

Федеральное агентство по образованию
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет информатики
Кафедра прикладной информатики

УДК 681.03

Допустить к защите в ГАК
Зав. кафедрой, проф., д.т.н.
_____ С.П.Сущенко
«__» _____ 2006 г.

Ивашкина Наталья Алексеевна

Создание комплекса тренажеров и игровых обучающих программ по математике

Дипломная работа

Научный руководитель,
Доцент ФИнф

Поддубная Т.Н

Исполнитель,
Студ, гр1412

Ивашкина Н.А.

Электронная версия дипломной работы помещена
В электронную библиотеку. Файл
Администратор

Томск – 2006

Реферат

Отчет – 41 с, 15 рис, 5 источников.

ОБУЧАЮЩАЯ ПРОГРАММА, ДИДАКТИЧЕСКИЕ ИГРЫ, КРАТНОСТЬ, ТАБЛИЦА УМНОЖЕНИЯ, ПРОСТЫЕ ЧИСЛА, ЧИСЛА ФИБОНАЧЧИ, ОБЫКНОВЕННЫЕ ДРОБИ, НОД, DELPHI.

Объект разработки – комплекс тренажеров и игровых обучающих программ по математике.

Цель работы – создать для учащихся 3-6 классов набор программ-тренажеров и игровых обучающих программ по математике.

Результат работы - создан комплекс тренажеров и игровых обучающих программ по математике. Данная программа знакомит пользователя с некоторыми математическими понятиями, дает возможность потренироваться в решении арифметических выражений и математических действий в игровой форме. Обучающая программа реализована в Delphi. Данная программа рассчитана на школьников 3-6 классов, владеющих знанием основных арифметических действий.

Содержание

Введение.....	3
1. Применение компьютерных технологий в образовании...	
1.1 Обучающие программы.....	4
1.2 Дидактические игры.....	6
1.3 Функциональная структура игровой обучающей программы.....	8
2. Игровая обучающая программа «С математикой в путь».....	10
2.1 Общие характеристики программы.....	10
2.2 Программы с числами.....	11
2.2.1 «Живая таблица умножения».....	11
2.2.2 «Простые числа».....	12
2.2.3 «Числа Фибоначчи».....	12
2.3 Программы - тренажеры.....	13
2.3.1 «Вычислить арифметическое выражение».....	13
2.3.2 «Вставить недостающий знак».....	14
2.3.3 «Сравнение обыкновенных дробей».....	16
2.3.4 «Нахождение НОД».....	16
2.4 Программы игры.....	17
2.4.1 «Узоры стоклеточного квадрата».....	17
2.4.2 «Расположи числа».....	17
2.4.3 «Найди числа».....	18
3. Руководство программиста.....	19
4. Руководство пользователя.....	25
Заключение.....	28
Список использованной литературы.....	29
Приложение.....	30

Введение

Одним из главных направлений процесса информатизации современного общества становится информатизация образования, обеспечивающая широкое внедрение в практику психолого-педагогических разработок, направленных на интенсивность процесса обучения, реализацию идей развивающего обучения, совершенствование форм и методов организации учебного процесса. Применение в образовании компьютеров и информационных технологий оказывает существенное влияние на содержание, методы и организацию учебного процесса по различным дисциплинам.

В связи с этим, в настоящее время происходит интенсивная разработка обучающих, игровых и контролирующих программ, которые помогают существенно сэкономить время преподавателя, развить творческие способности обучаемого путем создания благоприятной среды, тренироваться в решении задач определенного класса и получать адекватные оценки своих знаний.

Однако у процесса информатизации есть и негативные последствия: детей все больше интересуют компьютеры, компьютерные игры, следовательно, привлекать внимание к обучению ребенка следует через компьютерную реализацию развивающих игр.

В дошкольном возрасте игра является ведущей деятельностью ребенка. С приходом ребенка в школу ведущая деятельность из игровой превращается в учебную. Основным видом деятельности ребенка должно стать учение. Поэтому в начальной школе необходимо заложить у учащихся основы учебной деятельности. Однако этот процесс осложняется возрастными особенностями младших школьников: слабой переключаемостью внимания, его неустойчивостью, произвольностью памяти и мышления. Для преодоления этого в учении и должны широко использоваться игровые формы активности детей - учебная деятельность младшего школьника должна быть пронизана игровыми моментами.

В целях экономии времени преподавателя, повышения мотивации, стимулирования инициативы и творческого мышления учащихся, а также сближения учебной деятельности и игровой была поставлена цель – создать комплекс тренажеров и игровых обучающих программ по математике, который знакомит обучаемого с «интересными» свойствами чисел, дает возможность проверить усвоенный материал, и, играя, потренироваться в выполнении арифметических действий. Игра позволяет просто реализовать известную в педагогике концепцию «зоны ближайшего развития». В процессе игры ребенок иногда выполняет предметные действия, которые он «еще не проходил», поскольку с точки зрения своего развития он уже готов выполнять эти действия, пользуясь знаниями, полученными на бытовом уровне.

1. Применение компьютерных технологий в образовании

1.1 Обучающие программы

Информационная технология обучения является новой методической системой, позволяющей рассматривать учащегося не как объект, а как субъект обучения, а компьютер - как средство обучения. Обучаемый переходит в новую категорию потому, что по форме компьютерное обучение является индивидуальным, самостоятельным, но осуществляется по общей методике, реализованной в обучающей программе.

Обучающие программы бывают нескольких типов: информационные, справочные, контролирующие, комбинированные. Одни обучающие программы способны контролировать знания учащихся, другие содержат в себе элементы «учебного тренажа», третьи помогают овладению новым учебным материалом, четвертые предназначены для того, чтобы стимулировать интерес учащихся к изучаемому предмету.

Существует несколько основных принципов построения обучающих программ:

Принцип *доступности* при компьютерном обучении переходит от принципа всеобщей доступности, для определенной возрастной группы учащихся или для некоторого усредненного учащегося данного возраста, в принцип индивидуальной доступности и рассматривается как возможность достижения цели обучения. Учебный материал, реализованный в компьютерном обучении, предполагает наличие разветвлений, различных путей и скоростей прохождения учебного курса, оказание помощи в виде пояснений, подсказок, дополнительных указаний и задач, постоянно контролирует и поддерживает на необходимом уровне мотивацию обучаемого.

Принцип *наглядности* в компьютерном обучении позволяет увидеть то, что не всегда возможно в реальной жизни даже с помощью самых чувствительных и точных приборов.

Принцип *систематичности* и *последовательности* связан как с организацией учебного материала, так и с системой действий обучаемого по его усвоению. Компьютерное обучение характеризуется последовательностью специфических действий, часть которых присуща обучению в любых формах, а часть - только компьютерному. Такими действиями, например, являются восприятие информации с экрана дисплея, работа в знаковых моделях, ввод ответа с клавиатуры. Для обеспечения принципа последовательности учащемуся в начале сеанса компьютерного обучения полезно дать ориентировочную основу действия, сформулировать цель обучения. Независимо от сложности и длины пути, приводящего обучаемого к цели, это происходит систематично и последовательно. Понятие последовательности получило свой смысл в информационных технологиях обучения, под последовательностью понимается очередность выдачи учебных фрагментов обучающей программой.

Эффективное представление уроков на последовательности кадров является одним из центральных вопросов компьютерного обучения.

Перед выдачей обучаемому изображения необходимо позаботиться о виде каждой строки, каждого слова и даже каждого символа текста. Главным здесь является ясность сообщения, выдаваемого обучаемому. Сообщение в целом можно охарактеризовать четырьмя показателями читабельности: типом стиля, длиной строки, выравниванием текста, точками прерывания. Стилль включает отчетливость изображения, подчеркивание, выделение мерцанием и другими средствами важных моментов содержания, размеры текста, всевозможные способы чередования, курсив и цвет. Очевидно, что размер текста должен быть небольшим, по крайней мере таким, чтобы можно было различать сходные строчные буквы и знаки, длина текста обусловлена особенностями памяти обучаемого, в том числе возрастными.

При чтении текста усваивается сенсорная, синтаксическая, семантическая и прагматическая информации. Читающий использует различные процессы представления и обработки знаний для усвоения информации, содержащейся в тексте.

В обучающих программах существует три основных способа организации кадра.

При первом способе кадр разделяется на две, не обязательно равные, части. Одна часть может быть использована для ввода данных с клавиатуры, а другая - для вывода команд, подсказок.

В другом случае - окна экрана могут иметь разный размер и содержание в разные моменты обучения. Второй способ организации пространства кадра означает, что нет зарезервированных участков кадра, которые содержат некоторый бланк для выдачи информации. Требуемая информация выдается по мере необходимости, а отводимая для этого часть кадра может использоваться для других целей.

Третий способ - использование вложенных окон. Такое окно появляется в определенном месте экрана по требованию, оно исчезает либо когда на экране исчезает данное требование, либо появляется новое. Также необходимо иметь на кадре участок для ввода ответа обучаемого на вопрос системы. Желательно, чтобы и вопросы задавались в определенном месте кадра. Постоянно в одном и том же месте должна появляться (или постоянно присутствовать) ориентировочная информация. Необходимо так же иметь список управляющих воздействий, таких как прерывание обучения, очистка экрана.

Обучающие программы бывают нескольких типов: информационные, справочные, контролируемые, комбинированные, обучающие игровые.

Одни обучающие программы способны контролировать знания учащихся, другие содержат в себе элементы «учебного тренажера», третьи помогают овладению новым учебным материалом, четвертые предназначены для того, чтобы стимулировать интерес учащихся к изучаемому предмету.

Обучающие игровые программы используются для решения широкого диапазона педагогических задач, поскольку главная их особенность состоит в комплексном воздействии на интеллектуальную, эмоциональную, нравственную и поведенческую стороны личности обучаемого.

В обучающих играх можно отметить целый ряд психолого-педагогических достоинств:

во-первых, активность их участников - обучаемые увлеченные игрой, не замечают, что учатся, они познают, запоминают новое, ориентируются в необычных ситуациях, пополняют запас представлений и понятий, развивают фантазию, приобретают умения и навыки и применяют их в процессе игры;

во-вторых, динамичность, которая выражает влияние фактора времени на игровую деятельность человека - время приобретает такое же первостепенное значение, как и в жизни;

в-третьих, занимательность, которая отражает увлекательное проявление человеческой деятельности и потому оказывает сильное эмоциональное воздействие на человека, побуждая его к участию в игре или заинтересованному наблюдению за ее ходом и усиливая тем самым познавательные интерес и активность.

Основным элементом обучающей игры является выполнение ею дидактической задачи. Она определяется общей целью обучения и тесно связана с целью и задачами занятий. Все остальные элементы (содержание, правила, игровые действия) подчинены дидактической задаче и обеспечивают ее реализацию.

1.2 Дидактические игры

Дидактическая игра является формой обучения, применяемой и в докомпьютерную эру, для этого использовалась бумажная технология.

Дидактическая игра наиболее часто применяется для маленьких детей. В ходе игры учащиеся незаметно для себя выполняют различные упражнения, где им самим приходится сравнивать, выполнять арифметические действия, тренироваться в устном счете, решать задачи, она развивает самостоятельность и активность мышления у детей.

Игра ставит учащихся в условия поиска, пробуждает интерес к победе, следовательно, дети стремятся быть быстрыми, находчивыми, четко выполнять задания, соблюдать правила игры. Интерес к ней, стремление выполнить её активизируется игровыми действиями. Чем они разнообразнее и содержательнее, тем интереснее для детей сама игра и тем успешнее решаются познавательные и игровые задачи.

Например, задание на усвоение счета до 10 можно предложить в обычной форме «Заполни таблицу» рис1.1

1	2			
10	9			

Рис.1.1 – «Заполни таблицу».

А можно создать игровую ситуацию: «Помоги Незнайке украсить новогоднюю елочку числовыми шарами». На рисунке 1.2 можно увидеть иллюстрацию данного задания.

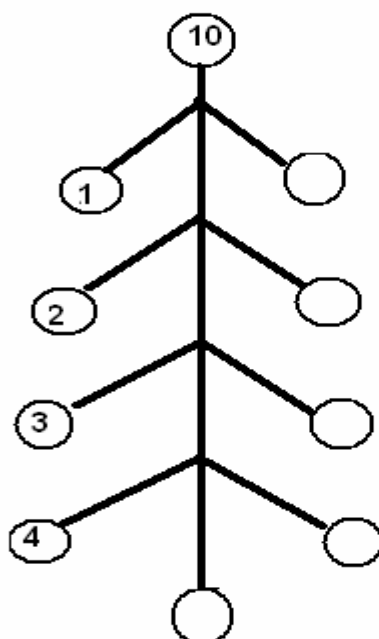


Рис.1.2 – «Украсьте елочку».

Результат дидактической игры - показатель уровня достижения детей в усвоении знаний, в развитии умственной деятельности, взаимоотношений, а не просто выигрыш, полученный любым путём.

Дидактическая игра оказывает влияние на формирование такого качества ума как его подвижность и гибкость, развивает внимание, воображение, формирует волю детей.

Для младших школьников учение - новое и непривычное дело. Поэтому при знакомстве со школьной жизнью игра способствует снятию барьера между «внешним миром знания» и психикой ребенка.

Игровое действие позволяет осваивать то, что заранее вызывает у младшего школьника страх неизвестности, постоянно внушаемое уважение к школьной премудрости, что мешает свободному освоению знаний. Кроме того, установка на выполнение учебной работы у детей еще не сформирована. Поэтому основным типом дидактических игр, используемых на начальных этапах, являются игры, формирующие устойчивый интерес к учению и снимающие напряжение, которое возникает в период адаптации ребенка к школьному режиму. Игра помогает сделать любой учебный материал увлекательным, вызывает у учеников глубокое удовлетворение, создает радостное рабочее настроение, облегчает процесс усвоения знаний. Игровые задания развивают у детей смекалку, находчивость, сообразительность. Вначале дети проявляют интерес только к игре, а затем и к тому учебному материалу, без которого игра невозможна.

Особенность дидактической игры заключается в том, что учебные задачи в них выступают не в явном виде – на первый план для учащегося выходит игровая задача.

На рисунке 1.3 можно увидеть структуру дидактической игры.

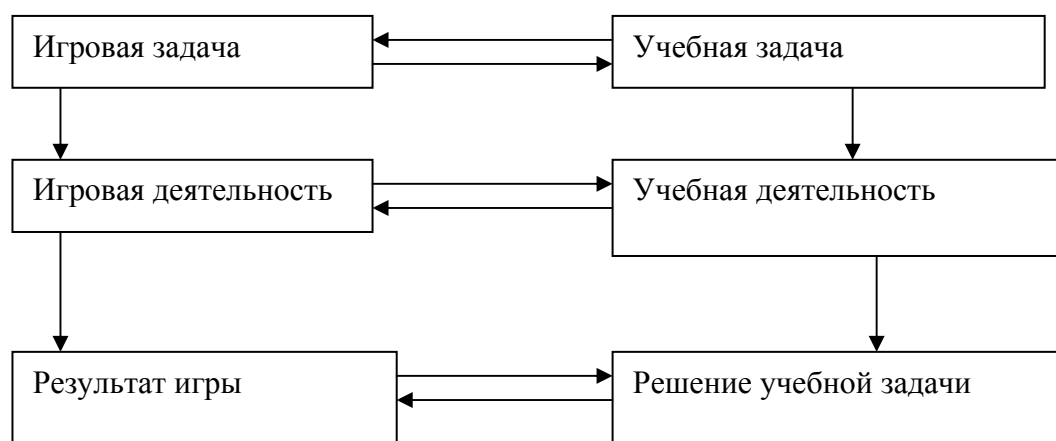


Рис1.3 - Структура дидактической игры

1.3 Функциональная структура игровой обучающей программы

Игры в обучении – одна из форм компьютерных обучающих программ.

Психологическое обоснование использования игр в обучении определяется фактом использования наиболее подходящего вида деятельности учащегося для решения основной учебной задачи – получения и усвоения новых знаний и закрепления этих знаний в результате многократного их использования в процессе игры.

Игра с точки зрения играющего представляет собой определенную задачу – в ней имеются условия, цель и средства её достижения. В дипломной работе рассматриваются математические игры, относящиеся к классу логических игр или головоломок.

Решение задачи – процесс, управляемый играющим в реальном времени по установленным правилам. Течение процесса оценивается по установленному критерию.

В связи с этим в игровой обучающей программе (далее ИОП) можно выделить три функциональные компоненты: игровую среду (далее ИС), алгоритм взаимодействия компьютера с играющим и оценку результатов игровой ситуации.

При очевидной связи и взаимозависимости этих компонентов их можно рассматривать как самостоятельные, что облегчит работу по их программированию.

Игровая среда – это совокупность всех объектов и связей в игре и законов их изменения. Другими словами, ИС – это основа, "мир", в котором развивается игровое действие. Так, в шахматах игровой средой будет совокупность, в которую входят: доска, два набора фигур, правила перемещения фигур по доске, а также правила взятия (и перемещения) фигур. В игровых средах, предназначенных для изучения нового учебного материала, игровая среда должна представлять собой удобную (чаще всего, графическую) модель предметной области.

Взаимодействие с играющим – это совокупность средств, предоставляемых играющему для изменений параметров игровой среды, т. е. для действий и изменений в игровой среде, которые происходят, когда играющий выполняет некоторые действия с объектами игровой среды, нажимая на определенные управляющие компоненты, находящиеся на поле среды. Заметим, что обучающие игры, в отличие игр реального времени, всегда в своей основе содержат некоторую интеллектуальную деятельность учащегося, связанную с предметной областью.

Важным свойством игровых обучающих программ является то, что эта предметная область является скрытой от учащегося совокупностью внешних действий с объектами игры.

Игра, являющаяся деятельностью, интересной для учащегося, заставляет его без педагогического насилия овладевать новыми знаниями (так называемое латентное научение).

Оценка игровой ситуации – это соотношения и условия, которые определяют задачу для играющего в данной игре. Сюда включается система очков и штрафов за игровые действия, описание начальной и конечной игровой ситуации. В структуре игровой обучающей программы эту задачу решает функциональный блок классификации и регистрации действий учащегося в процессе игры по выполнению действий предметной области.

Из перечисленных компонент важнейшая, безусловно, – это игровая среда. Если она найдена удачно, то, изменяя другие компоненты, можно создать вариант игры с теми или иными свойствами (темпом, сложностью и др.), иными словами, игру, адаптирующуюся к возможностям учащегося.

После разработки игры на функциональном уровне (когда создана ИС, определены средства взаимодействия с играющим и намечена система оценок) начинается этап разработки компьютерной программы, реализующей данную игру.

Игровая программа состоит, как правило, из двух частей: первая реализует ее внутреннюю (логическую) структуру, т. е. отображение игры в системе алгоритмов. Вторая

– отображает процесс игры на экране дисплея для играющего. Эта часть, определяющая форму протекания игры, связана больше с эстетическими и эргономическими критериями, и хотя и не является, на наш взгляд, основной при создании ИОП, ей придается огромное значение.

В настоящее время имеется достаточно большое число средств, позволяющих визуально создавать необходимое отображение игры (игровое поле), что делает эту часть создания игровой обучающей программы достаточно не сложной.

Основу игровой программы составляет её логическая структура. Подавляющее большинство игровых программ рассматриваемого нами типа (обучающие программы) имеет более или менее сходную логическую структуру. В ней можно выделить три блока:

- **блок оперативного слежения**, реагирующий на каждое действие игрока;
- **тактический блок**, определяющий совокупность игровых действий, ведущих к достижению какой-либо локальной цели. Этот блок также определяет моменты перехода на новый раздел работы с предметным материалом или возврата к уже пройденному материалу.
- **стратегический блок** позволяет учащемуся либо самостоятельно выбирать желаемый путь обучения и значение параметров игры, либо делать это по рекомендациям методики, заложенной в ИОП. Эти уровни находят, как правило, адекватное представление в структуре компьютерной программы.



Рисунок 1.4 – Функциональные схемы обучающих программ.

2.Игровая обучающая программа «С математикой в путь»

2.1 Общие характеристики программы

Данная обучающая программа предназначена для ребенка 3-6 классов, который уже знаком с основными арифметическими операциями: сложение, умножение, вычитание, деление. И нуждается в занимательной тренировке данных умений.

Также она дает возможность повторить таблицу умножения в игровой форме. Обучающая программа позволяет работать учащемуся с числами, лежащими в диапазоне от -100 до 100, так как не каждый ребенок начальных классов умеет оперировать большими числами.

Наряду со знанием арифметических операций, обучаемый должен владеть операциями сравнения обыкновенных дробей.

Обучающая программа предназначена для детей с разными математическими знаниями и способностями, поэтому во многих заданиях – играх ребенок сам выбирает знаки арифметических операций, а так же диапазон, тип и количество чисел, которыми будет оперировать, в соответствии со своими математическими способностями.

При отображении каждого задания, на кадре присутствует подробная информация, которая может понадобиться учащемуся при выполнении текущего задания.

Обучающая программа представляет собой набор заданий - игр, с учебным материалом по текущей теме. Также реализованы развивающие задания-игры, которые позволяют тренироваться в выполнении арифметических операций в игровой форме.

Данная программа состоит из следующих блоков заданий:

1. «Темы» - здесь предлагается учебный материал по некоторым темам.
2. «Контрольные» – в данном блоке реализованы задания - тренажеры, позволяющие обучаемому проверить свое умение совершать арифметические операции.
3. «Игры»- это блок развивающих игр.

Данная обучающая программа построена в соответствии с принципами построения обучающих программ, о которых говорилось ранее.

2.2 Программы с числами

2.2.1 Живая таблица умножения

Всем известно, что знание таблицы умножения очень важно для ребенка. Пробелы в знании таблицы умножения могут создать серьезные проблемы в изучении математики в дальнейшем. Таблица умножения – основа устного умножения.

В настоящее время, каждый ребенок может воспользоваться «услугами» калькулятора, даже для перемножения однозначных чисел. Это очень плохо, так как ребенок, который постоянно использует калькулятор, забывает о необходимости знания таблицы умножения, рассчитывая на «умного друга».

Именно это явилось причиной включения в данную обучающую программу повторения таблицы умножения, чтобы ребенок еще раз увидел ее, так как давно ушли в прошлое тетради с таблицей умножения на обороте.

Если посмотреть на таблицу умножения внимательнее, то можно заметить, что она симметрична относительно главной диагонали. Таким образом для освоения таблицы умножения достаточно знать числа, получившиеся в результате перемножения множимого и множителя над главной диагональю и применять принцип симметричности. Наблюдения показывают, что некоторые дети, не знающие данного свойства таблицы умножения, 5 умножить на 6 могут, а вот какое число получится в результате умножения 6 на 5 – затрудняются дать ответ.

Данная математическая игра предназначена для закрепления знания таблицы умножения. Игровая среда представляет собой кадр, на котором постепенно «оживает» таблица умножения, заполняя клетки. Даже в этот момент обучаемый может проверить свои знания таблицы умножения предугадывая число, которое появится на экране в следующий момент. В обучающих целях на данный кадр добавлен фрагмент с учебным материалом, который способствует запоминанию таблицы умножения. Кадр данного задания представлен на рисунке А.1.

После прочтения учебного материала, ребенок может проверить свои знания таблицы умножения: для этого на кадре появляется «пустая» таблица умножения – изображены только множимые и множители, произведения этих чисел обучаемый должен ввести сам. При вводе чисел пользователем используется принцип симметричности таблицы умножения, что несомненно «напомнит» обучаемому, что числа над главной диагональю равны числам под главной диагональю.

После заполнения пустых клеток таблицы умножения, ребенок должен проверить правильность своих вычислений. Неправильно введенные числа программа считает ошибками и выделяет их красным цветом, что бы обучаемый смог сам увидеть оплошности в своих вычислениях. Пример проверки знания таблицы умножения представлен на рисунке А.2.

Так как программа рассчитана не на определенный возраст, а на некоторый возрастной диапазон обучаемого, программа считает ошибкой только неправильно введенные данные, а незаполненные клетки таблицы ошибкой не считаются. Данная функциональность объясняется тем, что даже в школе от ученика не требуют усвоения «за раз» всей таблицы умножения. Запоминание таблицы умножения должно происходить постепенно, путем ее разбиения на 2-3 части.

Не забывая о том, что пользователем данной обучающей программы является ребенок, который при вводе чисел в таблицу умножения, или по невнимательности, или специально, может ввести недопустимые значения, например букву или дробное число, реализована защита от ввода недопустимых символов. Недопустимые символы просто не отражаются на экране, давая понять пользователю, что он делает неверный шаг.

2.2.2 Простые числа

Простые числа является одним из любопытных наблюдений над числовыми последовательностями, накопившимися в математике от глубокой древности до нашего времени.

Простые числа являются первичными элементами из которых составляются все числа, поэтому понятен интерес к простым числам со стороны математиков.

Данная тема знакомит ребенка с понятием «простое число». Кадр этой игры, для улучшения восприятия обучаемым, представляет собой красочно выделенные простые числа на столклеточном квадрате и учебный материал по теме «простые числа»: определение и способ нахождения изучаемых чисел. Соблюдая все правила оформления обучающих программ, учебный материал разбит на несколько фрагментов для более удобного запоминания обучаемым данной темы. Кадр данной задачи можно увидеть на рисунке А.3.

После прочтения материала, ребенку предлагается проверить усвоение изучаемого материала. Проверка усвоения учебного материала: на столклеточном квадрате обучаемый должен выбрать числа, которые являются простыми. Выбранные ребенком числа выделяются цветом. Ребенок может убрать ранее сделанное выделение, если понял, что допустил ошибку.

После выделения простых чисел, ребенок может проверить правильность своих действий. Для этого на этом же столклеточном квадрате изображаются простые числа, но уже другим цветом. Таким образом, ребенок сам может сравнить, какие именно по его мнению «простые числа», действительно «простые», и тем самым проверить усвоение пройденного материала.

2.2.3. Числа Фибоначчи

Данная тема так же является одной из любопытных наблюдений над числовыми соотношениями, накопившимися в математике от глубокой древности до нашего времени.

Как известно, Фибоначчи составил ряд из натуральных чисел, который впоследствии оказался полезным в науке. Закон образования членов этого ряда очень прост: первые два члена - единицы, а затем каждый последующий член получается путем сложения двух непосредственно ему предшествующих. Ряд Фибоначчи известен не только математикам, но и природоведам.

Данная игра знакомит ребенка с числами ряда Фибоначчи. Кадр этой игры представляет собой красочно выделенные числа Фибоначчи на столклеточном квадрате и учебный материал по теме «ряд Фибоначчи»: способ его вычисления, свойства чисел ряда.

После прочтения материала, ребенок должен проверить усвоение пройденной темы: задание данной игры заключается в выделении чисел ряда Фибоначчи на столклеточном квадрате – выбранное обучаемым число выделяется цветом, повторный выбор числа убирает выделение.

Для проверки правильности выбранных чисел на этом же столклеточном квадрате выделяются другим цветом правильные числа ряда Фибоначчи, таким образом обучаемый может сравнить свои варианты ответов с правильными.

2.3 Программы - тренажеры

2.3.1 Вычислить арифметическое выражение

Данное задание относится к блоку «контрольные», оно является заданием - тренажером.

Оно предназначено для закрепления навыков устного счета.

Суть этого задания заключается в следующем: обучаемый должен решить арифметическое выражение, сгенерированное обучающей программой. Потребность в данном рода задании возникла по причине не умения учеников считать в уме. Данное задание позволяет потренироваться обучаемому в совершении арифметических операций.

Игровое задание состоит из двух этапов:

1. Конструирование арифметического выражения.
2. Вычисление полученного арифметического выражения.

Конструирование заключается в установлении учащимся параметров арифметического выражения, которое ему необходимо решить. Это позволяет настраивать генерируемое выражение под конкретного пользователя, и исключает использование «шаблонного» выражения, ответ на который пользователь может просто запомнить из предыдущего опыта прохождения обучающей программы. Каждый раз генерируется новое выражение, поэтому вероятность «запомнить» правильный ответ сведена к нулю.

Параметры генерируемого выражения приведены ниже

1. Количество слагаемых:
 - 1.1 Задается случайно из диапазона от 2 до 5
 - 1.2 Задаёт ребенок - при выборе данной опции пользователю предоставляется возможность указать, какое количество слагаемых будет присутствовать в генерируемом выражении;
2. Диапазон слагаемых:
 - 2.1 Задается случайно из диапазона от -100 до 100
 - 2.2 Задаёт ребенок из диапазона -100 до 100 - при выборе данной опции пользователь должен ввести точный диапазон слагаемых, так как уже говорилось ранее, не у всех обучаемых равные возможности математических вычислений;
3. Арифметические операции:
 - 3.1 Задается случайно
 - 3.2 Задаёт ребенок – сложение, умножение, вычитание, деление. При выборе данной опции, пользователю предоставляется возможность выбора одного или нескольких знаков арифметических операций, которые будут присутствовать в генерируемом выражении;
4. Тип слагаемых:
 - 4.1 Задается случайно – либо целые либо дробные. Количество знаков после запятой для дробных чисел выбирается случайно из диапазона от 1 до 5
 - 4.2 Задаёт ребенок – целые либо дробные. Если пользователь выбрал «дробные», то ему необходимо указать количество знаков после запятой;
5. Ограниченность выполнения задания по времени:
 - 5.1 Время на решение арифметического выражения ограничено – пользователю нужно ввести время на выполнение в секундах, по умолчанию, временное ограничение равно 30 сек
 - 5.2 Время на выполнение арифметического выражения не ограничено;

После установления параметров задания, происходит проверка правильности введения данных: диапазон может быть введен только указанием сначала левой границы диапазона, а затем правой границы диапазона. Если выбран вариант собственного указания знаков арифметических операций, а не указано какие/какой именно знак должен присутствовать в генерируемом выражении, то пользователю выдается сообщение о совершенных ошибках при вводе, и возвращение к кадру конструктора задания для исправления совершенной ошибки. Кадр конструктора задания представлен на рисунке А.4.

После успешного прохождения процедуры проверки кадр конструктора закрывается и начинается второй этап данного задания.

Второй этап заключается в вычислении сгенерированного арифметического выражения.

По установленным параметрам генерируется выражение, затем программа решает его.

Сгенерированное выражение выводится на экран, ребенку предлагается решить данное выражение и ввести ответ в поле для ввода. Пример данного задания можно увидеть на рисунке А.5.

Как говорилось ранее, не забывая о том, что с данной обучающей программой в первую очередь работает ребенок, все поля защищены от ввода не допустимых значений.

Если задание выполняется на время, то на кадре отображаются часы, показывающие прошедшее и оставшееся время в процентах. И если учащийся хочет успеть решить выражение за отведенное ему время, он должен внимательно наблюдать за тем, сколько у него осталось времени. Если пользователь за отведенное ему на решение время не успевает ввести ответ, то поле для ввода ответа ребенка становится недоступным и показывается правильный ответ.

После решения выражения, обучаемый должен проверить правильность своего ответа, для этого он должен ввести результат в соответствующее поле и нажать кнопку проверки.

После завершения вычислений данного выражения, можно сгенерировать новое выражение, по прежним параметрам, например, если на первой этапе были выбраны только целые числа, то новое выражение будет состоять только из целых чисел.

2.3.2 Вставить недостающий знак

Данное задание относится к блоку «контрольные», оно является заданием - тренажером.

Суть данного задания заключается в том, что обучаемый должен найти в сгенерированном выражении недостающий знак арифметической операции или число, которое при выведении на кадр обучающая программа заменила знаком «?».

Отличительной чертой данного задания от предыдущего является то, что обучаемому требуется не только складывать, вычитать, делить, умножать, но и переносить числа с левой стороны знака равенства в правую и наоборот, совершая соответствующие преобразования над данным выражением.

Если компьютером загадан знак арифметической операции, то обучаемому придется решать данное задание полным перебором возможных вариантов, количество возможных вариантов можно значительно сократить, если пользоваться принципом исключения, но знает ли об этом обучаемый?

При выполнении данного задания, обучаемому понадобится не только знание арифметических операций, но и смекалка для ускорения процесса решения поставленной задачи.

Данная математическая игра состоит из двух этапов:

1. Конструирование арифметического выражения.
2. Нахождение недостающего числа или знака арифметической операции.

Конструирование заключается в задании параметров арифметического выражения, в котором будет загадано число или арифметическая операция. Параметрами выражения являются:

1. Количество слагаемых:
 - 1.1 Задается случайно из диапазона от 2 до 5
 - 1.2 Задаёт ребенок;
2. Диапазон слагаемых:
 - 2.1 Задается случайно из диапазона от -100 до 100
 - 2.2 Задаёт ребенок из диапазона -100 до 100;
3. Арифметические операции:
 - 3.1 Задается случайно
 - 3.2 Задаёт ребенок – сложение, умножение, вычитание, деление;
4. Тип слагаемых:
 - 4.1 Задается случайно – либо целые либо дробные. Количество знаков после запятой для дробных чисел выбирается случайно из диапазона от 1 до 5.
 - 4.2 Задаёт ребенок – целые либо дробные. Количество знаков после запятой для дробных чисел ребенок выбирает сам;
5. Выполнение задания на время:
 - 5.1 Время на выполнение данного задания ограничено
 - 5.2 Нахождение загаданного числа или знака арифметической операции происходит без учета времени;

Все параметры и принципы их установления идентичны параметрам задания «Реши выражение», рассмотренным в предыдущей главе.

Второй этап наступает только при успешном завершении первого этапа. После закрытия окна конструктора, обучаемому показывается кадр второго этапа данного задания.

Игра на этом этапе заключается в том, что ребенок должен найти недостающее число или знак арифметической операции. После того, как ребенок задал параметры выражения, программа генерирует его, затем оно решается с помощью функции-решателя. Из выражения случайным образом выбирается знак арифметической операции или число, и заменяется символом «?», после чего выводится на экран. Кадр задания иллюстрирует рисунок А.6.

Ребенок должен найти замену символу «?». После того как найдено «возможное» число, обучаемый должен проверить его правильность.

Поле ввода ответа пользователя защищено от ввода не допустимых значений, то есть, если загадано число, то при нажатии клавиш в поле ввода отражаются только числа, если загадана арифметическая операция, то возможен ввод только знаков «+», «-», «*», «/».

Если задание выполняется на время, то после его истечения, поле для ввода ответа обучаемого становится недоступным и показывается правильный ответ загаданного числа или арифметической операции.

В данной программе-тренажере предусмотрена подсказка загаданного числа, и по желанию обучаемого возможен ее вывод. Данная функциональность программы определяет принадлежность загаданного числа к какой либо последовательности и выводит

получившейся результат пользователю. Возможно, это поможет обучаемому принять правильное решение и найти правильное загаданное число.

2.3.3 Сравнение обыкновенных дробей

Данное задание относится к блоку «контрольные», оно является заданием - тренажером.

Включение данного задания-тренажера в обучающую программу обосновывается тем, что ребенок начальных классов хуже всего из четырех арифметических операций владеет делением. Кадр данного задания можно увидеть на рисунке А.7.

Данное задание заключается в том, что ребенок должен расположить в порядке возрастания сгенерированные программой обыкновенные дроби. Количество обыкновенных дробей для сравнения обучаемый задает сам.

После установления параметров пользователем, обучающая программа генерирует обыкновенные дроби и выводит на экран. Выполнение данного задания не ограничено временем, так как сравнение обыкновенных дробей - это достаточно трудоемкий процесс, а часы, отражающие оставшееся время для решения данного задания, будут только отвлекать пользователя и мешать сосредоточиться на задании.

Для того чтобы расположить обыкновенные дроби в порядке возрастания, ребенок должен сначала привести их к десятичному виду, а затем сравнить их между собой и расположить в правильном порядке. Это позволяет обучаемому упражняться в операции деления и операции сравнения.

После того, как обучаемый выполнит задание, он должен проверить правильность своего расположения дробей, для этой возможности, на кадре есть кнопка, при нажатии на которую ребенок может увидеть правильное расположение обыкновенных дробей.

2.3.4 Нахождение НОД

Данное задание относится к блоку «контрольные», оно является заданием - тренажером.

Для того, что бы ребенок закрепил умение выполнять операцию деления, данная обучающая программа содержит задание – тренажер «Нахождение НОД».

Кадр данного задания состоит из 2-ух частей, в одной части, располагается учебный материал по данной теме, вторая часть является проверочной частью владения операцией деления и усвоения предложенного учебного материала. Кадр данного задания можно увидеть а рисунке А.8.

Данное задание заключается в том, что ребенок выбирает количество чисел, для которых нужно будет найти НОД, программа случайно выбирает указанное количество чисел и отображает их пользователю. В то же время, реализованная программе процедура сама вычисляет для сгенерированных чисел НОД. Ребенок после вычисления для предложенных чисел НОДа, вводит его в поле ввода, которое защищено от ввода недопустимых данных, воспринимая только положительные числа.

После ввода своего варианта НОД, ребенок может проверить его правильность – для этого на кадре есть кнопка, нажав на которую программа сравнивает введенный пользователем НОД и НОД, вычисленный ею. Результат сравнения выводится на экран.

По желанию учащегося программа выводит на экран правильный НОД.

2.4 Программы игры

2.4.1 Узоры стоклеточного квадрата

Данное задание относится к блоку «игры», оно является заданием – развивающей игрой.

Идея этого задания взята из книги [1].

Данная игра знакомит обучаемого с понятием «кратность числа». Кадр данной игры состоит из 2 частей, в одной части обучаемому предлагается учебный материал по теме «Кратность», а так же правила данной игры. Вторая часть кадра предназначена для отображения стоклеточного квадрата, на котором будет происходить проверка усвоения материала, а так же игра с узорами.

Для отображения чисел используется стоклеточный квадрат. Числа, кратные одному числу – выделяются одним цветом, таким образом рисуются узоры. Данная игра возможна в двух вариантах:

1 – ребенок задает число, программа выделяет одним цветом числа, кратные введенному. Кадр первого варианта игры можно увидеть на рисунке А.9.

2 – программа случайно выбирает число из диапазона от 1 до 100, и выделяет одним цветом на стоклеточном квадрате числа, кратные выбранному. Задание данного варианта игры «Узоры стоклеточного квадрата» заключается в том, что ребенок должен вычислить, какое именно число из диапазона от 1 до 100 является кратным для раскрашенных чисел. Ребенок может проверить правильность своего варианта ответа, если введет «предполагаемое число» в поле ввода. Если введенное число равно загаданному, то программа подтверждает правильность вычислений ребенка. Кадр второго варианта игры показан на рисунке А.10.

По желанию ребенка, программа может показывать правильный вариант загаданного числа.

2.4.2 Расположи числа

Данное задание относится к блоку «игры», оно является заданием – развивающей игрой.

Правила данной игры схожи с правилами игры «пятнашки».

Числа перемещаются по игровому полю за счет того, что одно из мест на поле – свободно.

Данная игра требует сообразительности, благодаря чему обучаемый может получить удовольствие от сознания своего высокого интеллектуального уровня. Кадр данной игры можно увидеть на рисунке А.11.

В этой игре заложен математический смысл: расположение является правильным, если сумма чисел по строке равно заданному числу. Поэтому обучаемому требуется сначала посчитать, каким образом нужно расположить числа, что бы собрать требуемую сумму, а только после этого нужно расположить числа в нужном порядке. После того, как обучаемый расположит числа в правильном порядке, игра заканчивается.

Данная игра дает обучаемому возможность потренироваться в операции сложения, так как если обучаемый найдет один из вариантов расположения чисел, а поставить на данное место данное число не сможет, ему придется искать новый вариант расположения чисел, и опять считать, считать, считать....

2.4.3 Найди числа

Данное задание относится к блоку «игры», оно является заданием – развивающей игрой.

Кадр данной игры состоит из 2-ух частей: на первой части кадра представлена информация с правилами данной игры, вторая часть кадра представляет собой квадрат, размером 4×4 , с 16 числами.

Кадр данной игры можно увидеть на рисунке А.12.

Задача обучаемого заключается в отыскании чисел - слагаемых для получения требуемой суммы. Сложность этой игры для пользователя заключается в том, что все 16 чисел закрыты, программа отображает одновременно «открытыми» только 2-а числа. При попытке пользователя открыть еще одно число - предыдущие числа закрываются. Если сумма «открытых» в данный момент чисел, равна заданному числу, то данные числа удаляются из таблицы, и происходит пересчет суммы, которую необходимо найти по тому же принципу.

Таким образом, обучаемый, играющий в данную игру тренирует не только свои математические способности, но и память, ведь ему необходимо запоминать, в какой клеточке находить требуемое число.

3. Руководство программиста.

Обучающая программа содержится в модуле Unit1.pas. На форме модуля Unit1.pas, расположены следующие основные компоненты:

1. WebBrowser1: TWebBrowser - для отображения учебного материала в виде htm-страниц.
2. PageControl1: TPageControl – для отображения набора страниц.
3. TabSheet1, TabSheet2, TabSheet3, TabSheet4, TabSheet5, TabSheet7, TabSheet8, TabSheet10: TTabSheet; - для отображения кадров заданий-игр.
4. MainMenu1: TMainMenu - меню для выбора заданий-игр.
5. Timer1, Timer2: TTimer – для контроля времени выполнения задания.
6. Timer3:TTimer – для создания эффекта постепенного заполнения таблицы умножения числами.

«Таблица умножения»

Кадр данного задания представляет собой TabSheet1 с расположенным на нем StringGrid1 – для отображения таблицы умножения. При активизации кадра с данным заданием включается Timer3, который запускает процедуру timertable, данная процедура с определенной скоростью последовательно заполняет клетки таблицы умножения, создавая тем самым эффект «оживления».

WebBrowser1- для вывода учебного материала, Button32 – кнопка для перехода из текущего задания к содержанию. Кнопка Button2 - для смены кадра учебного материала. Для проверки знания таблицы умножения пользователю нужно ввести значения в StringGrid1 и нажать Button1 – при нажатии на эту кнопку, вызывается процедура TestTabl(Sender: TObject), которая считывает значения, введенные в StringGrid1 и сравнивает их с таблицей умножения. Если найдено расхождение в значениях, то данная клетка выделяется красным цветом, обозначая ошибку.

«Простые числа»

Кадр данного задания представляет собой TabSheet2, с расположенным на нем StringGrid2 – для отображения столбчатого квадрата.

WebBrowser1- для вывода учебного материала, Button32 – кнопка для перехода из текущего задания к содержанию.

Первоначально на StringGrid2 выделяются цветом простые числа до 100. Кнопка Button7 меняет кадр с фрагментом учебного материала. Когда все фрагменты учебного материала заканчиваются, StringGrid2 очищается и пользователю предлагается выделить простые числа.

Для выделения используется процедура StringGrid3DrawCell(Sender: TObject; ACol, ARow: Integer; Rect: TRect; State: TGridDrawState), которая закрашивает синим цветом выбранные числа.

Процедура StringGrid3SelectCell(Sender: TObject; ACol, ARow: Integer; var CanSelect: Boolean) выделяет выбранные числа, при повторном выборе числа - снимает выделение.

Процедура PrimNumDrawCell(Sender: TObject; ACol, ARow: Integer; Rect: TRect; State: TGridDrawState) закрашивает красным цветом простые числа.

Функция PrimeNumber(Number: integer):boolean; - возвращает true, если число простое, false – в противном случае.

«Числа Фибоначчи»

Кадр данного задания представляет собой TabSheet2, с расположенным на нем StringGrid2 – для отображения стоклеточного квадрата.

WebBrowser1- для вывода учебного материала, Button32 – кнопка для перехода из текущего задания к содержанию.

Первоначально, на StringGrid2 выделяются цветом числа ряда Фибоначчи до 100. Кнопка Button7 меняет кадр учебного материала, когда учебный материал заканчивается, StringGrid2 очищается и пользователю предлагается выделить числа ряда Фибоначчи для подтверждения усвоения пройденного материала. Для выделения используется процедура StringGrid3DrawCell(Sender: TObject; ACol, ARow: Integer; Rect: TRect; State: TGridDrawState), которая закрашивает синим цветом выбранные числа.

Процедура StringGrid3SelectCell(Sender: TObject; ACol, ARow: Integer; var CanSelect: Boolean) выделяет выбранные числа, при повторном выборе числа - снимает выделение.

Процедура FibDrawCell(Sender: TObject; ACol, ARow: Integer; Rect: TRect; State: TGridDrawState), закрашивает розовым цветом числа ряда Фибоначчи.

«Вычислить арифметическое выражение»

Каждый из этапов этого задания представляется на отдельном кадре.

Первый этап задания:

Кадр данного этапа представляет собой TabSheet9, с расположенными на нем GroupBox'ами для группировки параметров задания, Button8 – для считывания введенных параметров задания, WebBrowser1 – для вывода «инструкции», Button32 – для возврата к содержанию.

GroupBox1 – для выбора способа генерации типа слагаемых:

RadioButton1 – если количество слагаемых генерируется случайно.

RadioButton2 – если количество слагаемых будет вводить пользователь.

SpinEdit3 – кнопка-счетчик для ввода количества слагаемых.

Для улучшения наглядности, если выбран RadioButton1, то SpinEdit3 не отражается на экране. Если пользователь выбрал RadioButton2, то появляется SpinEdit3.

GroupBox2 – для выбора способа генерации диапазона слагаемых:

RadioButton3 – если диапазон слагаемых от -100 до 100.

RadioButton4 – если диапазон слагаемых будет устанавливать сам пользователь.

SpinEdit5 и SpinEdit6 – для ввода диапазона.

Для улучшения наглядности, если выбран RadioButton3, то SpinEdit5 и SpinEdit6 не отображаются, при выборе RadioButton4, SpinEdit5 и SpinEdit6 появляются на экране.

GroupBox3 – для выбора способа генерации арифметических операций.

RadioButton5 – если все четыре арифметических операций.

RadioButton6 – если пользователь сам выбирает, какие арифметические операции будут включены в формируемое выражение.

CheckBox1, CheckBox2, CheckBox3, CheckBox4 – Флажки для выбора арифметических операции: сложение, вычитание, умножение, деление, соответственно. Они видимы только при выбранном RadioButton6.

GroupBox4 – для установки временного ограничения на выполнение задания.

RadioButton6 – если выполнение задания ограничено по времени. Если пользователь выбирает данный переключатель, то на форме становится виден SpinEdit7, который служит для ввода времени на выполнения задания в секундах.

RadioButton7 – выполнение задания не ограничено по времени.

GroupBox5 – для выбора способа генерации типа слагаемых.

RadioButton8 – если тип слагаемых выбирается случайно: либо целые, либо дробные.

RadioButton9 – если пользователь сам задает тип слагаемых.

CheckBox5 – слагаемые в выражении будут целые.

CheckBox6 – слагаемые в выражении будут дробные.

SpinEdit6 – для ввода количества знаков дробной части, в случае выбора CheckBox6.

После того, как пользователь задал все требуемые параметры для задания, он должен нажать кнопку Button8. Процедура Button8Click(Sender: TObject) – считывает введенные параметры и проверяет их допустимость. Диапазон можно задавать только указанием сначала левой границы, а затем правой границы диапазона, если пользователь выбрал RadioButton9, то обязательно должен указать желаемый тип слагаемых, если пользователь выбрал RadioButton6, то обязательно должен перечислить арифметические операции.

Если пользователь установил не все параметры арифметического выражения, то открывается диалоговое окно Form2, в текстовом поле которого содержится сообщение об ошибке заполнения параметров выражения.

В данной процедуре заполняются поля структуры type TTask = record Make : string; // что делаем(решаем выражение или вставляем недостающий знак); Num : integer; // кол-во слагаемых; DFrom , DTo : integer;// диапазон; types : string; // тип слагаемых; oper : string ; // арифм.действия; timer : boolean; // если на время; tim:integer- // для указания времени на выполнение, Hin : boolean; // подсказка; fract : integer; // кол-во цифр после запятой; M : integer; end, хранящая параметры задания.

После успешного формирования структуры параметров задания, кадр конструктора закрывается и начинается выполнение второго этапа игры «вычислить выражение».

Второй этап данного задания представляет собой TabSheet5, с расположенными на нем компонентами: Label29 – для показа арифметического выражения, которое нужно решить, LabeledEdit5 для ввода ответа пользователя. Для защиты от ввода недопустимых значений используется процедура Edit3KeyPress(Sender: TObject; var Key: Char). Edit1 – поле, в которое выводится правильный ответ сгенерированного выражения.

Кнопка Button9 обрабатывается процедурой Button9Click(Sender: TObject), которая формирует арифметическое выражение с заданными параметрами - заполняет массивы случайными числами и знаками возможных арифметических операций. Из полученных массивов формирует строку выражения. Затем выводит полученную строку в Label29. Если задание выполняется на время, то включается таймер

На форме также расположена компонента Gauge1: TGauge – индикатор хода процесса, отображает остаток времени. Компонента Timer1: TTimer – изменяет положение Gauge1. Timer2: TTimer – отсчитывает время, выделенное для выполнения задания. После истечения времени, процедура Timer2Timer(Sender: TObject) выводит в поле правильный ответ, и делает недоступным поле ввода ответа.

Кнопка Button10 обрабатывается процедурой Button10Click(Sender: TObject) – делает недоступным поле ввода ответа, вызывает функцию Calculate(sMyExpression, T.fract), которая вычисляет арифметическое выражение, сравнивает возвращаемое данной функцией значение с ответом пользователя. Результат сравнения выводит в Label1.

«Найти недостающее число или арифметическую операцию»

Первый этап данного задания аналогичен первому этапу задания «Вычисление выражения».

Второй этап представляет собой TabSheet4, с расположенными на нем компонентами:

Кнопка Button14 обрабатывается процедурой Button14Click(Sender: TObject), которая формирует арифметическое выражение, выбирает из него случайно число либо знак, заменяет выбранное значение в выражении знаком «?» и полученное выражение выводит в Label21. Затем вызывает функцию Calculate(sMyExpression, T.fract), возвращаемое ею значение выводит в Label20.

Label21 – для вывода выражения с загаданным числом или знаком арифметической операции, LabeledEdit2 – для ввода варианта загаданного числа или знака пользователем. Данное поле обрабатывается процедурой Edit7KeyPress(Sender: TObject; var Key: Char), которая проверяет вводимые значения, защищая от ввода недопустимых. Если загадано число, то ввод знака невозможен, если загадан знак, то невозможен ввод числа.

Кнопка Button13 обрабатывается процедурой Button13Click(Sender: TObject), в этой процедуре происходит формирование строки-подсказки. Если загаданное число – дробное, то оно округляется и в строку-подсказку заносится информация об округлении, далее проверка свойств загаданного числа происходит с округленным значением.

Для выявления свойств загаданного числа, вызывается функция PrimeNumber(Number: integer):boolean, которая возвращает true, если число простое, false - в противном случае. Результат выполнения функции записывается в строку-подсказку. Затем проверяется, не является ли данное число числом ряда Фибоначчи – результат заносится в строку-подсказку. После этого вычисляются все делители данного числа, результат также заносится в строку-подсказку. После выявления свойств числа – строка-подсказка выводится в statictext1.

Кнопка Button15 предназначена для вывода исходного выражения(без знака «?»), она обрабатывается процедурой Button15Click(Sender: TObject), в которой вызывается функция Calculate(sMyExpression, T.fract), результат выполнения функции выводится в LabeledEdit4. В данном поле пользователь может увидеть загаданное число или знак.

«Сравнение обыкновенных дробей»

На кадре данного задания расположены: WebBrowser1- для вывода справочного материала, Button32 – кнопка для перехода из текущего задания к содержанию, TabSheet7 с расположенными на нем SpinEdit8 – для ввода количества обыкновенных дробей для сравнения, SpinEdit10 и SpinEdit11 – для ввода левой и правой границ диапазона, из которого будут выбираться числитель и знаменатель. Button23 – для генерирования обыкновенных дробей, Button25 – для показа дробей в правильном порядке, ListBox1 – для отображения сгенерированных обыкновенных дробей, ListBox2 – для отображения дробей в правильном порядке.

Кнопка Button23 обрабатывается процедурой Button23Click(Sender: TObject), которая генерирует обыкновенные дроби и выводит их в ListBox1.

Процедура ListBox1MouseDown предназначена для вывода пользователю подсказки (Десятичное представление обыкновенной дроби).

Кнопка Button25 обрабатывается процедурой Button25Click(Sender: TObject), в которой обыкновенные дроби располагаются в правильном порядке с помощью алгоритма сортировки, результат выводится в ListBox2.

«Нахождение НОД»

На кадре данного задания расположены: WebBrowser1- для вывода учебного материала, Button32 – кнопка для перехода из текущего задания к содержанию, TabSheet6 с расположенными на нем SpinEdit9 – для ввода количества чисел, для которых нужно найти НОД, StringGrid4 – для отображения чисел, для которых нужно найти НОД, Button27 – для генерирования чисел, LabeledEdit8 – для ввода пользователем найденного для данных чисел НОДа, Button28- для сравнения введенного пользователем и вычисленного программой НОДа, LabeledEdit10 – для отображения результата сравнения, Button29 – для вывода правильного НОДа в поле LabeledEdit9,

Button30 – для очищения текущего задания.

Кнопка Button27 обрабатывается процедурой Button27Click(Sender: TObject), которая генерирует числа, выводит их в StringGrid4. Из этой процедуры вызывается функция Nodarray(A: array of Integer; lmas: integer): integer, которая возвращает НОД для указанных чисел.

Для выбора заданий пользователем на форме расположена компонента MainMenu1: TMainMenu.

«Стоклеточный квадрат»

Кадр данного задания представляет собой TabSheet3 с расположенными на нем: StringGrid3- для отображения стоклеточного квадрата, WebBrowser1- для вывода правил игры, Button32 – кнопка для перехода из текущего задания к содержанию.

Для выбора варианта данной игры используется ComboBox1.

Для первого варианта игры, когда ребенок задает число, программа выделяет одним цветом числа, кратные введенному, на TabSheet3 для ввода требуемой кратности расположен компонент SpinEdit1, в свойстве которого установлено максимальное значение 100. Кнопка Button5 предназначена для раскраски стоклеточного квадрата. Числа, которые кратны введенному пользователем числу, выделяются одним цветом. Кнопка Button4 предназначена для очищения выделенных чисел.

Для второго варианта игры, когда случайно выбирается число, и отображаются числа кратные ему, на TabSheet3 расположен компонент SpinEdit2, для ввода предполагаемой кратности выделенных чисел. Кнопка Button3 – предназначена для проверки загаданного числа и числа, введенного пользователем, ответ после их сравнения выводится в LabeledEdit1.

«Расположи числа»

На кадре данной игры расположены WebBrowser1- для вывода правил игры, Button32 – кнопка для перехода из текущего задания к содержанию, TabSheet8 с расположенными на нем Image1 для отображения чисел.

Процедура Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer) – процедура для перемещения чисел. Процедура NewGame1 – для генерирования чисел. Процедура add – для вычисления контрольных сумм. Функция Finishgam – для определения окончания игры. Процедура ShowPole – для перерисовки чисел.

«Найди числа»

На кадре данной игры расположены WebBrowser1- для вывода правил игры, Button32 – кнопка для перехода из текущего задания к содержанию, TabSheet10 с расположенными на нем Image2 для отображения чисел.

Move(col,row:integer) – процедура открытия\ закрытия клеток с числами и определения нахождения требуемой суммы. Процедура NewGame2 – для генерирования чисел. Процедура MakeX – для вычисления суммы. Процедура Kletka(col,row: integer) – рисования клетки с числом. Процедура ShowPole – для перерисовки чисел.

4. Руководство пользователя

Обучающая программа создана для детей 3-6 классов, знающих числа от -100 до 100, а также владеющих арифметическими операциями: сложение, умножение, вычитание, деление и операциями сравнения чисел.

Для того, чтобы начать работу с обучающей программой, нужно запустить файл Project1.exe. После появления кадра приветствия обучающей программы, в меню можно выбрать одно из предложенных заданий-игр:

Меню «Темы» / подменю «Таблица умножения»: в данном задании вниманию пользователя представлена «Живая таблица умножения». Можно прочитать интересные свойства таблицы умножения. Нажав на кнопку «Проверить», можно проверить свои знания таблицы умножения. Для этого нужно заполнить таблицу умножения числами – произведениями, стоящими на пересечении строки и столбца множимого и множителя. После заполнения, нужно нажать кнопку «Проверить», если число введено не правильно, оно будет выделено красным цветом.

Меню «Темы» / подменю «Простые числа»: в данном задании можно познакомиться с понятием «Простые числа» и проверить усвоение пройденного материала, для этого нужно нажать кнопку «Дальше». После прочтения учебного материала, можно проверить знание простых чисел, для этого нужно выделить их на столбчатом квадрате. После выделения чисел нужно нажать кнопку «Простые числа». На столбчатом квадрате красным цветом выделяются простые числа, синим цветом неправильно вычисленные простые, малиновым цветом правильно вычисленные простые числа.

Меню «Темы» / подменю «Числа Фибоначчи»: в данном задании можно познакомиться с понятием «Числа Фибоначчи» и проверить усвоение пройденного материала. После прочтения учебного материала, нужно выделить на столбчатом квадрате числа Фибоначчи, а затем нажать кнопку «Числа Фибоначчи». На столбчатом квадрате розовым цветом выделяются числа Фибоначчи, синим цветом неправильно вычисленные числа, малиновым цветом выделяются правильно вычисленные обучаемым числа Фибоначчи.

Меню «Контрольные» / подменю «Решить выражение»: при выборе данного подменю, открывается конструктор для установления параметров арифметического выражения. После заполнения всех полей, нужно нажать кнопку «Готово». Если ошибок в заполнении не обнаружено, окно конструктора закрывается и открывается окно с выражением, которое требуется решить. После вычисления выражения, ответ нужно ввести в поле «Ответ». Чтобы проверить правильность своего ответа, нужно нажать кнопку «Показать правильный ответ». Если нажать на кнопку «Сгенерировать», будет сгенерировано новое выражение. Чтобы закончить работу с текущим заданием, нужно нажать кнопку «К содержанию».

Меню «Контрольные» / подменю «Вставить недостающий знак»: в данном задании будет загадано число или знак арифметической операции. Выбор данного подменю открывает окно конструктора для заполнения параметров выражения, в котором будут загаданы значения. После заполнения всех признаков выражения, нужно нажать кнопку «Готово». Если ошибок в заполнении не обнаружено, окно конструктора закрывается и открывается окно задания: вычислите, какое число или знак заменен символом «?». Введите свой вариант в поле «Вместо знака «?» в выражении должно быть...» и нажмите кнопку «Показать результат». Данный вариант можно очистить, нажав соответствующую кнопку. При желании, можно увидеть подсказку, нажав соответствующую кнопку. Чтобы увидеть полный вариант выражения(без знака «?»), нужно нажать кнопку «Правильный ответ». При нажатии на кнопку «Сгенерировать», будет предложено решить новое выражение. Для того, что бы закончить работу с текущим заданием, нужно нажать кнопку «К содержанию».

Меню «Контрольные» / подменю «Сравнить дроби»: в данном задании требуется расположить обыкновенные дроби в порядке возрастания. После выбора этого подменю открывается окно задания, в которое следует ввести количество обыкновенных дробей и

диапазон числителей и знаменателей. После этого нужно нажать кнопку «Сгенерировать», сгенерированные дроби будут выведены на экран. Выделяя мышью дробь, можно перемещать ее вверх и вниз. При выделении мышью дроби, можно увидеть подсказку, в которой представлена данная дробь в десятичном виде. Расположите обыкновенные дроби в порядке возрастания и нажмите кнопку «Показать правильное расположение». Можно продолжить работу с данным заданием, нажав на кнопку «Сгенерировать» или выйти из задания «К содержанию».

Меню «Контрольные» / подменю «НОД»: в данном задании требуется найти НОД для нескольких чисел. После выбора этого подменю открывается окно задания. Введите для какого количества чисел искать НОД и нажмите «Показать числа», найдите НОД для сгенерированных чисел и нажмите «Проверить». Для вывода на экран правильного НОДа, необходимо нажать «Показать правильный НОД». Для выхода из данного задания, нажмите «К содержанию».

Меню «Игры» / подменю «Узоры стоклеточного квадрата»: В данном задании можно посмотреть красивые узоры на стоклеточном квадрате.

Если в поле «Что будем делать?» выбрать вариант «Посмотреть», то для отображения узора, требуется ввести в поле «Введи кратность» число от 1 до 100 и нажать «Раскрасить». После этого можно нажать кнопку «Очистить выделение».

Если в поле «Что будем делать?» выбрать вариант «Разгадать», то нужно сначала нажать кнопку «Раскрасить». Некоторые числа стоклеточного квадрата выделяются красным цветом. Нужно вычислить, какое число является кратным для выделенных чисел. Если такое число найдено, его нужно ввести в поле «Угадай кратность» и нажать кнопку «Проверить догадку», в поле «Ответ» выведется результат сравнения.

Меню «Игры» / подменю «Расположи числа»: В данной игре требуется расположить числа в правильном порядке, т.е, что бы сумма чисел по строке равнялась числам после знака « \Rightarrow ». Способом перемещения цифр в данной игре является выбор указателем мышки числа, которое требуется передвинуть, но только в том случае, если рядом стоящее с ним место является пустым. После того как вы правильно расположите числа, программа определит это и сообщит вам, что игра закончена!

Меню «Игры» / подменю «Найди числа»: В данной игре требуется найти в закрытых клетках числа, сумма которых равняется заданному числу. В каждый момент времени количество «открытых» цифр не более 2. После того, как вы откроете одновременно два числа, сумма которых равна загаданному числу, программа определит это и сотрет клетки, в которых находились найденные числа, и пересчитает сумму для следующего этапа данной игры. Игра считается законченной только после того, как будут удалены все клетки.

ЗАКЛЮЧЕНИЕ

Использование компьютерных технологий в образовании является способом повышения мотивации и индивидуализации обучения ребенка, развития творческих способностей и создания доброжелательного эмоционального фона. При реализации компьютерных дидактических игр должны использоваться дополнительные возможности, предоставляемые компьютерной средой: мультимедийные эффекты, интерактивное взаимодействие.

Компьютерные технологии позволили широко применять в обучении игровые программы. Всякая игровая программа по своей сути дидактическая игра. Предметная область для ученика не выступает явно, но он овладевает ею по мере выполнения игровых действий.

В результате дипломной работы построена игровая обучающая программа по математике для учащихся 3-6 классов, состоящая из трех разделов, в каждом из которых реализованы учебные, тренажерные и игровые задания, способствующие усвоению, тренировке и развитию математических навыков. Благодаря красочному оформлению и интересной постановке задач, реализованная обучающая игровая программа способна мотивировать ее использование детьми многократно.

Список использованных источников

1. Лэнгдон.Н, Снейп Ч. С математикой в путь:Пер.Л33 с англ. - М.: Педагогика,1987. – 48с.
2. Кордемский Б.А. Математическая смекалка. - М.: Наука. 1965. – 567с.
3. Выгодский М.Я. Справочник по элементарной математике. – М.: Наука. 1986. – 416с.
4. Кордемский Б.А. Увлечь школьников математикой. - М.: Просвещение.1981. – 111с.
5. И.А. Барташникова, А.А. Барташников. Учись играя. Харьков, ФОЛИО, 1997г. – 95с.

Приложение

С математикой в путь...
Темы Контрольные Игры

Таблица умножения

Х	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28		
5								
6								
7								
8								
9								

Проверить Узнать

Таблица умножения:

Ты можешь узнать интересные свойства таблицы умножения, если нажмешь кнопку **"Узнать"**. Но ты можешь проверить свои знания таблицы умножения сразу, нажав на кнопку **"Проверить"**.

К содержанию

Рисунок А.1 – Кадр темы «Живая таблица умножения».

С математикой в путь....
Темы Контрольные Игры

Таблица умножения

×	2	3	4	5	6	7	8	9
2	4	6	8	10	12			
3	6	9	12	15	18			
4	8	12	15	21	24			
5	10	15	21	25	30			
6	12	18	24	30	36			
7								
8								
9								

Начало задания

Результат проверки знания таблицы умножения

Числа выделенные красным цветом - это твои ошибки. Ты можешь начать задание сначала, нажав соответствующую кнопку - "Начало задания".

К содержанию

Рисунок А.2 – Кадр проверки знания таблицы умножения.

С математикой в путь...
 Темы Контрольные Игры

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17	18	19	20
	21	22	23	24	25	26	27	28	29	30
	31	32	33	34	35	36	37	38	39	40
	41	42	43	44	45	46	47	48	49	50
	51	52	53	54	55	56	57	58	59	60
	61	62	63	64	65	66	67	68	69	70
	71	72	73	74	75	76	77	78	79	80
	81	82	83	84	85	86	87	88	89	90
	91	92	93	94	95	96	97	98	99	100

Простые числа

На этом столбчатом квадрате выделены красным цветом **простые числа**. Все целые числа, кроме 1, имеют по меньшей мере два делителя: единицу и самого себя. Те из них, которые не имеют никаких других делителей, называются "**простыми**". Те числа, которые имеют еще и другие делители, называются "**составными**".

Дальше

К содержанию

Рисунок А.3 – Кадр темы «Простые числа».

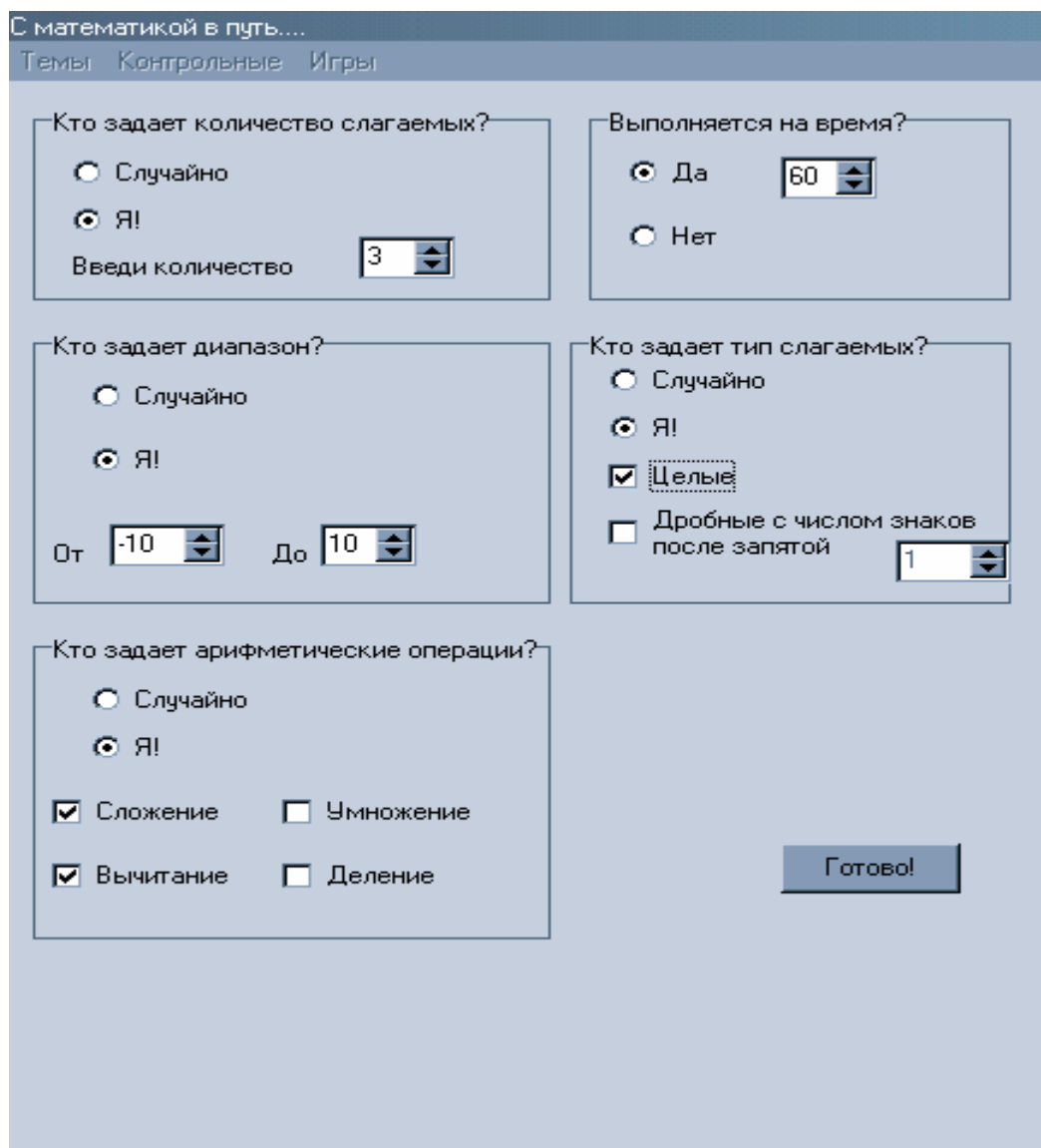


Рисунок А.4 – Кадр конструктора выражения.

С математикой в путь....
Темы Контрольные Игры

Вычисли значение выражения

Введи свой ответ

$(-7)-2-2 =$

Проверить

Проверка вычисления выражения

Неправильно $(-7)-2-2 = -11$

Следующее выражение

Вычисли выражение

Вычисли предложенное тебе выражение. После этого введи свой ответ в поле "Введи свой ответ". Для того, чтобы проверить правильность своих вычислений, нажми кнопку "Проверить".

К содержанию

Рисунок А.5 – Пример задания «Решить выражение».

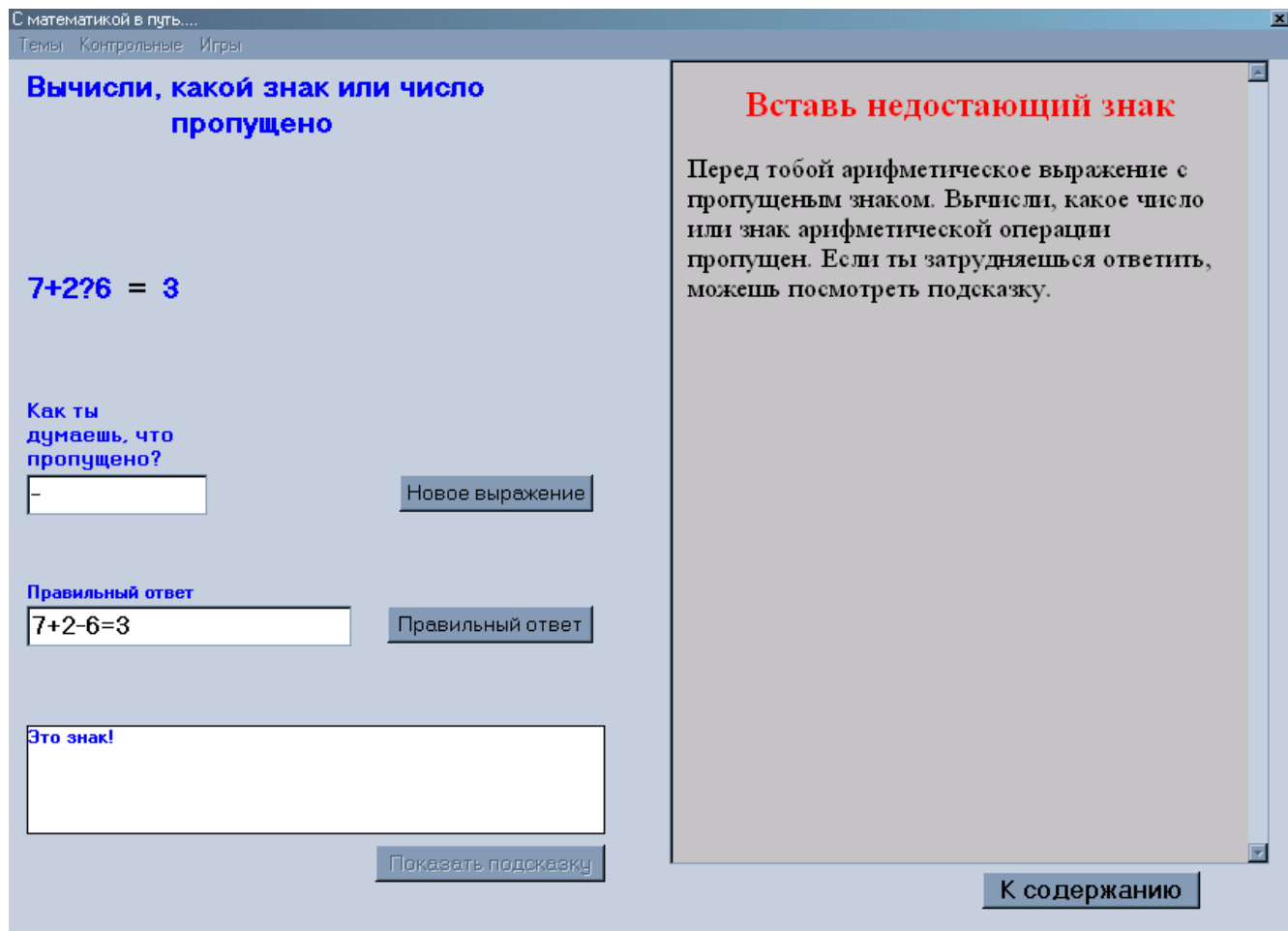


Рисунок А.: - Кадр задания «Вставить недостающий знак».

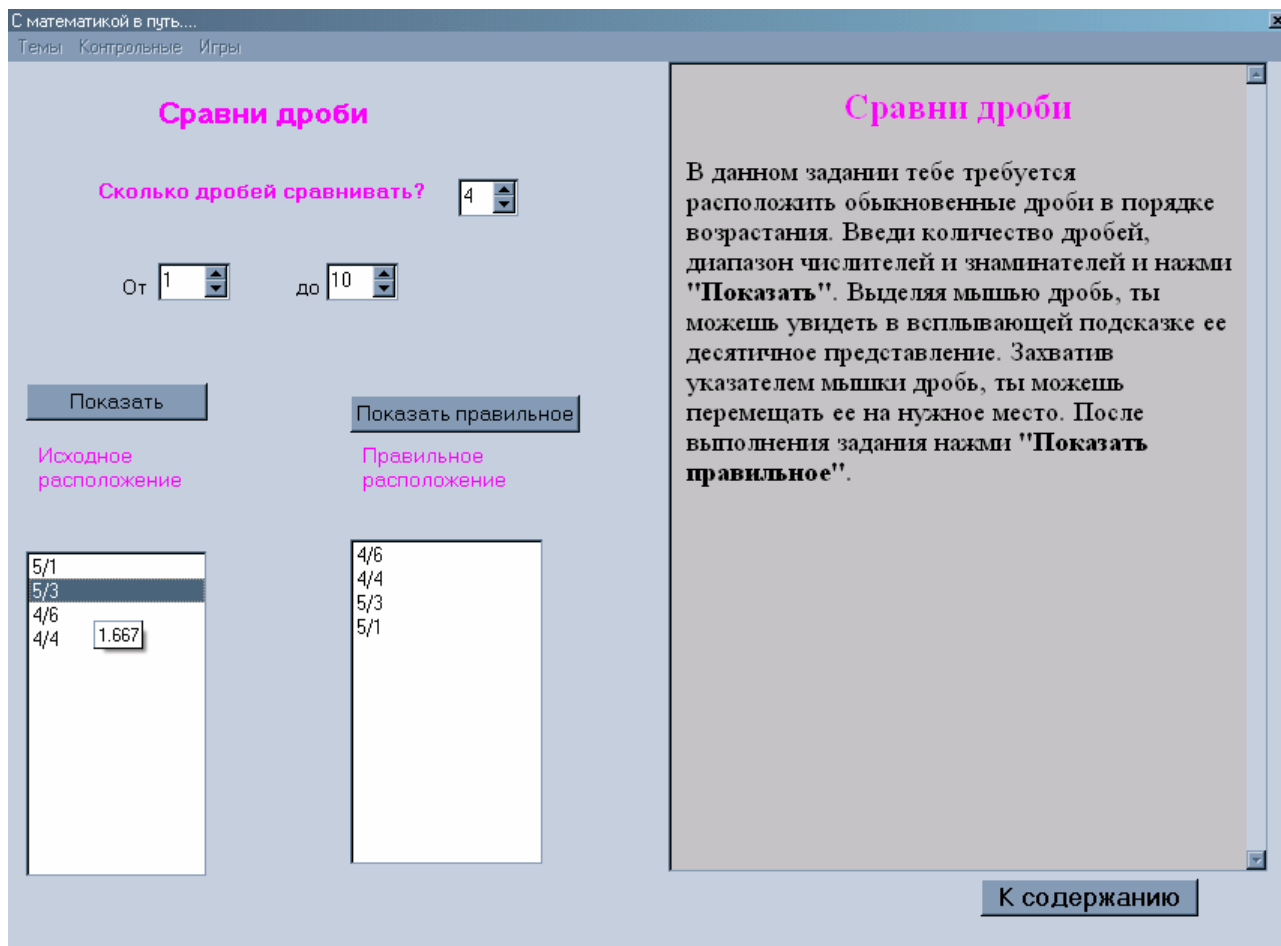


Рисунок А.7. – Кадр задания «Сравнить дроби».

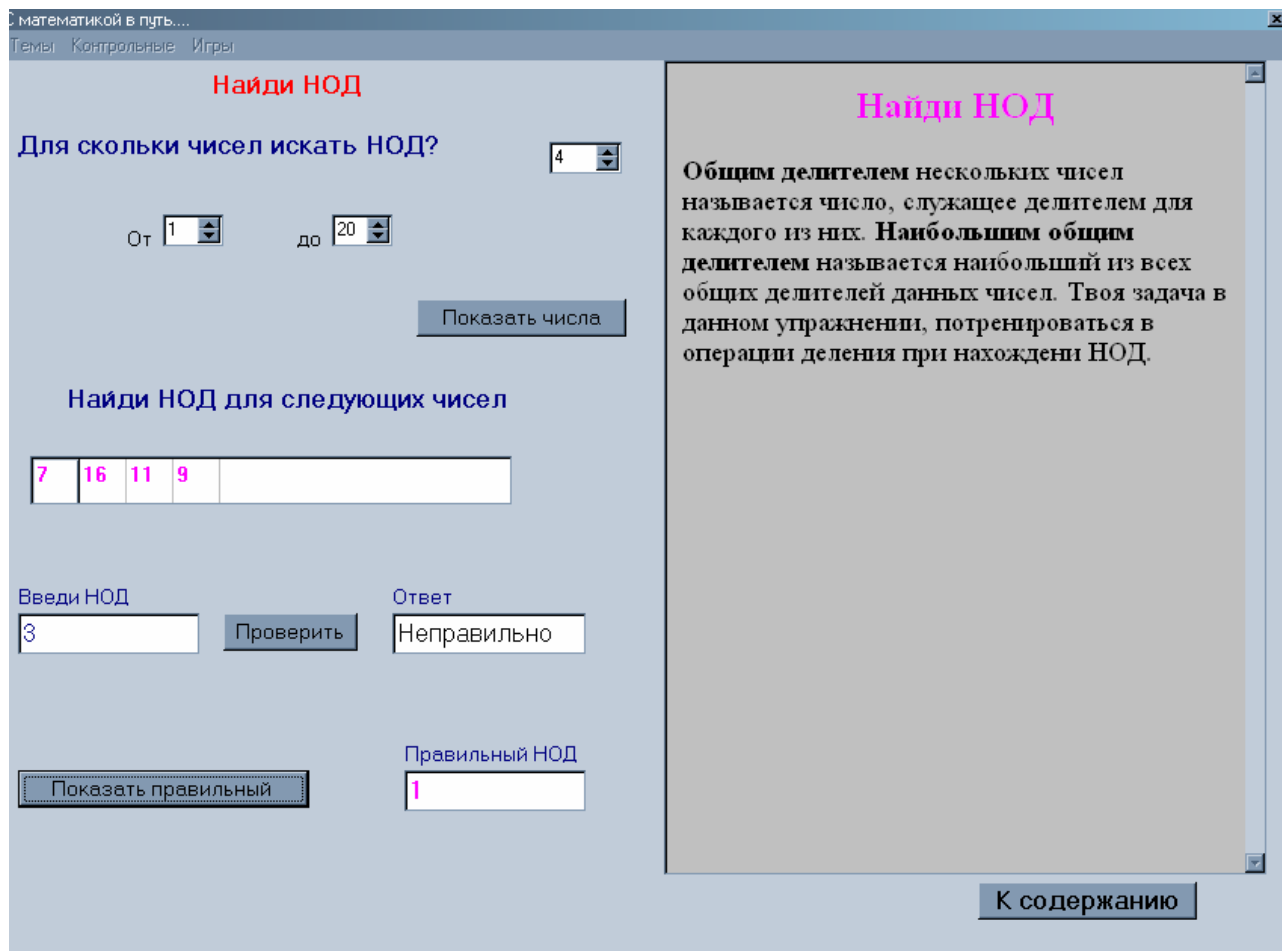


Рисунок А.8. – Кадр задания «Найди НОД».

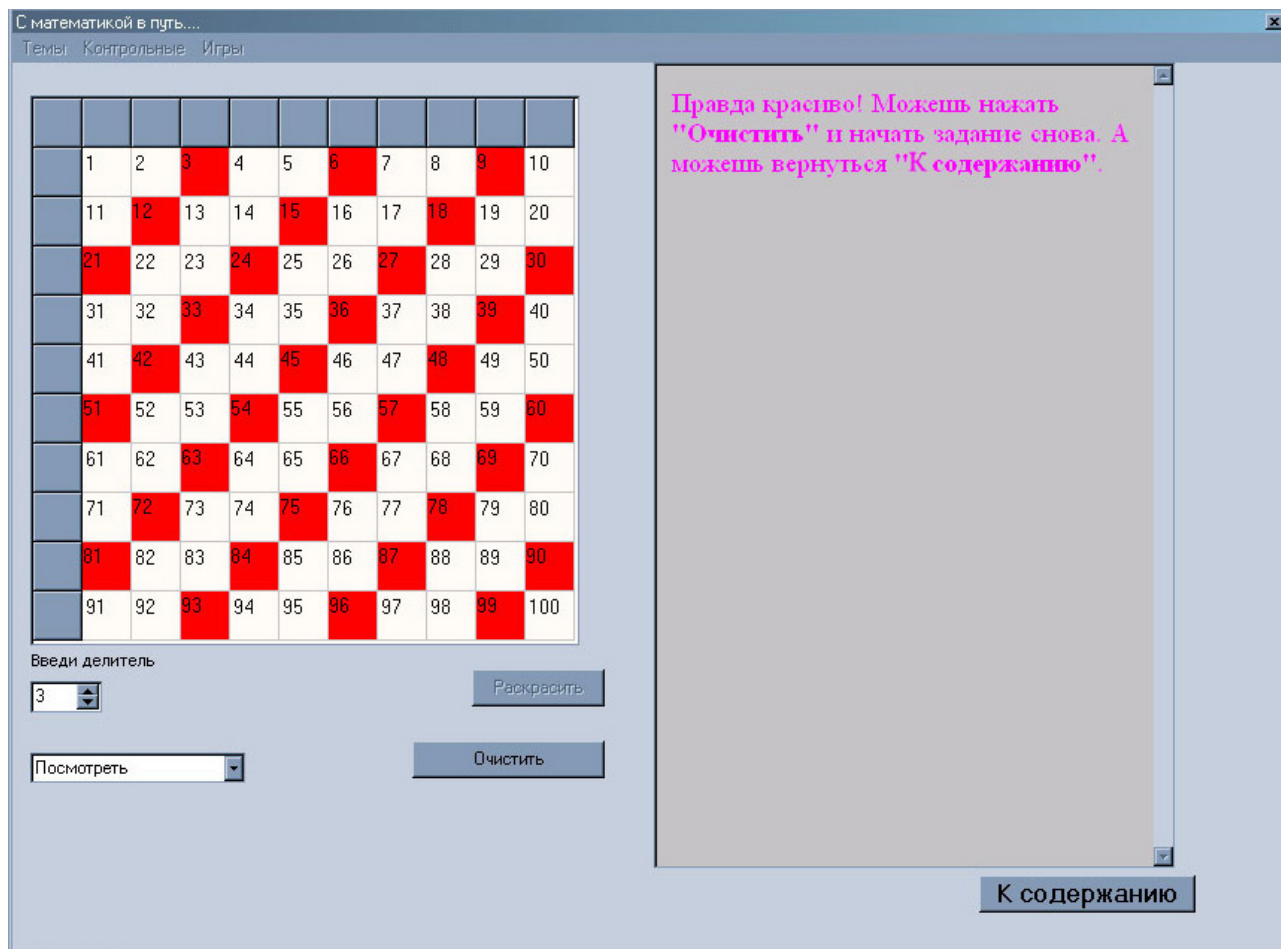


Рисунок А.9. – Кадр первого варианта игры «Столчатый квадрат».

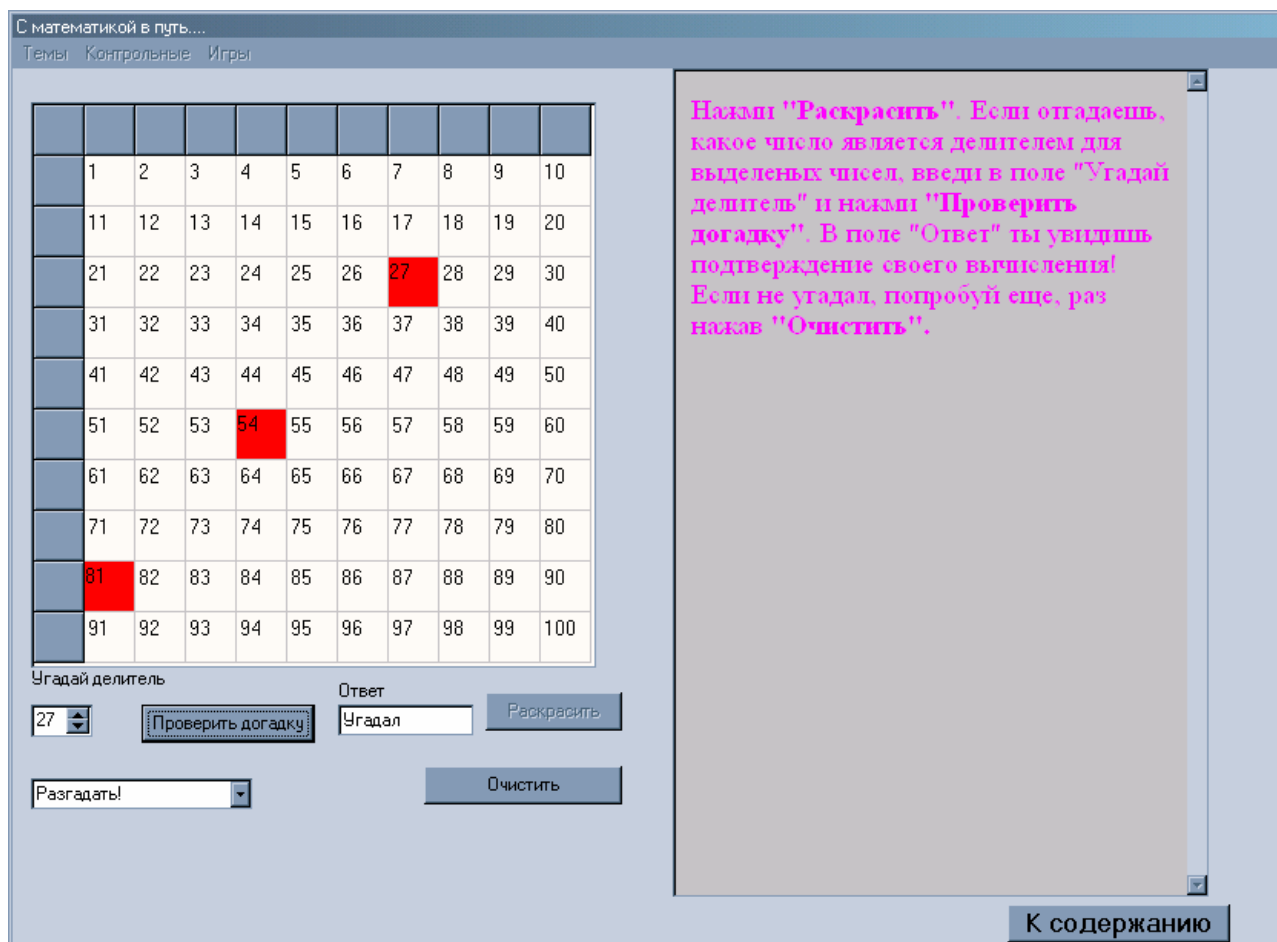


Рисунок А.10. – Кадр второго варианта игры «Стоклеточный квадрат».

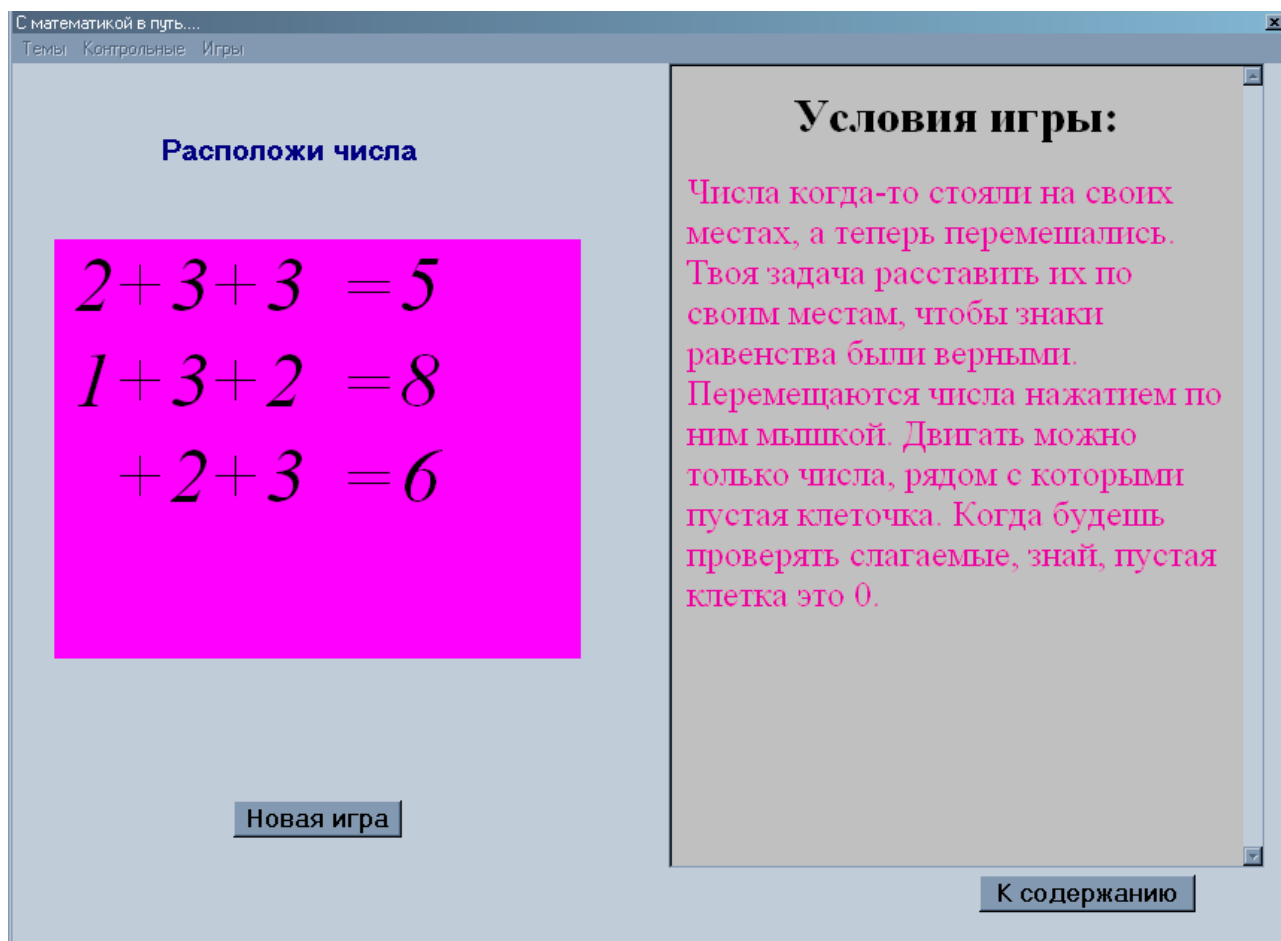


Рисунок А.11 – Кадр игры «Расположи числа».

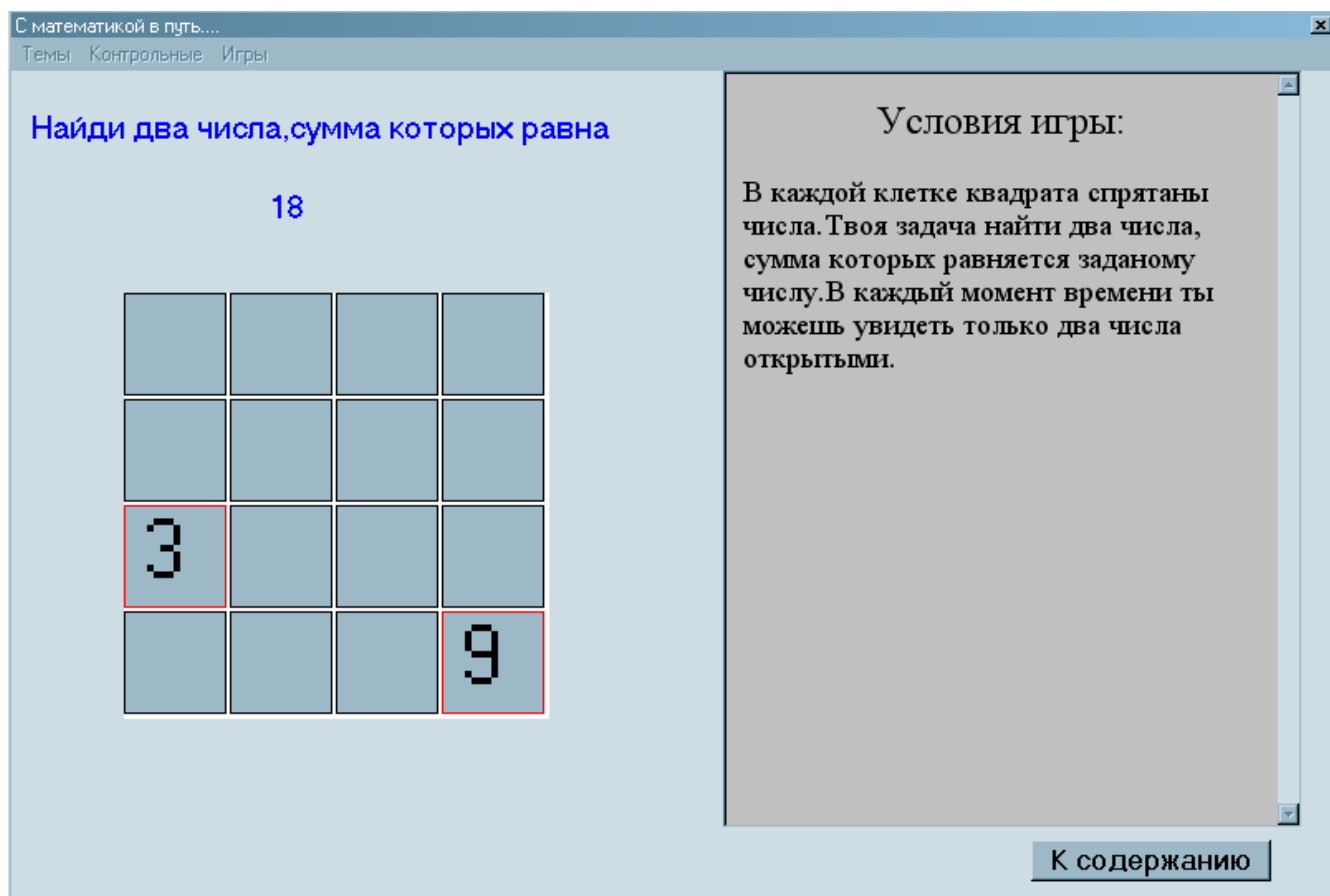


Рисунок А.12 – Кадр игры «Найди числа».