Муниципальное бюджетное общеобразовательное учреждение
«лицей «Созвездие» № 131» городского округа Самара
443083, г. Самара, ул. Промышленности,319, тел.: 261-14-40, 261-14-90.
E-mail: school131@bk.ru

ОБЛАСТНАЯ НАУЧНО-ФАНТАСТИЧЕСКАЯ КОНФЕРЕНЦИЯ ШКОЛЬНИКОВ И СТУДЕНТОВ «НАМ ЖИТЬ В БУДУЩЕМ»

Тема проекта:

«Разработка компьютерной игры "Повторим математику"»

Выполнил: Беляков Даниил Андреевич,
класс 10 «А»
Руководитель: Макарова Ирина Сергеевна,
доцент кафедры ПМИИС
СамГУПС

Самара 2019

ВВЕДЕНИЕ

Учащиеся всегда считали математику сложным предметом, изучение которого требует значительных умственных усилий. Как считают учителя математики, 5-6 классы – “критический возраст” в математическом развитии, т.к. освоение необходимых вычислительных навыков делает учебу однообразной, а саму математику неинтересной.

Великий французский математик и философ 17 века Блез Паскаль говорил: «Предмет математики настолько серьезен, что надо не упускать случая, сделать его немного занимательным». Поэтому важнейшим фактором успеха в изучении математики является интерес учеников к предмету. Умение заинтересовать математикой – дело непростое, поэтому игровая форма проведения занятий – это мощный стимул обучения. Игра может способствовать активизации мыслительной деятельности, повысить внимательность, настойчивость, работоспособность школьника.

Сначала учащихся интересует только сам процесс игры, а уже потом тот материал, без которого нельзя участвовать в игре. Тем не менее, нужно отметить, что в связи с высокой загруженностью на уроках математики, нет возможности включить игру непосредственно в процесс урока. Однако игру можно использовать на внеурочных занятиях и для самостоятельной работы школьников. В ходе игры учащиеся незаметно для себя выполняют различные упражнения, где им самим приходится выполнять арифметические действия, тренироваться в округлении, решать логические задачи. Использование дидактических игр не только способствует лучшему усвоению программного материала по математике, но и развитию логического мышления, речи, развитию наблюдательности, внимания и интереса к предмету [1].

Сегодня существует достаточно много компьютерных игр по различным предметам, в том числе по математике, что позволяет использовать эти игры, как на уроках, так и во внеклассной работе.

Компьютерные игры особенно популярны у детей школьного возраста, т.к. они позволяют учащемуся находиться в привычной для него среде, использовать хорошо освоенные навыки работы с компьютерными программами.

Цель проекта: разработка компьютерной игры по математике для учеников 5-6-х классов.

Задачи проекта:

- исследовать список тем, изучаемых на уроках математики в 5-6-м классе, и выделить из них те, по которым было бы целесообразно провести повторение материала при помощи компьютерной игры;

- исследовать возможности среды программирования PascalABC.Net, на основе которых в дальнейшем разработать игру, направленную на повторение того или иного материала для уроков математики в 5-6-х классах;

Гипотеза: с помощью внесения игровых элементов в учебный процесс можно выработать у ученика мотивацию к обучению и, как итог, повысить его успеваемость.

Инструментальные средства: программная среда PascalABC.NET

§1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Обоснование выбора темы проекта

Целью изучения курса математики в 5-6-х классах является систематическое развитие понятия числа, выработка умений выполнять устно и письменно арифметические действия над числами, переводить практические задачи на язык математики, подготовка учащихся к изучению систематических курсов алгебры и геометрии. В ходе изучения курса учащиеся развивают навыки вычислений с натуральными числами, овладевают навыками с обыкновенными и десятичными дробями, положительными и отрицательными числами, получают представление об использовании букв для записи выражений и свойств арифметических действий, составлении уравнений.

Основным звеном в процессе обучения математике являются математические упражнения. Следовательно, при проведении уроков обязательным является наличие системы обучающих, тренировочных, творческих математических упражнений, которые будут способствовать познавательной активности, интересу учеников к самостоятельной работе и формированию общих приемов учебной деятельности в изучении математики.

Однако просто выполнять тренировочные упражнения учащимся 5-6-х классов скучно, т.к. дети в возрасте 11-12 лет не имеют сильной мотивации к самостоятельной работе. По возрасту им гораздо интереснее играть в игры, особенно в компьютерные, т.к. для современных школьников компьютер фактически превратился в привычный бытовой прибор. Поэтому было принято решение разработать компьютерную игру для учащихся 5-6-х классов, при помощи которой они смогли бы повторить отдельные темы математики.

При изучении базисного учебного плана по математике для 5-х и 6-х классов было принято решение в качестве основных тем выбрать следующие [2], [3]:

- задачи на округление чисел;
- выполнение простейших арифметических операций (сложение/вычитание/умножение/деление) как с обычными числами, так и с десятичными;
- решение линейных уравнений;
- логические задачи.

Выбор тем не случаен, т.к. их освоение требует от школьника систематического выполнения достаточно однотипных действий, не отличающихся большим разнообразием, что делает изучение перечисленных тем неинтересным и скучным. Кроме того перечисленные навыки необходимы для уверенного владения приемами математических вычислений, что особенно важно для учащихся при подготовке к ГИА и ЕГЭ.

1.2. Среда программирования PascalABC.Net как платформа для создания компьютерной игры по математике

В качестве платформы для создания игры было решено использовать PascalABC.NET – язык программирования Паскаль нового поколения, включающий классический Паскаль, большинство возможностей языка Delphi, а также ряд собственных расширений. Язык реализован на платформе Microsoft.NET и содержит все современные языковые средства: классы, перегрузку операций, интерфейсы, обработку исключений, обобщенные классы и подпрограммы, сборку мусора, средства параллельного программирования.

PascalABC.NET – это:

- современный язык программирования, основанный на Delphi (Object Pascal) и сочетающий простоту языка Паскаль и огромные возможности платформы .NET;

- бесплатная, простая и мощная среда разработки, ориентированная на обучение программированию;
- уникальная Web-среда, позволяющая разрабатывать и запускать программы на языке Паскаль из окна браузера, а также иметь личный каталог программ на сервере.

Основные достоинства PascalABC.NET:

- бесплатность [6];
- возможность использования большинства библиотек платформы .NET;
- возможность интерактивного ввода-вывода (приостановка программы на сервере в процессе ввода данных на клиенте);
- для зарегистрированных пользователей - возможность поддерживать файловую структуру на сервере;
- возможность создания графических приложений (модуль Graph);
- возможность опубликовать программу в общий каталог. В результате программа может быть открыта по специальной ссылке и моментально выполнена;
- возможность открыть публичный доступ на чтение к своей программе;
- возможность скомпилировать файл и скачать его к себе на компьютер в виде exe файла.

PascalABC.NET имеет подсказки, которые предоставляются при написании кода, автоформатирование, отладчик и дизайнер форм. Благодаря платформе .NET, программное обеспечение поддерживает все возможности Microsoft.NET и ее же библиотеки. Можно смело использовать последние в своих программах, опираясь даже на те, которые созданы на другом языке программирования. Для того чтобы создать графические элементы в программе PascalABC.NET используются модули векторной и растровой графики. Среда программирования имеет дизайнер форм, благодаря которому можно создать оконные приложения.

Модуль GraphABC представляет собой простую графическую библиотеку и предназначен для создания несобытийных графических и

анимационных программ в процедурном и частично в объектном стиле. Рисование осуществляется в специальном графическом окне, возможность рисования в нескольких окнах отсутствует. Кроме этого, в модуле GraphABC определены простейшие события мыши и клавиатуры, позволяющие создавать элементарные событийные приложения. Модуль GraphABC основан на графической библиотеке GDI+, но запоминает текущие перо, кисть и шрифт, что позволяет не передавать их в качестве параметров при вызове графических примитивов. К свойствам пера, кисти и шрифта можно получать доступ как в процедурном, так и в объектном стиле. В модуле GraphABC можно управлять самим графическим окном и компонентом GraphABCControl, на котором осуществляется рисование. По умолчанию компонент GraphABCControl занимает всю клиентскую часть графического окна, однако, на графическое окно можно добавить элементы управления, уменьшив область, занимаемую графическим компонентом.

Для работы с рисунками используется класс Picture, позволяющий рисовать на себе те же графические примитивы, что и на экране. Режим блокировки рисования на экране (LockDrawing) позволяет осуществлять прорисовку лишь во внеэкранном буфере, после чего с помощью метода Redraw восстанавливать все графическое окно. Данный метод используется для ускорения анимации и создания анимации без мерцания. В модуле GraphABC определен ряд констант, типов, процедур, функций и классов для рисования в графическом окне.

Модуль ABCObjects реализует векторные графические объекты с возможностью масштабирования, наложения друг на друга, создания составных графических объектов и многократного их вложения друг в друга. Каждый векторный графический объект корректно себя перерисовывает при перемещении, изменении размеров и частичном перекрытии другими объектами. [4]

§2. ПРОЦЕСС РАЗРАБОТКИ КОМПЬЮТЕРНОЙ ИГРЫ «ПОВТОРИМ МАТЕМАТИКУ» ДЛЯ УЧЕНИКОВ 5-6-Х КЛАССОВ

2.1. Постановка задачи

Целью данной работы является программная реализация игры «Повторим математику».

Для реализации поставленной цели необходимо выполнить следующие задачи:

1) создать графическое окно, которое будет являться игровым полем;

2) создать визуализированный и интуитивно понятный интерфейс;

3) проработать алгоритмы управления графическим окном, генерирования задач на каждую из тематик, режимов игры (которые будут иметь свои те или иные нюансы);

4) создать методы, с помощью которых результат каждой игры будет сравниваться с другими, более лучшими результатами, и, в случае чего, будет заноситься в отведённую для этого простейшую базу данных, включающую в себя имена и количество баллов 10 игроков, показавших наилучшие результаты;

5) Создать класс переменных, которые могут использоваться для вышеописанных алгоритмов в программе повсеместно;

2.2. Интерфейс игры

На рисунке 1 представлено игровое поле, которое состоит из нескольких частей: окна с задачей/выражением для решения, поле для ввода ответа с клавиатуры и в двух отдельных окошках – количество набранных баллов, равное количеству решённых задач, а также время, которое прошло с момента начала игры либо время, оставшееся до конца игры (в зависимости от выбранного режима):

Рис.1.

При старте игры программа в зависимости от выбранного типа задач для решения и сложности генерирует случайным образом задачу и выводит на экран условия, сохраняя в отдельную переменную значение правильного ответа к ней. Игрок вводит ответ с помощью клавиатуры – событие, считывающее нажатие как числовых, так и символьных клавиш, заносит данные в строчную переменную. Он также может его исправить, стерев ненужные символы кнопкой Backspace. Для подтверждения нужно нажать клавишу Enter.

Полученная строчная переменная сверяется со значением переменной, соответствующей правильному ответу. Если ответ неверный, он начинает моргать красным цветом и игра завершается. Если же ответ правильный, он также моргает, но уже зелёным цветом, при этом игрок получает 1 балл и переходит к новой задаче.

2.3. Алгоритмическое конструирование

В ходе работы были созданы следующие 5 библиотек с процедурами [5], сходными по выполняемым ими действиями:

Библиотека	Выполняемые задачи
StartGame	Основная программа. Внутри неё – лишь открытие графического окна и выход на главное меню.
MainActions	Основные действия по управлению программой, включающие в себя работу с главным меню, настройкой параметров игры, с самим игровым процессом
Tasks	Генерирование необходимых задач в зависимости от выбранного типа и их сложности
Graphics	Работа с графикой: прорисовка спрайтов кнопок и их анимации, а также вывод определённого текста внутри них
MouseAndKeyBoardActions	События, связанные с взаимодействием с мышью и клавиатурой во время игры

Рассмотрим алгоритмы внутри библиотек MainActions и Tasks

MainActions

Алгоритм Start запускает программу. В графическом окне отображается главный экран, представляющий из себя основное меню игры. Кнопка «Начать» выводит игрока на окна для выбора параметров игры, кнопка «Выход» позволяет покинуть игру. Обе кнопки имеют три спрайта: когда на них не наведён курсор, когда на них наведён курсор и когда кнопка нажата (рис. 2):

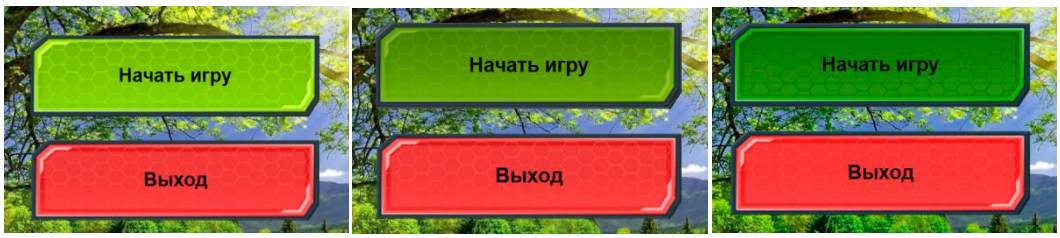

Рис. 2.

Алгоритм RazmMode запускается каждый раз, когда игрок начинает игру из главного меню или хочет поиграть ещё раз после завершённой игры, но в другом режиме. Алгоритм отвечает за выбор игроком таких параметров, как режим игры (“На время”/”Ни единой ошибки”), тип заданий (“Округление”/”Сложение/Вычитание”/”Деление/Умножение”/”Уравнения”/” На логику”), а также их сложность (легко/средне/трудно/тяжело). Рядом с кнопкой каждого параметра имеется маркер, который показывает, выбран ли он (параметр) на данный момент (см. рис.3).

Перед началом игрок должен ввести своё имя для дальнейшего сохранения в таблице рекордов в случае, если он побьёт чей-либо лучший результат. С главной страницы можно будет перейти в таблицу рекордов или узнать, какие задания будут генерироваться на каждом уровне сложности.

Рис. 3.

Алгоритм RecordLink перенаправляет с предыдущей процедуры на таблицу лидеров по очкам в пределах данного режима - типа заданий-сложности. С помощью алгоритма ConvertInfo значения данных вытаскиваются в переменные, после чего выводятся на экран (см.рис.4).

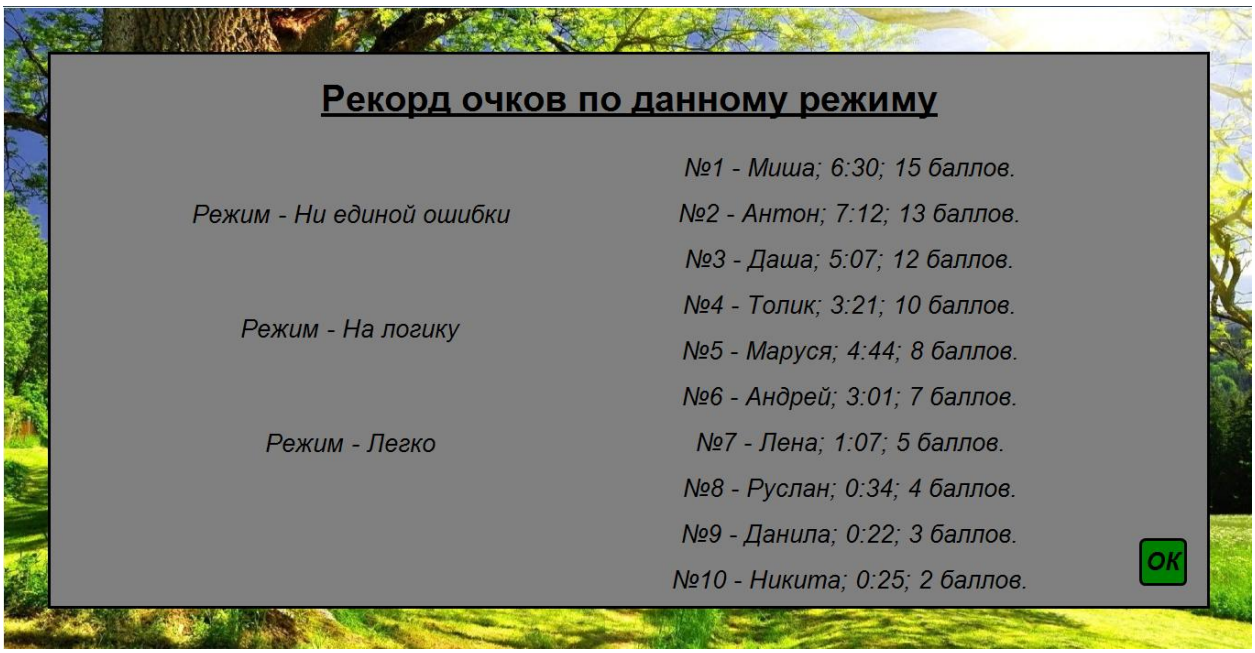

Рис. 4.

Аналогично, элементарный алгоритм DifficultyInfo переводит нас на информационное окно о соответствии разных заданий одного типа сложностям.

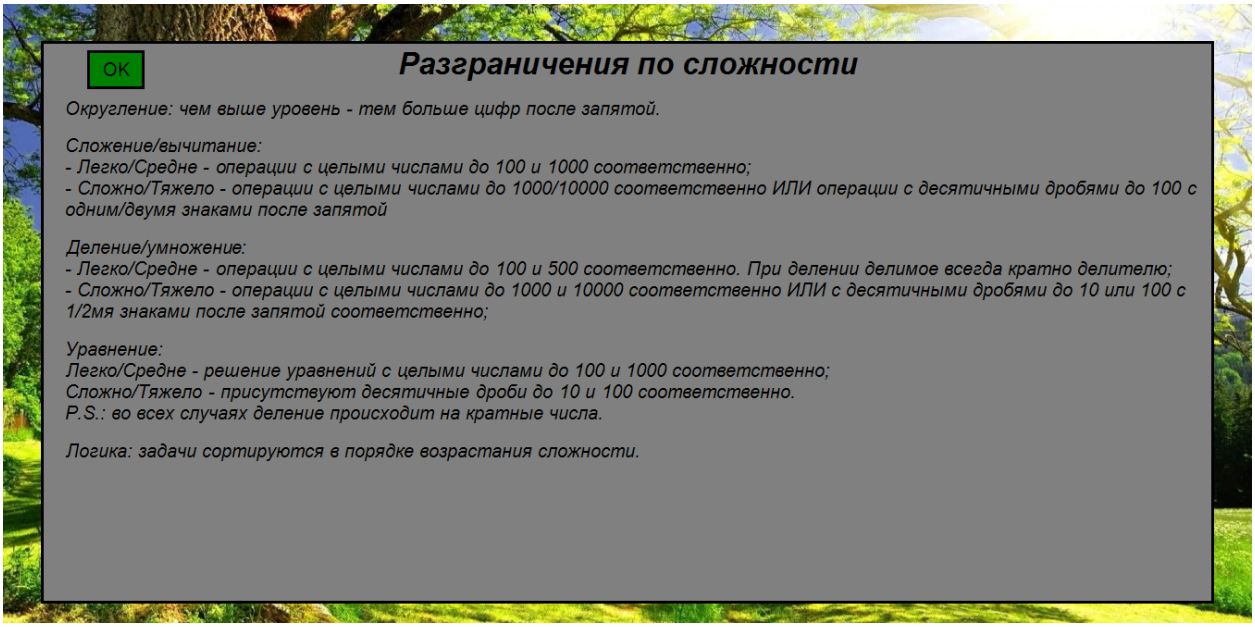

Рис. 5.

Алгоритм ForTime запускает режим игры на время. Его суть – набрать наибольшее количествово баллов за 10 минут (исключение – логические задачи, где на решение отводится 15 минут). Время высвечивается в нижнем правом окошке, а за его обратный отсчёт и вывод на экран отвечает алгоритм

Timer. Игрок может допускать сколько угодно ошибок. Когда время истекает, в окошке ‘0:00’ подсвечивается красным цветом, а игроку даётся возможность закончить решение последнего задания до окончания игры.

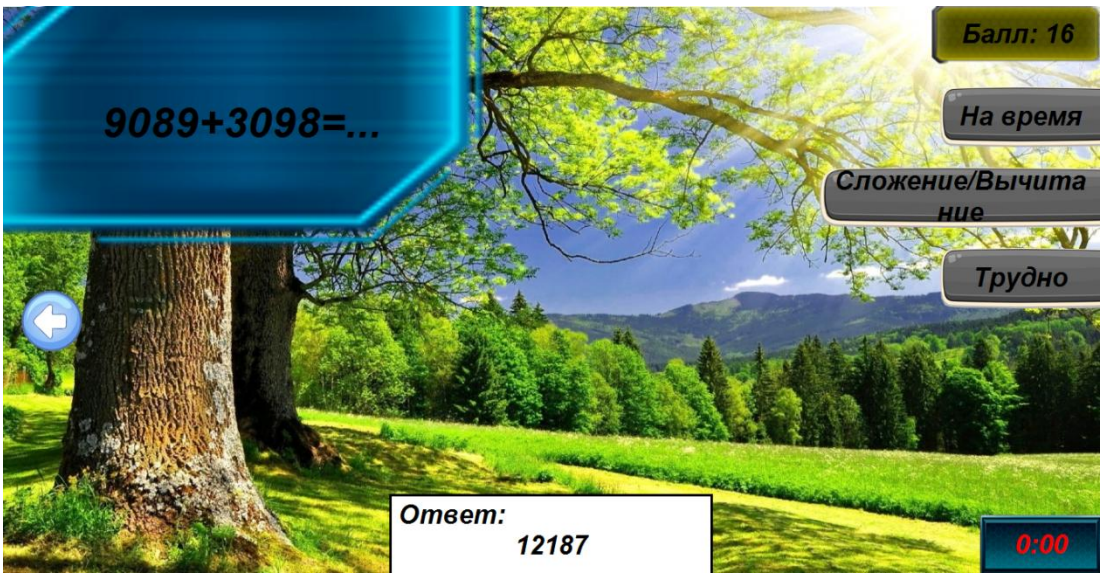

Рис. 6.

Алгоритм NoError запускает режим игры “Ни единой ошибки”. Цель – набрать наибольшее количество баллов, не допустив при этом ни единой ошибки. Время при этом неограниченно. Задачи генерируются до тех пор, пока игрок не даст неверный ответ на одну из них. Потраченное время засекается, и по окончанию игры записывается в базу вместе с количеством набранных баллов и именем игрока.

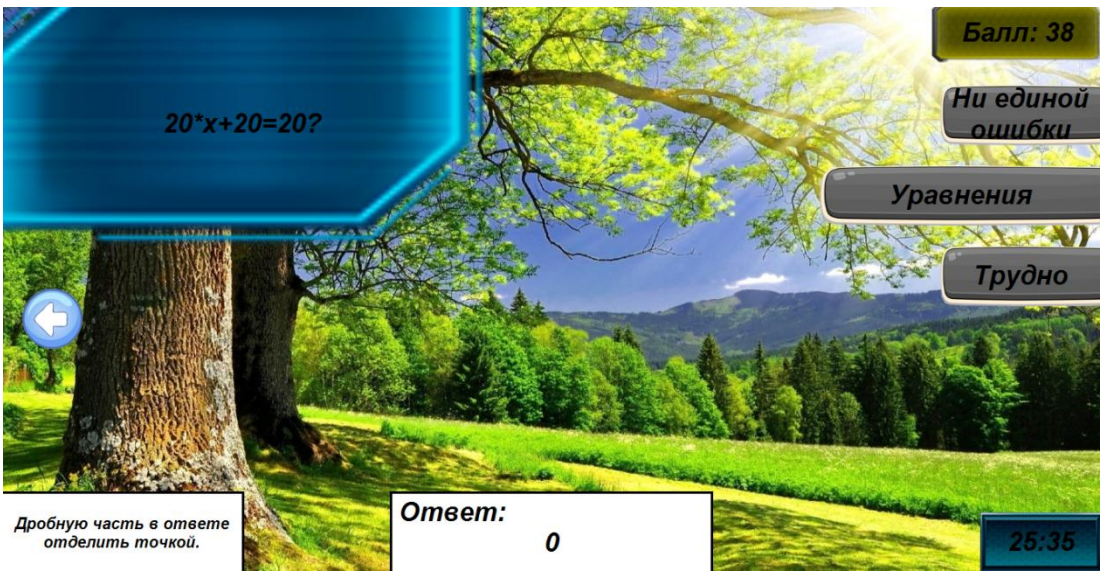

Рис. 7.

Оба предыдущих алгоритма по окончании игры выводят диалоговое окно, которое показывает игроку его результаты и предлагает выполнить одно из 3-х действий: начать игру заново, выйти в меню выбора параметров, чтобы начать игру сначала или выйти из программы.

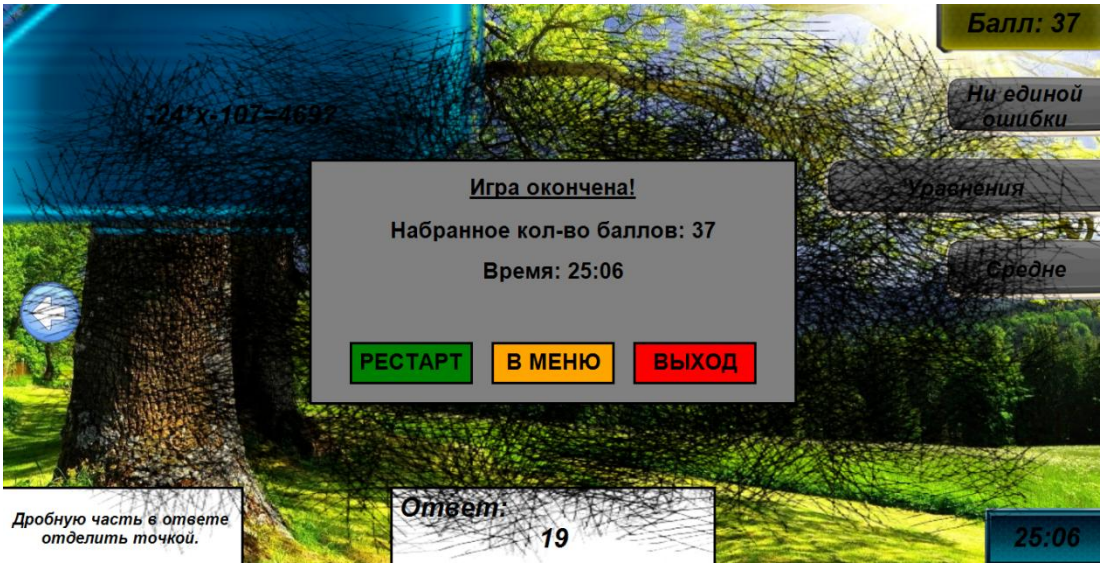

Рис. 8.

Алгоритм SaveResults взаимодействует с базой данных рекордов (находится в папке Data в каталоге вместе с самой игрой; данные рассортированы по папкам) вызывается каждый раз, когда игра заканчивается. Он извлекает из базы данные с помощью алгоритма ConvertInfo по рекордам на данном режиме и сравнивает: если игрок кого-то обогнал по очкам, его результат вносится в этот список. Затем происходит обратная конвертация, из переменных – в текстовый файл, с помощью алгоритма ConvertBack.

Tasks

Алгоритм Rounding генерирует задачу на округление числа. В зависимости от выбранной сложности генерируется несколько знаков после запятой и необходимо совершить округление до целых/десятых/сотых/тысячных (что зависит от сложности). (рис.9)

Рис.9

Алгоритм AdditionSubtraction генерирует задачу на сложение/вычитание чисел. В зависимости от сложности подбираются малые или большие целые числа, а также десятичные числа с 1-2 знаками после запятой. Все числа, а также выполняемое действие (сложение или вычитание) определяются оператором random(). (рис.10)

Рис.10

Алгоритм MultiplicationDivision работает аналогично, но уже с умножением и делением чисел: тоже в зависимости от сложности случайным образом подбираются целые или десятичные числа. Однако при делении часто будет случаться так, что ответ будет периодическим или иррациональным числом. Для упрощения игроку нужно в этом случае написать ответ, округлённый до сотых долей. (рис.11)

Рис.11

Алгоритм Equations генерирует простейшие, с приведёнными слагаемыми, линейные уравнения. Здесь также будут попадаться как целые числа до 100 и 1000, так и вещественные числа. Для удобства было принято решение, чтобы деление внутри уравнения было на кратное число. Таким образом, в любом случае корень уравнения будет являться целым числом (см. рис.12).

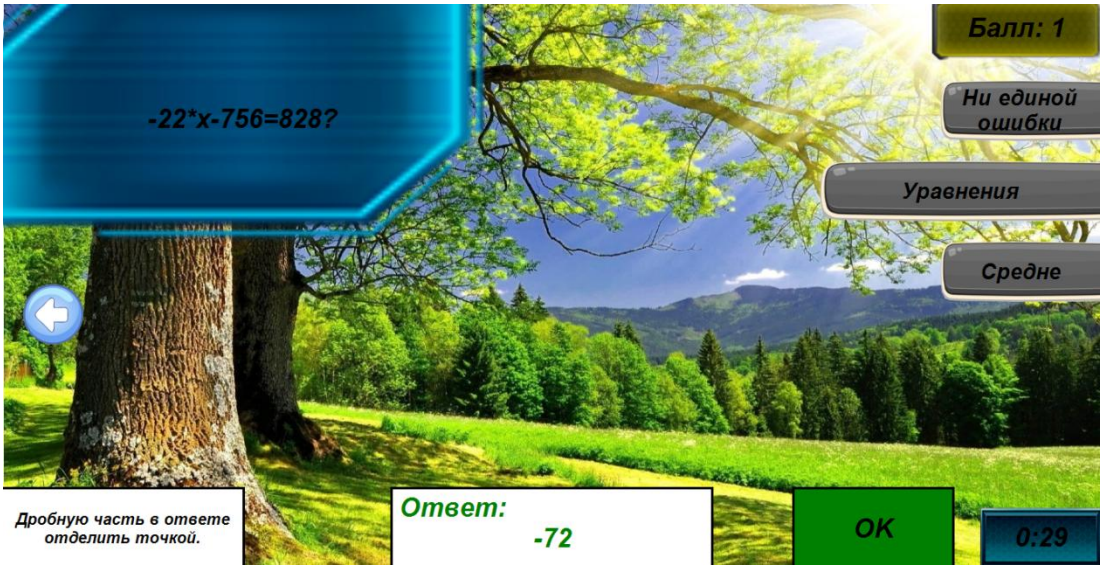

Рис.12

Алгоритм LogicTask берёт предварительно загруженные в базу заданий BaseOfTasks условия логических заданий, тематические картинки к ним значения правильных ответов и так же, как и в случае с предыдущими алгоритмами, формирует графическое поле. Отличительной особенностью является то, что ответом здесь может быть не только число (рис.13).

Рис. 13.

2.4. Таблицы методов

Для удобства взаимодействия между библиотеками во время работы программы была создан класс Task, содержащий в несколько методов, которые используются между несколькими библиотеками (отчего было бы неудобно их объявлять как простые глобальные переменные).

Таблица 1 - таблица свойств класса Task

Свойство	Тип	Описание
Protv	string	Соответствует правильному значению того или иного задания внутри игры. Задаётся алгоритмами из библиотеки Tasks и затем сравнивается с ответом, данным учеником.
Answ	string	Соответствует тексту решения данной логической задачи. Предварительно задаётся в BaseOfTasks, и, когда игрок решил задачу, выводится на экран.
Otv	string	Соответствует тексту ответа на задачу, даваемого учеником. Может редактироваться благодаря событийным процедурам CharAnswerInput и AnswerInput из библиотеки MouseAndKeyboardActions
Ppoint	shortint	Некое подобие флажка. Используется в основном в циклах while в случаях, когда нужно подождать, пока игрок не совершит какое-либо действие. В этом случае благодаря событийным процедурам значение этого флажка меняется на необходимое и дальше происходит заданное действие
Difficulty, typ, mode	shortint	Сложность заданий, тип заданий и режим игры соответственно. Все 3 параметра выборочно изменяются в меню по настройке параметров для игры и, соответственно, необходимы для того, чтобы задать настройки дальнейшей игры или просто чтобы посмотреть в таблице рекордов, кто лидирует в игре по данным параметрам
Column,line	shortint	Положение курсора относительно спрайтов кнопок в главном меню и меню настройки параметров игры (дальше – просто НПИ). Названы так, потому что эти самые кнопки (в НПИ, по большей мере) расположены так, что напоминают таблицу. Используются в событийных процедурах ModeChoose и ChooseSettings, отвечающих за передвижение курсора относительно экрана.
Cwork, Lwork	boolean	производные от Column и Line. Показывают, наведён ли впринципе курсор на какую-либо из “линий” и “столбцов”. При условиях Cwork=true и Lwork=true

		считается, что курсор наведён на какую-то из кнопок в главном меню или НПИ.
Ball	integer	Балл, набранный учеником на протяжении всей игры. Если по окончанию игры он оказывается рекордным, то процедурой SaveResults заносится его значение внутрь таблицы рекордов.
Name	string	Имя игрока, введённое им же в НПИ. Необходимо лишь в том случае, если данный игрок побил чей-либо рекорд (вследствие чего его балл заносится в таблицу рекордов вместе со значением этой переменной, то есть вместе с именем)
Time	real	Содержит в себе время, оставшееся до конца игры (режим “ForTime”) или время, прошедшее с начала игры (режим “NoError”. В данном случае, если балл ученика окажется рекордным, оно также заносится в таблицу рекордов).
Modename, typetaskname, difficultname	string	Название выбранных режима, типа заданий и сложности соответственно. Определяется событийной процедурой UnpressedSettings, вместе со значениями методов razmode, typ и difficulty сооответсвенно. Применяется в случае необходимости вывода данной информации на экран (таблица рекордов/игровое окно).

ВЫВОДЫ

В ходе проведённой научной работы:

1) Было выдвинуто предположение о том, что благодаря игровой визуализации наиболее «скучных» математических тем для учащихся 5-6 класса, только что вышедших из начальной школы, можно усилить интерес к предмету и, следовательно, повысить успеваемость. Поэтому в качестве темы научно-исследовательской работы была выбрана разработка компьютерной игры «Повторим математику».

2) На основании изучения базисного учебного плана по математике для 5-6-го класса были отобраны темы, которые требуют дополнительного повторения на уроках математики и во время внеурочных занятий.

3) Были исследованы возможности программной среды PascalABC.NET, выбранной в качестве платформы для разработки компьютерной игры по математике.

4) Была разработана компьютерная игра «Повторим математику», которая удовлетворила поставленной цели и задачам.

О перспективах дальнейшего развития

Полученная в результате исследовательской работы компьютерная игра может быть использована для повторения и закрепления вычислительных навыков как на уроках математики, так и во время выполнения домашней работы.

Открытый код разработанной игры дает следующие возможности:

- *Добавление новых типов задач:* будет полезно в случае низкой успеваемости учеников по какой-то конкретной теме. В этом случае к доработке игры можно будет подключить учащихся старших классов, поскольку PascalABC.NET – наиболее распространённый язык, на котором учат программированию на уроках информатики. Всё, что нужно будет уметь – это

составлять алгоритмы, которые будут генерировать случайным образом те или иные задания, и переводить их с листка в компилятор;

- *Доработка функционала:* здесь можно отметить возможность сделать процесс ещё более игровым и интересным для простого ученика (например, добавление одноразовых подсказок, которые можно будет использовать посреди игрового процесса); также по мере увеличения размеров кода и количества алгоритмов может возникнуть вопрос об оптимизации программы;
- *Переход на более профессиональную среду программирования:* поскольку первоначальная цель среды PascalABC.NET – обучение школьников и студентов азам программирования, не удивительно, что она не предназначена для создания какого-то цельного программного обеспечения. По мере необходимости можно будет перейти, например, на гораздо более распространённый в современном программировании язык C++, который обладает большим функционалом и универсальностью по сравнению с Паскалем.

СПИСОК ЛИТЕРАТУРЫ

1. Варченко, В.И. Фуксон Л.М. Особенности использования дидактических компьютерных игр при изучении математики в начальной школе / В.И. Варченко, Л.М. Фуксон. [Электронный ресурс] http://ito.edu.ru/2000/II/1/18.html.
2. БУП по математике, 5класс [Электронный ресурс] https://nsportal.ru/shkola/materialy-k-attestatsii/library/2015/06/06/uchebnaya-programma-po-matematike-5kl-vilenkin

3. БУП по математике, 6класс [Электронный ресурс] - https://nsportal.ru/shkola/algebra/library/2015/09/21/rabochaya-programma-po-matematike-6-klass-po-fgos-vilenkin-2015
4. Возможности среды программирования PascalABC.NET [Электронный ресурс] - http://fb.ru/article/255779/sreda-programmirovaniya-paskal-abc; https://ru.wikipedia.org/wiki/PascalABC.NET#Особенности_языка
5. Создание библиотек подпрограмм на основе родственной PascalABC.NET среде TurboPascal [Электронный ресурс] - https://www.examen.ru/add/manual/school-subjects/mathematics-and-computer-science/computer-science/reshenie-zadach-po-informatike,-osnovyi-programirovaniya/osnovyi-programirovaniya/sozdanie-bibliotek-podprogramm-v-turbo-pascal/
6. Ссылка на бесплатное скачивание компилятора для среды PascalABC.NET [Электронный ресурс] – http://pascalabc.net/downloads/pab

Приложение 1

Библиотека StartGame.pas

```
program StartGame;
uses MainActions;

begin
start;
end.
```

Приложение 2

Библиотека MainActions.pas

```
Unit MainActions;
interface
/// Имя и набранный балл каким-либо игроком, записанные в таблицу рекордов
Type results=record
ball: integer;
name: string;
time: string;
end;
type tasksvariables = record
      protv: string; // правильный ответ на задачи.
      answ: string; // разъяснение ответа в логической задаче
      otv: string; // ответ, вводимый учеником с клавиатуры. используется
только в процедуре answervvod; при завершении ввода ответа процедурой val
переводится в числовое значение
      ppoint: shortint; // что-то типа флажка, изменение значения к-рого
определяет дальнейшие действия программы. используется повсеместно
      difficulty: shortint; // выборная сложность в заданиях
(легко/средне/сложно/тяжело)
      typ: shortint; // тип заданий (округление/решение
примеров/уравнений/логических задач)
      mode: shortint; // режим игры (на время/без ошибок)
      column,line:shortint;   // переменные для динамичных спрайтов в
основном меню. Первые две проверяют, на какой столбец/линию наведён
курсор, а другие две показывают, наведён ли вообще на них курсор
      cwork,lwork:boolean;
      time: real; // время на таймере/секундомере в ForTime/NoError
      ball: integer; // набранный игроком балл, который, в случае чего,
потом вносится в таблицу рекордов
```

```
      name: string;                        // имя игрока
      modename: string;                    // наименование режима
      typetaskname: string;                // наименование типа заданий
      difficultname: string;               // наименование сложности
      end;
var task: tasksvariables;
type resultat=array [1..11] of results;
/// 10 позиций в т-це рекордов - массив записей
var res:resultat;
/// процедура прорисовки заднего фона
procedure BackgroundDraw;
/// Выход в главное меню игры
procedure Start;
/// Выход в меню настроек игры
procedure RazmMode;
/// Переход из меню настроек в таблицу рекордов
Procedure RecordLink;
/// Выход в режим игры "Ни единой ошибки". Его суть - набрать как можно
большее кол-во баллов за одну игру, не допустив ни единой ошибки.
procedure NoError(typet,difficulty: integer; mname,tname,dname:string);
/// Выход в режим игры "На время". Суть - за 5 минут набрать наибольшее
количество баллов. Ошибки допустимы.
procedure ForTime(typet,difficulty: integer; mname,tname,dname:string);
/// Секундомер - за какое кол-во времени игрок решил n-ое кол-во задач в
пределах NoError
procedure Stopwatch(var time: real);
/// Таймер для режима ForTime
procedure Timer(var time:real);
/// Информация по поводу разграничения типов заданий по сложности.
procedure DifficultyInfo;
/// Сохранение результатов в папку Data относительно режима и сложности,
на которых играл ученик.
procedure SaveResults(mode,typetask,difficulty: shortint);
/// Производная от SaveResults. Подбирает путь к определённому текстовому
файлу - базе данных рекордов по определенному режиму-сложности.
procedure PickDataAddress(mode,typetask,difficulty: shortint; var address:
string);
/// Производная от SaveResults. Вычленение из текстового файла кол-ва
баллов
procedure ConvertInfo(var res:resultat; var address:string);
/// Производная от SaveResults. Обратный перевод данных в текстовый файл
procedure ConvertBack(var res:resultat; var address: string);
```

```
implementation
uses GraphABC,Tasks,Graphics,Timers,Events,MouseAndKeyboardActions;

   procedure BackgroundDraw;
   begin
     var bg: picture;
     bg:=Picture.Create('C:\Игра года\Images\background.jpg');
     bg.Load('C:\Игра года\Images\background.jpg');
     bg.Draw(1,1,WindowWidth,WindowHeight);

   end;


   procedure DifficultyInfo;
   begin
    graphic(7);
    task.ppoint:=1;
    while (task.ppoint=1) do OnMouseDown:=DifficultyQuit;
    RazmMode;
   end;



   procedure Stopwatch;
   begin
       var sec, min: integer;
       var pict:=Picture.Create('Sprites\time.png');
       pict.Load('Sprites\time.png');
       Font.Size:=25;
       sleep(1000);
       time+=1;
       sec:=round(time) mod 60;
       min:=round(time) div 60;
       if (frac(time)=0) then begin
            pict.draw(Window.Width-150,Window.Height-75,150,75);
            DrawTextCentered(Window.Width-150,Window.Height-
75,Window.Width,Window.Height,concat(min.ToString,':',(sec<10?'0':''),sec.
ToString));
       end;
     end;
```

```
procedure Timer;
begin
var sec, min: integer;
var pict:=Picture.Create('Sprites\time.png');
pict.Load('Sprites\time.png');
sleep(1000);
if (time>0) then time-=1;
sec:=round(time) mod 60;
min:=round(time) div 60;
if (frac(time)=0) then begin
   if (time>0) then begin // если время >0, то каждую секунду
обновляем его значение на экране
   pict.draw(Window.Width-150,Window.Height-75,150,75);
   DrawTextCentered(Window.Width-150,Window.Height-
75,Window.Width,Window.Height,concat(min.ToString,':',(sec<10?'0':''),sec.
ToString));
   end
   else begin
   Font.Color:=clRed; // иначе - выводим "0:00" красным цветом и
замораживаем
   pict.draw(Window.Width-150,Window.Height-75,150,75);
   DrawTextCentered(Window.Width-150,Window.Height-
75,Window.Width,Window.Height,'0:00');
   Font.Color:=clBlack;
   end;
end;
end;

procedure PickDataAddress;
begin
case mode of
1: address+='Data\ForTime';
2: address+='Data\NoError';
end;
case typetask of
1: address+='\Rounding';
2: address+='\AddictionSubtraction';
3: address+='\MultiplicationDivision';
4: address+='\Equations';
5: address+='\LogicTask';
end;
case difficulty of
```

```
1: address+='\easy.txt';
2: address+='\medium.txt';
3: address+='\hard.txt';
4: address+='\veryhard.txt';
end;
end;



procedure SaveResults;
var address:string;
var maxindex,i: shortint;
begin
PickDataAddress(mode,typetask,difficulty,address); // 1. Находим тот
документ-базу, который нам нужен
ConvertInfo(res,address);                          // 2. Переводим
данные из него в массив переменных res.ball и res.name - кол-ва набранных
баллов и имени игрока из рекордов
maxindex:=0;
i:=1;
while (i<=10) and (maxindex=0) do begin
if (res[i].ball<task.ball) then begin              // 3. Перебираем
данные: если мы побили чей-то рекорд - обновляем их, ставя на место
предыдущего рекордсмена результат игрока, а всё, что ниже, сдвигаем на
позицию назад
maxindex:=i;
var p:=11;
while (p>maxindex) do begin
res[p].ball:=res[p-1].ball;
if (task.mode=2) then res[p].time:=res[p-1].time;
res[p].name:=res[p-1].name;
p-=1;
end;
res[maxindex].ball:=task.ball;
if (task.mode=2) then res[maxindex].time:=concat((round(task.time) div
60).ToString,':',((round(task.time) mod 60)<10?'0':''),(round(task.time)
mod 60).ToString);
res[maxindex].name:=task.name;
end;
i+=1;
end;
```

```
   ConvertBack(res,address);                          // 4. После того,
как перебрали данные, возвращаем их на место, обновляя документ-базу
   end;


   procedure ConvertBack;
   var f:text;
   begin
   assign(f,address);
// связываем файл с текстом
   erase(f);
// сносим имеющийся текст в виде файла на диске
   Rewrite(f);
// сносим текст в самой файловой переменной
   for var i:=1 to 10 do writeln(f,concat(res[i].ball.ToString, '
',(task.mode=2?concat(res[i].time,' '):''),res[i].name));  // вносим её в
пустой файл
   close(f);
// done.
   end;

   procedure ConvertInfo;
   var f: text;
   var s:string;
   var err: integer;
   begin
   assign(f,address);
   reset(f);
   var n:=0;
    while (not eof(f)) or (n<=10) do // пока не дойдём до конца файла -
считываем со строк балл-имя рекордсмена, каждый раз переходя на новую
     begin
      readln(f,s);
   n:=n+1;
   val(copy(s,1,pos(' ',s)-1),res[n].ball,err);
   Delete(s,1,res[n].ball.tostring.length+1);
   if (task.mode=2) then begin
   res[n].time:=copy(s,1,pos(' ',s)-1);
   Delete(s,1,res[n].time.length+1);
   end;
   res[n].name:=copy(s,1,pos(' ',s)-1);
   end;
```

```
   close(f);
   end;


   procedure RecordLink;
   var address:string;
   begin
   graphic(6);
   DrawTextCentered(75,Window.Height div 2 - 175,Window.Width div
2,Window.Height div 2 - 75,concat('Режим - ',task.modename.tostring));
   DrawTextCentered(75,Window.Height div 2 - 50,Window.Width div
2,Window.Height div 2+50,concat('Режим - ',task.typetaskname.tostring));
   DrawTextCentered(75,Window.Height div 2 + 75,Window.Width div
2,Window.Height div 2 + 175,concat('Режим -
',task.difficultname.tostring));
   PickDataAddress(task.mode,task.typ,task.difficulty,address);
   ConvertInfo(res,address);
   for var i:=1 to 10 do DrawTextCentered(Window.Width div 2 - 150, 75+(i-
1)*50,Window.Width-50,Window.Height div 2 - 75+(i-
1)*50,concat('№',i.ToString,' - ',res[i].name,';
',(task.mode=2?concat(res[i].time,'; '):''),res[i].ball.ToString,'
баллов.'));
   task.ppoint:=1;
   while (task.ppoint=1) do OnMouseDown:=RecordLinkExit;
   RazmMode;
   end;

    procedure NoError;
    type Arr=array [1..8] of shortint;
    var otv:real;
    var error: boolean;
    var err: integer;
    begin
    graphic(8);
    error:=false;
    task.time:=100;
    task.ball:=2;
    task.otv:=nil;
    while (error<>True) do begin // пока игрок не допустит ошибку - каждый
раз даём ему задания
    case typet of // в зависимости от выбранного типа задач - запускаем
определённую процедуру
```

```
    1: Rounding(task.difficulty,task.protv);
    2: AdditionSubtraction(task.difficulty,task.protv);
    3: MultiplicationDivision(task.difficulty,task.protv);
    4: Equations(task.difficulty,task.protv);
    5: LogicTask(task.difficulty,task.protv);
    end;
    Font.Size:=25;
    DrawTextCentered(Window.Width-210,-3,Window.Width+3,75,concat('Балл:
',task.ball.tostring));
    OnKeyDown:=AnswerInput; // пока игрок не нажмёт на энтер - ввод с
клавиатуры
    OnMouseDown:=GameButtons; // также - возможен выход из игры во время
неё
    OnKeyPress:=CharAnswerInput;
    task.ppoint:=0;
    while (task.ppoint<>1) and (task.ppoint<>4) do Stopwatch(task.time);
    OnKeyDown:=nil;
    if (task.ppoint=4) then break
    else
    if (task.otv<>task.protv) then error:=true // если ответ неверный -
выходим из игры
    else begin
    task.ball+=1; // если ответ правильный - +балл
    graphic(9);
    task.ppoint:=3;
    OnKeyDown:=GoNextEnter;
    while (task.ppoint=3) do OnMouseDown:=GameButtons; // пока кнопка не
нажмётся - программа не переключится на следующую задачу
    if (task.ppoint=4) then break; // кнопка выхода из игры
    Task.otv:=nil;
    graphic(8);
    end;
    end;
    if (task.ppoint<>4) then graphic(11);
    SaveResults(2,typet,task.difficulty); // сохранение рез-тов в базу
рекордов
    task.ppoint:=2;
    while (task.ppoint<>1) and (task.ppoint<>0) do OnMouseDown:=restart;
    case task.ppoint of
    1: NoError(typet,difficulty, mname, tname, dname);
    0: Razmmode();
    end;
```

```
    end;

    procedure ForTime;
    var otv:real;
    var err: integer;
    begin
        task.ppoint:=0;
        task.ball:=0;
        if (typet=5) then task.time:=900
        else task.time:=600;
        while (task.time>0) do begin // пока не закончилось время - каждый
раз крутим данный цикл
        graphic(8);
        task.ppoint:=0;
        case typet of // в зависимости от выбранного типа задач -
запускаем определённую процедуру
        1: Rounding(task.difficulty,task.protv);
        2: AdditionSubtraction(task.difficulty,task.protv);
        3: MultiplicationDivision(task.difficulty,task.protv);
        4: Equations(task.difficulty,task.protv);
        5: LogicTask(task.difficulty,task.protv);
        end;
        Font.Size:=25;
        DrawTextCentered(Window.Width-210,-
3,Window.Width+3,75,concat('Балл: ',task.ball.tostring));
        task.otv:=nil;
        OnKeyDown:=AnswerInput; // пока игрок не нажмёт на энтер - ввод с
клавиатуры
        OnKeyPress:=CharAnswerInput;
        OnMouseDown:=GameButtons;
        task.ppoint:=2;
        while (task.ppoint<>1) and (task.ppoint<>4) do Timer(task.time);
// пока игрок не захочет ответить/выйти из игры - таймер работает
        OnKeyDown:=nil;
        if (task.ppoint=4) then break // выход из игры
        else
        if (task.otv=task.protv) then begin
        task.ball+=1;
        graphic(9);
        end
        else graphic(10);
        end;
```

```
        SaveResults(1,typet,task.difficulty); // сохранение рез-тов в базу
рекордов
        if (task.ppoint<>4) then graphic(11);
        task.ppoint:=2;
        while (task.ppoint=2) do OnMouseDown:=restart;
        end;

    procedure RazmMode(); // окно выбора настроек игры
    begin
    BackgroundDraw;
    task.mode:=0; // режим
    task.typ:=0; // задания
    task.difficulty:=0; // сложность
    task.name:=nil;
    task.ppoint:=4;
    graphic(3);
    graphic(4);
    while ((task.mode=0) or (task.typ=0) or (task.difficulty=0)) and
(task.ppoint<>0) do begin // пока не будет выбрано по варианту из столбца
и не будет введено имя - кнопка 'OK'+'Таблиа рекордов' будет заблокирована
    OnMouseMove:=ChooseSettings;
    OnMouseDown:=PressedSettings;
    OnMouseUp:=UnpressedSettings;
    OnKeyPress:=NameInput;
    OnKeyDown:=NameSet;
    end;
    if (task.ppoint=0) then start();
    graphic(5);
    while (task.name=nil) and (task.ppoint=4) do
OnMouseUp:=UnpressedSettings; // пока не будет введено имя - начать игру
будет нельзя (хотя таблицу рекордов по данным настройкам можно будет
посмотреть)
    if (task.ppoint=0) then start();
    if (task.ppoint=1) then RecordLink();
    task.ppoint:=3;
    sleep(200);
    OnKeyPress:=nil;
    graphic(5);
    OnKeyPress:=NameInput;
    while (task.ppoint=3) do OnMouseUp:=UnpressedSettings; // пока игрок
не нажал на кнопку 'ок', он всё ещё может переключать настройки игры
    OnMouseMove:=nil;
```

```
    OnKeyPress:=nil;
    OnKeyDown:=nil;
    if (task.ppoint=0) then start()
    else if (task.ppoint=1) then RecordLink()
    else case task.mode of
    1:
ForTime(task.typ,task.difficulty,task.modename,task.typetaskname,task.diff
icultname);
    2:
NoError(task.typ,task.difficulty,task.modename,task.typetaskname,task.diff
icultname);
    end;
    end;

    procedure start; // по сути, основная программа. Вынес в процедуру,
дабы было удобно её вызывать каждый раз через рекурсию (в осн. проге надо
было бы использовать goto, а для меня это глаз мозолит)
    var point: boolean;
    begin
    Window.Maximize;
    graphic(1); // прорисовка стартового окна
    task.ppoint:=0; // непосредственно самы режимы этого окна
    point:=false; // флажок, который, когда становится равным true,
выходит из второго while (см. ниже), тем самым выводя игру вновь на
стартовый экран
    OnMouseMove:=nil;
    OnMouseDown:=nil;
    OnMouseUp:=nil;
    OnKeyPress:=nil;
    OnKeyDown:=nil;
    while (task.ppoint=0) do begin
    OnMouseMove:=ModeChoose;
    OnMouseDown:=PressedModeChoose;
    OnMouseUp:=UnpressedModeChoose;
    end;
    OnMouseUp:=nil;
    OnMouseMove:=nil;
    while (task.ppoint<>100) do razmmode;
    start; // перевызов текущей процедуры, т.е. в данном случае - выход на
стартовое окно
    end;
    end.
```

Приложение 3

Библиотека Tasks.pas

```
Unit Tasks;
interface
/// база заданий для логической задачи
procedure BaseOfTasks(textn: shortint);
/// задания на округление чисел
procedure Rounding(difficulty: shortint; var protv: string);
/// задания на сложение/вычитание целых/десятичных чисел
procedure AdditionSubtraction(difficulty: shortint; var protv: string);
/// задания на умножение/деление целых/десятичных чисел
procedure MultiplicationDivision(difficulty: shortint; var protv: string);
/// задания на решение простейших линейных уравнений
procedure Equations(difficulty: shortint; var protv: string);
/// логические задания
procedure LogicTask(difficulty: shortint; var protv: string);
implementation
Uses GraphABC,ABCObjects,Graphics,MainActions;


procedure BaseOfTasks;
var tasktext,answ:string;
var protv: real;
begin
case task.difficulty of
1: case textn of
    1: begin
        tasktext:='Изделие весит 89,4 грамма. Сообразите в уме, сколько
тонн весит миллион таких изделий.';
        task.protv:='89,4';
        task.answ:='Очевидно, миллион таких изделий весит 89.4*1000000
грамм = 89.4*1000 килограмм = 89.4 тонны.';
        end;
    ...
        end;
    end;
2:  case textn of
    1: begin
        tasktext:='Когда Бесс спросили, сколько ей лет, она ответила:
«Через два года я буду в два раза старше, чем была пять лет назад».
Сколько ей лет?';
        task.protv:='12';
```

```
        task.answ:='12. Действительно, пусть Бесс сейчас x лет, тогда
имеет место уравнение: x+2= 2(x-5), откуда x =12.';
        end;
   ...
   end;
 end;
 Font.Size:=10;
 DrawTextCentered(0,0,500,250,tasktext);
end;

procedure LogicTask; // вследствие вынесения в отдельный тип задач чуть
порезал её. здесь - просто генерирование задачи с вынесением правильного
ответа
var pict: picture;
var textn: shortint;
begin
randomize;
case task.difficulty of
1: textn:=random(12)+1;
2: textn:=random(6)+1;
3: textn:=random(4)+1;
4: textn:=random(1)+1;
end;
BaseOfTasks(textn); // вытаскиваем из базы всю информацию по заданию
(текст/ответ)
protv:=task.protv;
task.ppoint:=0;
pict:=Picture.Create(concat('C:\Игра
года\Images\task',task.difficulty.ToString,textn.ToString,'.jpg'));
pict.Load(concat('C:\Игра
года\Images\task',task.difficulty.ToString,textn.ToString,'.jpg'));
pict.Draw(Window.Width-1100, 300,450,300);
end;

procedure Rounding;
type number=record
      tochn: shortint; // знач-е, определяющее точность необходимого
округления
      vallue: real;    // само число. в зависимости от поля является либо
округляемым, либо округлением
      end;
var num1, num2: number;
```

```
          temp: string; // для выдачи условия (до какого числа округлять);
begin
num1.tochn:=random(task.difficulty)+1;
// задаём точность округляемого
num2.tochn:=task.difficulty+1;
// задаём точность округления. т.к. она обязательно должна быть меньше
первой (т.е. не можем 943.31 округлить до тысячных),
while (num2.tochn>num1.tochn) do num2.tochn:=random(task.difficulty);
// сначала задаём как максимальное значение num1.tochn, а затем рандомом
подбираем число, пока оно не будет меньше текущего num1.tochn
num1.vallue:=random(1000)+round(random(power(10,num1.tochn)))/power(10,num
1.tochn); // сначала рандомим целую часть, а затем прибавляем дробную, в
которой кол-во знаков после запятой зависит от выбранной сложности
num2.vallue:=round(num1.vallue*power(10,num2.tochn))/power(10,num2.tochn);
// округляем num1.tochn в зависимости от num2.tochn. Это - округлённое
значение.
temp:=(num2.tochn=0) ? 'целых: ' : concat(power(10,num2.tochn).ToString,'-
x: ');
DrawTextCentered(-3,-3,600,300,concat('Округлить до ', temp,
num1.vallue.tostring,'≈...'));
task.protv:=num2.vallue.ToString;
end;

procedure AdditionSubtraction;
var AorS: boolean;
    fst, snd: real;
begin
if (random(2)=0) then AorS:=true else AorS:=false; // рандомный выбор
того, что будет производиться в примере - вычитание или разность
if (difficulty<3) then begin                        // если сложность 1я/2я
- рандомное число до 100/1000 соответственно
fst:=round(random(power(10,difficulty+1)));
snd:=round(random(power(10,difficulty+1)));
end
else begin
// иначе:
if (random(2)=0) then begin
// 1) - числа до 100 с 1м/2мя цифрами после запятой соответственно
fst:=round(random(100)*exp(ln(10)*(difficulty-2))/power(10,difficulty-2));
snd:=round(random(100)*exp(ln(10)*(difficulty-2))/power(10,difficulty-2));
end
```

```
else begin
// 2) числа до 1000/10000 соответственно
fst:=round(random(power(10,difficulty+1)));
snd:=round(random(power(10,difficulty+1)));
end;
end;
Font.size:=40;
DrawTextCentered(-3,-3,600,300,concat(fst.tostring,(AorS?'+':'-
'),snd.tostring,'=...'));
if (AorS=true) then task.protv:=(fst+snd).ToString
else task.protv:=(fst-snd).ToString;
end;

procedure MultiplicationDivision;
var MorD: boolean;
    fst, snd: real;
begin
if (random(2)=0) then MorD:=true else MorD:=false;  // так же, как и в
прошлой процедуре, рандомим умножение/деление чисел
if (difficulty<3) then begin                        // при 1й или 2й
сложности - умножение/деление чисел до 10 или 100 соотв-но
fst:=round(random(power(10,difficulty)));
snd:=round(random(power(10,difficulty)));
if (MorD=false) then fst-=round(frac(fst/snd)*snd); // на 1 и 2й сложности
делимое кратно делителю
end
else begin
// иначе:
if (random(2)=0) then begin
// 1) умножение/деление больших чисел (до 1000 и 10000 соот-но.)
fst:=round(random(power(10,difficulty)));
snd:=round(random(100));
end
else begin
// 2) обе операции с десятичными числами до 10 и 100, 1/2мя знаками после
запятой
fst:=round(random(power(10,difficulty-2))*(difficulty-2))/(difficulty-2);
snd:=round(random(power(10,difficulty-2))*(difficulty-2))/(difficulty-2);
end;
end;
DrawTextCentered(-3,-
3,600,300,concat(fst.tostring,(MorD?'*':'/'),snd.tostring,'=...'));
```

```
Rectangle(-3,Window.Height-100,300,Window.Height+3);
Font.Size:=15;
DrawTextCentered(-3,Window.Height-100,300,Window.Height+3,'При делении результат округлить до сотых, дробную часть отделить точкой.'); // совет по вводу ответа
task.protv:=((MorD?(fst*snd):(round(fst/snd*100)/100))).ToString;
end;

procedure Equations;
var k,a,b: real;
begin
case difficulty of
1: begin
k:=round(random(-30,30)); // рандомим коэффициенты для уравнения, не забывая про то, что могут быть и отриц. зн-я
a:=round(random(-100,100));
b:=round(random(-100,100));
end;

2: begin
k:=round(random(-30,30));
a:=round(random(-1000,1000));
b:=round(random(-1000,1000));
end;

3: begin
k:=round(random(-30,30)*10)/10;
a:=round(random(-30,30)*10)/10;
b:=round(random(-30,30)*10)/10;
end;

4: begin
k:=round(random(-100,100)*100)/100;
a:=round(random(-100,100)*100)/100;
b:=round(random(-100,100)*100)/100;
end;
end;
b-=frac((b-a)/k)*k;
DrawTextCentered(-3,-3,600,300,concat(k.tostring,'*x',(a>0?'+':''),a.tostring,'=',b.tostring,'?'));
task.protv:=(round((b-a)/k*100)/100).ToString;
```

```
Font.Size:=15;
Rectangle(-3,Window.Height-100,300,Window.Height+3);
DrawTextCentered(-3,Window.Height-100,300,Window.Height+3,'Дробную часть в
ответе отделить точкой.');
end;
end.
```

Приложение 4

Библиотека MouseAndKeyboardActions.pas

```
Unit MouseAndKeyboardActions;
interface
// три процедуры для курсора в главном меню: вождения по экрану, нажатия и
отжатия ЛКМ на опред-м спрайте:
/// Смена спрайтов кнопок при вождении по ним курсором на стартовом экране
procedure ModeChoose(x,y,mb:integer);
/// Смена спрайтов кнопок при нажатии на них на стартовом экране
procedure PressedModeChoose(x,y,mb:integer);
/// Смена спрайтов кнопок + выполнение определённых действий при их
отжатии на стартовом экране
procedure UnpressedModeChoose(x,y,mb:integer);
// Аналогичные процедуры, но уже для курсора, находящегося в меню выбора
настроек игры:
/// Смена спрайтов кнопок при вождении по ним курсором на экране выбора
настроек игры
procedure ChooseSettings(x,y,mb:integer);
/// Смена спрайтов кнопок при нажатии на них на экране выбора настроек
игры
procedure PressedSettings(x,y,mb:integer);
/// Смена спрайтов кнопок + выполнение определённых действий при их
отжатии на экране выбора настроек игры
procedure UnpressedSettings(x,y,mb:integer);
/// управление кнопками на экране во время игры (выход на главное
меню/смена задачи в NoError)
procedure GameButtons(x,y,mb:integer);
/// процедура в режиме NoError, которая по нажатию Enter на экране меняет
задачу после решения предыдущей
procedure GoNextEnter(key:integer);
///диалоговое окно рестарта/выхода в меню/из программы после проигрыша в
игре
procedure Restart(x,y,mb:integer);
///процедура, отвечающая за кнопку выхода из окна с объяснениями по
разграничению сложностей в режимах
procedure DifficultyQuit(x,y,mb:integer);
///закрытие информационного окна в главном меню
procedure ExitGameInfo(x,y,mb:integer);
/// выход из таблицы рекордов
procedure RecordLinkExit(x,y,mb:integer);
/// удаление бэкспейсом символов из имени в меню настройки
procedure NameSet(key:integer);
```

```
/// ввод символами имени в меню настройки
procedure NameInput(ch:char);
/// ввод игроком ответа в процессе самой игры
procedure AnswerInput(key:integer);
/// символьный ввод ответа внутри игры
procedure CharAnswerInput(ch:char);
implementation
uses GraphABC,MainActions,Graphics;


procedure ModeChoose;
   begin
        task.lwork:=true;
        if (y>Window.Height div 2 - 250) and (y<Window.Height div 2 - 110)
then task.line:=1
        else if (y>Window.Height div 2 - 70) and (y<Window.Height div 2 +
70) then task.line:=2
        else task.lwork:=false;

        if (x>Window.Width div 2 - 250) and (x<Window.Width div 2 + 250)
then task.cwork:=true
        else task.cwork:=false;

        if (task.line>=1) and (task.line<=3) then begin
            if (mb=0) and (task.cwork=true) and (task.lwork=true) then
ButtonSprite((task.line=1?1:4),1,task.line,Window.Width div 2 -
250,Window.Height div 2 - 250+(task.line-1)*180,500,140)
            else ButtonSprite((task.line=1?1:4),0,task.line,Window.Width
div 2 - 250,Window.Height div 2 - 250+(task.line-1)*180,500,140);
            end;
    end;

    procedure PressedModeChoose;
    begin
       if (mb=1) and (task.lwork=true) and (task.cwork=true) and
(task.line<>0) then
ButtonSprite((task.line=1?1:4),2,task.line,Window.Width div 2 -
250,Window.Height div 2 - 250+(task.line-1)*180,500,140);
    end;

    procedure UnpressedModeChoose;
    begin
```

```
        if (mb=1) and (task.lwork=true) and (task.cwork=true) and
(task.line<>0) then begin
          ButtonSprite((task.line=1?1:4),1,task.line,Window.Width div 2 -
250,Window.Height div 2 - 250+(task.line-1)*180,500,140);
          if (task.line=1) then task.ppoint:=1
          else if (task.line=2) then halt;
          end;
        end;


procedure ChooseSettings; // выбор настроек в игре, v.1. вторая версия в
program1
begin
      task.cwork:=true; // флажок, обозначающий, наведён ли курсор на
какой-либо столбец (0 - не наведён ни на один из них)
        if (x>Window.Width div 2 - 550) and (x<Window.Width div 2 - 250)
then task.column:=1
        else if (x>Window.Width div 2 - 150) and (x<Window.Width div 2 +
250) then task.column:=2
        else if (x>Window.Width div 2 + 350) and (x<Window.Width div 2 +
600) then task.column:=3
        else task.cwork:=false;
      task.lwork:=true; // флажок, обозначающий, наведён ли курсор на
какую-либо линию (0 - не наведён ни на одну из них)
        if (y>Window.Height div 2 - 225) and (y<Window.Height div 2 - 150)
then task.line:=1
        else if (y>Window.Height div 2 - 125) and (y<Window.Height div 2 -
50) then task.line:=2
        else if (y>Window.Height div 2 - 25) and (y<Window.Height div 2 +
50) then task.line:=3
        else if (y>Window.Height div 2 + 75) and (y<Window.Height div 2 +
150) then task.line:=4
        else if (y>Window.Height div 2 + 175) and (y<Window.Height div 2 +
250) then task.line:=5
        else task.lwork:=false;

      if (task.ppoint<>1) then begin
        if (mb=0) then begin // кнопка мыши не нажата
          if ((task.column=1) and (task.line<=2)) then begin // если
наведён на какой-то столбец и линию (проще говоря, на одну из кнопок) -
спрайт определённой кнопки становится чуть темнее
```

```
              if (task.lwork=true) and (task.cwork=true) then
ButtonSprite(task.column,1,task.line+3,Window.Width div 2 -
550,Window.Height div 2 - 225+(task.line-1)*100,300,75)
              else ButtonSprite(task.column,0,task.line+3,Window.Width div
2 - 550,Window.Height div 2 - 225+(task.line-1)*100,300,75);
              end
            else if (task.column=2) then begin
              if (task.lwork=true) and (task.cwork=true) then
ButtonSprite(task.column,1,task.line+5,Window.Width div 2 -
150,Window.Height div 2 - 225+(task.line-1)*100,400,75)
              else ButtonSprite(task.column,0,task.line+5,Window.Width div
2 - 150,Window.Height div 2 - 225+(task.line-1)*100,400,75);
              end
            else if (task.column=3) and (task.line<=4) then begin
              if (task.lwork=true) and (task.cwork=true) then
ButtonSprite(task.column,1,task.line+10,Window.Width div 2 +
350,Window.Height div 2 - 225+(task.line-1)*100,250,75)
              else ButtonSprite(task.column,0,task.line+10,Window.Width
div 2 + 350,Window.Height div 2 - 225+(task.line-1)*100,250,75);
              end
            end;
        end;
   end;

   procedure PressedSettings;
   begin
   if (mb=1) and (task.lwork=true) and (task.cwork=true) then begin
      if (task.column=1) then
ButtonSprite(task.column,2,task.line+3,Window.Width div 2 -
550,Window.Height div 2 - 225+(task.line-1)*100,300,75)
      else if (task.column=2) then
ButtonSprite(task.column,2,task.line+5,Window.Width div 2 -
150,Window.Height div 2 - 225+(task.line-1)*100,400,75)
      else if (task.column=3) then
ButtonSprite(task.column,2,task.line+10,Window.Width div 2 +
350,Window.Height div 2 - 225+(task.line-1)*100,250,75);
      end;
   end;


   procedure UnpressedSettings;
   begin
```

```
        if (mb=1) then begin // если кнопка мыши нажата - спрайт
становится ещё более тёмным, НО действия по выбору настроек никакого не
происходит
          if (task.lwork=true) and (task.cwork=true) then begin
            if (task.column=1) and (task.line<=2) then begin
                  Brush.Color:=clWhite;
                  Circle(Window.Width div 2 - 215,Window.Height div 2 -
190+(task.mode=0?0:(task.mode-1))*100,25);
                  if (task.line=1) then begin task.mode:=1;
task.modename:='На время'; end
                  else if (task.line=2) then begin
task.mode:=2;task.modename:='Ни единой ошибки'; end;
                  ButtonSprite(task.column,1,task.line+3,Window.Width div
2 - 550,Window.Height div 2 - 225+(task.line-1)*100,300,75);
              end
            else if (task.column=2) then begin
                Brush.Color:=clWhite;
                Circle(Window.Width div 2 + 285,Window.Height div 2 -
190+(task.typ=0?0:(task.typ-1))*100,25);
                if (task.line=1) then begin task.typ:=1;
task.typetaskname:='Округление'; end
                else if (task.line=2) then begin task.typ:=2;
task.typetaskname:='Сложение/Вычитание'; end
                else if (task.line=3) then begin task.typ:=3;
task.typetaskname:='Деление/Умножение'; end
                else if (task.line=4) then begin task.typ:=4;
task.typetaskname:='Уравнения'; end
                else if (task.line=5) then begin task.typ:=5;
task.typetaskname:='На логику'; end;
                ButtonSprite(task.column,1,task.line+5,Window.Width div 2
- 150,Window.Height div 2 - 225+(task.line-1)*100,400,75);
              end
            else if (task.column=3) and (task.line<=4) then begin
                 Brush.Color:=clWhite;
                 Circle(Window.Width div 2 + 635,Window.Height div 2 -
190+(task.difficulty=0?0:(task.difficulty-1))*100,25);
                 if (task.line=1) then begin task.difficulty:=1;
task.difficultname:='Легко'; end
                 else if (task.line=2) then begin task.difficulty:=2;
task.difficultname:='Средне'; end
                 else if (task.line=3) then begin task.difficulty:=3;
task.difficultname:='Трудно'; end
```

```
                else if (task.line=4) then begin task.difficulty:=4;
task.difficultname:='Тяжело'; end;
                ButtonSprite(task.column,1,task.line+10,Window.Width div
2 + 350,Window.Height div 2 - 225+(task.line-1)*100,250,75);
            end;
            end

          else if (x>Window.Width div 2 - 625) and (x<Window.Width div 2 -
550) and (y>Window.Height div 2 - 325) and (y<Window.Height div 2 - 250)
then task.ppoint:=0
          else if (x>Window.Width div 2 + 350) and (x<Window.Width div 2 +
500) and (y>Window.Height div 2 + 275) and (y<Window.Height div 2 + 350)
then task.ppoint:=1
          else if (x>Window.Width div 2 + 200) and (x<Window.Width div 2 +
300) and (y>Window.Height div 2 + 250) and (y<Window.Height div 2 + 325)
then task.ppoint:=2
          else if (x>Window.Width div 2 + 500) and (x<Window.Width div 2 +
560) and (y>Window.Height div 2 - 300) and (y<Window.Height div 2 - 240)
then DifficultyInfo;
          end;
        end;

   procedure GoNextEnter;
   begin
   if (key=13) then task.ppoint:=2;
   end;

   procedure Restart;
   begin
   if (y>420) and (y<480) and (mb=1) then begin
         if (x>Window.Width div 2 - 250) and (x<Window.Width div 2 - 100)
then task.ppoint:=1
         else if ((x>Window.Width div 2 - 75) and (x<Window.Width div 2 +
75)) or (task.ppoint=4) then task.ppoint:=0
         else if (x>Window.Width div 2 + 100) and (x<Window.Width div 2 +
250) then halt;
   end;
   end;

    procedure DifficultyQuit;
   begin
```

```
    if (x>100) and (x<160) and (y>60) and (y<100) and (mb=1) then
task.ppoint:=0;
   end;

   procedure ExitGameInfo;
    begin
    if (x>Window.Width div 2 - 500) and (x<Window.Width div 2 - 400) and
(y>Window.Height div 2 + 275) and (y<Window.Height div 2 + 325) then
task.ppoint:=1;
    end;

    procedure GameButtons;
    begin
    if (mb=1) then begin
      if (x>25) and (x<100) and (y>Window.Height div 2) and
(y<Window.Height div 2+75) then task.ppoint:=4
      else if (x>Window.Width div 2+300) and (x<Window.Width div 2+500)
and (y>Window.Height-100) and (y<Window.Height+3) and (task.ppoint=3) then
task.ppoint:=2;
      end;
    end;

   procedure RecordLinkExit;
   begin
   if (x>Window.Width-125) and (x<Window.Width-75) and (y>Window.Height-
125) and (y<Window.Height-75) then task.ppoint:=0;
   end;


    procedure AnswerInput; // ввод с клавиатуры. на нампад не реагирует -
только цифры с основной клавиатуры. стирать бэкспейсом; подтвердить, что
набранный ответ верный - энтером
    begin
    if (key>=48) and (key<=57) then task.otv+=(key-48).ToString
    else if (key=188) then task.otv+=','
    else if (key=189) then task.otv+='-'
    else if (key=8) then Delete(task.otv,task.otv.Length,1)
    else if (key=13) then task.ppoint:=1;
    if (key=8) or (key=13) or (key=189) or (key>=48) or (key<=57) then
begin
    Font.Color:=clBlack;
    Font.Style:=fsBoldItalic;
```

```
    Font.Size:=25;
    Brush.Color:=clWhite;
    Rectangle(Window.Width div 2 - 200,Window.Height-100,Window.Width div
2 + 200,Window.Height+3);
    TextOut(Window.Width div 2 - 190,Window.Height-95,'Ответ:');
    DrawTextCentered(Window.Width div 2 - 200,Window.Height-
75,Window.Width div 2 + 200,Window.Height+3,task.otv.tostring);
    end;
    end;

    procedure CharAnswerInput;
    begin
    if (ord(ch)<>8)  and ((ord(ch)<48) or (ord(ch)>57)) and (ord(ch)<>13)
and (ord(ch)<>188) and (ord(ch)<>189) then begin
    task.otv+=ch;
    Brush.Color:=clWhite;
    Rectangle(Window.Width div 2 - 200,Window.Height-100,Window.Width div
2 + 200,Window.Height+3);
    TextOut(Window.Width div 2 - 190,Window.Height-95,'Ответ:');
    DrawTextCentered(Window.Width div 2 - 200,Window.Height-
75,Window.Width div 2 + 200,Window.Height+3,task.otv);
    end;
    end;

   procedure NameInput;
   begin
   if (ord(ch)<>8) then begin
   task.name+=ch;
   if (task.ppoint<>1) then begin
   LockDrawing;
   Brush.Color:=clWhite;
   Rectangle(Window.Width div 2 -150, Window.Height div 2 + 295,
Window.Width div 2 + 100, Window.Height div 2 + 330);
   DrawTextCentered(Window.Width div 2 -150, Window.Height div 2 + 295,
Window.Width div 2 + 100, Window.Height div 2 + 330,task.name);
   Redraw;
   UnlockDrawing;
   end;
   end;
   end;

   procedure NameSet;
```

```
   begin
   if (key=8) then begin
   Delete(task.name,task.name.Length,1);
   Brush.Color:=clWhite;
   Rectangle(Window.Width div 2 -150, Window.Height div 2 + 295,
Window.Width div 2 + 100, Window.Height div 2 + 330);
   DrawTextCentered(Window.Width div 2 -150, Window.Height div 2 + 295,
Window.Width div 2 + 100, Window.Height div 2 + 330,task.name);
   end;
   end;
   end.
```

Приложение 5

Библиотека Graphics.pas

```
Unit Graphics;
interface
/// Прорисовка графики в процессе игры:
/// 1 - стартовое окно
/// 3 - выбор настроек игры (сложность/режим/тип заданий)
/// 4 - заштрихованные (то есть недоступные, пока не будут выбраны все
параметры) кнопки подтверждения настроек для игры/просмотра таблицы
рекордов
/// 5 - те же кнопки, но уже цветные, т.е. когда к ним уже имеется доступ
/// 6 - окошко с таблицей рекордов. вызывается из окна по выбору
параметров игры
/// 7 - информационное окошко, толкующее о том, какие задания на какой
сложности будут генерироваться
/// 8 - окно самой игры (показывается сам вопрос, окошко с вводом ответа,
кол-во баллов и время, которое уходит на решение/которое осталось до конца
игры)
/// 9 - взаимодействие с окошком игры после того, как игрок дал правильный
ответ (смена балла, "мигание" ответа и т.п.)
/// 10 - мигание ответа красным, если он неверный (эксклюзивно для
ForTime)
/// 11 - вывод диалогового окошка об окончании игры
procedure Graphic(index: shortint);
/// Для меню настроек игры. Показывает, какой параметр на данный момент
выбран, в виде кружков около них.
procedure Confirm;
/// Прорисовка спрайтов кнопок внутри игры
procedure ButtonSprite(color,status,text: shortint; x,y,w,h:integer);
implementation
Uses GraphABC,MainActions,Tasks,ABCObjects;
procedure Graphic;
begin
LockDrawing;
case index of
1: begin
    BackgroundDraw;
    var pict:=Picture.Create('Sprites\10.png');
    pict.load('Sprites\10.png');
    pict.Draw(Window.Width div 2 - 250, Window.Height div 2 -
250,500,140);
    pict.Load('Sprites\40.png');
```

```
    pict.Draw(Window.Width div 2 - 250, Window.Height div 2 - 70,500,140);
    Font.Size:=25;
    Font.Style:=fsBold;
    DrawTextCentered(Window.Width div 2 - 250, Window.Height div 2 - 250,
Window.Width div 2 + 250, Window.Height div 2 - 110, 'Начать игру');
    DrawTextCentered(Window.Width div 2 - 250, Window.Height div 2 - 70,
Window.Width div 2 + 250, Window.Height div 2 + 70, 'Выход');
        Redraw;
    end;
3:  begin
    Font.Style:=fsBoldItalic;
    Font.Size:=20;
    Font.Color:=clBlack;
    var pict:=Picture.Create('Sprites\10.png');
    pict.Load('Sprites\10.png');
    pict.Draw(Window.Width div 2 - 550,Window.Height div 2 - 225,300,75);
    pict.Draw(Window.Width div 2 - 550,Window.Height div 2 - 125,300,75);
    pict.Load('Sprites\20.png');
    pict.Draw(Window.Width div 2 - 150,Window.Height div 2 - 225,400,75);
    pict.Draw(Window.Width div 2 - 150,Window.Height div 2 - 125,400,75);
    pict.Draw(Window.Width div 2 - 150,Window.Height div 2 - 25,400,75);
    pict.Draw(Window.Width div 2 - 150,Window.Height div 2 + 75,400,75);
    pict.Draw(Window.Width div 2 - 150,Window.Height div 2 + 175,400,75);
    pict.Load('Sprites\30.png');
    pict.Draw(Window.Width div 2 + 350,Window.Height div 2 - 225,250,75);
    pict.Draw(Window.Width div 2 + 350,Window.Height div 2 - 125,250,75);
    pict.Draw(Window.Width div 2 + 350,Window.Height div 2 - 25,250,75);
    pict.Draw(Window.Width div 2 + 350,Window.Height div 2 + 75,250,75);
    pict.Load('Sprites\back.png');
    pict.Draw(Window.Width div 2 - 625,Window.Height div 2 - 325,75,75);
    pict.Load('Sprites\infobutton.png');
    pict.Draw(Window.Width div 2 + 500,Window.Height div 2 - 300,60,60);
    DrawTextCentered(Window.Width div 2 - 550,Window.Height div 2 -
225,Window.Width div 2 - 250,Window.Height div 2 - 150,'На время');
    DrawTextCentered(Window.Width div 2 - 550,Window.Height div 2 -
125,Window.Width div 2 - 250,Window.Height div 2 - 50,'Ни единой ошибки');
    DrawTextCentered(Window.Width div 2 - 150,Window.Height div 2 -
225,Window.Width div 2 + 250,Window.Height div 2 - 150,'Округление');
    DrawTextCentered(Window.Width div 2 - 150,Window.Height div 2 -
125,Window.Width div 2 + 250,Window.Height div 2 -
50,'Сложение/Вычитание');
```

```
    DrawTextCentered(Window.Width div 2 - 150,Window.Height div 2 -
25,Window.Width div 2 + 250,Window.Height div 2 + 50,'Деление/Умножение');
    DrawTextCentered(Window.Width div 2 - 150,Window.Height div 2 +
75,Window.Width div 2 + 250,Window.Height div 2 + 150,'Уравнения');
    DrawTextCentered(Window.Width div 2 - 150,Window.Height div 2 +
175,Window.Width div 2 + 250,Window.Height div 2 + 250,'На логику');
    DrawTextCentered(Window.Width div 2 + 350,Window.Height div 2 -
225,Window.Width div 2 + 600,Window.Height div 2 - 150,'Легко');
    DrawTextCentered(Window.Width div 2 + 350,Window.Height div 2 -
125,Window.Width div 2 + 600,Window.Height div 2 - 50,'Средне');
    DrawTextCentered(Window.Width div 2 + 350,Window.Height div 2 -
25,Window.Width div 2 + 600,Window.Height div 2 + 50,'Трудно');
    DrawTextCentered(Window.Width div 2 + 350,Window.Height div 2 +
75,Window.Width div 2 + 600,Window.Height div 2 + 150,'Тяжело');
    Brush.Color:=clWhite;
    Pen.Width:=3;
    for var i:=1 to 2 do Circle(Window.Width div 2 - 215,Window.Height div
2 - 190+100*(i-1),25);
    for var i:=1 to 5 do Circle(Window.Width div 2 + 285,Window.Height div
2 - 190+100*(i-1),25);
    for var i:=1 to 4 do Circle(Window.Width div 2 + 635,Window.Height div
2 - 190+100*(i-1),25);
    Brush.Color:=clGray;
    Rectangle(Window.Width div 2 - 500, Window.Height div 2 + 275,
Window.Width div 2 + 150, Window.Height div 2 + 350);
    DrawTextCentered(Window.Width div 2 - 500, Window.Height div 2 + 275,
Window.Width div 2 - 250, Window.Height div 2 + 350,'Введите ваше имя: ');
    Brush.Color:=clWhite;
    Rectangle(Window.Width div 2 -150, Window.Height div 2 + 295,
Window.Width div 2 + 100, Window.Height div 2 + 330);
        Redraw;
    end;
4: begin
    Brush.Style:=bsHatch;
    Brush.Hatch:=bhForwardDiagonal;
    Pen.Width:=3;
    RoundRect(Window.Width div 2 + 200,Window.Height div 2 +
275,Window.Width div 2 + 300,Window.Height div 2 + 350,10,10);
    DrawTextCentered(Window.Width div 2 + 200,Window.Height div 2 +
275,Window.Width div 2 + 300,Window.Height div 2 + 350,'OK');
    RoundRect(Window.Width div 2 + 350, Window.Height div 2 + 275,
Window.Width div 2 + 500, Window.Height div 2 + 350,10,10);
```

```
    DrawTextCentered(Window.Width div 2 + 350, Window.Height div 2 + 275,
Window.Width div 2 + 500, Window.Height div 2 + 350,'Таблица рекордов');
    Brush.Style:=bsSolid;
        Redraw;
    end;
5: begin
    Pen.Width:=3;
    if (task.ppoint=3) then begin
    Brush.Color:=clGreen;
    RoundRect(Window.Width div 2 + 200,Window.Height div 2 +
275,Window.Width div 2 + 300,Window.Height div 2 + 350,10,10);
    DrawTextCentered(Window.Width div 2 + 200,Window.Height div 2 +
275,Window.Width div 2 + 300,Window.Height div 2 + 350,'ОК');
    end
    else begin
    Brush.Color:=clYellow;
    RoundRect(Window.Width div 2 + 350, Window.Height div 2 + 275,
Window.Width div 2 + 500, Window.Height div 2 + 350,10,10);
    DrawTextCentered(Window.Width div 2 + 350, Window.Height div 2 + 275,
Window.Width div 2 + 500, Window.Height div 2 + 350,'Таблица рекордов');
    Brush.Color:=clWhite;
    end;
        Redraw;
    end;
6: begin
    BackgroundDraw;
    Brush.Color:=clGray;
    Rectangle(50, 50, Window.Width - 50, Window.Height - 50);
    Brush.Color:=clGreen;
    RoundRect(Window.Width-125,Window.Height-125,Window.Width-
75,Window.Height-75,10,10);
    DrawTextCentered(Window.Width-125,Window.Height-125,Window.Width-
75,Window.Height-75,'ОК');
    Brush.Color:=clWhite;
    Font.Style:=fsBoldUnderline;
    Font.Size:=30;
    DrawTextCentered(50,50, Window.Width - 50, Window.Height div 2 -
200,'Рекорд очков по данному режиму');
    Font.Style:=fsItalic;
    Font.Size:=20;
        Redraw;
    end;
```

```
7: begin
    BackgroundDraw;
    Brush.Color:=clGray;
    Rectangle(50,50,Window.Width-50,Window.Height-50);
    Font.Style:=fsBoldItalic;
    Font.Size:=25;
    DrawTextCentered(50,50,Window.Width-50,100,'Разграничения по
сложности');
    Font.Style:=fsItalic;
    Font.Size:=15;
    TextOut(75,110,'Округление: чем выше уровень - тем больше цифр после
запятой.');
    TextOut(75,150,concat('Сложение/вычитание: ',#10,'- Легко/Средне -
операции с целыми числами до 100 и 1000 соответственно;',#10,'-
Сложно/Тяжело - операции с целыми числами до 1000/10000 соответственно ИЛИ
операции с десятичными дробями до 100 с',#10,'одним/двумя знаками после
запятой'));
    TextOut(75,260,concat('Деление/умножение: ',#10,'- Легко/Средне -
операции с целыми числами до 100 и 500 соответственно. При делении делимое
всегда кратно делителю;',#10,'- Сложно/Тяжело - операции с целыми числами
до 1000 и 10000 соответственно ИЛИ с десятичными дробями до 10 или 100
с',#10,'1/2мя знаками после запятой соответственно;'));
    TextOut(75,370,concat('Уравнение: ',#10,'Легко/Средне - решение
уравнений с целыми числами до 100 и 1000 соответственно;', #10,
'Сложно/Тяжело - присутствуют десятичные дроби до 10 и 100
соответственно.',#10,'P.S.: во всех случаях деление происходит на кратные
числа.'));
    TextOut(75,480,concat('Логика: задачи сортируются в порядке
возрастания сложности.'));
    Font.Style:=fsNormal;
    Brush.Color:=clGreen;
    Rectangle(100,60,160,100);
    DrawTextCentered(100,60,160,100,'OK');
        Redraw;
    end;
8: begin
    BackgroundDraw;
    var pict:=Picture.Create('Sprites\question.png');
    pict.Load('Sprites\question.png');
    pict.Draw(-50,-50,650,350);
    pict.Load('Sprites\ball.png');
    pict.draw(Window.Width-210,-3,210,75); //кол-во баллов
```

```
    pict.Load('Sprites\time.png');
    pict.draw(Window.Width-150,Window.Height-75,150,75); // время, которое
ушло на решение заданий
    DrawTextCentered(Window.Width-150,Window.Height-
75,Window.Width,Window.Height,concat((round(task.time) div
60).ToString,':',((round(task.time) mod 60)<10?'0':''),(round(task.time)
mod 60).ToString));
    pict.Load('Sprites\back.png');
    pict.Draw(25,Window.Height div 2,75,75);
    pict.Load('Sprites\modeinfo.png');
    pict.Draw(Window.Width-200,100,225,75);
    DrawTextCentered(Window.Width-200,100,Window.Width,175,task.modename);
    pict.Draw(Window.Width-350,200,375,75);
    DrawTextCentered(Window.Width-
350,200,Window.Width,275,task.typetaskname);
    pict.Draw(Window.Width-200,300,225,75);
    DrawTextCentered(Window.Width-
200,300,Window.Width,375,task.difficultname);
    Brush.Color:=clWhite;
    Font.Size:=25;
    Font.Style:=fsBoldItalic;
    Font.Color:=clBlack;
    Rectangle(Window.Width div 2 - 200,Window.Height-100,Window.Width div
2 + 200,Window.Height+3);
    TextOut(Window.Width div 2 - 190,Window.Height-95,'Ответ:');
        Redraw;
    end;
9: begin // табличка с кол-вом баллов
    UnlockDrawing;
    var pict:=Picture.Create('Sprites\ball.png');
    pict.Load('Sprites\ball.png');
    pict.draw(Window.Width-210,-3,210,78);
    DrawTextCentered(Window.Width-210,-3,Window.Width,75,concat('Балл:
',task.ball.ToString));
    Font.Color:=clGreen;
    Font.Size:=25;
    Brush.Color:=clWhite;
    for var i:=1 to 5 do begin // моргающий зелёным цветом правильно
введённый ответ
    sleep(200);
    Rectangle(Window.Width div 2 - 200,Window.Height-100,Window.Width div
2 + 200,Window.Height+3);
```

```
    sleep(200);
    TextOut(Window.Width div 2 - 190,Window.Height-95,'Ответ:');
    DrawTextCentered(Window.Width div 2 - 200,Window.Height-
75,Window.Width div 2 + 200,Window.Height+3,task.otv.tostring);
    end;
    if (task.mode=2) then begin
    Brush.Color:=clGreen;
    Font.Color:=clBlack;
    Rectangle(Window.Width div 2+300,Window.Height-100,Window.Width div
2+500,Window.Height+3);
    DrawTextCentered(Window.Width div 2+300,Window.Height-100,Window.Width
div 2+500,Window.Height+3,'OK'); // после того, как ответ "померцал" -
появляется кнопка. (только в NoError)
    end;
    Font.Color:=clBlack;
    end;
10: begin
    UnlockDrawing;
    Font.Color:=clRed;
    for var i:=1 to 5 do begin
    sleep(200);
    Rectangle(Window.Width div 2 - 200,Window.Height-100,Window.Width div
2 + 200,Window.Height+3);
    sleep(200);
    TextOut(Window.Width div 2 - 190,Window.Height-95,'Ответ:');
    DrawTextCentered(Window.Width div 2 - 200,Window.Height-
75,Window.Width div 2 + 200,Window.Height+3,task.otv);
    end;
    Font.Color:=clBlack;
        Redraw;
    end;
11: begin
    var pict:=Picture.Create('Images\error.png');
    pict.Load('Images\error.png');
    pict.Draw(0,0,Window.Width,Window.Height);
    Brush.Style:=bsClear;
    Brush.Hatch:=bhBackwardDiagonal;
    Rectangle(-10,-10,Window.Width+10,Window.Height+10);
    Brush.Style:=bsSolid;
    Brush.Color:=clGray;
    Font.Size:=20;
```

```
    Rectangle(Window.Width div 2 - 300, Window.Height div 2-150,
Window.Width div 2 + 300, Window.Height div 2+150);
    Font.Style:=fsBoldUnderline;
    DrawTextCentered(Window.Width div 2 - 300,Window.Height div 2-
140,Window.Width div 2 + 300,Window.Height div 2-90,'Игра окончена!');
    Font.Style:=fsBold;
    DrawTextCentered(Window.Width div 2 - 300,Window.Height div 2-
75,Window.Width div 2 + 300,Window.Height div 2-50,concat('Набранное кол-
во баллов: ',task.ball.ToString));
    if (task.mode=2) then DrawTextCentered(Window.Width div 2 -
300,Window.Height div 2-25,Window.Width div 2 + 300,Window.Height div
2,concat('Время: ',(round(task.time) div
60).ToString,':',(round(task.time) mod 60<10?'0':''),(round(task.time) mod
60).ToString));
    Brush.Color:=clGreen;
    Rectangle(Window.Width div 2 - 250,Window.Height div 2+75,Window.Width
div 2-100,Window.Height div 2+125);
    Brush.Color:=clOrange;
    Rectangle(Window.Width div 2 - 75,Window.Height div 2+75,Window.Width
div 2+75,Window.Height div 2+125);
    Brush.Color:=clRed;
    Rectangle(Window.Width div 2 + 100,Window.Height div 2+75,Window.Width
div 2 + 250,Window.Height div 2+125);
    Font.Style:=fsBold;
    DrawTextCentered(Window.Width div 2 - 250,Window.Height div
2+75,Window.Width div 2-100,Window.Height div 2+125,'РЕСТАРТ');
    DrawTextCentered(Window.Width div 2 - 75,Window.Height div
2+75,Window.Width div 2+75,Window.Height div 2+125,'В МЕНЮ');
    DrawTextCentered(Window.Width div 2 + 100,Window.Height div
2+75,Window.Width div 2 + 250,Window.Height div 2+125,'ВЫХОД');
        Redraw;
    end;
    end;
    UnlockDrawing;
    end;

    procedure Confirm;
    begin
    LockDrawing;
        Brush.Color:=clWhite;
        case task.column of
```

```
        1: if (task.mode<>0) then Circle(Window.Width div 2 -
215,Window.Height div 2 - 190+(task.mode-1)*100,25);
        2: if (task.typ<>0) then Circle(Window.Width div 2 +
285,Window.Height div 2 - 190+(task.typ-1)*100,25);
        3: if (task.difficulty<>0) then Circle(Window.Width div 2 +
635,Window.Height div 2 - 190+(task.difficulty-1)*100,25);
        end;
        Brush.Color:=clBlack;
        case task.column of
        1: if (task.mode<>0) then Circle(Window.Width div 2 -
215,Window.Height div 2 - 190+(task.mode-1)*100,15);
        2: if (task.typ<>0) then Circle(Window.Width div 2 +
285,Window.Height div 2 - 190+(task.typ-1)*100,15);
        3: if (task.difficulty<>0) then Circle(Window.Width div 2 +
635,Window.Height div 2 - 190+(task.difficulty-1)*100,15);
        end;
    Redraw;
    UnlockDrawing;
    end;

    procedure ButtonSprite;
    var pict:picture;
    var info:string;
    begin
    LockDrawing;
    pict:=Picture.Create('Images/background.jpg');
    pict.Load(concat('Sprites\',color.ToString,status.ToString,'.png'));
    pict.Draw(x,y,w,h);
    case text of
     1: info:='Начать игру';
     2: info:='Выход';
     3: info:='Выход';
     4: info:='На время';
     5: info:='Ни единой ошибки';
     6: info:='Округление';
     7: info:='Сложение/Вычитание';
     8: info:='Деление/Умножение';
     9: info:='Уравнения';
    10: info:='На логику';
    11: info:='Легко';
    12: info:='Средне';
    13: info:='Трудно';
```

```
14: info:='Тяжело';
end;
DrawTextCentered(x,y,x+w,y+h,info);
Confirm();
Redraw;
UnlockDrawing;
end;
end.
```