| ChooseAnswers | Предоставляет пригласившему игроку выбор темы следующих вопросов. |
| ExitPage | Закрывает приложение. |
| GetActiveUsers | Предоставляет пользователю список активных пользователей. |
| GetStatistic | Предоставляет пользователю статистику на отвеченные вопросы. |
| PlayGamePage | Реализует ожидание пользователем приглашения в игру. |
| ShowResaltPage | Демонстрирует результаты игры и определяет исход игры. |
| ShowErPage | Информирует пользователя что его противник покинул игру. |
| TasksPage | Демонстрирует условие задач, считывает и обрабатывает ответы пользователя. |

# ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 5.1

Описание полей, методов и свойств класса ServerObject.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| tcpListener | | private | | TcpListener | | Подключение новых пользователей к серверу. | |
| ip | | private | | string | | IP адрес сервера. | |
| connectionString | | private | | string | | Строка подключения к БД. | |
| rnd | | private | | Random | | Получение списка задач для игры. | |
| games | | public | | List<Game> | | Список игр в данный момент, | |
| clients | | public | | List<ClientObject> | | Радиус описываемой окружности, при движении астероида. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| AddConnection | protected internal | | void | | ClientObject  clientObject | | Добавляет пользователя в список clients. |
| RemoveConnection | protected internal | | void | | string id | | Удаляет объект ClientObject из clients и закрывает подключения. |
| Listen | protected internal | | void | | - | | Прослушивает подключения, добавляет пользователей. |
| Disconnect | protected internal | | void | | - | | Отключает все подключения к серверу и заканчивает работу сервера. |
| SignIn | protected internal | | bool | | string msg, string id | | Осуществляет вход пользователя в приложения, согласно информации полученной из БД. |
| SignUp | protected internal | | bool | | string msg, string id | | Осуществляет регистрацию пользователя в игре, добавляет информацию в БД. |
| CallBack | protected internal | | void | | string message, ClientObject client | | Отправляет клиенту сообщение с сервера |
| RemoveActive | protected internal | | void | | string msg | | Изменяет информацию о пользователе в БД, меняет поле IsActive в таблице Users. |
| GetActiveUsers | protected internal | | string | | - | | Возвращает список пользователей, кто находится в приложении, получая информацию из БД. |
| GetStatistic | protected internal | | string | | string nickname | | Возвращает статистику, по решенным задачам, получая информацию из БД |
| PossibleInvite | protected internal | | bool | | String usName | | Проверяет возможно ли пригласить пользователя с именем usName в игру |
| GetTasks | protected internal | | void | | string s, Game game | | Получает список задач и ответов на задачи из БД, для одной игры. |
| SetInGame | public | | void | | string nickname | | Меняет значение поля InGame у пользователя с ником, nickname, устанавливает True. |
| RemoveInGame | public | | void | | string nickname | | Меняет значение поля InGame у пользователя с ником, nickname, устанавливает False. |
| AddStatistic | public | | void | | string nick, string[] idT, string[] answ | | Добавляет информацию после игры, на отвеченные пользователем задачи. |
| GetIdUser | public | | string | | string nickname | | Возращает Id игрока согласно информации в БД, по его имени. |
| AddGame | protected internal | | void | | string nickname1, string nickname2, string idGame | | Добавляет информацию об игре в БД, |
| DeleteGame | protected internal | | void | | String IdGame | | Удаляет из БД информацию о данной игре. |
| GetNumbers | private | | int[] | | string s, int count | | Генерирует номера задач для игры. |

Таблица 5.2

Описание полей, методов и свойств класса СlientObject.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| userName | | public | | string | | Хранит в себе логин и пароль пользователя | |
| acception | | public | | string | | Хранит в себе информацию о принятии игроком приглашения в игру. | |
| answers | | public | | string[] | | Ответы на игру, в которой находится пользователь. | |
| idTasks | | public | | string[] | | Номера задач, из которых состоит текущая игра пользователя. | |
| client | | private | | TcpClient | | Обьект клиента, взаимодействующего с сервером. | |
| srever | | private | | ServerObject | | Обьект сервера, к которому подключем пользователь. | |
| numinGame | | public | | string | | Номер игрока в игре. | |
| **Свойства** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Доступность | | Назначение |
| Id | protected internal | | string | | get set | | Хранит Id пользователя |
| Stream | protected internal | | NetworkStream | | get set | | Хранит поток, по которому передается информация от пользователя. |
| **Методы** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Process | public | | void | | - | | Обрабатывает сообщения от пользователя. |
| GetMessage | private | | string | | - | | Получает сообщения от пользователя. |
| Close | protected internal | | void | | - | | Закрывает подключение с пользователем. |
| InviteInGame | protected internal | | void | | - | | Отправялет другому пользователею приглашение в игру. |
| AcceptInviting | protected internal | | void | | string message | | Отправляет ответ пользователя на приглашение в игру. |
| GetName | protected internal | | string | | - | | Возвращает имя пользователя. |
| ParseAnsw | protected internal | | string[] | | string s | | Обрабатывает ответы пользователя на игру. |

Таблица 5.3

Описание полей, методов и свойств класса Game.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| player1 | | public | | ClientObject | | Хранит объект клиента, который находится в данной игре. | |
| player2 | | public | | ClientObject | | Хранит объект клиента, который находится в данной игре. | |
| server | | public | | ServerObject | | Хранит объект сервера, с которым взаимодействуют клиенты. | |
| tasks | | public | | string[] | | Хранит задания для текущей игры. | |
| answ | | public | | string[] | | Хранит ответы на задания для данной игры. | |
| idTasks | | public | | string[] | | Хранит номера заданий, согласно информации в БД. | |
| idGame | | public | | string | | Хранит номер игры. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SendTasks | public | | void | | - | | Отправляет задания для данной игры пользователям, находящимся в этой игре. |

Таблица 5.4

Описание полей, методов и свойств класса Program.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| server | | private | | ServerObject | | Хранит обьект сервера, по которому происходит обмен данными. | |
| listenThread | | private | | Thread | | Хранит поток, по котором происходит прослушивание подключений. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Main | private | | void | | string[] args | | Запускает работу сервера. |

Таблица 5.5

Описание полей, методов и свойств класса User.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| userName | | public | | string | | Хранит имя пользователя. | |
| host | | private | | string | | Хранит информацию об IP адресе хоста. | |
| port | | private | | string | | Хранит информацию о порте, по которому передается информация. | |
| opponent | | public | | string | | Хранит имя соперника. | |
| iswait | | public | | string | | Отвечает за ожидание пользователем приглашения в игру. | |
| opflag | | public | | string | | Хранит ответ противника на приглашение в игру. | |
| ActUsers | | public | | string | | Хранит строку, в которой хранятся имена активных пользователей. | |
| stat | | public | | string | | Хранит строку, которая содержит данные о статистике. | |
| tasks | | public | | string[] | | Список задач на игру. | |
| answers | | public | | string[] | | Список ответов на задачи игры. | |
| useranswers | | public | | string[] | | Список ответов пользователя на задачи игры. | |
| flag | | public | | bool | | Хранит информацию, прошел ли пользователь регистрацию или вход успешно. | |
| s | | public | | string | | Хранит информацию об успешном принятии сообщения. | |
| client | | private | | TcpClient | | Хранит объект клиента сервера, по которому | |
| stream | | private | | NetworkStream | | Поток, по которому происходит обмен информацией между клиентом и срвером. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SendMessage | public | | void | | string message | | Отправления сообщения на сервер. |
| RecieveMessage | public | | void | | - | | Получение сообщения с сервера. |
| Disconnect | public | | void | | - | | Закрытие подключения к серверу. |
| SignIn | public | | void | | string message | | Вход пользователя в приложение. |
| SignUp | public | | void | | string message | | Регистрация пользователя в приложении. |
| GetActiveusers | public | | void | | - | | Получение списка активных пользователей |
| GetStatistic | public | | void | | - | | Получение статистики игры. |
| InviteUser | public | | void | | string s | | Приглашение другого пользователя в игру. |
| AcceptionInvating | public | | void | | string s | | Принятие приглашения в игру. |
| TransformTasks | public | | void | | string message | | Обрабатывает полученные задания и ответы на игру. |
| ParseAnsw | public | | string[] | | string answ | | Обрабатывает сообщение с ответами противника. |

Таблица 5.6

Описание полей, методов и свойств класса TasksPage.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| But\_Click | public | | void | | object sender, EventArgs e | | Считывает данные при нажатии на кнопки ввода ответов |
| DelLastSymbol | private | | string | | string s | | Удаление последнего символа строки. |
| AnswToStr | private | | string | | - | | Создает строку, состоящую из ответов игрока, |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.7

Описание полей, методов и свойств класса ShowResaltPage.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| yourcount | | private | | int | | Количество правильных ответов игрока. | |
| countoponnent | | private | | int | | Количество правильных ответов противника. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Ok\_Click | public | | void | | object sender, EventArgs e | | Отчищает данные об игре и открывает главное меню. |
| GetArrAnsw | private | | string[] | | string[] answers, string name | | Преобразует строку в массив, который заполняет таблицы на странице. |
| GetWinner | private | | void | | - | | Определяет победителя игры. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Item | public | | ObservableCollection<string> | | get, set | | Коллекция, заполняющая данными одну таблицу. |

Таблица 5.8

Описание полей, методов и свойств класса ShowErPage.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |
| Ok\_click | public | | void | | object sender, EventArgs e | | Открытие страницы главного меню. |

Таблица 5.9

Описание полей, методов и свойств класса PlayGamePage.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| WaitBut\_Click | public | | void | | object sender, EventArgs e | | Ожидает приглашения в игру. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.10

Описание полей, методов и свойств класса MainPage.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| SignIn\_Click | public | void | object sender, EventArgs e | Вход Пользователя. |
| SignUp\_Click | public | void | object sender, EventArgs e | Регистрация пользователя. |
| IsCorrectNick | public | string | string message | Проверяет корректность имени. |
| IsCorrectPassword | public | string | string message | Проверяет корректность пароля, введенного пользовтаелем. |

Таблица 5.11

Описание полей, методов и свойств класса MainMenuMenuItem.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Свойства** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Id | public | | Int | | get, set | | Хранит ID раздела меню. |
| Title | public | | string | | get, set | | Хранит заголовок раздела меню, отображающего страницу. |
| TargetType | public | | Type | | get, set | | Хранит тип страницы для этого элемента. |

Таблица 5.12

Описание полей, методов и свойств класса MainMenuMaster.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Свойства** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | доступ | | Назначение |
| MenuItems | public | | ObservableCollection<MainMenuMenuItem> | | get,set | | Коллекция элементов меню. |

Таблица 5.13

Описание полей, методов и свойств класса MainMenuDetail.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Invite\_click | public | | void | | object sender, EventArgs e | | Приглашение пользователя в игру. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.14

Описание полей, методов и свойств класса MainMenu.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| ListView\_ItemSelected | private | | void | | object sender, EventArgs e | | Обрабатывает выбор элементов в панели меню. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.15

Описание полей, методов и свойств класса GetStatistic.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Show | public | | void | | - | | Выводит данные о статистики на экран. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.16

Описание полей, методов и свойств класса GetActiveUsers.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Show | public | | void | | - | | Выводит данные об активых пользователях на экран. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.17

Описание полей, методов и свойств класса ExitPage.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Yes\_click | public | | void | | object sender, EventArgs e | | Заканчивает работу приложения. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |

Таблица 5.18

Описание полей, методов и свойств класса CurrentUser.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| currentUser | public | User | Хранит текущего пользователя. |
| OponEx | public | bool | Показывает вышел ли противник из игры. |
| GetOponAnsw | public | bool | Показывает получил ли игрок ответы противника |
| IsWaiting | public | bool | Показывает находится ли пользователь в ожидании игры. |

Таблица 5.19

Описание полей, методов и свойств класса ChooseAnswers.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| ChooseComplexyty\_Toggled | private | | void | | object sender, ToggledEventArgs e | | Задает сложность задач. |
| Algebra\_click | public | | void | | object sender, EventArgs e | | Отправляет сообщение с выбранным типом. |
| Geom\_click | public | | void | | object sender, EventArgs e | | Отправляет сообщение с выбранным типом. |
| Comb\_click | public | | void | | object sender, EventArgs e | | Отправляет сообщение с выбранным типом. |
| OnBackButtonPressed | protected | | bool | | - | | Блокирует нажатие кнопки «назад». |
| ShowTasks | public | | void | | - | | Ожидает получение задач. |

Таблица 5.20

Описание полей, методов и свойств класса App.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| OnStart | protected | | void | | - | | Срабатывает при запуске программы. |
| OnSleep | protected | | void | | - | | Срабатывает при переключении приложения в режим «сна». |
| OnResume | protected | | void | | - | | Срабатывает при возобновлении работы приложения. |

Таблица 5.21

Описание полей, методов и свойств класса AcceptPage.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Yes\_Click | public | void | object sender, EventArgs e | Принимает приглашение в игру и запускает игру. |
| No\_Click | public | void | object sender, EventArgs e | Отклоняет приглашение в игру. |
| WaitMethod | private | bool | - | Ожидает получения задач. |
| OnBackButtonPressed | protected | bool | - | Блокирует нажатие кнопки «назад». |

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 | Аннулированных |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |