**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 следующему тесту, 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.

**2.2 Описание входных и выходных данных**

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.

**2.3 Разработка интерфейса пользователя**

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

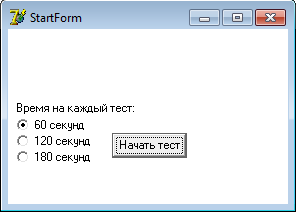
Форма начала работы программы представлена на рисунке 1.

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 |

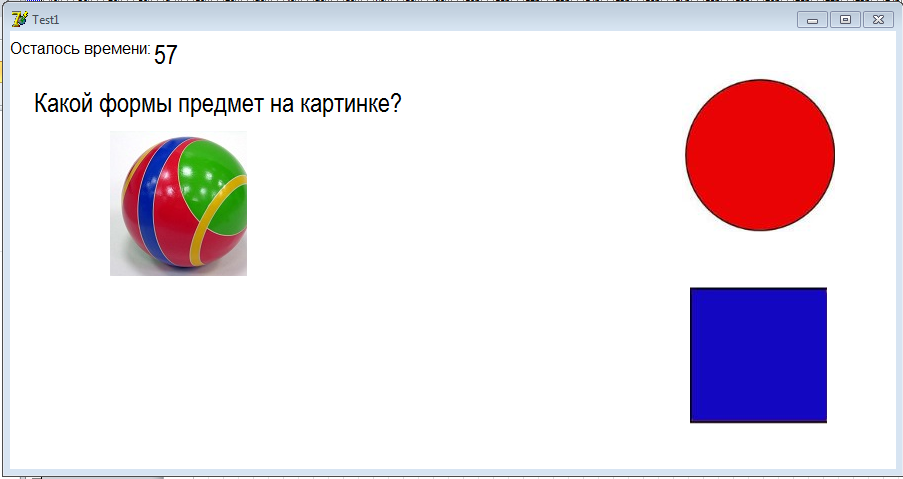
Форма «Test1» представлена на рисунке 2.

Рисунок 2 – Форма «Test2»

Компоненты и свойства данной формы представлены в таблице 2.

Таблица 2 – 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 3.

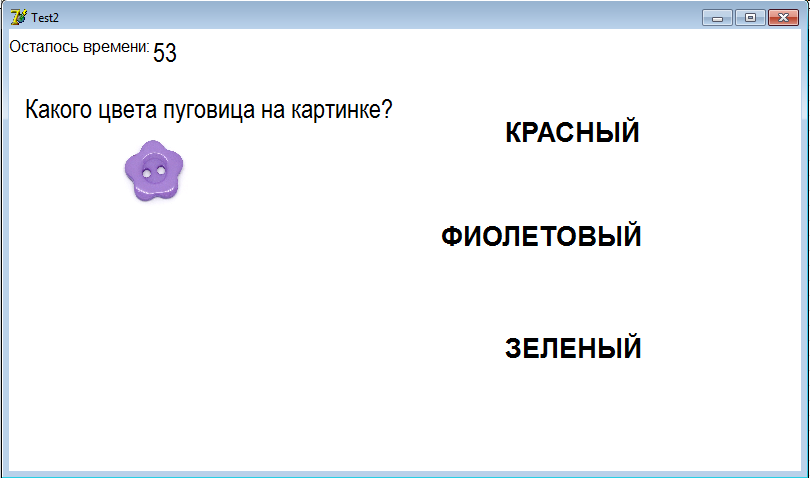


Рисунок 3 – Форма «Test2»

Компоненты и свойства данной формы представлены в таблице 3.

Таблица 3 – Компоненты и свойства формы «Test2»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Осталось времени | 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 |

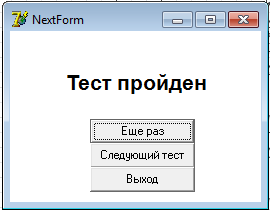
Форма «NextForm» представлена на рисунке 4.

Рисунок 4 – Форма «NextForm»

Компоненты и свойства данной формы представлены в таблице 4.

Таблица 4 – Компоненты и свойства формы «NextForm»

|  |  |  |  |
| --- | --- | --- | --- |
| **Компоненты** | **Свойства** | **Значение** | **Применение** |
| Label1 | Caption | Тест пройдет | Подпись |
| Button1 | Caption | Еще раз | Пройти тест еще  раз |
| Button1 | Caption | Следующий тест | Перейти  к следующему  тесту |
| Button1 | Caption | Выход | Выйти из приложения |

**2.4 Разработка программного кода**

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.

Техническим заданием является разработка программы – развивающей программы для детей дошкольного возраста, что подразумевает наличие в себе тестов, созданных для развития детей, кнопки управления и грамотный интуитивно понятный пользовательский интерфейс, всё это возможно создать с помощью среды разработки Delphi 10.

Проанализировав осуществляемую работу, было принято решение разработать следующий программный код для выполнения основных функций программы.

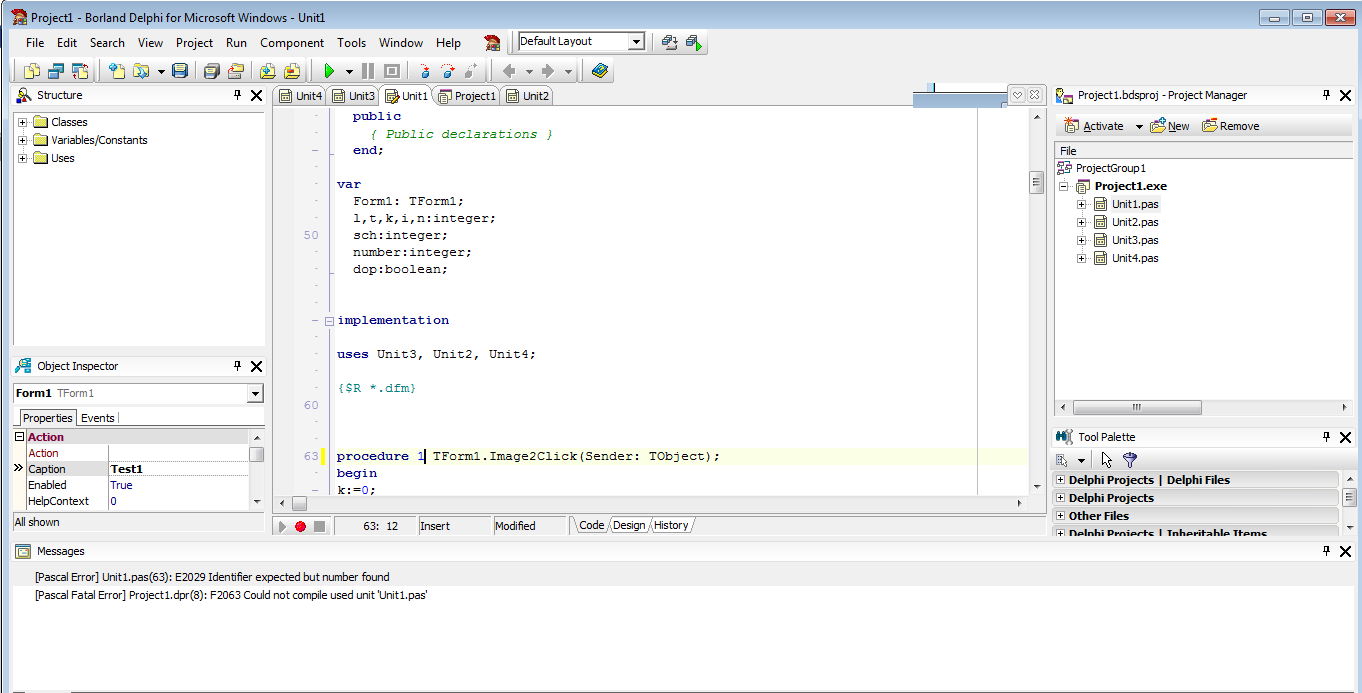
Программный код состоит из 4 unit (Test1, Test2, NextUnit, StartUnit) и 4 форм (Test1, Test2, NextForm, StartForm).

StartUnit – данный раздел предназначен для выбора времени для выполнения теста и кнопку, запускающую раздел «Test1». Test1 – данный раздел содержит обработку отрисовки задания теста, а так же обработку правильности выбранного варианта ответа. Test2 – данный раздел аналогичен разделу Test1. Отличием является лишь задание и варианты ответа. NextUnit – данный раздел начинает свое выполнение по завершению выполнения разделов Test1 и Test2. Главными его функциями являются переходы к следующему тесту, возвращение к только-что пройденному тесту, для его повторения и выход из программы.

* 1. **Отладка программы**

Отладка программы представляет собой поиск возможных ошибок в коде программы.

В ходе разработки данной программы мы столкнулись с синтаксическими ошибками, пример ошибки изображён на рисунке 6.

****Рисунок 6 - Синтаксическая ошибка в коде программы

Также были логические ошибки, которые изображены на рисунке 7.

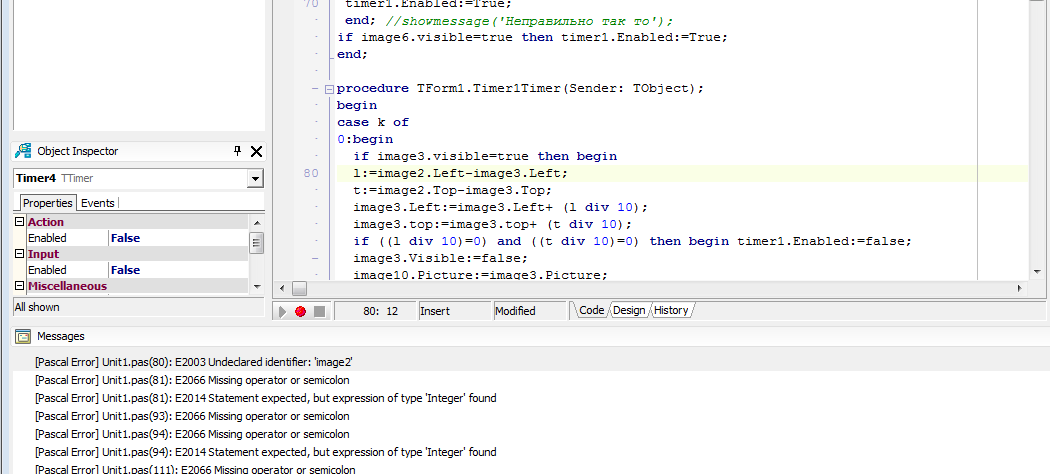


Рисунок 7 - Логическая ошибка в коде программы

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 8.

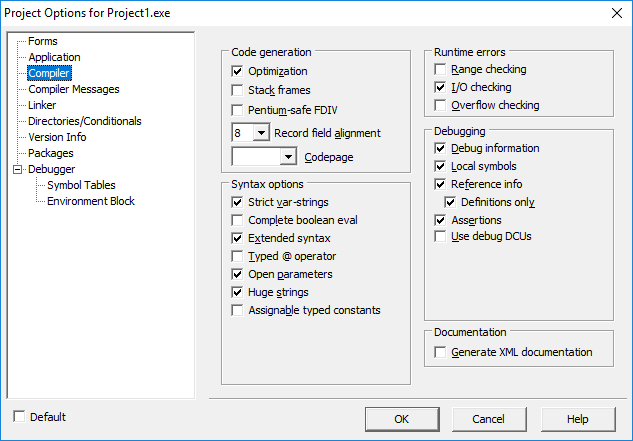


Рисунок 8 - Компилятор в Delphi10

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 9.

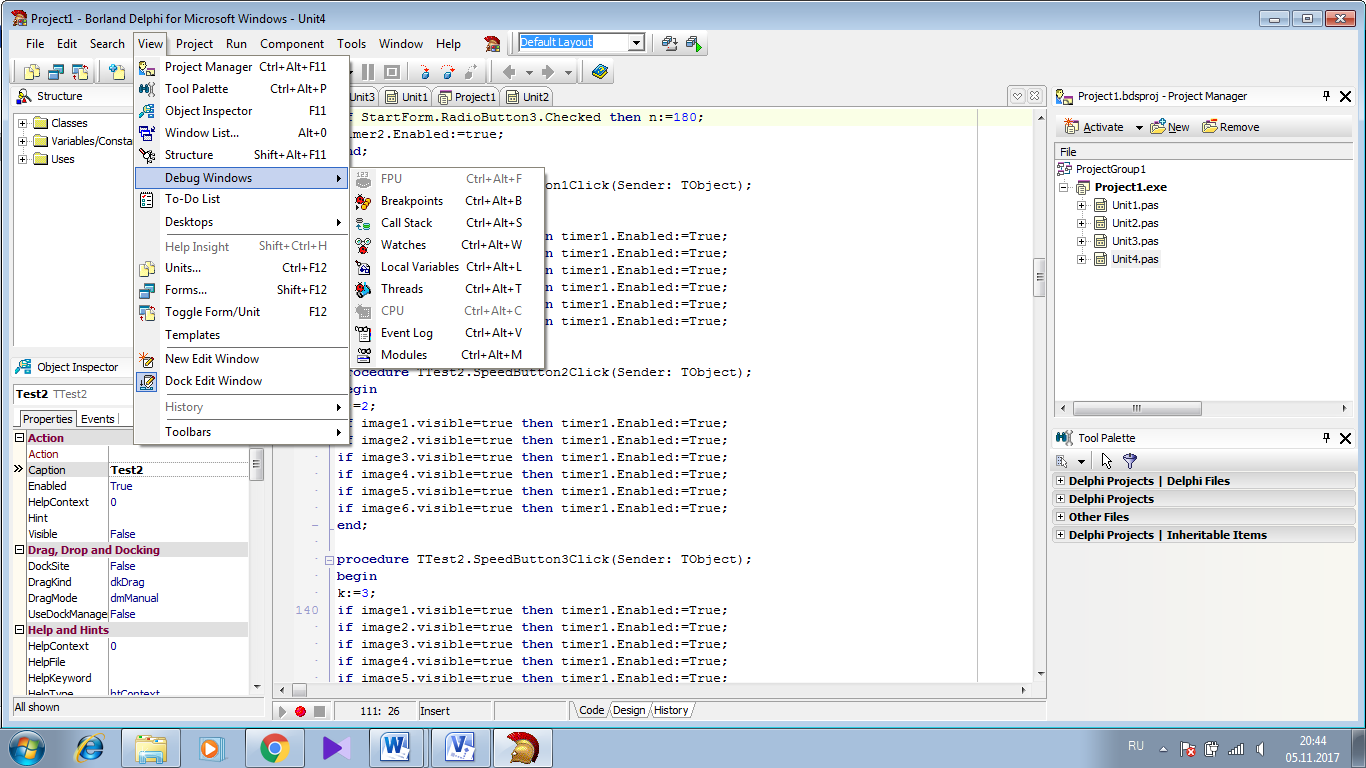


Рисунок 9 - Возможности Debug Windows

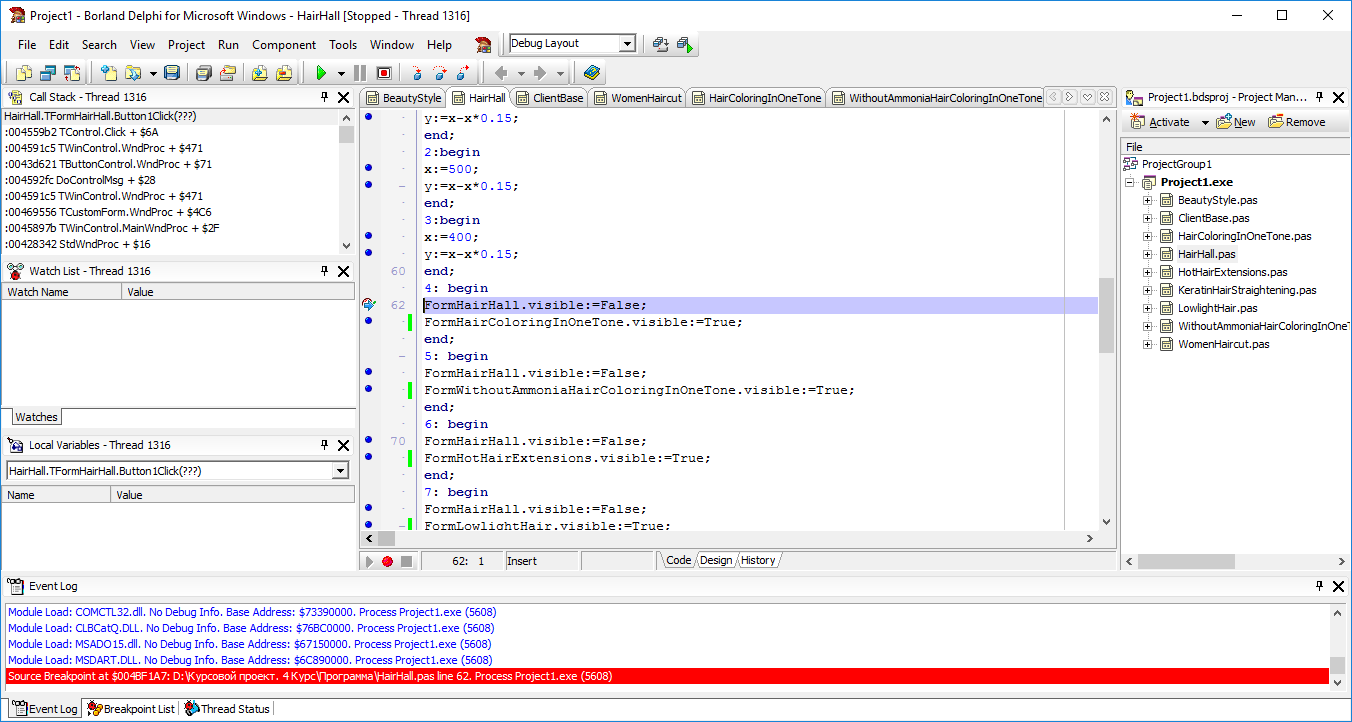
Пример работы программы до точки останова изображён на рисунке 10.

Рисунок 10 – 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.

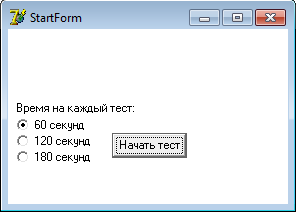
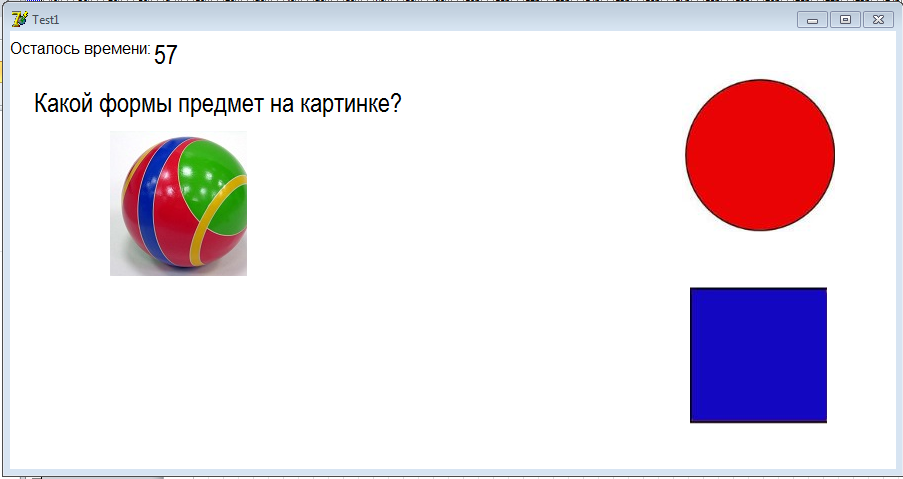
После старта работы программы она запускает форму стартового окна программы. Форма представлена на рисунке 11.

Рисунок 11 – Стартовая форма программы

При нажатии на переключатель времени задается время для прохождения каждого теста. Переключатель может быть выбран только один. По нажатию на кнопку «Начать тест» запускается первый тест программы. Форма теста представлена на рисунке 12.

Рисунок 12 – Первый тест программы

На форме представлены: время до окончания теста, при завершении которого тест будет считаться проваленным. Для прохождения теста необходимо кликнуть по нужному варианту ответа (представлены справа в виде круга и квадрата). При верном ответе появляется следующая фигура и процедура ответа повторяется. 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.

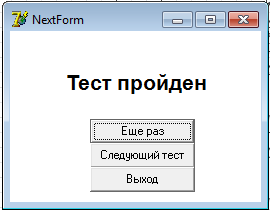
После прохождения теста запускается форма перехода к следующему тесту. Форма представлена на рисунке 13.

Рисунок 13 – Форма перехода к следующему тесту

Форма содержит в себе 3 кнопки: «Еще раз», «Следующий тест» и «Выход».

При нажатии на кнопку «Еще раз» предыдущий тест будет запущен еще раз с возможностью пройти его еще раз.

При нажатии на кнопку «Следующий тест» программа перейдет к следующему тесту, если таковой имеется, если тестов больше нет – программа завершит свою работу.

При нажатии на кнопку «Выход» программа завершит свою работу.

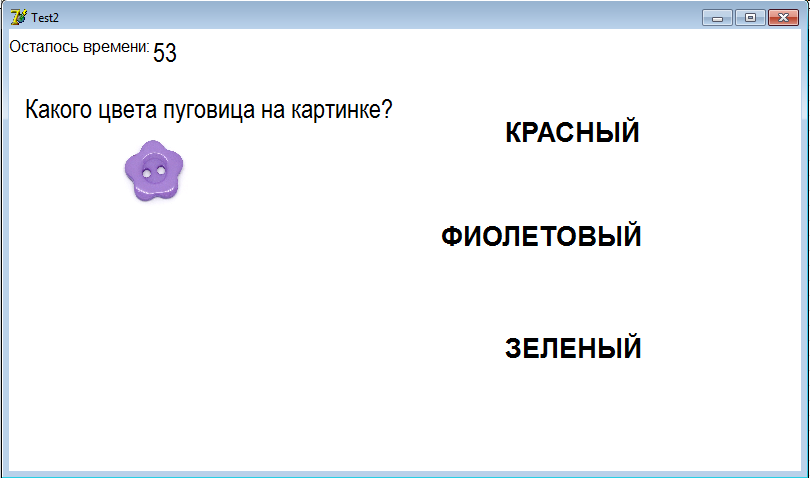
Форма теста представлена на рисунке 14.

Рисунок 14 – Второй тест программы

На форме представлены: время до окончания теста, при завершении которого тест будет считаться проваленным. Для прохождения теста необходимо кликнуть по нужному варианту ответа (представлены справа в виде кнопок с названиями цветов). При верном ответе появляется следующая пуговица другого или этого же цвета и процедура ответа повторяется. 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.

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