**Введение**

В условиях современного научно-технического прогресса, массовой информатизации и коммуникации общество испытывает острую потребность интенсивно познавать и реализовывать возможности информационных и коммуникационных технологий. Безусловно, что общая тенденция применения ИКТ не могла не коснуться сферы образования, в частности дошкольного. В России возможности информационных и коммуникационных технологий в дошкольных учреждениях применяются недостаточно, особенно это касается регионов, однако, развитие в этом направлении ведется и ему уделяется все большее внимание. Кроме того, ИКТ можно рассматривать как неотъемлемый элемент современной домашней образовательной среды, поскольку у большинства семей средства ИКТ используются дома, и дети с младшего дошкольного возраста, а иногда и намного раньше, активно взаимодействуют с ними. Особое место в таком взаимодействии принадлежит компьютеру и компьютерным играм. В обществе широко распространено мнение, что такие игры оказывают негативное влияние на развитие дошкольников. Такое мнение кажется не совсем верным, т.к. процесс общения с компьютером в дошкольном возрасте целиком зависит от взрослых и при его правильной организации предоставляет большие образовательные и развивающие возможности с максимальным исключением возможных негативных последствий.

Целью дипломного проекта является разработка программы, позволяющей преподавателям или родителям использовать компьютерную технику для развития и образования детей дошкольного возраста, а так же технической документации на программный продукт. Актуальность и практическая значимость проекта позволяют ускорить процесс обучения, освободить преподавателя от бумажной работы. А так же использования компьютера, как средство обучения, имеет такое преимущество, как привлечение внимания ребенка.

Процесс воспитания и образования детей должен быть подчинен конкретным целям. Педагогу или родителю необходимо четко представлять результат, который он хочет получить в ходе этого процесса, отбирать и применять конкретные средства и методы соответственно поставленной цели. Сейчас основной задачей образования и воспитания становится формирование личности ребенка. Наиболее подходящим является период дошкольного возраста, т.к. именно в это время происходит формирование основных свойств личности. Самой естественной и доступной формой познания в этом возрасте является игра. Игра, как особый вид деятельности, представляет собой процесс, суть которого можно обозначить как отображение и преображение действительности, в ходе которого происходит усвоение общественного опыта, отработка умений и становление навыков. Компьютерная игра - это компьютерная программа, направлена на организацию этого процесса. Существует огромное множество компьютерных игр, основную разновидность, пожалуй, представляют развлекательные игры различных вариантов. В случае же организации процесса воспитания и образования дошкольников используются компьютерные игры, не просто направленные на игровую деятельность, а являющиеся средством образования и воспитания, представленным в игровой форме. Это развивающие и обучающие компьютерные игры.

Важное значение имеет то, что развивающие и обучающие компьютерные игры не только эмоционально привлекательны для детей, но и содержательны, а значит их можно подчинить педагогическим целям. Правильное их применение позволяет скорректировать развитие ребенка, сделать процесс усвоения знаний более успешным. Нужно понимать, что компьютерные игры должны являться лишь дополнением основных видов деятельности детей, а не замещать их. Кроме того, представляя собой наглядно-символический тип информации, они отвечают особенностям познавательных процессов дошкольников, т.к. у них преобладает наглядно-образный тип мышления, но, не смотря на это, следует соблюдать принцип постепенного перехода от простого к сложному. Также важно подбирать такие игры соответственно возрасту и общему развитию. Вот здесь как раз и проявляется замечательная возможность реализации индивидуального подхода к обучению. Ведь квалифицированный педагог, да и просто внимательный родитель, с легкостью может подобрать компьютерные игры, соответствующие индивидуальным особенностям и потребностям ребенка.

Актуальность и практическая значимость разработки состоит в том, что использование данной программы позволяет в игровой форме концентрировать внимание ребенка на различных объектах, обучать его. Программа имеет дружественный пользовательский интерфейс, удобный для работы самых маленьких детей. Программа может быть использована как в домашних условиях, так и в детских садах, других организациях, связанных с работой психологов и воспитателей с детьми.

**1 Проектирование программы**

**1.1 Анализ технического задания**

В ходе дипломного проекта необходимо разработать развивающую программу для детей. Программа должна включать в себя тесты для обучения детей в различных областях. При неверном ответе программа будет возвращать тест к исходному положению с возможностью ответить заново.

Программа должна иметь 2 режима работы: «Преподаватель» и «Ребенок».

В режиме «Преподаватель» преподавателю будет предоставлена возможность просмотреть отчет по детям. Отчет должен содержать: Ф.И.О. ребенка, количество попыток, время на прохождение.

В режиме «Ребенок» ребенка необходимо будет зарегистрировать преподавателю (ввести Ф.И.О, выбрать тест), затем ребенок выполняет тест.

Разработать интуитивно-понятный пользовательский интерфейс.

Программный продукт может быть реализован с помощью различных средств разработки. Из массы программных средств в настоящее время существует большое количество способов создания программного обеспечения аналогичного «Развивающей программы для детей». Для решения поставленной задачи можно выделить такие программные средства как языки программирования, табличные процессоры и даже базы данных.

С помощью любого из этих средств можно обеспечить полное выполнение технического задания на курсовое проектирование.

Для создания программного продукта «Развивающая программа для детей» наиболее удобным средством является инструментальная программная среда разработки программных продуктов, так как это наиболее наглядное и простое средство. В Borland Delphi можно разработать наиболее удобный пользовательский интерфейс с помощью технологии Drag&Drop. Эта технология удобна своей наглядностью и удобностью использования.

Таким образом, для разработки программного продукта наиболее эффективно использовать инструментальные программные средства создания программных продуктов, позволяющие учитывать все требования технического задания проекта.

Развивающая программа для детей, разработанная в дипломном проекте, может использоваться не только как самостоятельный продукт, но и совместно с образовательным процессом. Это обстоятельство позволяет рассматривать данную программу как универсальную.

**1.2 Описание предметной области**

**Внести информацию о диагностике и результатах тестированя**

Возможности компьютера позволяют увеличить объём предлагаемого для ознакомления материала. Яркий светящийся экран привлекает внимание, даёт возможность переключить у детей аудиовосприятие на визуальное, анимационные герои вызывают интерес, в результате снимается напряжение. Но на сегодня, к сожалению, существует недостаточное количество хороших компьютерных программ, которые предназначены для детей данного возраста.

Специалисты выделяют ряд требований, которым должны удовлетворять развивающие программы для детей:

- развитие широкого спектра навыков и представлений;

- занимательность.

Существующие на рынке обучающие программы для данного возраста можно классифицировать следующим образом:

1. Игры для развития памяти, воображения, мышления и др.

2. "Говорящие" словари иностранных языков с хорошей анимацией.

3. АРТ-студии, простейшие графические редакторы с библиотеками рисунков.

4. Игры-путешествия, "бродилки".

5. Простейшие программы по обучение чтению, математике и др.

Использование таких программ позволяет не только обогащать знания, использовать компьютер для более полного ознакомления с предметами и явлениями, находящимися за пределами собственного опыта ребенка, но и повышать креативность ребенка; умение оперировать символами на экране монитора способствует оптимизации перехода от наглядно-образного к абстрактному мышлению; использование творческих и режиссерских игр создает дополнительную мотивацию при формировании учебной деятельности; индивидуальная работа с компьютером увеличивает число ситуаций, решить которые ребенок может самостоятельно.

Таким образом, применение компьютерной техники позволяет оптимизировать коррекционно-педагогический процесс, индивидуализировать обучение детей с нарушениями развития и значительно повысить эффективность любой деятельности.

Кроме того, в процессе замысла, создания новых заданий для коррекционно-развивающих занятий с использованием компьютера и мультимедийного проектора, развиваются и совершенствуются креативные качества педагога, растёт уровень его профессиональной компетентности. Желание взрослого разнообразить деятельность детей, сделать занятия ещё более интересными и познавательными, выводит их на новый виток общения, взаимопонимания, развивает личностные качества детей, способствует отличной автоматизации полученных на занятиях навыков на новом коммуникативном этапе педагогического и коррекционного воздействия. Таким образом, информатизация образования открывает воспитателям и учителям новые пути и средства педагогической работы.

Компьютер, мультимедийные средства – инструменты для обработки информации, которые могут стать мощным техническим средством обучения, коррекции, средством коммуникации, необходимыми для совместной деятельности педагогов, родителей и дошкольников.

Любого педагога и родителя волнует вполне закономерный вопрос о возможном негативном воздействии компьютерной техники на организм ребенка. Обычно детям дошкольного возраста рекомендуют находиться от 15 до 20 минут, а то и не более 10 минут. Многие дети проводят за компьютером гораздо больше времени без каких-либо вредных последствий. Признавая, что компьютер — ново-мощное средство интеллектуального развития детей, необходимо помнить, что его использование в развитии детей старшего дошкольного возраста требует тщательной организации, как самих занятий, так и всего режима в целом.

Вместе с тем, педагог должен в совершенстве знать содержание всех компьютерных программ, их операционную характеристику (специфику технических правил действия с каждой из них). Построение каждой игры имеет свои особенности. Занятие с одной подгруппой, включающее деятельность детей за компьютером, познавательную беседу, игру, гимнастику для глаз и др. может длиться от 20 до 25 минут. При этом дети могут быть за экраном не более 7-10 минут. Учитывая, что одновременно занимаются не более 3-5 детей (это напрямую зависит от количества компьютеров и выбранной формы работы), для обслуживания одной возрастной группы требуется не менее 2 часов.

Только педагог — мастер может вынести такую нагрузку и правильно использовать компьютер как мощное средство индивидуального воздействия на интеллектуальное, эмоциональное и нравственное развитие каждого малыша. Педагоги стараются работать творчески. Педагог — всегда исследователь. Он умело переключает внимание с поведения ребенка, его действий, эмоциональных проявлений, речи, на результаты, полученные в ходе игры, побуждая его к самостоятельному поиску путей и достижению поставленных в игре целей. Побуждая детей к принятию или постановке игровых задач, поиску своих вариантов ее выполнения, педагог развивает у них инициативу, творческий потенциал. Ребенок в игре всегда предстает не как исполнитель, как творец своей деятельности. Основная цель педагога — не выучить ту или иную компьютерную программу с детьми, а использовать ее игровое содержание для развития памяти, мышления, воображения, речи у конкретного ребенка. А этого можно достигнуть, если сам малыш с удовольствием выполняет всю программу. Каждая компьютерная математическая игра проводится с учетом основных компонентов комплексного метода. От педагогического мастерства зависит то, как ненавязчиво и незаметно оживить, расширить, закрепить полученный детьми опыт. Игровые проблемные ситуации, связанные с содержанием игровых задач, могут быть созданы посредством незначительных изменений в содержании математических игр.

Говоря об использовании компьютера детьми раннего возраста, встает вопрос о сохранении здоровья и зрения. Разумно сделать ограничения занятий с ПК по времени, но непроизвольное внимание у детей данного возраста очень мало (10-15 минут), поэтому, как правило, дети не могут долго находиться за компьютером. Для детей 5-6 лет норма не должна превышать 10 минут. Периодичность занятий 2 раза в неделю. Нормально развивающийся ребенок в этом возрасте двигается 70-80% времени бодрствования, поэтому пока вопрос о "засиживании" за компьютером не актуален.

**1.3 Обоснование выбора инструментальных программных средств реализации программы**

Для реализации программного проекта рассмотрим следующие инструментальные средства: Borland Pascal, Borland Delphi.

Особенностями языка Borland Pascal являются строгая типизация и наличие средств структурного (процедурного) программирования. Паскаль был одним из первых таких языков. Язык способствует дисциплинированию программирования, поэтому, наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис интуитивно понятен даже при первом знакомстве с языком.

К недостаткам среды можно отнести: невозможность передачи функциям массивов переменной длины, отсутствие нормальных средств работы с динамической памятью, ограниченная библиотека ввода-вывода, отсутствие средств для подключения функций написанных на других языках, отсутствие средств раздельной компиляции и т. п.

Учитывая все выше сказанное среду Borland Pascal неэффективно использовать как средство реализации программного продукта дипломного проекта.

Delphi — это среда быстрой разработки, в которой в качестве языка программирования используется язык Delphi. Язык Delphi — строго типизированный объектно-ориентированный язык, в основе которого лежит хорошо знакомый программистам Object Pascal.

Delphi — это комбинация нескольких важнейших технологий:

высокопроизводительный компилятор в машинный код;

– объектно-ориентированная модель компонент;

– визуальное (а, следовательно, и скоростное) построение приложений из программных прототипов;

– масштабируемые средства для построения баз данных.

Borland Delphi позволяет создавать самые различные программы: от простейших однооконных приложений до программ управления распределенными базами. В состав пакета включены разнообразные утилиты, обеспечивающие работу с базами данных, XML-документами, создание справочной системы, решение других задач. Отличительной особенностью седьмой версии является поддержка технологии .NET.

В Delphi максимально производительно можно использовать код. Это позволяет очень быстро разрабатывать приложения, так как уже существуют заранее подготовленные объекты, а так же объекты, созданные самим пользователем. Язык Delphi — строго типизированный объектно-ориентированный язык, в основе которого лежит Object Pascal.

Delphi обладает удобным интерфейсом для создания графических оболочек.

Таким образом, для создания программного продукта наиболее эффективно использовать инструментальное программное средство Borland Delphi.

**2 Разработка программы**

**2.1 Разработка алгоритма для корректного выполнения программы.**

Блок-схема работы программы приведена на рисунке А.1 в приложении А.

Рассмотрим описание блок-схемы алгоритма работы программы.

При нажатии на «RadioButton» с соответствующими им подписями в «Label» задается время для выполнения каждого теста. При нажатии на «BeginButon» «Начать тест» вызывается процедура «Unit1.pas» привязанная к этой кнопке. Далее заполняется рабочее поле элементами, необходимыми для выполнения задания. По нажатии на вариант ответа, программа обрабатывается правильность ответа, если ответ является верным, тест переходит на следующий этап, запуская следующий вопрос или переходит к окну отчета, если вопросов и заданий больше нет. Если ответ является неверным, то программа восстанавливает все элементы данного задания на этом этапе в исходное положение и пользователь может ответить еще раз. Количество попыток не ограничено.

По истечении времени на выполнение теста, программа выводит сообщение о том, что время вышло и тест не пройден.

По окончании всех тестов программа подсчитывает количество попыток пользователя на каждом тесте и выводит их в финальном окне отчета.

Входные данные – это величины, которые задаются до начала работы алгоритма или определяются динамически во время его работы, которые берутся из определённого набора объектов.

Входными данными для данного программного продукта являются выбор времени на выполнение теста, вариант ответа, выбранный пользователем.

Выходными данными являются результаты прохождения теста.

**2.2 Разработка интерфейса пользователя**

Для данного программного продукта был создан интуитивно понятный и дружественный интерфейс, представленный в виде нескольких форм с определенными компонентами.

Форма начала работы программы представлена на рисунке 1.

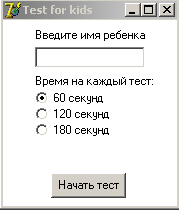


Рисунок 1 – Форма «StartForm»

Компоненты и свойства данной формы представлены в таблице 1.

Таблица 1 – Компоненты и свойства формы «StartForm»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Время на каждый  тест | Подпись выбора времени |
| RadioButton1 | Caption | 60 секунд | Задает время на выполнение теста |
| RadioButton2 | Caption | 120 секунд | Задает время на выполнение теста |
| RadioButton3 | Caption | 180 секунд | Задает время на выполнение теста |
| BeginButton | Отсутствует | Отсутствует | Выполняет действие, прописанное в коде |

Форма «Выбор теста» представлена на рисунке 2.

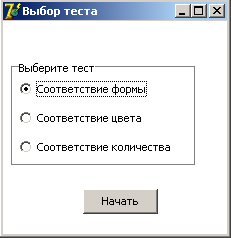


Рисунок 2 – Форма «Выбор теста»

Компоненты и свойства данной формы представлены в таблице 2.

Таблица 2 – Компоненты и свойства формы «Выбор теста»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| RadioGroup1 | Caption | Выберите тест | Название панели выбора |
| RadioGroup1 | Item 0 | Соответствие формы | Выбор соответствующего теста |
| RadioGroup1 | Item 1 | Соответствие цвета | Выбор соответствующего теста |
| RadioGroup1 | Item 2 | Соответствие количества | Выбор соответствующего теста |
| Button1 | Caption | Начать | Начинает выбранный тест |

Форма «Test1» представлена на рисунке 3.

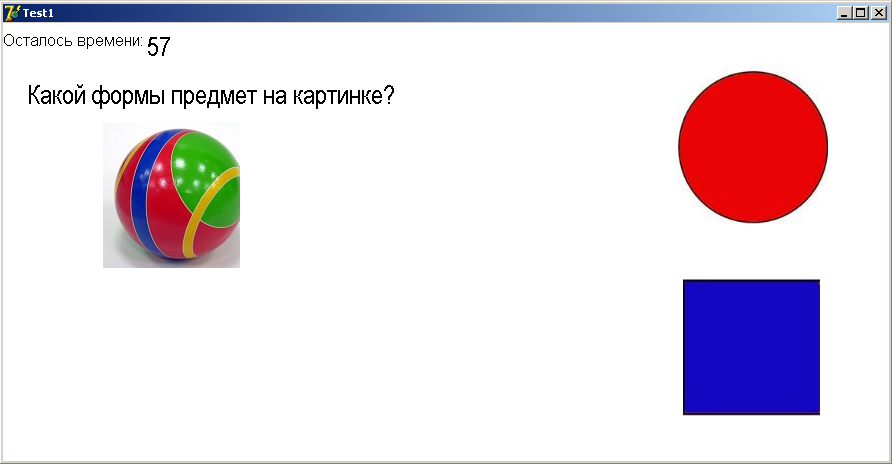


Рисунок 3 – Форма «Test2»

Компоненты и свойства данной формы представлены в таблице 3.

Таблица 3 – Компоненты и свойства формы «Test1»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Осталось времени | Подпись |
| Label2 | Caption | Результат кода | Выводит время до окончания теста |
| Label3 | Caption | Какой формы предмет на картинке? | Задание вопроса |
| Image1 | Image | Выбранная картинка | Задание |
| Image2 | Image | Выбранная картинка | Вариант ответа |
| Image3 | Image | Выбранная картинка | Вариант ответа |

Форма «Test2» представлена на рисунке 4.

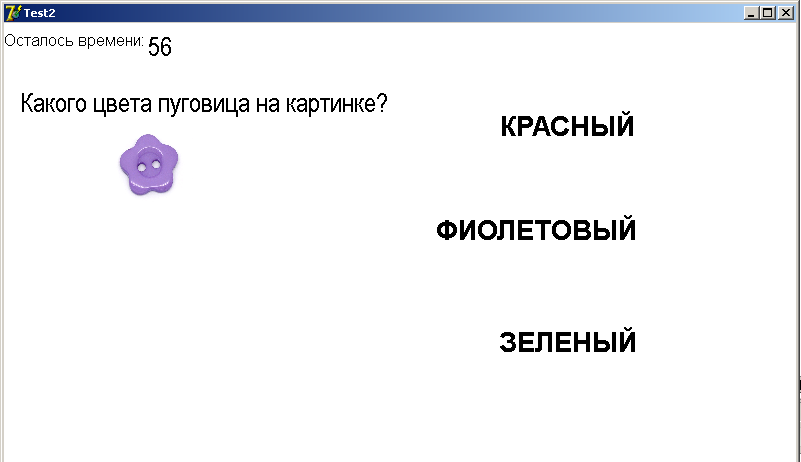


Рисунок 4 – Форма «Test2»

Компоненты и свойства данной формы представлены в таблице 4.

Таблица 4 – Компоненты и свойства формы «Test2»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Осталось | Подпись |
| Label2 | Caption | Результат кода | Выводит время до окончания теста |
| Label3 | Caption | Какого цвета пуговица на  картинке | Задание вопроса |
| Image1 | Image | Выбранная картинка | Задание |
| SpeedButton1 | Caption | Красный | Вариант ответа |
| SpeedButton2 | Caption | Фиолетовый | Вариант ответа |
| SpeedButton3 | Caption | Зеленый | Вариант ответа |

Форма «Test3» представлена на рисунке 5.

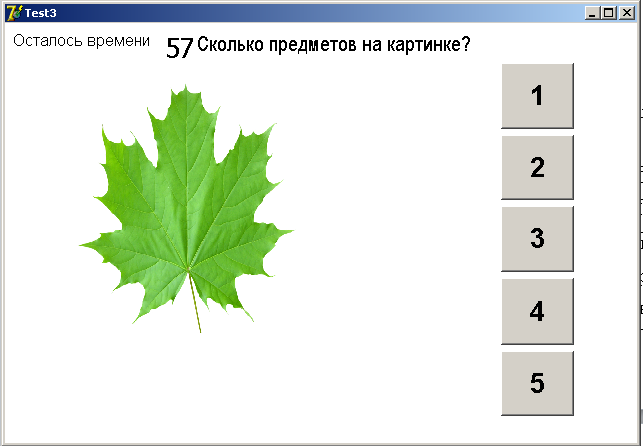


Рисунок 5 – Форма «Test3»

Компоненты и свойства данной формы представлены в таблице 5.

Таблица 5 – Компоненты и свойства формы «Test3»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Осталось времени |  |
| Label2 | Caption | Результат кода | Показывает время до конца теста |
| Label3 | Caption | Сколько предметов на картинке? | Задание вопроса |
| Image1 | Image | Выбранная картинка | Задание |
| Button1 | Caption | 1 | Вариант ответа |
| Button2 | Caption | 2 | Вариант ответа |
| Button3 | Caption | 3 | Вариант ответа |
| Button4 | Caption | 4 | Вариант ответа |
| Button5 | Caption | 5 | Вариант ответа |

Форма «Отчетная форма» представлена на рисунке 6.

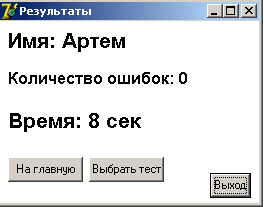


Рисунок 6 – «Отчетная форма»

Компоненты и свойства данной формы представлены в таблице 6.

Таблица 6 – Компоненты и свойства формы «Отчетная форма»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Имя |  |
| Label2 | Caption | Артем (имя ребенка, введенное в начале) | Выводит имя ребенка, проходившего тест |
| Label3 | Caption | Количество ошибок |  |
| Label4 | Caption | Результат кода | Выводит количество ошибок |
| Label5 | Caption | Время |  |
| Label6 | Caption | Результат кода | Выводит затраченное время на прохождение теста |
| Button1 | Caption | На главную | Возвращает программу к первой форме |
| Button2 | Caption | Выбрать тест | Возвращает программу к форме выбора теста |
| Button3 | Caption | Выход | Выходит из программы |

**2.3 Разработка программного кода**

**Добавить про формы и компоненты**

Программный код – это компьютерная программа, написанная на определённом языке программирования по алгоритму, заданному педагогическим и технологическим сценарием.

Появление современных систем визуального проектирования, таких как Visual

Basic и Delphi, в значительной степени упростили работу в данной среде, поскольку они позволяют разрабатывать интерфейс в интерактивном режиме. В то же время они не ограничивают свободу готовыми решениями, позволяя подстраивать работу под разный стиль.

Проанализировав техническое задание, было принято решение разработать следующий программный код для выполнения основных функций программы.

Программный код состоит из 6 unit (Test1, Test2, Test3 ResultUnit, StartUnit, ChooseUnit) и 6 форм (Test1, Test2, Test3 ResultForm, StartForm, ChooseForm).

StartUnit – данный раздел предназначен для выбора времени для выполнения теста, записи имени ребенка и кнопку, запускающую раздел «ChooseUnit». ChooseUnit – предназначен для выбора одного из трех тестов, который переходит к следующим разделам соответственно: Test1, Test2, Test3. Test1 – данный раздел содержит обработку отрисовки задания теста, а так же обработку правильности выбранного варианта ответа. Test2 и Test3 – данные разделы аналогичны разделу Test1. Отличием является лишь задание и варианты ответа. ResultUnit – данный раздел начинает свое выполнение по завершению выполнения разделов Test1, Test2 или Test3. Главными его функциями являются отображение результатов прохождения теста, возвращение к выбору теста, возвращение к главной форме программы и выход из программы.

**2.4 Отладка программы**

Отладка программы представляет собой поиск возможных ошибок в коде программы.

В ходе разработки данной программы были обнаружены синтаксические ошибки. Пример ошибки изображён на рисунке 7.

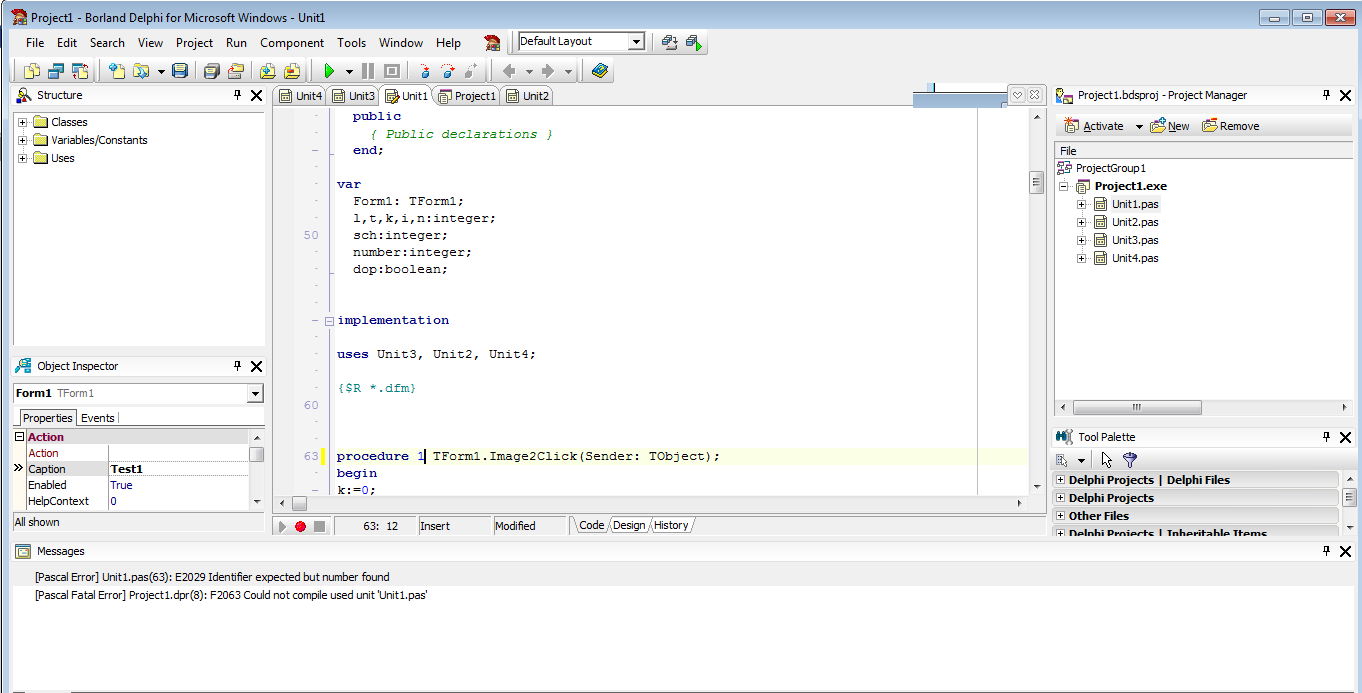


Рисунок 7 – Синтаксическая ошибка в коде программы.

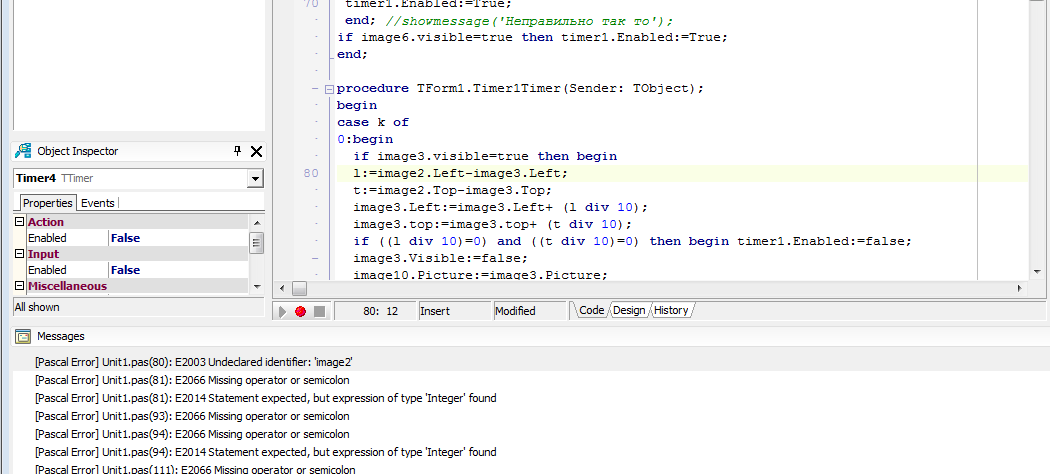
Также были логические ошибки, которые изображены на рисунке 8.

Рисунок 8 - Логическая ошибка в коде программы

Для отладки программы был использован интегрированный в среду разработки Delphi10 отладчик, вызвать его можно выполнив переход по адресу: Project → Options → Compiler. Перечень его возможностей изображён на рисунке 9.

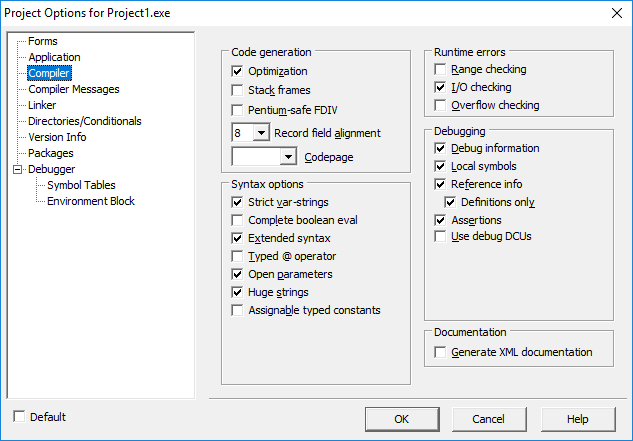


Рисунок 9 - Компилятор в Delphi10

Отладка была проведена с помощью следующих функций:

- Step Over (F8), которая выполняет следующую строку кода;

- Trace Into (F7), пошагово запускает код функции или подпрограммы, заходя во все вызываемые подпрограммы и запуская их в пошаговом режиме. Delphi выделяет строку, которая будет выполняться следующей, другим цветом и маленьким значком в виде стрелки, чтобы вы видели, что делает ваша программа;

Run to Cursor (F4), программа будет выполняться, пока не достигнет этой строки, так что это аналогично установке временной точки останова.

Функции Debug можно просмотреть по адресу: View → Debug Windows. Список всех возможностей Debug Windows изображён ниже на рисунке 10.

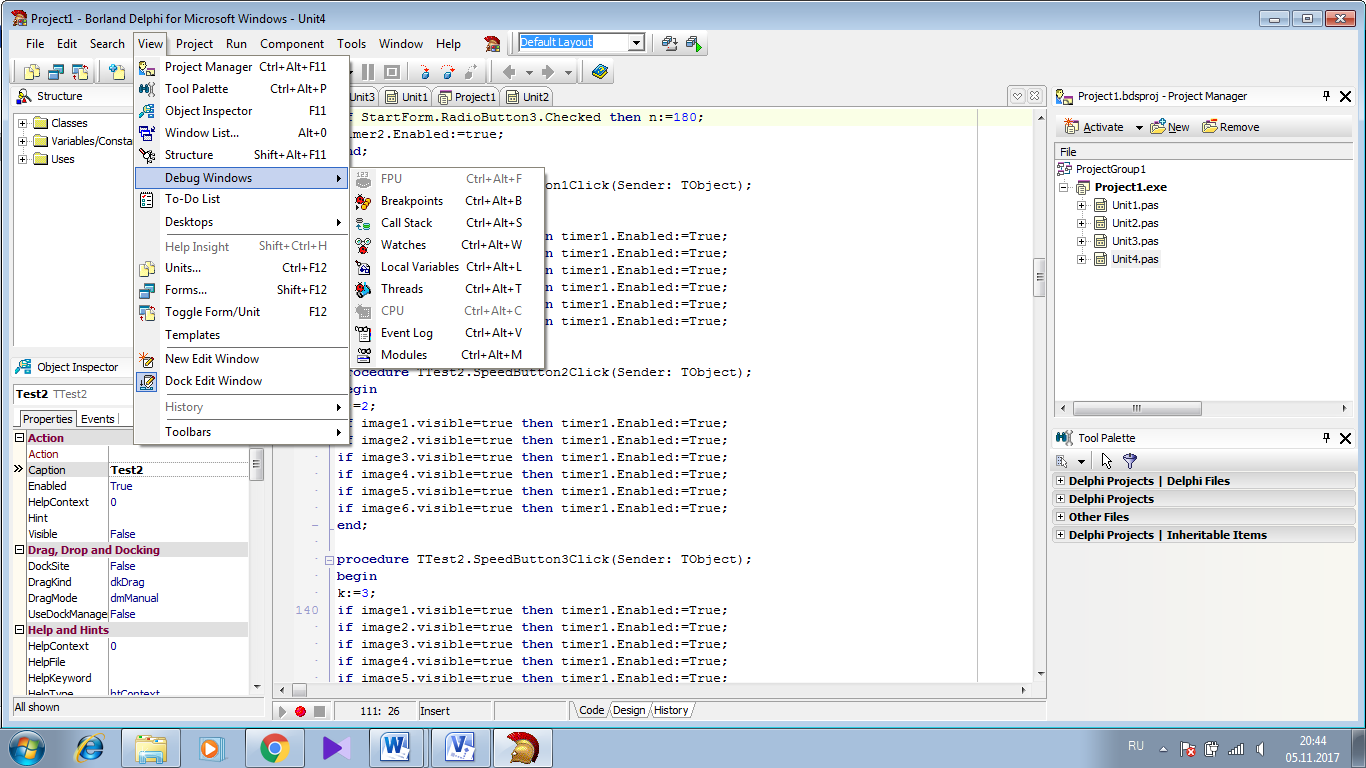


Рисунок 10 - Возможности Debug Windows

Пример работы программы до точки останова изображён на рисунке 11.

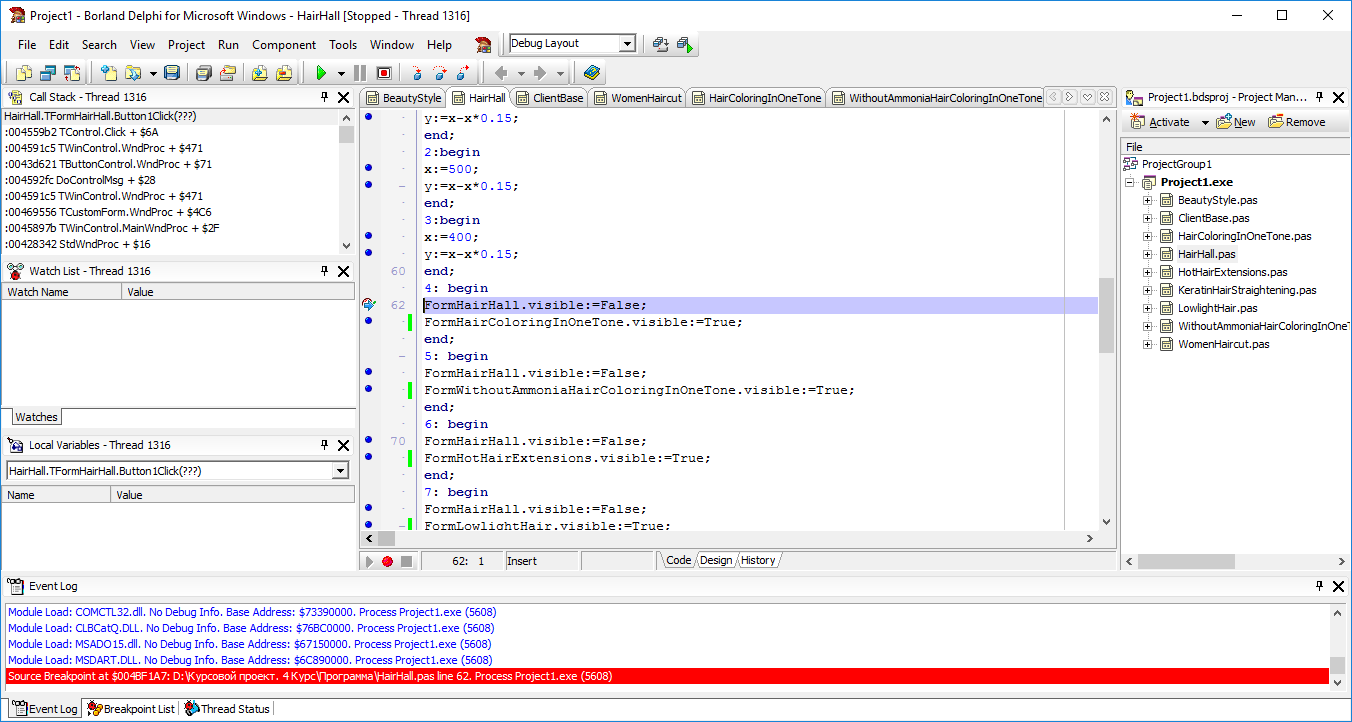


Рисунок 11 – Точка останова в Delphi10

Таким образом, была описана отладка данной программы, описаны все проблемы, встретившиеся при работе на программой, описаны используемые функции интегрированного в Delphi10 компилятора.

* 1. **Тестирование программы**

Для тестирования работы программы необходимо рассмотреть её функциональные возможности и корректность отображения данных.

Работа данной программы была протестирована с помощью динамического метода, который заключается в том, что все испытания проводились в разное время, на разных компьютерах с разными операционными системами.

Все тестирования проводилось на компьютерах, имеющие следующие технические и программные характеристики:

* ОС Windows 7 и ОС Windows 10;
* Система программирования Delphi 2010;
* Объем оперативной памяти не менее 4 ГБ.

В ходе данного тестирования ошибок и проблем в работе программы выявлено не было.

После старта работы программы она запускает форму стартового окна программы. Форма представлена на рисунке 12.

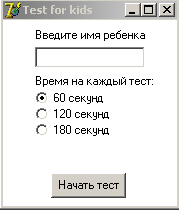


Рисунок 12 – Стартовая форма программы

При нажатии на переключатель времени задается время для прохождения каждого теста. Переключатель может быть выбран только один. В поле введите имя ребенка вводится имя ребенка, который должен проходить тест. По нажатию на кнопку «Начать тест» запускается форма выбора теста. Форма показана на рисунке 13.

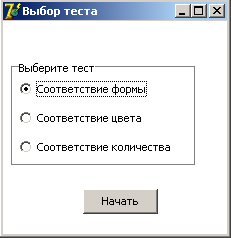


Рисунок 13 – Форма «Выбор теста»

На форме так же находится 3 переключателя, выбрать необходимо один из них, по нажатию на кнопку «Начать» запустится соответствующий тест. Тесты представлены на рисунках 14, 15, 16.

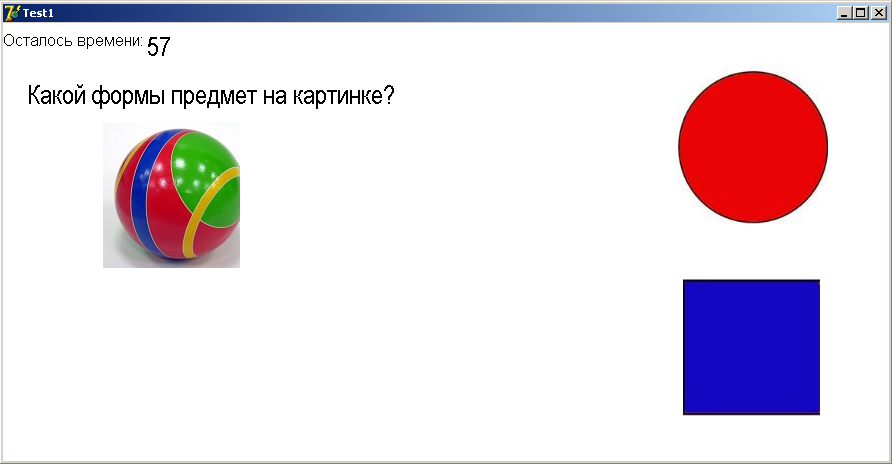


Рисунок 14 – Форма «Test1»

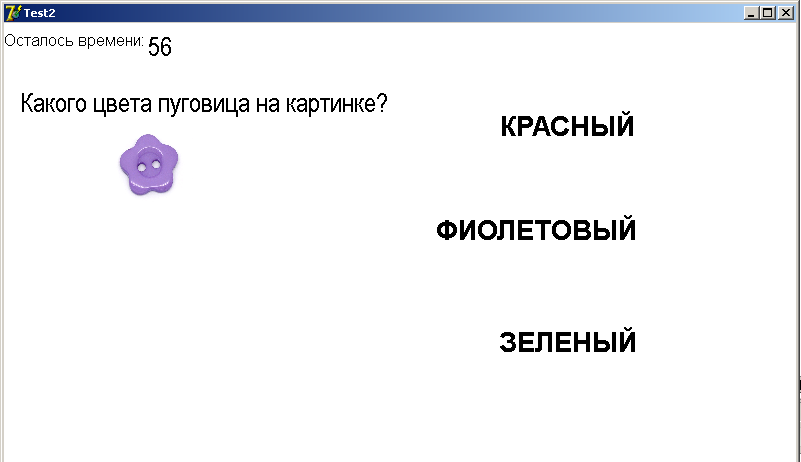


Рисунок 15 – Форма «Test2»

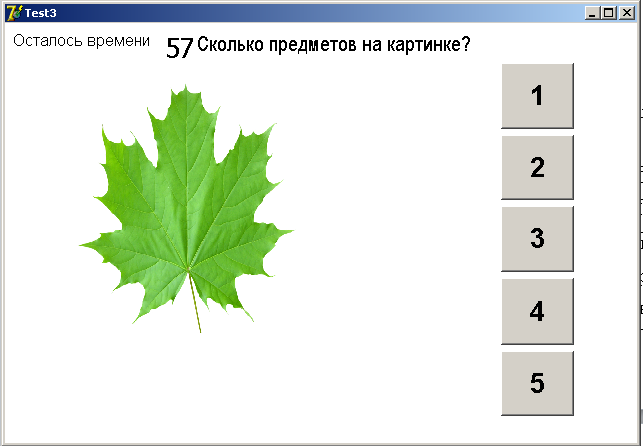


Рисунок 16 – Форма «Test3»

На формах представлено время до окончания теста, при завершении которого тест будет считаться проваленным. Для прохождения теста необходимо кликнуть по нужному варианту ответа. В случае верного ответа программа отобразит следующую картинку. При неверном результате программа возвращает все элементы к исходному положению и необходимо ответить еще раз, и так до тех пор, пока не будет пройден тест правильно. После прохождения теста запускается форма отчета. Форма представлена на рисунке 17.

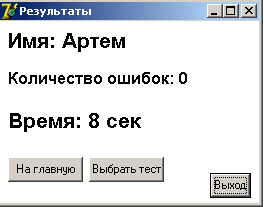


Рисунок 17 – Форма отчета

Форма содержит в себе результаты прохождения теста, а именно: имя тестируемого, количество допущенных ошибок и время, затраченное на прохождение теста.

Для управления программой на форму помещены 3 кнопки: «На главную», «Выбрать тест» и «Выход». По нажатию на кнопку «На главную» программа переходит к стартовой форме программы. При нажатии на кнопку «Выбрать тест» программа возвращается к форме выбора теста. При нажатии на кнопку «Выход» программа завершает свою работу.

Таким образом, было выполнено полное тестирование алгоритма работы программы. По результатам данного тестирования все компоненты и функции работают правильно, алгоритм выполняется корректно.

1. **Разработка технической документации**

**3.1 Программный код с комментариями**

Ниже представлен фрагмент листинга программы.

//формирование интерфейса теста №2 и задание значений необходимых переменных

procedure TTest2.FormActivate(Sender: TObject);

begin

time := 0;

errors := 0;

number:=2;

i:=0;

sch:=0;

image1.Visible:=true;

image1.Left:=100;

image1.top:=100;

image2.Left:=100;

image2.top:=100;

image3.Left:=100;

image3.top:=100;

image4.Left:=100;

image4.top:=100;

image5.Left:=100;

image5.top:=100;

image6.Left:=100;

image6.top:=100;

image7.Picture:=form1.image15.Picture;

image8.Picture:=form1.image15.Picture;

image9.Picture:=form1.image15.Picture;

image10.Picture:=form1.image15.Picture;

image11.Picture:=form1.image15.Picture;

image12.Picture:=form1.image15.Picture;

image13.Picture:=form1.image15.Picture;

image14.Picture:=form1.image15.Picture;

image15.Picture:=form1.image15.Picture;

image16.Picture:=form1.image15.Picture;

image17.Picture:=form1.image15.Picture;

image18.Picture:=form1.image15.Picture;

image19.Picture:=form1.image15.Picture;

image20.Picture:=form1.image15.Picture;

image21.Picture:=form1.image15.Picture;

image22.Picture:=form1.image15.Picture;

image23.Picture:=form1.image15.Picture;

image24.Picture:=form1.image15.Picture;

if StartForm.RadioButton1.Checked then n:=60;

if StartForm.RadioButton2.Checked then n:=120;

if StartForm.RadioButton3.Checked then n:=180;

Timer2.Enabled:=true;

end;

//Выбор первого варианта ответа и отслеживание правильности, а также отрисовка анимации

procedure TTest2.SpeedButton1Click(Sender: TObject);

begin

k:=1;

if image1.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image2.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image3.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

if image4.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image5.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

if image6.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

end;

//Выбор первого варианта ответа и отслеживание правильности, а также отрисовка анимации

procedure TTest2.SpeedButton2Click(Sender: TObject);

begin

k:=2;

if image1.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

if image2.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image3.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image4.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image5.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image6.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

end;

//Выбор первого варианта ответа и отслеживание правильности, а также отрисовка анимации

procedure TTest2.SpeedButton3Click(Sender: TObject);

begin

k:=3;

if image1.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image2.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

if image3.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image4.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; end;

if image5.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

if image6.visible=true then begin timer1.Enabled:=True; timer4.Enabled:=false; errors := errors + 1; end;;

end;

//Анимация движения объектов на форме, а так же подсчет правильных ответов и ошибок

procedure TTest2.Timer1Timer(Sender: TObject);

begin

case k of

1:begin

if image1.visible=true then begin

l:=image7.Left-image1.Left;

t:=image7.Top-image1.Top;

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image1.Left);

t:=abs(100-image1.Top);

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

timer1.Enabled:=false;

//image1.Visible:=false;

//image7.Picture:=image1.Picture;

//image2.Visible:=true;

//sch:=sch+1;

end;

end;

if image2.visible=true then begin

l:=image7.Left-image2.Left;

t:=image7.Top-image2.Top;

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image2.Left);

t:=abs(100-image2.Top);

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

timer1.Enabled:=false;

{image2.Visible:=false;

image8.Picture:=image2.Picture;

image3.Visible:=true;

sch:=sch+1;}

end;

end;

if image3.visible=true then begin

l:=image7.Left-image3.Left;

t:=image7.Top-image3.Top;

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image3.Visible:=false;

image9.Picture:=image3.Picture;

image4.Visible:=true;

i:=i+1;

sch:=sch+1;

end;

end;

if image4.visible=true then begin

l:=image7.Left-image4.Left;

t:=image7.Top-image4.Top;

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image4.Left);

t:=abs(100-image4.Top);

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

timer1.Enabled:=false;

{image4.Visible:=false;

image10.Picture:=image4.Picture;

image5.Visible:=true;

sch:=sch+1; }

end;

end;

if image5.visible=true then begin

l:=image7.Left-image5.Left;

t:=image7.Top-image5.Top;

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image5.Visible:=false;

image11.Picture:=image5.Picture;

image6.Visible:=true;

i:=i+1;

sch:=sch+1;

end;

end;

if image6.visible=true then begin

l:=image7.Left-image6.Left;

t:=image7.Top-image6.Top;

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image6.Left);

t:=abs(100-image6.Top);

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

timer1.Enabled:=false;

{image6.Visible:=false;

image12.Picture:=image6.Picture;

sch:=sch+1; }

end;

end;

end;

2:begin

if image1.visible=true then begin

l:=image13.Left-image1.Left;

t:=image13.Top-image1.Top;

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image1.Visible:=false;

image13.Picture:=image1.Picture;

image2.Visible:=true;

i:=i+1;

sch:=sch+1;

end;

end;

if image2.visible=true then begin

l:=image13.Left-image2.Left;

t:=image13.Top-image2.Top;

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image2.Left);

t:=abs(100-image2.Top);

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

timer1.Enabled:=false;

{image2.Visible:=false;

image14.Picture:=image2.Picture;

image3.Visible:=true;

sch:=sch+1; }

end;

end;

if image3.visible=true then begin

l:=image13.Left-image3.Left;

t:=image13.Top-image3.Top;

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image3.Left);

t:=abs(100-image3.Top);

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

timer1.Enabled:=false;

{image3.Visible:=false;

image15.Picture:=image3.Picture;

image4.Visible:=true;

sch:=sch+1;}

end;

end;

if image4.visible=true then begin

l:=image13.Left-image4.Left;

t:=image13.Top-image4.Top;

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image4.Left);

t:=abs(100-image4.Top);

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

timer1.Enabled:=false;

{image4.Visible:=false;

image16.Picture:=image4.Picture;

image5.Visible:=true;

sch:=sch+1;}

end;

end;

if image5.visible=true then begin

l:=image13.Left-image5.Left;

t:=image13.Top-image5.Top;

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image5.Left);

t:=abs(100-image5.Top);

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

timer1.Enabled:=false;

{image5.Visible:=false;

image17.Picture:=image5.Picture;

image6.Visible:=true;

sch:=sch+1; }

end;

end;

if image6.visible=true then begin

l:=image13.Left-image6.Left;

t:=image13.Top-image6.Top;

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image6.Visible:=false;

image18.Picture:=image6.Picture;

sch:=sch+1;

i:=i+1;

end;

end;

end;

3:begin

if image1.visible=true then begin

l:=image19.Left-image1.Left;

t:=image19.Top-image1.Top;

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image1.Left);

t:=abs(100-image1.Top);

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

timer1.Enabled:=false;

{image1.Visible:=false;

image19.Picture:=image1.Picture;

image2.Visible:=true;

sch:=sch+1; }

end;

end;

if image2.visible=true then begin

l:=image19.Left-image2.Left;

t:=image19.Top-image2.Top;

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image2.Visible:=false;

image20.Picture:=image2.Picture;

image3.Visible:=true;

sch:=sch+1;

i:=i+1;

end;

end;

if image3.visible=true then begin

l:=image19.Left-image3.Left;

t:=image19.Top-image3.Top;

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image3.Left);

t:=abs(100-image3.Top);

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

timer1.Enabled:=false;

{image3.Visible:=false;

image21.Picture:=image3.Picture;

image4.Visible:=true;

sch:=sch+1; }

end;

end;

if image4.visible=true then begin

l:=image19.Left-image4.Left;

t:=image19.Top-image4.Top;

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin timer1.Enabled:=false;

image4.Visible:=false;

image22.Picture:=image4.Picture;

image5.Visible:=true;

sch:=sch+1;

i:=i+1;

end;

end;

if image5.visible=true then begin

l:=image19.Left-image5.Left;

t:=image19.Top-image5.Top;

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image5.Left);

t:=abs(100-image5.Top);

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

timer1.Enabled:=false;

{ image5.Visible:=false;

image23.Picture:=image5.Picture;

image6.Visible:=true;

sch:=sch+1; }

end;

end;

if image6.visible=true then begin

l:=image19.Left-image6.Left;

t:=image19.Top-image6.Top;

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then begin

timer4.Enabled:=true;

l:=abs(100-image6.Left);

t:=abs(100-image6.Top);

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

timer1.Enabled:=false;

{image6.Visible:=false;

image24.Picture:=image6.Picture;

sch:=sch+1; }

end;

end;

end;

end;

if sch=6 then

Timer3.Enabled:=true;

end;

//Реализация таймера

procedure TTest2.Timer2Timer(Sender: TObject);

begin

if n<>0 then begin

label1.Caption:=inttostr(n);

n:=n-1;

time := time + 1;

end else

begin

timer2.enabled:=false;

showmessage('Время вышло');

if i<>6 then showmessage('Тест не пройден') else

showmessage('Тест пройден');

end;

end;

//Реализация окончания работы программы

procedure TTest2.Timer3Timer(Sender: TObject);

begin

timer2.enabled:=false;

NextForm.show;

Test2.Hide;

Timer3.Enabled:=false;

NextForm.Label1.Caption := 'Имя: ' + kidName;

NextForm.Label2.Caption := 'Количество ошибок: ' + IntToStr(errors);

NextForm.Label3.Caption := 'Время: ' + IntToStr(time) + ' сек';

end;

//Анимация возврата к исходному положению

procedure TTest2.Timer4Timer(Sender: TObject);

begin

if image1.Visible=true then begin

l:=100-image1.Left;

t:=100-image1.Top;

image1.Left:=image1.Left+ (l div 10);

image1.top:=image1.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

if image2.Visible=true then begin

l:=100-image2.Left;

t:=100-image2.Top;

image2.Left:=image2.Left+ (l div 10);

image2.top:=image2.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

if image3.Visible=true then begin

l:=100-image3.Left;

t:=100-image3.Top;

image3.Left:=image3.Left+ (l div 10);

image3.top:=image3.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

if image4.Visible=true then begin

l:=100-image4.Left;

t:=100-image4.Top;

image4.Left:=image4.Left+ (l div 10);

image4.top:=image4.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

if image5.Visible=true then begin

l:=100-image5.Left;

t:=100-image5.Top;

image5.Left:=image5.Left+ (l div 10);

image5.top:=image5.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

if image6.Visible=true then begin

l:=100-image6.Left;

t:=100-image6.Top;

image6.Left:=image6.Left+ (l div 10);

image6.top:=image6.top+ (t div 10);

if ((l div 10)=0) and ((t div 10)=0) then

timer4.Enabled:=false;

end;

end;

**3.2 Руководство пользователя**

Для корректной работы программы необходимо осуществить запуск программы с помощью двойного нажатия на ярлык «Test for Kids». Ярлык программы представлен на рисунке 18.



Рисунок 18 – Ярлык программы

При запуске программы запустится окно выбора времени, ввода имени и начала теста, представлено на рисунке 19.

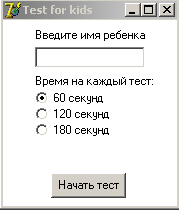


Рисунок 19 – Окно выбора времени, ввода имени и начала теста

После выбора времени, ввода имени ребенка и нажатия на кнопку «Начать тест» запустится Форма выбора теста. Форма представлена на рисунке 20.

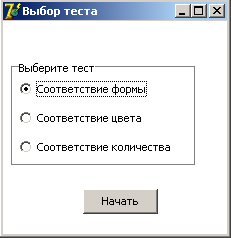


Рисунок 20 – Форма выбора теста

Для выбора необходимо кликнуть на интересующий тест и нажать на кнопку «Начать». Тест можно выбрать только один. По окончанию теста можно вернуться на данную форму и выбрать другой тест на форме результатов, представлена на рисунке 21.

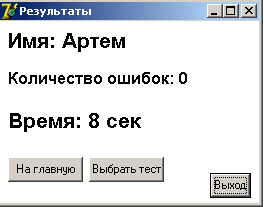


Рисунок 21 – Форма результатов

На данной форме представлен своего рода отчет, состоящий из имени ребенка, количества совершенных ошибок и времени, затраченного на прохождение теста. При нажатии на кнопку «На главную» программа отобразит окно выбора времени и ввода имени ребенка. По нажатию на кнопку «Выбрать тест» программа отобразит форму выбора тестов. При нажатии на кнопку «Выход» программа завершит свою работу.

**Заключение**

В рамках данного дипломного проекта было разработано программное приложение для развития детей дошкольного возраста.

С помощью разработанной программы преподаватель или родитель может сэкономить время в обучении детей, проводить анализ прогресса развития ребенка, основываясь на результатах прохождения теста в начале обучения, в середине и конце, а так же освободиться от бумажной работы.

Пользовательское приложение разработано в среде программирования Borland Delphi 10. Одним из этапов проектирования являлось проведение тестирования программы, в ходе которого выявлено, что программа является логически завершенной и не содержит явных ошибок.

Предметной областью разрабатываемого программного продукта является педагогическая деятельность в детских садах и школах.