

РАЗРАБОТКА СЕТЕВОЙ ТЕСТИРУЮЩЕЙ ПРОГРАММЫ

Каранаев Л.М.

научный руководитель канд. физ.-мат. наук Сабитова Г.С.
*Стерлитамакская государственная педагогическая академия
им. Зайнаб Бишевой*

Целью данной работы является разработка сетевой тестирующей программы состоящей из трех модулей: конструктор тестов, модуль тестирования и журнал тестирования.

Актуальность темы обусловлена необходимостью создания элементов внутривузовской системы оценки качества подготовки студентов на основе результатов тестирования. Тестирование – это очень удобная форма проведения мониторинга качества образовательных услуг, предоставляемых вузом, дающая результаты проверки практически сразу и позволяющая, таким образом, проводить оперативную корректировку учебного процесса.

При выполнении данной работы нами ставились **следующие задачи:**

- создаваемая программа должна иметь интуитивно понятный интерфейс;
- программа должна качественно и полностью давать необходимую информацию;
- программа должна обеспечивать необходимую точность получаемых результатов;
- программа должна обеспечивать наиболее быстрое получение результатов обработки ответов, даваемых тестируемыми.

Основная задача заключается в том, чтобы дать возможность студентам потренироваться в выполнении тех видов заданий, которые входят в обязательный минимум содержания дисциплины предметной подготовки согласно федеральным государственным образовательным стандартам.

Научная новизна данной работы состоит в том, что программа позволяет организовать как локальное, так и сетевое тестирование. Программа поддерживает пять типов тестовых заданий: одиночный выбор, множественный выбор, тест на соответствие, тест на упорядочение и открытый тест.

Практическое значение работы. Предлагаемый автором проект программы позволяет объективно проверить и оценить знания студентов при очной и дистанционной формах обучения, а также может быть использован для подготовки к итоговому экзамену по дисциплине.

Описание программы. Программа состоит из трех модулей: конструктор тестов (TesterEditor), модуль тестирования (TesterStudent) и журнал тестирования (TesterServer).

Конструктор тестов предназначен для создания и редактирования тестов. С его помощью можно как создать новый файл тестов, так и открыть существующий для редактирования. Также можно настроить параметры теста: ограничение по времени прохождения тестирования, шкалу оценивания, количество тестовых заданий, на которые должен ответить студент, пароль автора теста.

Модуль тестирования позволяет открыть файл с тестовыми заданиями или получить его по сети и пройти тестирование. Тестируемый может сохранить файл с результатом тестирования или отправить его по сети. Если тестируемый проходит тест на локальном компьютере, то он может отправить результаты тестирования по электронной почте. Данный вид тестирования актуален при дистанционном обучении.

Журнал тестирования предназначен для раздачи тестовых заданий по сети, а также получения результатов тестирования со всех компьютеров и для их обработки.

Возможности данной программы:

- тесты создаются и редактируются в самой программе;
- программа поддерживает как локальное, так и сетевое тестирование;
- программа поддерживает пять типов тестов;
- программа допускает неограниченное количество тестовых заданий;
- программа позволяет вывести результаты тестирования в Excel;
- предусмотрено домашнее использование программы с целью подготовки или контроля знаний по какой-либо дисциплине: студент открывает тестирующий модуль и проходит предложенный тест.

Пути решения и средства реализации задачи. Программа разработана и выполнена в объектно-визуальной среде Delphi 7.

Самая первая проблема, которую необходимо решить – как хранить тестовые задания. Самое простое решение – использовать базу данных. Но для них необходимы специальные надстройки (BDE, DAO и т. д., в зависимости от базы), что усложнит программу. В данном случае можно обойтись и без них. Для хранения теста и его параметров используется файл. Для работы с файлами используется класс TFileStream, который позволяет создать поток для работы с файлами без учета типа хранящихся в нем данных, что для этой задачи важно. Это обусловлено тем, что программа работает с пятью типами тестов и для их реализации используются структуры данных, и все эти структуры хранятся только в одном файле.

Как уже было сказано, для работы с различными типами тестов используются структуры данных. Эти структуры, в зависимости от типа теста, имеют следующие поля: текст вопроса (переменная типа String), тексты вариантов ответа (массив данных размерности 10 типа String), правильные ответы (массив данных, тип которого зависит от типа тестового задания, который он хранит) и количество вариантов задания. Например, для хранения теста с одиночным и множественным выбором используется следующая структура данных:

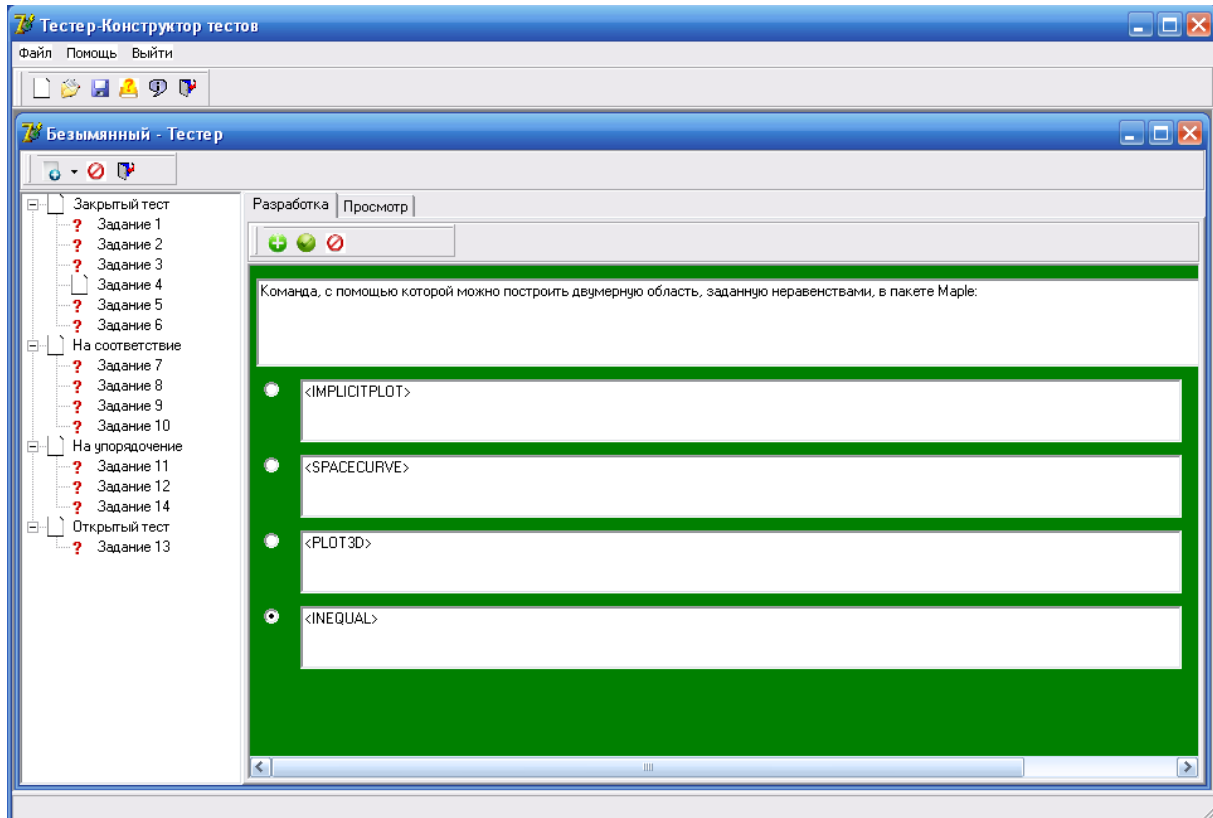
```
QuestionCloseType = ^TQuestionCloseType;  
TQuestionCloseType = record  
    TypeTest: byte;  
    Name: String[255];  
    ResultCount: integer;  
    ResultText: array[0..N-1] of String[255];  
    ResultValue: array[0..N-1] of Boolean;  
end;
```

Здесь объявлена переменная QuestionCloseType, которая является указателем на структуру TQuestionCloseType. Для хранения структур используется список CloseTestList. Для хранения остальных структур, хранящих другие типы тестов, используются свои списки. Эти списки инициализируются по событию OnShow и уничтожаются по событию OnClose.

В программе используются две основные формы. Главная форма является многодокументной. Вторая форма является дочерней. Она используется для редактирования определенного теста. Поэтому в конструкторе тестов можно редактировать сразу несколько тестов. Интерфейс этой формы следующий: в левой части находится дерево вопросов для быстрого перехода к нужному вопросу, в правой – окно редактирования. Вопросы можно создать, редактировать, удалить. Поля ввода

текста вопроса и вариантов создаются при выполнении программы, т. е. варианты ответов можно добавлять, перемещать вверх и вниз, удалять. Реализуется это с использованием списков:

```
TextList: TList; // списки для хранения элементов формы
TextList2: TList;
ButtonList: TList;
LabelList: TList;
TextQuestion: TMemo; // текст вопроса
```



Как уже отмечалось, компоненты, а именно TMemo, TRadioButton, TCheckBox, TLabel, TComboBox создаются и соответственно удаляются во время выполнения программы, т. е. они не рисуются на форме, а создаются при помощи программного кода во время выполнения программы. Например, для создания радиокнопки используется следующий код:

```
RadioButton:= TRadioButton.Create(Box);
RadioButton.Left:= _Left;
RadioButton.Top:= TempTop;
RadioButton.Tag:= ButtonList.Add(RadioButton);
Box.InsertControl(RadioButton);
```

Объясним описанный выше код программы. В первой строке кода создается радиокнопка и в скобках объявляется ее владелец – Box, во второй и третьей строках задаются левая и верхняя границы элемента, далее компонент RadioButton добавляется в список ButtonList, в последней строчке кнопка вставляется на форму. Удалить эту кнопку можно следующим кодом:

```
TRadioButton(ButtonList.Items[index]).Free;  
ButtonList.Delete(index);
```

При прохождении тестирования вопросы теста отбираются случайным образом из списка. Варианты ответов также отображаются в случайной последовательности. Для навигации используются кнопки, которые создаются во время выполнения программы (состояние runtime) программным кодом. Этот способ был описан выше. Для каждого вопроса создается своя кнопка. Таким образом, количество кнопок зависит от количества вопросов, на которые должен ответить тестируемый. Поэтому тестируемый, если даже ответил на некоторый вопрос теста, он может вернуться к этому заданию и исправить допущенную ошибку.

Для обработки результатов теста используются два списка. Один из них содержит правильные ответы теста, другой – ответы тестируемого. Как только пользователь завершит тестирование, программа проверяет результаты тестирования, сравнивая эти два списка. Далее программа подсчитывает количество правильных, неправильных и пропущенных заданий и выводит их на экран в процентном соотношении, а при необходимости выставляет оценку.

Если студент тестируется по сети, то результат тестирования отправляется также по сети преподавателю. В журнале тестирования предусмотрен экспорт результатов тестирования в Excel. Таким образом, преподаватель может при необходимости вывести отчет о тестировании студентов в Excel.

Реализованная программа в полной мере отвечает поставленным задачам. Программа позволяет объективно проверить и оценить знания студентов, позволяет легко создавать тестовые задания, допускает неограниченное число вопросов теста, поддерживает наиболее распространенные типы тестов. Данную программу можно использовать как для сетевого тестирования, так и для локального. Файл результатов тестирования может быть переслан по электронной почте.

Разработанная тестирующая программа полностью соответствует своему назначению, является готовым программным продуктом и может использоваться для проверки остаточных знаний студентов по соответствующей дисциплине.