Минский колледж предпринимательства

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

**ОБУЧАЮЩАЯ ИГРА «СЛОЖЕНИЕ И ВЫЧИТАНИЕ»**

**Пояснительная записка**

**МКП.3210901-01 81 01**



|  |  |  |
| --- | --- | --- |
|  |  | Учащийся |
|  |  |  |
|  |  | А.В.Баженов |
|  |  | \_\_\_\_\_\_\_\_\_\_2023 |
|  |  |  |
|  |  | Руководитель |
|  |  |  |
|  |  | И.В.Жилинский |
|  |  | \_\_\_\_\_\_\_\_\_\_2023 |
|  | 2023 |  |

АННОТАЦИЯ

Курсовой проект на тему: Обучающая игра «Сложение и вычитание», состоит из программного средства, пояснительной записки и графической части.

Программное средство предназначено для детей и взрослых.

Программное средство разработано в среде программирования С++ Builder с использованием языков программирования С++.

Пояснительная записка выполнена на 65 листах, содержит 4 раздела и одно приложение.

Графическая часть выполнена на двух листах, включает диаграмму BPMN и диаграмму вариантов использования.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_i7yh4o61tgur)

[1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_977466tha79s)

[2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_r30oaio49ec)

[2.1. Постановка задачи 6](#_xz1aacz5w40u)

[2.2. Описание организации данных 6](#_9sudwqapmurz)

[2.3. Описание программных средств 7](#_3r1axxjrj5m9)

[2.4. Проектирование интерфейса приложения 8](#_cytz2rbtof0y)

[2.5. Описание логической структуры программы 13](#_doopdcckium9)

[3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ 16](#_9q79q4a6l8bf)

[4. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА 20](#_blnxfaii7uzt)

[ЗАКЛЮЧЕНИЕ 24](#_wvgnxqbfkzr5)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_gm4hg7d33y20)

[ПРИЛОЖЕНИЕ 1 (ОБЯЗАТЕЛЬНОЕ) – ЛИСТИНГ ПРОГРАММЫ 26](#_jyqv47ahpkp3)

# ВВЕДЕНИЕ

На сегодняшний день обучающие игры стали важной частью жизни современных людей.

Обучающие игры  игры, в процессе которых происходит развитие или усовершенствование различных навыков. Понятие развивающих игр связано, в основном, с детским периодом жизни человека. Дети, играющие в обучающие игры, тренируют собственные мышление, изобретательность, воображение, креативность.

В данном курсовом проекте необходимо разработать обучающую игру «Сложение и вычитание чисел». Актуальность обучающих игр очень важна в современных условиях деградации нового поколения.

Целью выполнения курсового проекта является расширение, углубление знаний в области разработки на языке C++, формирование навыков научно-исследовательской деятельности.

Задачей выполнения курсового проекта является разработка игры «Сложение и вычитание чисел»

Данная пояснительная записка содержит четыре раздела и заключение.

Раздел «Назначение и область применения» – назначение данного проекта и возможности его использования.

Раздел «Технические характеристики» – постановка задачи с указанием всех функций проекта, определение круга задач, которые должны быть осуществлены в данном курсовом проекте, описание входных и выходных данных проекта для его использования, описание выбранного языка программирования и среды разработки для реализации проекта, их достоинств и недостатков. Обоснование выбранного языка и среды, описание внешнего пользовательского интерфейса − системы меню, диалоговых окон и элементов управления, описание назначения всех компонент приложения.

Раздел «Описание логической структуры программы» – описание объектов со всеми функциями, входящие в приложение – назначение и взаимодействие.

Раздел «Тестирование и анализ полученных результатов» – проверка работы программы на различных тестах – в нормальных и экстремальных условиях. Проверка обработки исключений. Проверка реакции программы на действия пользователя.

Раздел «Руководство по использованию программного средства» –сведения о назначении программного средства, области применения, классе решаемых задач, ограничениях, накладываемых на область применения.

Заключение – приводятся краткая формулировка проблемы и пути её решения, использованные методы и средства, полнота раскрытия проблемы.

# 1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

В данном курсовом проекте необходимо разработать обучающую игру «Сложение и вычитание чисел». Данная игра помогает пользователю с развитием нестандартного мышления, логики и банальных математических вычислений.

Обучающая игра «Сложение и вычитание чисел» может быть использована для развития логики и мышления, а так же, как игра для отдыха или соревновательная игра.

Аналогом данного проекта является игра «Игра сложение и вычитание чисел: Классическая».

Достоинствами данной игры являются разделы: понятный дизайн, простота использования. Недостатками игры являются: всплывающие рекламные блоки, мешающие просмотру информации.

Отличиями разрабатываемой программы от аналогов являются: отсутствие рекламы, возможность входа как зарегистрированный пользователь с последующим сохранением лучших результатов игры в базу данных, большее количество настроек.

Данное программное средство рассчитано на все категории людей.

Требования к аппаратным и операционным ресурсам для разработки:

1) Операционная система Windows 10.

2) Процессор Intel Core i5-7600.

3) Оперативная память 8GB.

4) SSD 256GB.

# 2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## 2.1. Постановка задачи

Разработать игру «Сложение и вычитание чисел», предназначенную для развития логики и математического мышления. Суть игры заключается в последовательном ответе на примеры за 1 минуту. Задача игрока ответить на как можно большее количество примеров.

В программе реализовать три режима игры: «лёгкий», «средний», «сложный».

Режим игры «лёгкий» представляет собой генерацию примеров, состоящих из однозначных чисел.

Режим игры «средний» представляет собой генерацию примеров, состоящих из двухзначных чисел.

Режим игры «сложный» представляет собой генерацию примеров, состоящих из трехзначных чисел.

Обеспечить возможность переключения режима игры, выбора цветовой темы игрового поля, включения\выключения музыки.

Реализовать способ входа в приложение: как пользователь с возможностью регистрации.

При регистрации предусмотреть автоматическое сохранение лучших результатов для режимов игры «лёгкий», «средний», «сложный», по окончанию игры в базу данных пользователей.

Предусмотреть возможность просмотра лучших результатов пользователей.

При вводе данных во время регистрации или входа обеспечить контроль вводимых данных, обработку исключений.

## 2.2. Описание организации данных

Входные данные – данные Логин, Пароль, генерация примеров.

Выходные данные – информация о времени игры.

Данные для проекта необходимо хранить в БД. Структура данных представлена в таблице 2.1.

Таблица 2.1 – Структура данных

| Наименование столбца | Тип данных | Описание |
| --- | --- | --- |
| Логин | Текст | Хранить логины пользователей |

*Продолжение таблицы 2.1*

| Наименование столбца | Тип данных | Описание |
| --- | --- | --- |
| Пароль | Текст | Хранить пароли пользователей |
| Легкий | Числовой | Хранить лучший результат пользователя в режиме игры «лёгкий» |
| Средний | Числовой | Хранить лучший результат пользователя в режиме игры «средний» |
| Сложный | Числовой | Хранить лучший результат пользователя в режиме игры «сложный» |

## 2.3. Описание программных средств

В данном проекте используется язык программирования C++ плюсами которого являются:

1. Большая безопасность.
2. Возможность писать обобщенный код с помощью шаблонов.
3. Возможность использовать объектно-ориентированный подход.
4. Масштабируемость. На языке C++ разрабатывают программы для самых различных платформ и систем.
5. Упрощение кода за счет перегрузки функций и операторов
6. Более простая обработка ошибок за счет исключений
7. Возможность работы на низком уровне с памятью, адресами, портами. что, при неосторожном использовании, может легко превратиться в недостаток.
8. Возможность создания обобщенных алгоритмов для разных типов данных, их специализация, и вычисления на этапе компиляции, используя шаблоны.

К минусам C++ можно отнести:

1. Долгая компиляция
2. Плохая поддержка модульности.
3. Недостаток информации о типах данных во время компиляции (CTTI).

C++Builder стал одним из самых популярных на сегодняшний день инструментов для создания как настольных, так и корпоративных информационных систем к плюсам которого можно отнести:

1. Быстрота разработки приложения.
2. Высокая производительность разработанного приложения.
3. Низкие требования разработанного приложения к ресурсам компьютера.
4. Простой и интуитивно понятный интерфейс.
5. Визуальная разработка с использованием программных каркасов C++Builder VCL.

Минусами C++Builder являются:

1. Долгая компиляция.
2. Непонятно показывает, где находится ошибка.
3. Имеет множество внутренних ошибок.

## 2.4. Проектирование интерфейса приложения

Данное приложение будет состоять из 4 форм: «Форма навигации», «Главная форма», «Форма настроек», «Игровая форма».

Макет формы навигации представлен на рисунке 2.1. На данной форме пользователь может зарегистрироваться и войти как пользователь или как гость.

Компоненты управления формы навигации представлены в таблице 2.2.

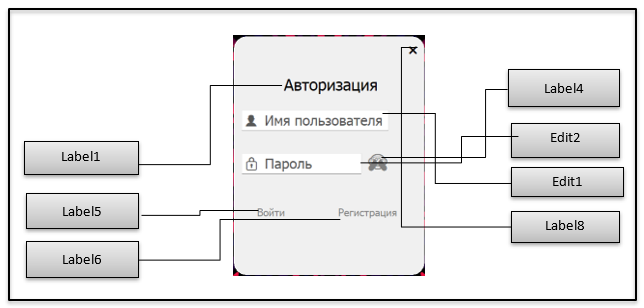


Рис. 2.1. Форма навигации

Таблица 2.2 – Элементы управления формы навигации

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Edit | Name = Edit1 | Поле ввода логина пользователя |
| Edit | Name = Edit2 | Поле ввода пароля пользователя |
| Label | Name = Label1 | Элемент оформления |
| Label | Name = Label4 | Значок для просмотра пароля |
| Label | Name = Label5 | Вход под указанным логином и паролем |

*Продолжение таблицы 2.2*

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Label | Name = Label6 | Регистрация под указанным логином и паролем |
| Label | Name = Label8 | Выход из приложения |

Макет главной формы представлен на рисунке 2.2. На главной форме пользователь может перейти в игровую форму игры «Сложение и вычитание чисел» и выйти из игры. Компоненты управления главной формы представлены в таблице 2.3.

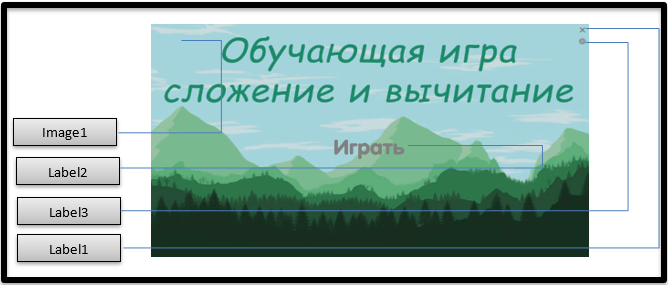


Рис. 2.2. Главная форма

Таблица 2.3 – Элементы управления главной формы

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Label | Name = Label1 | Выход из приложения |
| Label | Name = Label2 | Переход в игровую форму |
| Label | Name = Label3 | Переход в форму настроек |
| Image | Name = Image1 | Отображение фона |

Макет формы настроек представлен на рисунке 2.3. На данной форме пользователь может изменить настройки игрового поля. Также на данной форме можно изменить настройки интерфейса, просмотра рейтинга и музыки. Компоненты управления формы настроек представлены в таблице 2.4.

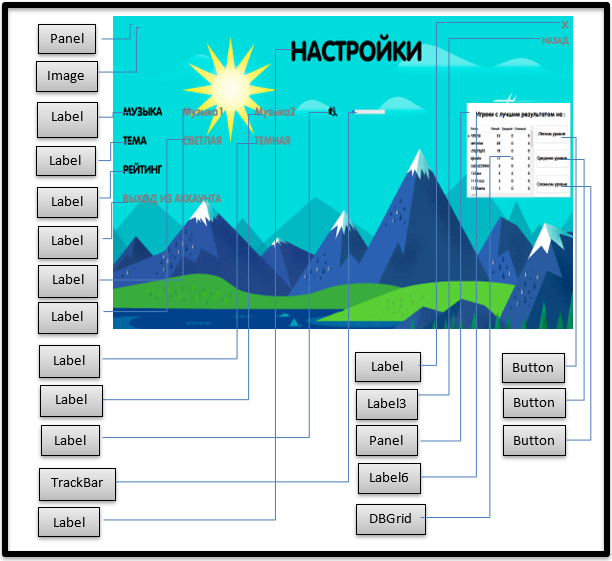


Рис. 2.3. Форма настроек

Таблица 2.4. – Элементы управления формы настроек

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Label | Name = Label1 | Элемент оформления |
| Label | Name = Label2 | Выход из игры |

*Продолжение таблицы 2.4*

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Label | Name = Label3 | Позволяет пользователю выбрать музыку |
| Label | Name = Label4 | Позволяет пользователю включить или выключить музыку1 |
| Label | Name = Label5 | Позволяет пользователю включить или выключить музыку2 |
| Label | Name = Label6 | Позволяет пользователю включить светлую тему |
| Label | Name = Label7 | Позволяет пользователю включить темную тему |
| Label | Name = Label8 | Значок для отображения включенного/выключенного звука |
| Label | Name = Label9 | Позволяет пользователю выбрать тему |
| Label | Name = Label10 | Позволяет пользователю выйти из аккаунта |
| Label | Name = Label11 | Позволяет пользователю посмотреть рейтинг пользователей |
| Label | Name = Label13 | Позволяет пользователю вернуться в предыдущую форму |
| Label | Name = Label16 | Элемент оформления |
| Panel | Name = Panel1 | Для расположения элементов настроек |
| Panel | Name = Panel2 | Для расположения элементов рейтинга |
| TrackBar | Name = TrackBar1 | Позволяет пользователю изменять громкость музыки |
| Image | Name = Image1 | Отображение фона |
| DBGrid | Name = DBGrid1 | Позволяет пользователю посмотреть рейтинг игроков |
| Button | Name = Button1 | Кнопка для сортировки по легкому уровню |
| Button | Name = Button2 | Кнопка для сортировки по среднему уровню |
| Button | Name = Button3 | Кнопка для сортировки по сложному уровню |

Макет игровой формы представлен на рисунке 2.4. На игровой форме пользователь может выбрать уровень и начать проходить игру «Сложение и вычитание чисел. После прохождения уровня будут показываться количество заданных примеров, количество примеров на которые пользователь ответил правильно и неправильно. Предусмотрен выход из игры и переход в настройки.

Компоненты управления игровой формы представлены в таблице 2.5.

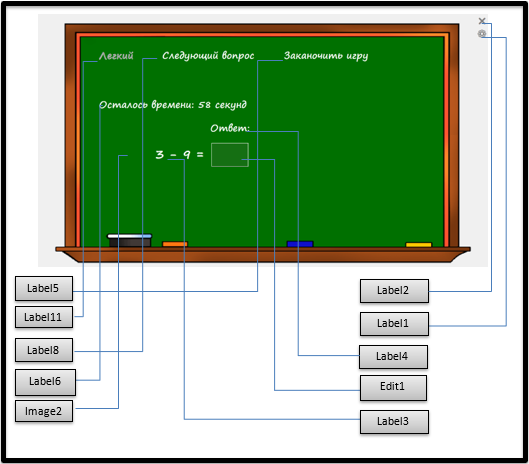


Рис. 2.4. Игровая форма

Таблица 2.5. – Элементы управления игровой формы

| Элементы управления формы навигации | Свойства | Назначение |
| --- | --- | --- |
| Label | Name = Label1 | Переход в форму настроек |
| Label | Name = Label2 | Выход из игры |
| Label | Name = Label3 | Отображение примера |
| Label | Name = Label4 | Элемент оформления |

*Продолжение таблицы 2.5*

| Label | Name = Label5 | Возможность пользователю выйти из запущенной игры |
| --- | --- | --- |
| Label | Name = Label6 | Отображение времени до конца игры |
| Label | Name = Label8 | Показать следующий вопрос |
| Label | Name = Label11 | Отображение выбранного уровня |
| Edit | Name = Edit1 | Ответ пользователя |

## 2.5. Описание логической структуры программы

На форме навигации размещены следующие функции:

* void \_\_fastcall Label4Click(TObject \*Sender) – показывает и скрывает пароль;
* void \_\_fastcall Label5Click(TObject \*Sender) – проверяет базу данных и переходит на следующую форму;
* void \_\_fastcall Label6Click(TObject \*Sender) – проверяет базу данных и переходит на следующую форму;
* void \_\_fastcall Edit1Change(TObject \*Sender) – отслеживает данные введенные в Edit1;
* void \_\_fastcall Edit2Change(TObject \*Sender) – отслеживает данные введенные в Edit2;
* void \_\_fastcall Timer1Timer(TObject \*Sender) – нужен для отображения сообщения на заданное количество времени об правильности ввода пароля и логина ;
* void \_\_fastcall FormCreate(TObject \*Sender) – для закругления краев формы;
* void \_\_fastcall Label8Click(TObject \*Sender) – выход из приложения;
* void \_\_fastcall LabelMouseEnter(TObject \*Sender) – для изменения цвета при наведении курсора;
* void \_\_fastcall LabelMouseLeave(TObject \*Sender) – для изменения цвета при отведении курсора;

На главной форме размещены следующие функции:

* void \_\_fastcall Label3Click(TObject \*Sender) – переход на форму настроек;
* void \_\_fastcall LabelMouseEnter(TObject \*Sender) – для изменения цвета при наведении курсора;
* void \_\_fastcall LabelMouseLeave(TObject \*Sender) – для изменения цвета при отведении курсора;
* void \_\_fastcall Label1Click(TObject \*Sender) – выход из игры;
* void \_\_fastcall Label2Click(TObject \*Sender) – переход в игровую форму;

На форме настроек размещены следующие функции:

* void \_\_fastcall Label12Click(TObject \*Sender) – выход из запущенной игры в главное меню;
* void \_\_fastcall LabelMouseEnter(TObject \*Sender); - для изменения цвета при наведении курсора:
* void \_\_fastcall LabelMouseLeave(TObject \*Sender); - для изменения цвета при отведении курсора:
* void \_\_fastcall Label2Click(TObject \*Sender) – закрытие приложения:
* void \_\_fastcall Label3Click(TObject \*Sender) – открытие доступной музыки для выбора;
* void \_\_fastcall Label4Click(TObject \*Sender) – включение Музыка1;
* void \_\_fastcall Label5Click(TObject \*Sender) – включение Музыка2;
* void \_\_fastcall TrackBar1Change(TObject \*Sender) – изменение громкости музыки;
* void \_\_fastcall Label9Click(TObject \*Sender) – открытие доступных тем;
* void \_\_fastcall Label10Click(TObject \*Sender) – выход из аккаунта;
* void \_\_fastcall Label11Click(TObject \*Sender) – просмотр рейтинга;
* void \_\_fastcall Label13Click(TObject \*Sender) – выйти на предыдущую форму;
* void \_\_fastcall Button1Click(TObject \*Sender) – сортировка рейтинга по легкому уровню;
* void \_\_fastcall Button2Click(TObject \*Sender) – сортировка рейтинга по среденму уровню ;
* void \_\_fastcall Button3Click(TObject \*Sender) – сортировка рейтинга по сложному уровню;
* void \_\_fastcall Label7Click(TObject \*Sender) – включение темной темы;
* void \_\_fastcall Label6Click(TObject \*Sender) – включение светлой темы;
* void \_\_fastcall DBGrid1DrawColumnCell(TObject \*Sender, const TRect &Rect, int DataCol, TColumn \*Column, TGridDrawState State) – раскрашивает выбранную ячейку в DBGrid1;

На форме настроек размещены следующие функции:

* void \_\_fastcall Label1Click(TObject \*Sender) – переход в настройки;
* void \_\_fastcall Edit1KeyDown(TObject \*Sender, WORD &Key, TShiftState Shift) – для нажатия на ENTER для переход на следующий вопрос;
* void \_\_fastcall Timer1Timer(TObject \*Sender) – для счетом оставшегося времени;
* void \_\_fastcall Label2Click(TObject \*Sender) – выйти из приложения;
* void \_\_fastcall LabelMouseEnter(TObject \*Sender) – для изменения цвета при наведении курсора;
* void \_\_fastcall LabelMouseLeave(TObject \*Sender) – для изменения цвета при отведении курсора;
* void \_\_fastcall Timer2Timer(TObject \*Sender) – для отслеживания окончания игры;
* void \_\_fastcall Label5Click(TObject \*Sender) – завершение игры при её запуске;
* void \_\_fastcall Label7Click(TObject \*Sender) – начать игру;
* void \_\_fastcall Label8Click(TObject \*Sender) – переход к следующему вопросу;
* void \_\_fastcall Label11Click(TObject \*Sender) – отображение всех доступных уровней;
* void \_\_fastcall Label12Click(TObject \*Sender) – выбор легкого уровня;
* void \_\_fastcall Label13Click(TObject \*Sender) – выбор среднего уровня;
* void \_\_fastcall Label14Click(TObject \*Sender) – выбор сложного уровня;

# 3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Метод «черного ящика». Основан на принципе «вход-выход». Программе подаются некоторые данные на вход и проверяются результаты, в надежде найти несоответствия. При этом как именно работает программа, считается несущественным.

Этапы процесса тестирования:

1. Проверка приложения в нормальных условиях. Представляет собой тестирование с исходными данными, требуемыми для нормального функционирования программы.
2. Проверка в экстремальных условиях. Предаставляет собой тестирование программы с граничными и неверными исходными данными. Программа должна работать с граничными исходными данными и отвергать неверные исходные данные. В проекте использовался метод тестирования «черного ящика».

Программа была протестирована на следующих тестах:

1. Тест №1. Проверка на запуск игры.
2. Тест №2. Проверка на корректное определение числа, которое ввел пользователь.
3. Тест №3. Проверка на работоспособность алгоритма генерации случайного вопроса.
4. Тест №4. Проверка на появление сообщений о количестве отвеченных вопросов.
5. Тест №5. Проверка на добавление результатов игры в базу данных.
6. Тест №6. Проверка на корректное изменение уровня игры.

Тест №1. При входе в программу должно создаваться изображение игровой зоны, представлено на рисунке 3.1.

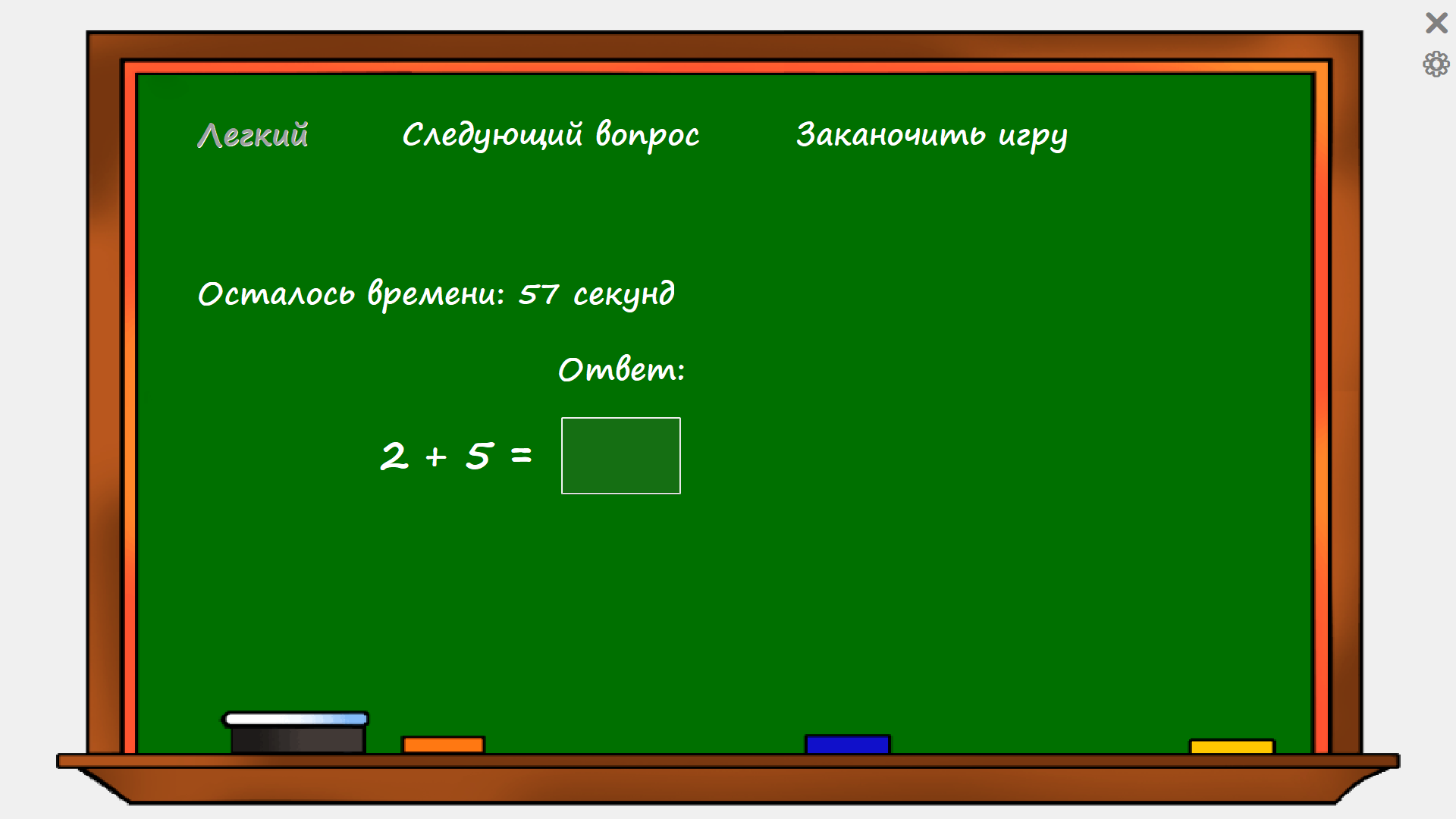


Рис. 3.1. Запуск игры

Тест №2. Ответ, введенный пользователем, должен сравниваться с ответом примера, представлено на рисунке 3.2.

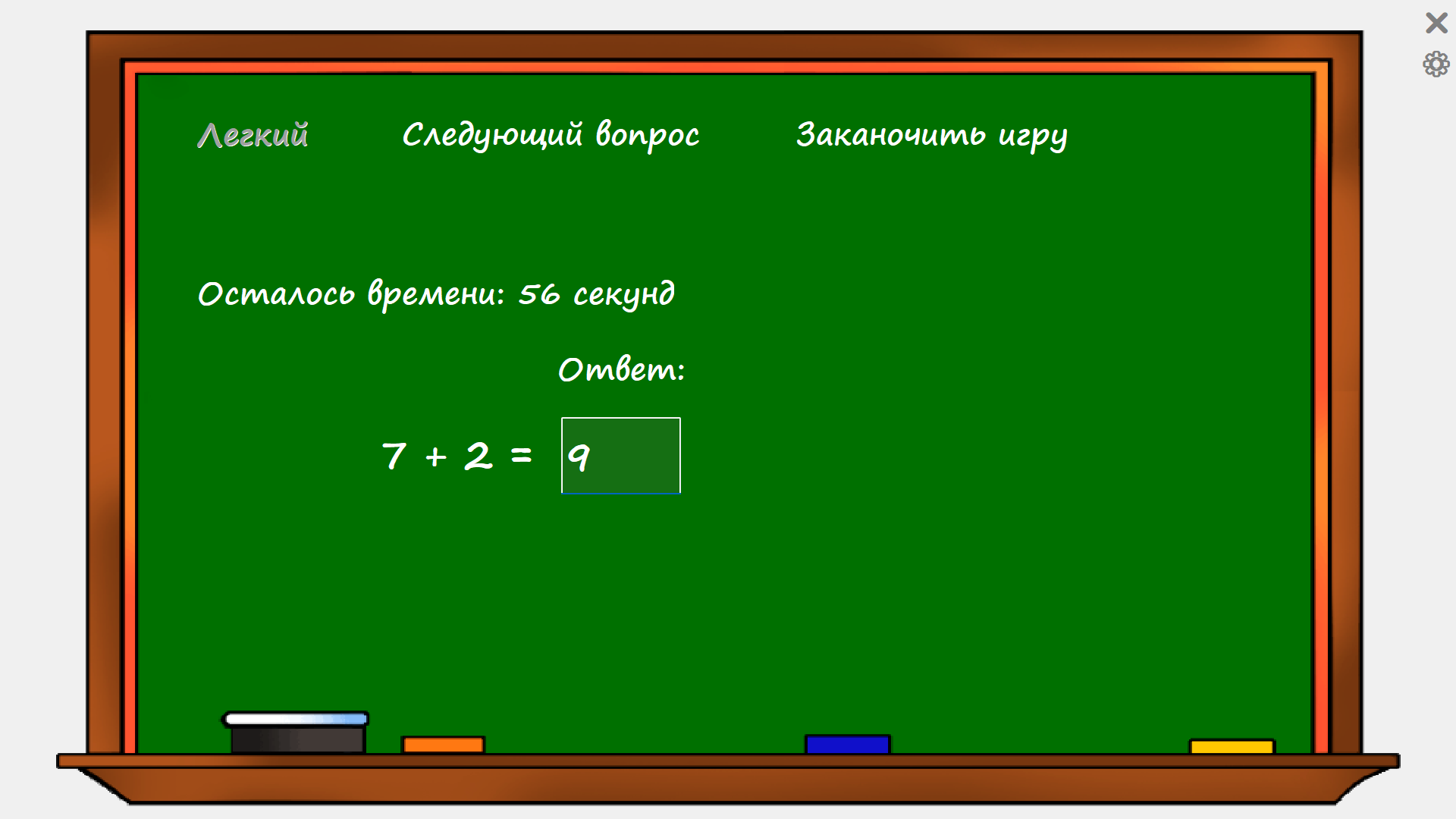


Рис. 3.2. Ввод ответа

Тест №3. При переходе на следующий вопрос генерируется случайный новый вопрос, представлено на рисунке 3.3.

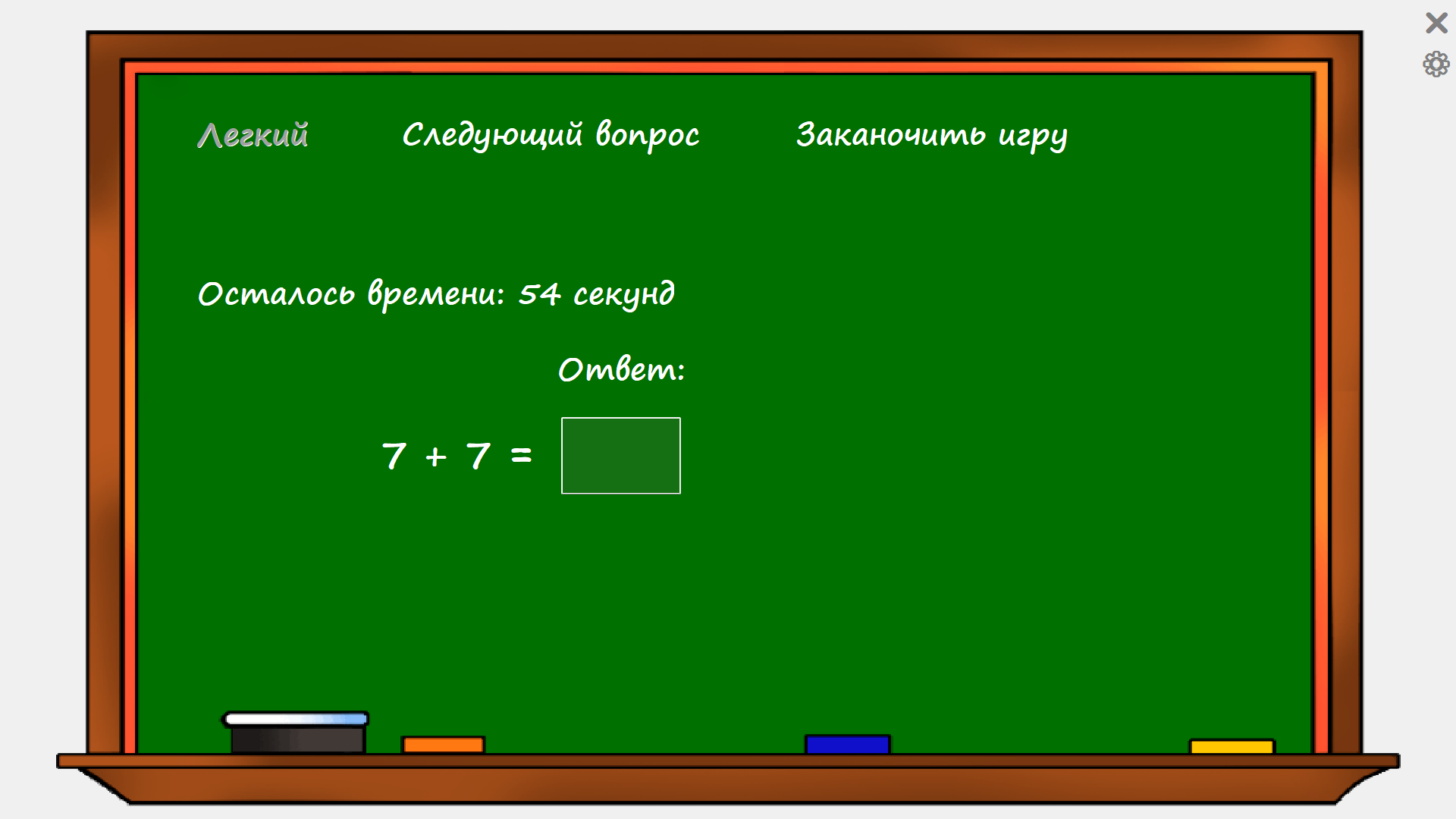


Рис. 3.3. Работоспособность алгоритма генерации случайного вопроса

Тест №4. При окончании уровня можно увидеть количество правильных, неправильных и заданных вопросов, представлено на рисунке 3.4.

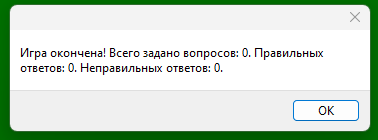


Рис. 3.4. Отображение сообщения с информацией об игре

Тест №5. После прохождения уровня происходит обновление результатов рейтинга в базе данных пользователей при условии, что новый результат лучше старого результата у данного пользователя или если вообще нет результата, представлено на рисунке 3.5, 3.6.

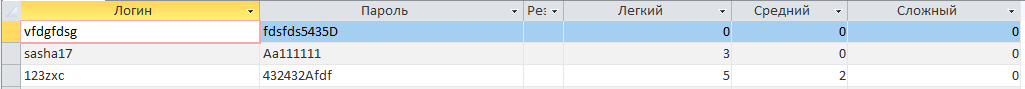


Рис. 3.5. Корректное добавление и отображение результатов игр

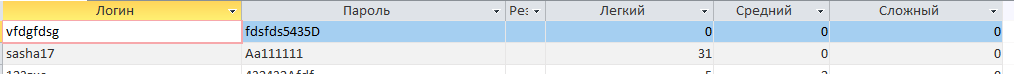


Рис. 3.6. Корректное обновление и отображение результатов игр

Тест №6. При выборе другого уровня появляются другие по сложности вопросы, представлено на рисунке 3.7. см рис. 3.2.

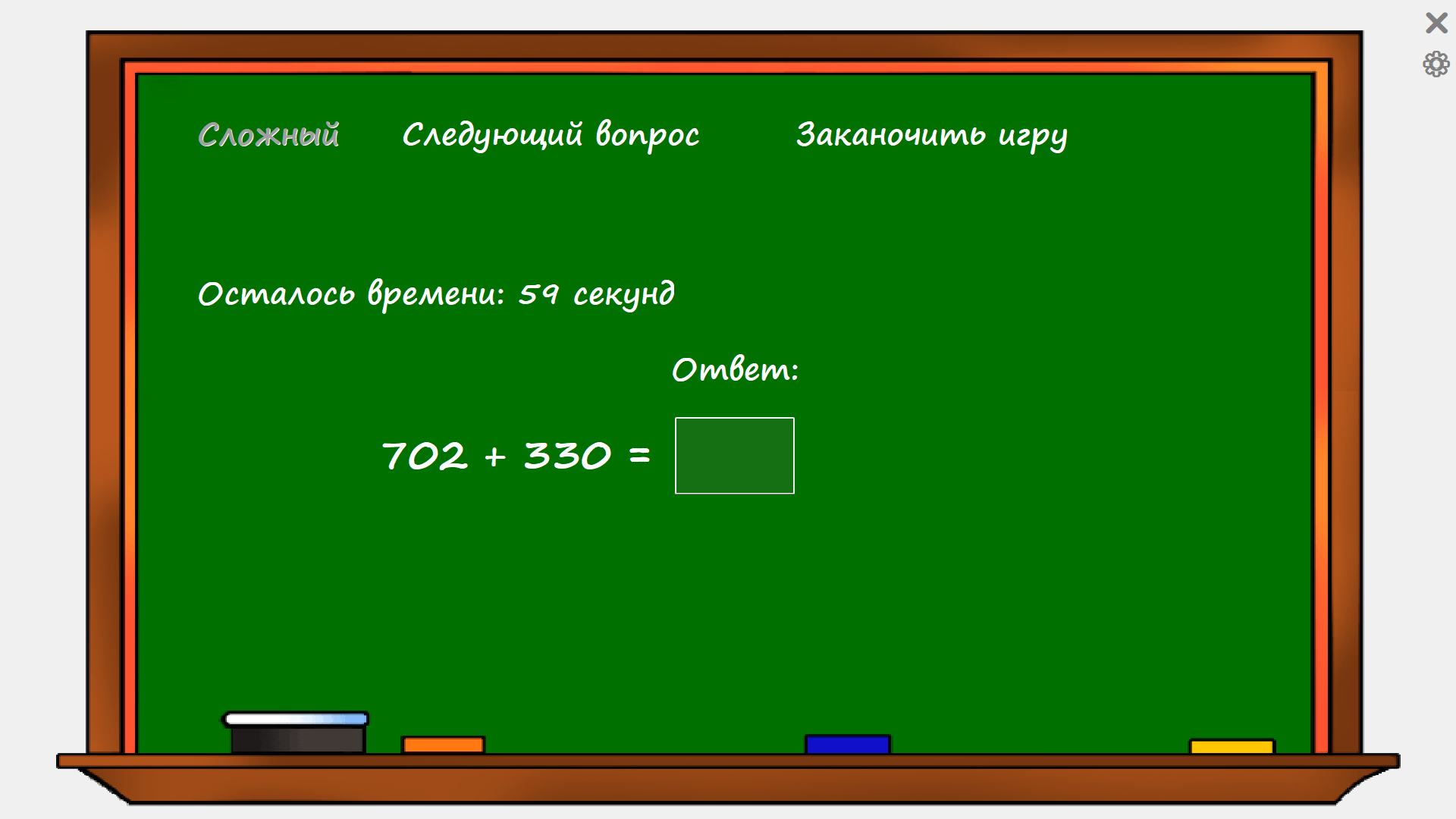


Рис. 3.7. Корректное изменение сложности примера

# 4. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА

Чтобы поиграть в игру «Сложение и вычитание чисел», необходимо пройти регистрацию или зайти как уже зарегистрированной пользователь, что представлено на рисунке 4.1.

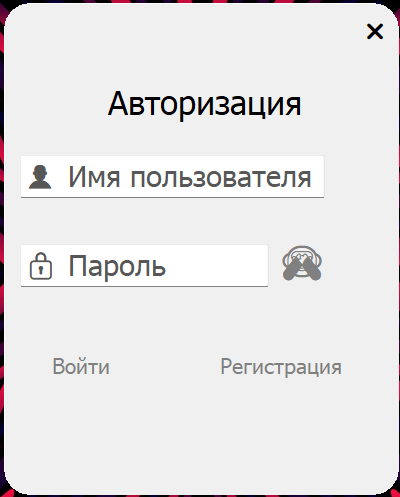


Рис. 4.1. Вход в приложение

Чтобы изменить настройки игры необходимо нажать на «Настройки» и выбрать необходимые параметры, что представлено на рисунке 4.2.

Чтобы просмотреть таблицу рекордов необходимо нажать на кнопку «Рейтинг», что представлено на рисунке 4.3.

Чтобы вернуться на главную форму необходимо нажать на кнопку «Назад», что представлено на рисунке 4.4.

Чтобы начать игру необходимо сначала выбрать уровень, а потом нажать на кнопку играть, что представлено на рисунке 4.5.

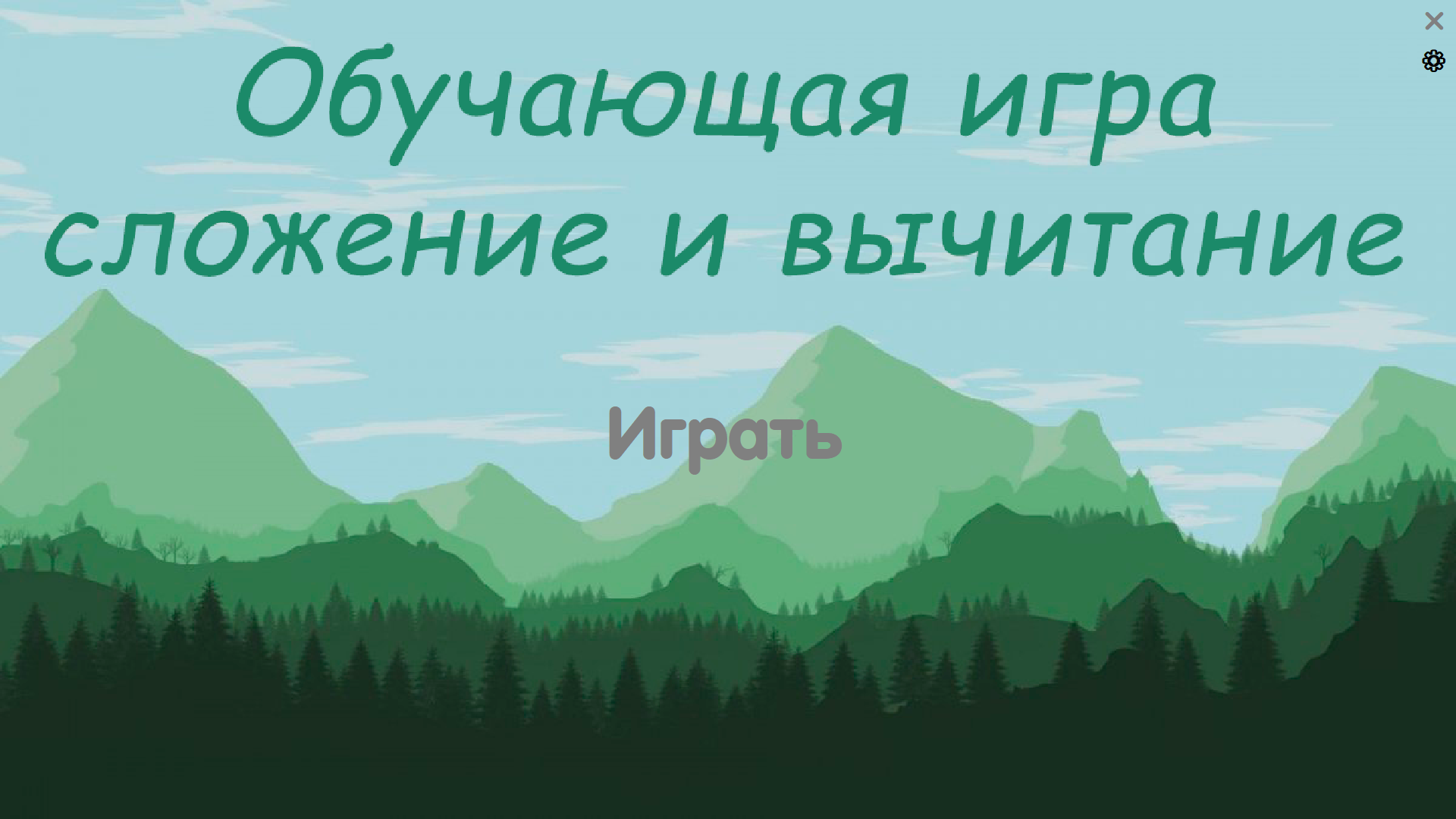


Рис. 4.2. Выбор кнопки «Настройки»

Чтобы вернуться на форму навигации, необходимо нажать на кнопку «Выход из аккаунта», что показано на рисунке 4.6.

Чтобы выйти из приложения необходимо нажать на крестик на любой форме, что представлено на рисунке 4.7.

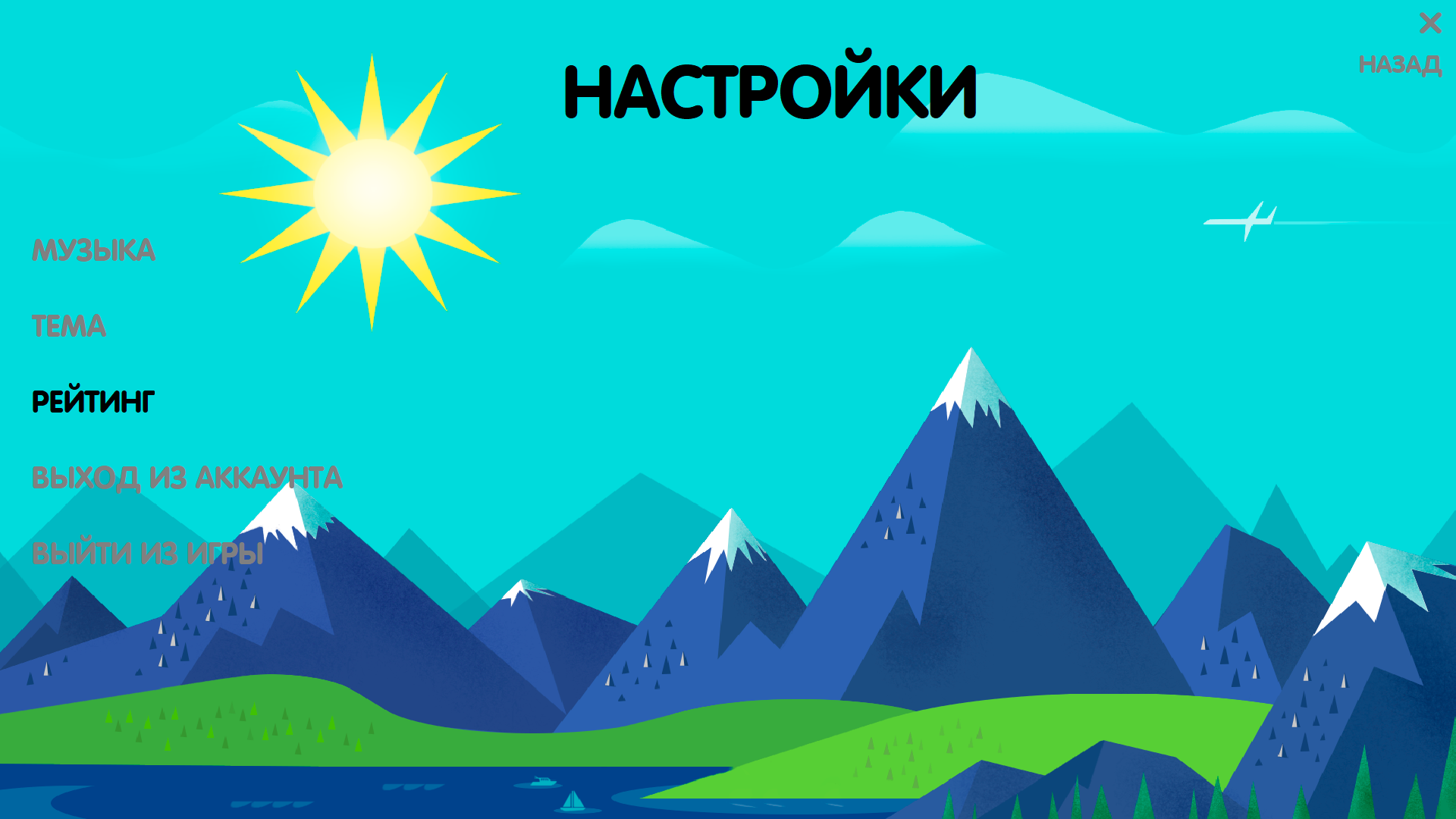


Рис. 4.3. Выбор кнопки «Рейтинг»

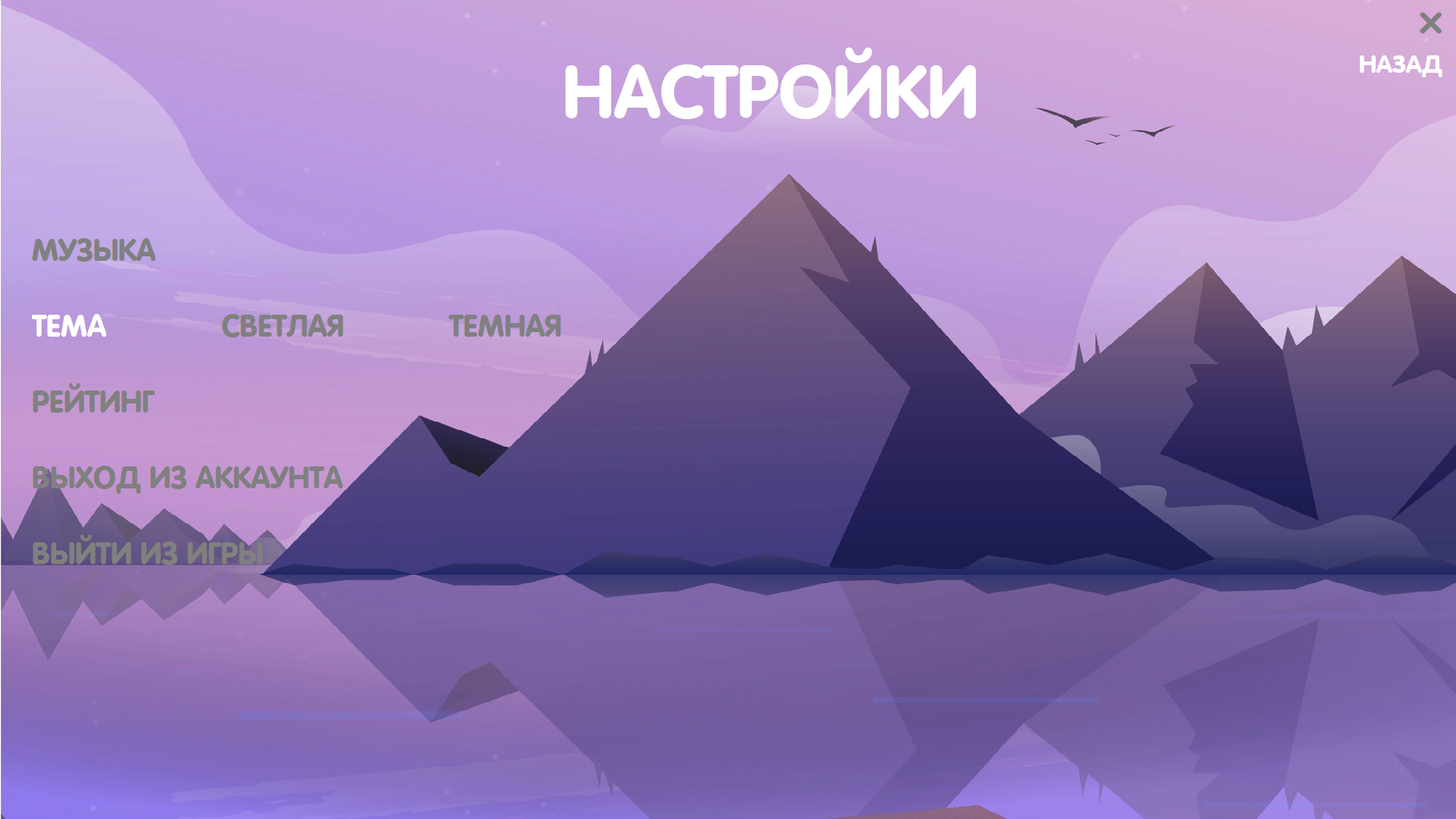


Рис. 4.4. Выбор кнопки «Назад»



Рис. 4.5. Выбор уровня и запуски игры

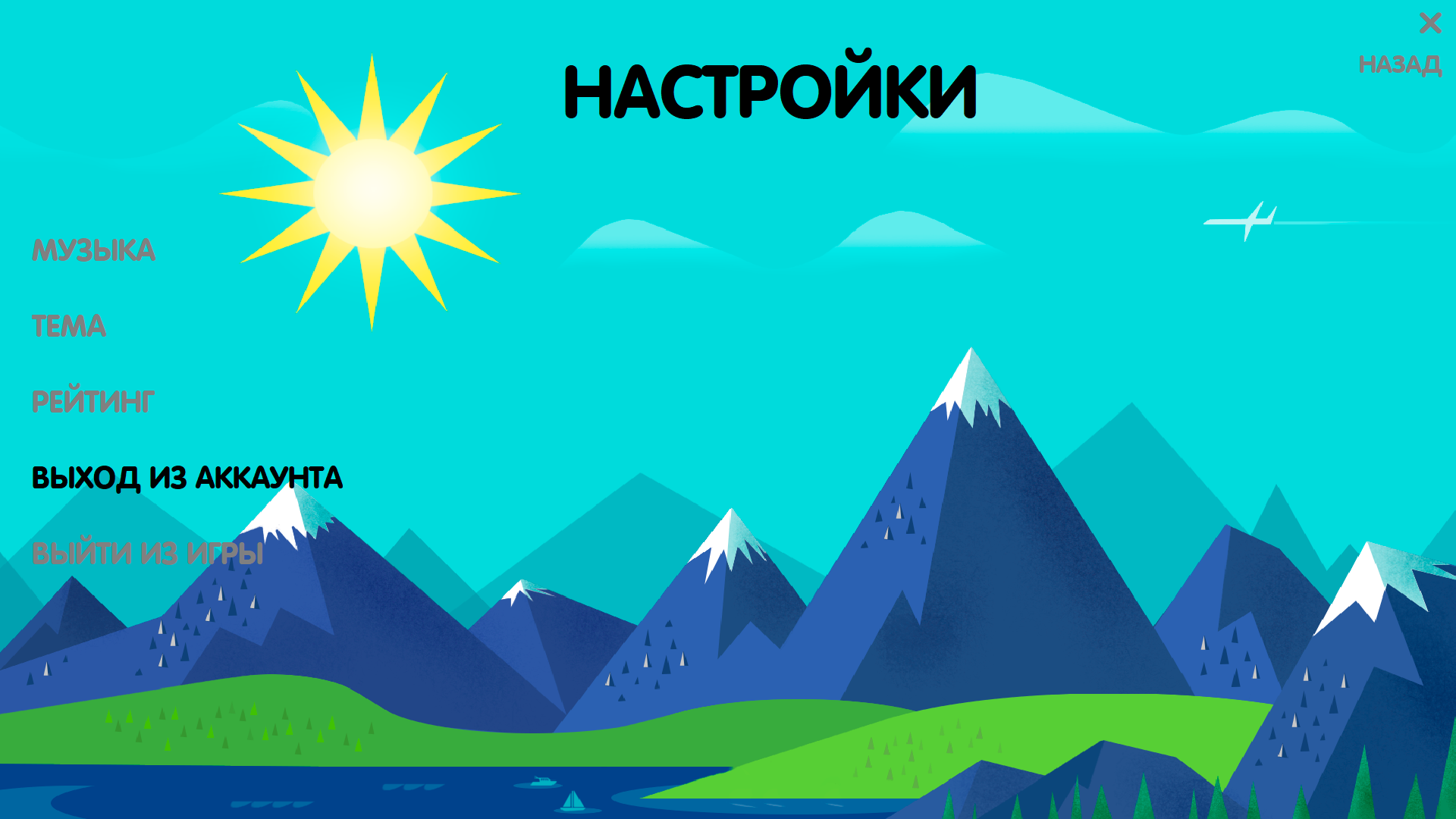


Рис. 4.6. Выбор кнопки «Выход из аккаунта»



Рис. 4.7. Выбор кнопки «Выход из приложения»

# ЗАКЛЮЧЕНИЕ

В ходе курсового проекта была разработана игра «Сложение и вычитание чисел». В приложении реализованы три режима игры: «лёгкий», «средний», «сложный». В режимах «лёгкий», «средний», «сложный» реализовано сохранение лучших результатов игры. Реализована возможность смены цветовой темы программы. Реализовано проигрывание музыки при включённом соответствующем параметре.

В качестве выбранного языка программирования используется C++.

В качестве выбранной среды разработки используется C++ Builder.

Программное средство было протестировано. Тестирование прошло успешно. Дальнейшую модификацию программы вижу в увеличении количества цветовых тем программы, возможности смены языка программы и добавлении новых режимов игры.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. CyberForum.ru [Электронный ресурс] / С++ Builder. Режим доступа: <https://www.cyberforum.ru/cpp-builder/thread832198.html> – Датa доступа 21.05.2023.
2. Ci-Builder.ru [Электронный ресурс] / MediaPlayer. Режим доступа: http://www.ci-builder.ru/04/Index03.html – 21.05.2023.
3. HABR [Электронный ресурс] / SQL запросы быстро. Часть 1. Режим доступа: <https://habr.com/ru/post/480838/> – Дата доступа: 21.03.2023.
4. Booksee.org [Электронный ресурс] / Язык программирования С++. Режим доступа: <https://booksee.org/book/1333580> – Дата доступа: 23.03.2023
5. Мedium.com [Электронный ресурс] / С++ Builder. Режим доступа: [Building Your First App Using C++ Builder (Part 1) | by Idera Dev Tools | Geek Culture | Medium](https://medium.com/geekculture/building-your-first-app-using-c-builder-part-1-e400789f03b3) – Датa доступа 22.05.2023.
6. Helloworld.ru [Электронный ресурс] / С++ Builder.Режим доступа: [Начинаем работать в Borland C++ Builder (helloworld.ru)](https://www.helloworld.ru/texts/comp/lang/builder/buid2/cb2.htm) – Дата доступа: 20.04.2023.

# ПРИЛОЖЕНИЕ 1 (ОБЯЗАТЕЛЬНОЕ) – ЛИСТИНГ ПРОГРАММЫ

Авторизация

//---------------------------------------------------------------------------

#include <windows.h>//- подключение библиотеки для работы с операционной системой Windows.

#include <vcl.h>//- подключение библиотеки Visual Component Library (VCL) для работы с графическим интерфейсом.

#pragma hdrstop//- указание компилятору остановить генерацию кода после заголовочного файла.

#include <System.RegularExpressions.hpp>//- подключение библиотеки для работы с регулярными выражениями.

#include "Unit1.h"//

#include "Unit2.h"// - подключение заголовочных файлов форм, которые будут использоваться в коде.

#include "Unit3.h"//

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;//- - создание глобальной переменной, которая будет хранить указатель на главную форму приложения.

bool passwordVisible = false;// - создание переменной-флага для отображения или скрытия пароля.

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)//- конструктор класса формы, который вызывается при создании экземпляра формы. В нем происходит инициализация таймеров и панели.

: TForm(Owner)

{

Timer2->Interval = 1000;//- установка интервала для таймера 2 в 1 секунду.

Timer2->Enabled = false;//- отключение таймера 2.

Timer3->Interval = 1000;//- установка интервала для таймера 3 в 1 секунду.

Timer3->Enabled = false;//- отключение таймера 3.

Panel1->Top = 190;//- установка позиции верхнего края панели на 190 пикселей.

Panel1->Left = 800;//- установка позиции левого края панели на 800 пикселей.

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label5Click(TObject \*Sender)//- обработчик события нажатия на кнопку "Войти". В этом методе происходит проверка введенных логина и пароля, а также открытие следующей формы при успешной авторизации.

{

AnsiString username = Edit1->Text;//- Получение введенного логина и пароля из соответствующих полей ввода

AnsiString password = Edit2->Text;

TADOQuery \*qry = new TADOQuery(NULL);//- Создание нового объекта запроса к базе данных и настройка его свойств

qry->Connection = ADOConnection1;

qry->SQL->Text = "SELECT \* FROM TablPolz WHERE Логин = :username AND Пароль = :password";

qry->Parameters->ParamByName("username")->Value = username;

qry->Parameters->ParamByName("password")->Value = password;

qry->Open();//- Выполнение запроса и открытие результирующего набора

if (!qry->IsEmpty())//- Если результирующий набор не пустой (логин и пароль введены верно)

{

ShowMessageWithTimer("Вы успешно зашли!", 1000);//- Отображение сообщения об успешной авторизации на заданное время

//MessageBeep(MB\_ICONASTERISK);//- Воспроизведение звукового сигнала

Timer2->Enabled = true;//- Запуск таймера, который будет выполнять дополнительные действия после авторизации

Form1->Visible = false;//- Скрытие текущей формы и открытие следующей формы

Form2->WindowState = wsMaximized;

Form2->Show();

}

else

{

//- Отображение сообщения об ошибке при неправильном вводе логина и пароля

ShowMessageWithTimer("Ошибка: Неверный логин или пароль, попробуйте зарегестрироваться", 3000);

//MessageBeep(MB\_ICONERROR);//- Воспроизведение звукового сигнала ошибки

Edit1->Text = "";//- Очистка полей ввода логина и пароля

Edit2->Text = "";

}

qry->Close();//- Закрытие результирующего набора и удаление объекта запроса

delete qry;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label6Click(TObject \*Sender)//- нажатие на кнопку зарегестрироваться)

{

AnsiString username = Edit1->Text; //- получаем введенный пользователем (логин, email)

AnsiString password = Edit2->Text; //- получаем введенный пользователем пароль

if (IsValidEmail(username))//- проверяем, что введенный (логин)email является допустимым

{

if (IsValidPassword(password))//- проверяем, что введенный пароль является допустимым

{

TADOQuery \*qry = new TADOQuery(NULL);// создаем новый запрос ADO

qry->Connection = ADOConnection1;// устанавливаем соединение ADO

qry->SQL->Text = "SELECT \* FROM TablPolz WHERE Логин = :username"; // формируем SQL запрос к базе данных

qry->Parameters->ParamByName("username")->Value = username;// задаем параметры запроса

qry->Open();// выполняем запрос

if (qry->IsEmpty())// проверяем, что результат запроса пустой, т.е. пользователя с таким логином(email) еще не зарегистрировали

{

ShowMessageWithTimer("Вы успешно зарегестрировались", 1000);// выводим сообщение об успешной регистрации с задержкой в 1 секунду

SaveUser(username, password);// сохраняем данные пользователя в базу данных

//MessageBeep(MB\_ICONASTERISK);// проигрываем звуковой сигнал при успешной регистрации

Timer2->Enabled = true;// запускаем таймер на форме для отображения текущего времени

Form1->Visible = false;// скрываем форму для авторизации

Form2->WindowState = wsMaximized;// устанавливаем состояние окна в "развернутое на весь экран"

Form2->Show();// отображаем следующую форму

}

else

{

ShowMessageWithTimer("Ошибка: Пользователь с таким логиным уже существует,\nвведите другой логин или попробуйте войти в свою учетную запись.", 2000);// выводим сообщение об ошибке при повторной регистрации пользователя

}

qry->Close();// закрываем запрос

delete qry;// освобождаем память, занятую запросом

}

else

{

ShowMessageWithTimer("Ошибка: Пароль должен содержать минимум 8 символов,\nвключая заглавные и строчные буквы и цифры.", 1500);// выводим сообщение об ошибке при некорректном вводе пароля

//MessageBeep(MB\_ICONERROR);// проигрываем звуковой сигнал при ошибке ввода пароля

}

}

else

{

ShowMessageWithTimer("Ошибка: Введите корректный Логин", 1000);// выводим сообщение об ошибке при некорректном вводе логина(email)

//MessageBeep(MB\_ICONERROR);

}

}

bool TForm1::IsValidEmail(const AnsiString& email)// Функция для проверки валидности логина

{

//TRegEx emailRegex("^\\w+([\\.-]?\\w+)\*@gmail\\.[a-z]{2,6}$", TRegExOptions());// Маска для проверки валидности email

TRegEx loginRegex("^[a-zA-Z0-9]{4,12}$", TRegExOptions());// Маска для проверки валидности логина

return loginRegex.IsMatch(email); // Проверка соответствия логина(email) маске

}

bool TForm1::IsValidPassword(const AnsiString& password)// Функция для проверки валидности пароля

{

// Переменные для отслеживания наличия заглавных букв, строчных букв и цифр

bool has\_uppercase = false;

bool has\_lowercase = false;

bool has\_digit = false;

if (password.Length() < 8)// Проверка длины пароля

{

return false;

}

for (int i = 1; i <= password.Length(); i++)// Проверка наличия заглавных букв, строчных букв и цифр

{

if (isupper(password[i]))

{

has\_uppercase = true;

}

if (islower(password[i]))

{

has\_lowercase = true;

}

if (isdigit(password[i]))

{

has\_digit = true;

}

}

return has\_uppercase && has\_lowercase && has\_digit;// Пароль валиден, если есть и заглавные буквы, и строчные буквы, и цифры

}

void TForm1::SaveUser(const AnsiString& username, const AnsiString& password)// Функция для сохранения пользователя в базу данных, если его email является валидным.

{

if (IsValidEmail(username))// Проверяем, является ли email валидным.

{

TADOQuery \*qry = new TADOQuery(NULL);// Создаем новый объект TADOQuery и устанавливаем ему соединение с базой данных.

qry->Connection = ADOConnection1;

qry->SQL->Text = "INSERT INTO TablPolz (Логин, Пароль) VALUES (:username, :password)";// Устанавливаем текст SQL-запроса для вставки нового пользователя в таблицу TablPolz.

qry->Parameters->ParamByName("username")->Value = username;// Задаем параметры запроса - email и пароль нового пользователя.

qry->Parameters->ParamByName("password")->Value = password;

qry->ExecSQL();// Выполняем запрос и освобождаем ресурсы.

delete qry;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Edit2Change(TObject \*Sender)// Функция для проверки пароля введенного в поле ввода и отображения статуса валидности пароля в Label3

{

AnsiString password = Edit2->Text;

if (IsValidPassword(password))

{

Label3->Caption = L"✅";

Label3->Font->Color = clGreen;

}

else

{

Label3->Caption = L"❎";

Label3->Visible = true;

Label3->Font->Color = clRed;

}

if (Edit2->Text == "")

{

Label3->Caption = "";

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Edit1Change(TObject \*Sender)// Функция для проверки логина введенного в поле ввода и отображения статуса валидности логина в Label2

{

AnsiString email = Edit1->Text;

if (IsValidEmail(email))

{

Label2->Caption = L"✅";

Label2->Font->Color = clGreen;

}

else

{

Label2->Caption = L"❎";

Label2->Visible = true;

Label2->Font->Color = clRed;

}

if (Edit1->Text == "")

{

Label2->Caption = "";

}

}

void \_\_fastcall TForm1::ShowMessageWithTimer(const AnsiString& message, int timeMs)// Функция для отображения сообщений на определенное количество времени

{

Label7->Caption = message;// отображение сообщения в TLabe

Timer1->Interval = timeMs;// установка времени таймера

Timer1->Enabled = true;// включение таймера

}

void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)

{

Timer1->Enabled = false;// отключение таймера

Label7->Caption = "";// cкрытие сообщения в TLabel

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label4Click(TObject \*Sender)// Функция для отображения глазика или заблокированного глазика, для показа/скрытия пароля

{

if (passwordVisible)

{

// скрыть пароль

Edit2->PasswordChar = '\*';

passwordVisible = false;

Label4->Caption = L"🙈";

}

else

{

// показать пароль

Edit2->PasswordChar = '\0';

passwordVisible = true;

Label4->Caption = L"🙉";

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Timer2Timer(TObject \*Sender)// Таймер для перехода на следующую форму, скрывает текущую форму

{

Timer2->Enabled = false;

Form1->Visible = false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Timer3Timer(TObject \*Sender)// Таймер для перехода на следующую форму, скрывает текущую форму

{

Timer3->Enabled = false;

Form1->Visible = false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)// Установка закругленных углов формы

{

HRGN rgn = CreateRoundRectRgn(0, 0, ClientWidth, ClientHeight, 50, 50);

SetWindowRgn(Handle, rgn, true);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label8Click(TObject \*Sender)// Обработчик события клика на кнопку выхода

{

Application->Terminate();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label8MouseEnter(TObject \*Sender)

{

Label8->Font->Color = clRed;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label8MouseLeave(TObject \*Sender)

{

Label8->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label5MouseEnter(TObject \*Sender)

{

Label5->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label5MouseLeave(TObject \*Sender)

{

Label5->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label6MouseEnter(TObject \*Sender)

{

Label6->Font->Color = clBlack;

}

//--------------------------------------------------------------------------

void \_\_fastcall TForm1::Label6MouseLeave(TObject \*Sender)

{

Label6->Font->Color = clGray;

}

void \_\_fastcall TForm1::Label4MouseEnter(TObject \*Sender)

{

Label4->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Label4MouseLeave(TObject \*Sender)

{

Label4->Font->Color = clGray;

}

//---------------------------------------------------------------------------

Начальная заставка

//---------------------------------------------------------------------------

/\*

В данном файле происходит объявление класса TForm2 и его методов.

Файл содержит также подключение необходимых библиотек, используемых в данном классе,

а также объявление объекта формы Form2.

Также в файле определены обработчики событий, такие как нажатие на Label и т.д.

При наведении курсора на некоторые элементы интерфейса происходит изменение их цвета

для улучшения визуального восприятия.

Файл содержит также использование внешней переменной previousform из другого модуля.

\*/

#include <vcl.h>

#include <PNGImage.hpp>

#pragma hdrstop

#include "Unit2.h"

#include "Unit3.h"

#include "Unit4.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm2 \*Form2;

extern int previousform;

bool Colors;

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

: TForm(Owner)

{

Form2->WindowState = wsMaximized;

Label1->Left = 1875;

Label1->Top = 10;

Label3->Left = 1871;

Label3->Top = 60;

Label2->Top = 500;

Label2->Left = 800;

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\день1.png");

Image1->Parent->DoubleBuffered = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label3Click(TObject \*Sender)

{

Form2->Visible = false;

Form3->WindowState = wsMaximized;

Form3->Show();

previousform = 2;

Form3->Label12->Visible = false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label1MouseEnter(TObject \*Sender)

{

Label1->Font->Color = clRed;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label1MouseLeave(TObject \*Sender)

{

Label1->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label1Click(TObject \*Sender)

{

Application->Terminate();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label2MouseEnter(TObject \*Sender)

{

Label2->Font->Color = clGreen;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label2MouseLeave(TObject \*Sender)

{

Label2->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label2Click(TObject \*Sender)

{

Form2->Visible = false;

Form4->WindowState = wsMaximized;

Form4->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label3MouseEnter(TObject \*Sender)

{

if(Colors == true)

{

Label3->Font->Color = clWhite;

}

else

{

Label3->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Label3MouseLeave(TObject \*Sender)

{

Label3->Font->Color = clGray;

}

//---------------------------------------------------------------------------

Настройки

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

// Подключение заголовочных файлов для форм

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

#include "Unit4.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm3 \*Form3;

// Объявление переменных

bool isPlaying = false;

bool isPlaying1 = false;

bool music1 = false;

bool music2 = false;

int previousform;

bool labelsVisible = false;

extern bool Colors;

extern int time\_left;

extern bool game\_started;

//---------------------------------------------------------------------------

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

: TForm(Owner)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\день2.png");

Image1->Parent->DoubleBuffered = true;

// Настройки формы на весь экран

Form3->BorderStyle = bsNone;

SetBounds(0,0,Screen->Width,Screen->Height);

Form3->WindowState = wsMaximized;

// Panel1->Width = Form3->Width;

// Panel1->Height = Form3->Height;

// Настройки расположения компонентов на форме

Label1->Top = 50;// настройки

Label1->Left = 750;

Label2->Left = 1875;// закрыть форму

Label2->Top = 10;

Label3->Left = 50;// музыка

Label3->Top = 300;

Label4->Left = 300;// музыка1

Label4->Top = 300;

Label5->Left = 600;// музыка2

Label5->Top = 300;

Label6->Left = 300;// светлая

Label6->Top = 400;

Label7->Left = 600;// темная

Label7->Top = 400;

Label8->Left = 900;// фон

Label8->Top = 300;

Label9->Left = 50;// тема

Label9->Top = 400;

Label10->Left = 50;// выход из аккаунта

Label10->Top = 600;

Label11->Left = 50;// рейтинг

Label11->Top = 500;

Label12->Left = 50;// выйти из игры

Label12->Top = 700;

Label13->Left = 1800;// назад

Label13->Top = 60;

TrackBar1->Left = 1000;// трэк бар для музыки

TrackBar1->Top = 320;

Panel2->Top = 300;

Panel2->Left = 1485;

//массив цветов для фона

colors[0] = clTeal;

colors[1] = clGray;

colors[2] = clSilver;

colors[3] = clWhite;

colorIndex = 0;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label12Click(TObject \*Sender)// выходим из данной игры

{

Form3->Visible = false;

Form2->Visible = true;

Form2->Show();

Form4->Label11->Enabled = true;

game\_started = false;

Form4->Timer1->Enabled = false;

Form4->Edit1->Visible = false;

Form4->Label4->Visible = false;

Form4->Label3->Visible = false;

Form4->Label6->Visible = false;

Form4->Label5->Visible = false;

time\_left = 60;

Form4->Label8->Visible = false;

Form4->Label7->Visible = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label2MouseEnter(TObject \*Sender)

{

Label2->Font->Color = clRed;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label2MouseLeave(TObject \*Sender)

{

Label2->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label2Click(TObject \*Sender)

{

Application->Terminate();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label3Click(TObject \*Sender)// музыка

{

Label4->Visible = !Label4->Visible;

Label5->Visible = !Label5->Visible;

Label8->Visible = !Label8->Visible;

TrackBar1->Visible = !TrackBar1->Visible;

labelsVisible = Label4->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label3MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label3->Font->Color = clWhite;

}

else

{

Label3->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label3MouseLeave(TObject \*Sender)

{

if (Label4->Visible == false)

{

Label3->Font->Color = clGray;

}

else

{

if (Colors == true)

{

Label3->Font->Color = clWhite;

}

else

{

Label3->Font->Color = clBlack;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label4MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label4->Font->Color = clWhite;

}

else

{

Label4->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label4MouseLeave(TObject \*Sender)

{

if (!music1)

{

Label4->Font->Color = clGray;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label5MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label5->Font->Color = clWhite;

}

else

{

Label5->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label5MouseLeave(TObject \*Sender)

{

if (!music2)

{

Label5->Font->Color = clGray;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label4Click(TObject \*Sender)

{

if (!isPlaying)

{

music1 = true;

Label8->Caption = L"🔊";

const char\* soundFile = "C:\\Users\\User\\Desktop\\Курсач\_в2.0\\Природа.wav";

WCHAR wideSoundFile[MAX\_PATH];

MultiByteToWideChar(CP\_ACP, 0, soundFile, -1, wideSoundFile, MAX\_PATH);

PlaySoundW(wideSoundFile, NULL, SND\_FILENAME | SND\_ASYNC);

isPlaying = true;

}

else

{

music1 = false;

Label8->Caption = L"🔇";

PlaySound(NULL, 0, 0);

isPlaying = false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label5Click(TObject \*Sender)

{

if (!isPlaying1)

{

music2 = true;

Label8->Caption = L"🔊";

const char\* soundFile = "C:\\Users\\User\\Desktop\\Курсач\_в2.0\\Йога.wav";

WCHAR wideSoundFile[MAX\_PATH];

MultiByteToWideChar(CP\_ACP, 0, soundFile, -1, wideSoundFile, MAX\_PATH);

PlaySoundW(wideSoundFile, NULL, SND\_FILENAME | SND\_ASYNC);

isPlaying1 = true;

}

else

{

music2 = false;

Label8->Caption = L"🔇";

PlaySound(NULL, 0, 0);

isPlaying1 = false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::TrackBar1Change(TObject \*Sender)// регулировка грромкости звука

{

double volume = TrackBar1->Position;

volume = volume / TrackBar1->Max \* 0xFFFF;

waveOutSetVolume(0, MAKELONG(volume, volume));

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label9MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label9->Font->Color = clWhite;

}

else

{

Label9->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label9MouseLeave(TObject \*Sender)

{

if(Label6->Visible == true)

{

if (Colors == true)

{

Label9->Font->Color = clWhite;

}

else

{

Label9->Font->Color = clBlack;

}

}

else

{

Label9->Font->Color = clGray;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label9Click(TObject \*Sender)

{

Label6->Visible = !Label6->Visible;

Label7->Visible = !Label7->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label10MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label10->Font->Color = clWhite;

}

else

{

Label10->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label10MouseLeave(TObject \*Sender)

{

Label10->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label10Click(TObject \*Sender)//выход из аккаунта

{

Form3->Visible = false;

Form1->Visible = true;

Form1->Show();

Form1->ShowMessageWithTimer("Вы успешно вышли из аккаунта", 1500);

Form1->Edit2->Text = "";

Form1->Edit1->Text = "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label11MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label11->Font->Color = clWhite;

}

else

{

Label11->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label11MouseLeave(TObject \*Sender)

{

if (Colors == true)

{

Label11->Font->Color = clGray;

if (Panel2->Visible == true)

{

Label11->Font->Color = clWhite;

}

else

{

Label11->Font->Color = clGray;

}

}

else

{

if (Panel2->Visible == true)

{

Label11->Font->Color = clBlack;

}

else

{

Label11->Font->Color = clGray;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label11Click(TObject \*Sender)// выдвижное memo1

{

ADOQuery1->Active = false;

ADOQuery1->SQL->Text = "SELECT \* FROM TablPolz ORDER BY Легкий DESC";

ADOQuery1->Active = true;

Panel2->Visible = !Panel2->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label13Click(TObject \*Sender)// для кнопки назад для разнах форм

{

if( previousform == 4)

{

Form3->Visible = false;

Form4->Visible = true;

Form4->Show();

Label12->Visible = false;

}

else if( previousform == 2)

{

Form3->Visible = false;

Form2->Visible = true;

Form2->Show();

}

else

{

Form3->Visible = false;

Form2->Visible = true;

Form2->Show();

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label13MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label13->Font->Color = clWhite;

}

else

{

Label13->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label13MouseLeave(TObject \*Sender)

{

Label13->Font->Color = clGray;

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label12MouseEnter(TObject \*Sender)

{

if (Colors == true)

{

Label12->Font->Color = clWhite;

}

else

{

Label12->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label12MouseLeave(TObject \*Sender)

{

Label12->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button1Click(TObject \*Sender)

{

ADOQuery1->Active = false;

ADOQuery1->SQL->Text = "SELECT \* FROM TablPolz ORDER BY Легкий DESC";

ADOQuery1->Active = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button2Click(TObject \*Sender)

{

ADOQuery1->Active = false;

ADOQuery1->SQL->Text = "SELECT \* FROM TablPolz ORDER BY Средний DESC";

ADOQuery1->Active = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button3Click(TObject \*Sender)

{

ADOQuery1->Active = false;

ADOQuery1->SQL->Text = "SELECT \* FROM TablPolz ORDER BY Сложный DESC";

ADOQuery1->Active = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label7Click(TObject \*Sender)

{

Colors = true;

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\ночь2.png");

Form4->Image2->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\доска4.png");

Form2->Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\ночь1.png");

Form2->Color = clBlack;

Panel1->Color = clBlack;

Form3->Color = clBlack;

Label8->Font->Color = clWhite;

Label9->Font->Color = clWhite;

Label1->Font->Color = clWhite;

Label16->Font->Color = clWhite;

if(Label4->Visible == true)

{

Label3->Font->Color = clWhite;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label6Click(TObject \*Sender)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\день2.png");

Colors = false;

Form4->Image2->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\доска3.png");

Form2->Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\день1.png");

Form2->Color = clWhite;

Form3->Color = clWhite;

Panel1->Color = clWhite;

Label8->Font->Color = clBlack;

Label9->Font->Color = clBlack;

Label1->Font->Color = clBlack;

if(Label4->Visible == true)

{

Label3->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label6MouseEnter(TObject \*Sender)

{

if(Colors == true)

{

Label6->Font->Color = clWhite;

}

else

{

Label6->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label6MouseLeave(TObject \*Sender)

{

Label6->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label7MouseEnter(TObject \*Sender)

{

if(Colors == true)

{

Label7->Font->Color = clWhite;

}

else

{

Label7->Font->Color = clBlack;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Label7MouseLeave(TObject \*Sender)

{

Label7->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::DBGrid1DrawColumnCell(TObject \*Sender, const TRect &Rect,

int DataCol, TColumn \*Column, TGridDrawState State)

{

if (DBGrid1->SelectedRows->CurrentRowSelected)

{

DBGrid1->Canvas->Brush->Color = clRed;

}

else

{

DBGrid1->Canvas->Brush->Color = clWhite;

DBGrid1->DefaultDrawColumnCell(Rect, DataCol, Column, State);

}

}

//---------------------------------------------------------------------------

Игровая форма

//---------------------------------------------------------------------------

#include <vcl.h>

#include <GIFImg.hpp>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

#include "Unit4.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm4 \*Form4;

extern int previousform;// объявление глобальной переменной, которая ссылается на предыдущую форму.

bool Colors;

//объявление глобальных переменных для уровня сложности, операции, чисел, результатов, количества вопросов и правильных ответов, времени на ответ и текущего состояния игры.

int level, operation, num1, num2, result, num\_of\_questions = 0, num\_of\_correct\_answers = 0;

int time\_left = 60;

bool game\_started = false;

int difficulty = -1; // 0 - легкий, 1 - средний, 2 - сложный

//---------------------------------------------------------------------------

\_\_fastcall TForm4::TForm4(TComponent\* Owner)

: TForm(Owner)

{

Edit1->Color = RGB(21, 111, 19);

Label9->Top = 565;

Label9->Left = 1000;

Label10->Top = 565;

Label10->Left = 1000;

Image2->Parent->DoubleBuffered = true;

Image2->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\доска3.png");

Image2->Top = 0;

Image2->Left = 0;

Form4->BorderStyle = bsNone;// форма во весь экран

SetBounds(0,0,Screen->Width,Screen->Height);

Form4->WindowState = wsMaximized;

Label1->Top = 60;

Label1->Left = 1871;

Label2->Top = 10;

Label2->Left = 1875;

Label3->Top = 550;

Label3->Left = 500;

Label4->Left = 735;

Label4->Top = 450;

Label5->Top = 140;

Label5->Left = 1050;

Label6->Top = 350;

Label6->Left = 260;

Label7->Top = 140;

Label7->Left = 530;

Label8->Top = 140;

Label8->Left = 530;

Edit1->Top = 550;

Edit1->Left = 800;

Label11->Top = 140;//уровни

Label11->Left = 260;

Label12->Top = 200;

Label12->Left = 260;

Label13->Top = 250;

Label13->Left = 260;

Label14->Top = 300;

Label14->Left = 260;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::generateExample()// функция, которая генерирует новый пример для игры

{

srand(time(NULL));

operation = rand() % 2;

switch (difficulty) //ComboBox1->ItemIndex

{

case 0:

num1 = rand() % 9 + 1;

num2 = rand() % 9 + 1;

result = operation == 0 ? num1 + num2 : num1 - num2;

break;

case 1:

num1 = rand() % 90 + 10;

num2 = rand() % 90 + 10;

result = operation == 0 ? num1 + num2 : num1 - num2;

break;

case 2:

num1 = rand() % 900 + 100;

num2 = rand() % 900 + 100;

result = operation == 0 ? num1 + num2 : num1 - num2;

break;

}

Label3->Caption = IntToStr(num1) + (operation == 0 ? " + " : " - ") + IntToStr(num2) + " = ";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label1Click(TObject \*Sender)//- функция, которая переводит игрока на предыдущую форму приложения.

{

Form4->Visible = false;

Form3->Visible = true;

Form3->WindowState = wsMaximized;

Form3->Show();

previousform = 4;

Form3->Label12->Visible = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Edit1KeyDown(TObject \*Sender, WORD &Key, TShiftState Shift)// функция, которая обрабатывает нажатие на клавишу Enter

{

if (Key == VK\_RETURN) // если нажата клавиша Enter

{

Label8Click(Sender); // вызываем обработчик события нажатия на кнопку

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Timer1Timer(TObject \*Sender)// функция, которая обновляет таймер игры

{

if (game\_started)

{

Label3->Visible = true;

Edit1->Visible = true;

Label4->Visible = true;

Label6->Visible = true;

Label5->Visible = true;

time\_left--;

Label6->Caption = "Осталось времени: " + IntToStr(time\_left) + " секунд";

if (time\_left == 0)

{

Label11->Caption = "Уровни";

Label11->Font->Color = clWhite;

Label11->Enabled = true;

Label3->Visible = false;

Label5->Visible = false;

Label4->Visible = false;

Edit1->Visible = false;

Label6->Visible = false;

Timer1->Enabled = false;

game\_started = false;

time\_left = 60;

Application->Title = " ";

ShowMessage("Игра окончена! Всего задано вопросов: " + IntToStr(num\_of\_questions) + ". Правильных ответов: " + IntToStr(num\_of\_correct\_answers) + ". Неправильных ответов: " + IntToStr(num\_of\_questions - num\_of\_correct\_answers) + ".");

Label8->Visible = false;

Label7->Visible = true;

String level;

switch (difficulty)

{

case 0: level = "Легкий"; break;

case 1: level = "Средний"; break;

case 2: level = "Сложный"; break;

}

difficulty = -1;

String tableName = "TablPolz";

String email = Form1->Edit1->Text;

TADOQuery \*query = new TADOQuery(NULL);

query->Connection = ADOConnection1;

// получение текущего значения из базы данных

query->SQL->Text = "SELECT " + level + " FROM " + tableName + " WHERE Логин = :email";

query->Parameters->ParamByName("email")->Value = email;

query->Open();

if (!query->Eof)

{

int currentValue = query->Fields->Fields[0]->AsInteger;

// сравнение текущего значения с новым значением

if (num\_of\_correct\_answers > currentValue)

{

// обновление значения в базе данных

query->Close();

query->SQL->Text = "UPDATE " + tableName + " SET " + level + " = :result WHERE Логин = :email";

query->Parameters->ParamByName("result")->Value = num\_of\_correct\_answers;

query->Parameters->ParamByName("email")->Value = email;

query->ExecSQL();

Application->Title = " ";

ShowMessage("Поздравляем! Dы побили свой рекорд");

Form3->ADOQuery1->Active = false;

Form3->ADOQuery1->Active = true;

}

else

{

Application->Title = " ";

ShowMessage("Нерастраивайтесь. Попробуйте побить рекорд в следующей игре");

}

}

delete query;

}

}

else

{

Timer1->Enabled = false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label2Click(TObject \*Sender)

{

Application->Terminate();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label1MouseEnter(TObject \*Sender)

{

Label1->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label1MouseLeave(TObject \*Sender)

{

Label1->Font->Color = clGray;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label2MouseEnter(TObject \*Sender)

{

Label2->Font->Color = clRed;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label2MouseLeave(TObject \*Sender)

{

Label2->Font->Color = clGray;

}

void \_\_fastcall TForm4::Timer2Timer(TObject \*Sender)

{

// Отключение таймера

Timer2->Enabled = false;

Label9->Visible = false;

Label10->Visible = false;

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label5Click(TObject \*Sender)

{

difficulty = -1;

Label11->Caption = "Уровни";

Label11->Font->Color = clWhite;

Label11->Enabled = true;

game\_started = false;

Timer1->Enabled = false;

Edit1->Visible = false;

Edit1->Clear();

Label4->Visible = false;

Label3->Visible = false;

Label6->Visible = false;

Label5->Visible = false;

time\_left = 60;

Label8->Visible = false;

Label7->Visible = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label5MouseEnter(TObject \*Sender)

{

Label5->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label5MouseLeave(TObject \*Sender)

{

Label5->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label7MouseEnter(TObject \*Sender)

{

Label7->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label7MouseLeave(TObject \*Sender)

{

Label7->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label7Click(TObject \*Sender)

{

if (difficulty == -1)

{

Application->Title = " ";

ShowMessage("Выберите уровень сложности!");

return;

}

Timer1->Enabled = true;

Label8->Visible = true;

Label3->Visible = true;

Label7->Visible = false;

game\_started = true;

level = difficulty;

num\_of\_questions = 0;

num\_of\_correct\_answers = 0;

generateExample();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label8Click(TObject \*Sender)

{

if (game\_started)

{

int user\_answer = StrToInt(Edit1->Text);// Получаем ответ пользователя из поля Edit1

if (user\_answer == result)// Проверяем правильность ответа и увеличиваем количество правильных ответов, если ответ верный

{

num\_of\_correct\_answers++;

Label9->Visible = true;

// Настройка TTimer для срабатывания через 1 секунду

Timer2->Interval = 700;

Timer2->Enabled = true;

}

else

{

Label10->Visible = true;

// Настройка TTimer для срабатывания через 1 секунду

Timer2->Interval = 700;

Timer2->Enabled = true;

}

num\_of\_questions++;// Увеличиваем количество заданных вопросов

generateExample();// Генерируем следующий пример

Edit1->Clear();// Очищаем поле Edit1

}

else

{

Label7Click(Sender);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label8MouseEnter(TObject \*Sender)

{

Label8->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label8MouseLeave(TObject \*Sender)

{

Label8->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label11Click(TObject \*Sender)

{

Label12->Visible = !Label12->Visible;

Label13->Visible = !Label13->Visible;

Label14->Visible = !Label14->Visible;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label11MouseEnter(TObject \*Sender)

{

Label11->Font->Color = clBlack;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label12MouseEnter(TObject \*Sender)

{

Label12->Font->Color = clLime;

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label13MouseEnter(TObject \*Sender)

{

Label13->Font->Color = clYellow;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label14MouseEnter(TObject \*Sender)

{

Label14->Font->Color = clRed;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label11MouseLeave(TObject \*Sender)

{

if(Label12->Visible == true)

{

Label11->Font->Color = clBlack;

}

else

{

Label11->Font->Color = clWhite;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label12MouseLeave(TObject \*Sender)

{

Label12->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label13MouseLeave(TObject \*Sender)

{

Label13->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label14MouseLeave(TObject \*Sender)

{

Label14->Font->Color = clWhite;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label12Click(TObject \*Sender)

{

difficulty = 0;

Label12->Visible = false;

Label13->Visible = false;

Label14->Visible = false;

Label11->Caption = Label12->Caption;

Label11->Enabled = false;

Label12->Font->Color = clLime;

Edit1->Left = 740;

Label4->Left = 735;

Label10->Left = 920;

Label9->Left = 920;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label13Click(TObject \*Sender)

{

difficulty = 1;

Label12->Visible = false;

Label13->Visible = false;

Label14->Visible = false;

Label11->Caption = Label13->Caption;

Label11->Enabled = false;

Label13->Font->Color = clYellow;

Edit1->Left = 820;

Label4->Left = 815;

Label10->Left = 1000;

Label9->Left = 1000;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm4::Label14Click(TObject \*Sender)

{

difficulty = 2;

Label12->Visible = false;

Label13->Visible = false;

Label14->Visible = false;

Label11->Caption = Label14->Caption;

Label11->Enabled = false;

Label14->Font->Color = clRed;

Edit1->Left = 890;

Label4->Left = 885;

Label10->Left = 1070;

Label9->Left = 1070;

}

ПРИЛОЖЕНИЕ 2 – ГАФИЧЕСКАЯ ЧАСТЬ

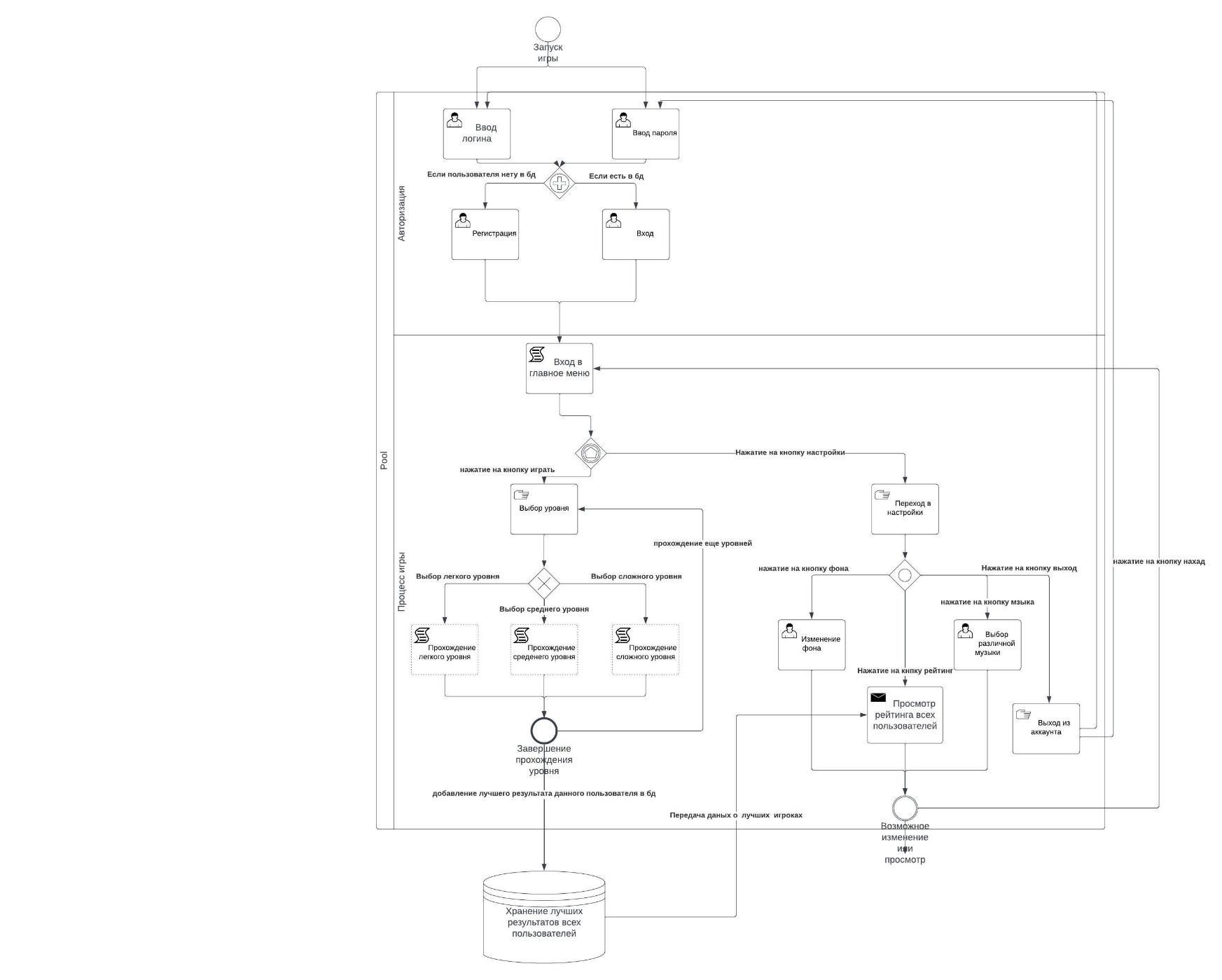


Рис. 5 Диаграмма BPMN

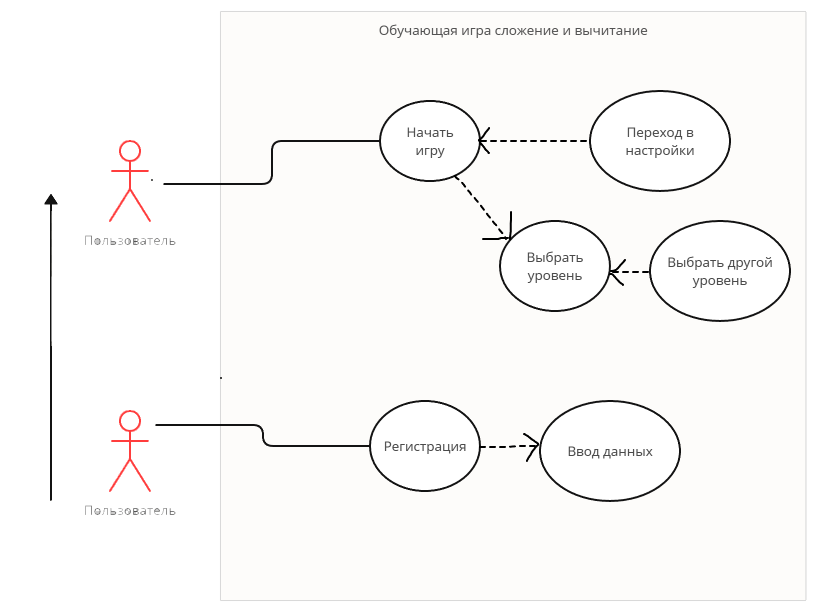


Рис. 5.1 Диаграмма вариантов использования